

# SAT 问题中局部搜索法的改进

杨晋吉<sup>1,2</sup>      苏开乐<sup>1,3</sup>

<sup>1</sup>(中山大学信息科学与技术学院计算机科学系 广州 510275)

<sup>2</sup>(华南师范大学计算机科学系 广州 510631)

<sup>3</sup>(河南科技大学电子信息工程学院 洛阳 471003)

(yangjj@scnu.edu.cn)

## Improvement of Local Research in SAT Problem

Yang Jinji<sup>1,2</sup> and Su Kaile<sup>1,3</sup>

<sup>1</sup>(Department of Computer Science, School of Information Science and Technology at Sun Yat-sen University, Guangzhou 510275)

<sup>2</sup>(Department of Computer Science, South China Normal University, Guangzhou 510631)

<sup>3</sup>(Electronic and Information Engineering College, Henan University of Science and Technology, Luoyang 471003)

**Abstract** Recently, local search methods for solving the SAT problem have attracted considerable attention because they are the best known approaches to several hard classes of satisfiability. In this paper, an efficient initial probability is presented, which can constrain the random initial assignment of variables in the local search method. Also discussed is how to use the initial probability to improve the current local search algorithms such as WSAT, NSAT, TSAT, SDF, etc.. The strategies are straightforward and intuitive, and are helpful for increasing the number of satisfied clauses at its beginning of local search, thus decreasing the number of flipping for finding a solution and making the local search process converge fast. For this method to be compared with the other current local algorithms, many general random 3-SAT problem instances, structured SAT problem instances in SATLIB and phase transition's SAT problem instances are tested. The actual experimental results indicate that the improved algorithms are more efficient than the current local search SAT solvers such as WSAT, and so on. Moreover, this method can also be applied to other related methods to improve their efficiency.

**Key words** SAT problem; local search; probability

**摘 要** 局部搜索方法在求解 SAT 问题的高效率使其成为一研究热点. 提出用初始概率的方法对局部搜索算法中变量的初始随机指派进行适当的约束. 使在局部搜索的开始阶段, 可满足的子句数大大增加, 减少了翻转的次数, 加快了求解的速度. 用该方法对目前的一些重要的 SAT 问题的局部搜索算法 (如 WSAT, TSAT, NSAT, SDF 等) 进行改进, 通过对不同规模的随机 3-SAT 问题的实例和一些不同规模的结构性的 SAT 问题的实例, 以及利用相变现象构造的难解 SAT 实例测试表明, 改进后的这些局部搜索算法的求解效率有了很大的提高. 该方法对其他局部搜索法的改进具有参考价值.

**关键词** SAT 问题; 局部搜索; 概率

中图法分类号 TP18

# 1 引言

命题逻辑公式的 CNF 范式的可满足性问题 (SAT) 是计算机科学的非常重要的核心问题, 许多研究方向如人工智能、模型检测、数据库检索、计算机网络系统设计、计算机辅助设计与制造、VLSI 设计等中的许多问题都是转为 CNF 范式的可满足性问题即 SAT 的问题来求解的. SAT 的问题被证明是 NP 难解的问题. 目前解决该问题的方法主要有完备的方法<sup>[1,2]</sup>和不完备的方法<sup>[3-9]</sup>两大类. 完备的方法优点是保证能正确地判断 SAT 问题的可满足性, 但其计算效率很低, 平均的计算时间为多项式阶, 最差的情况计算时间为指数阶, 不适用于求解大规模的 SAT 问题. 不完备的方法的优点是求解的时间比完备的方法快得多, 但在很少数的情况下不能正确地判断 SAT 问题的可满足性. 由于完备的方法效率低, 寻求高效实用的不完备的 SAT 问题的求解算法成为一研究热点, 目前主要有局部搜索法<sup>[3-7,9,10]</sup>、数学物理法<sup>[8]</sup>等. 随机法、多项式法及完备和不完备相结合的方法<sup>[11]</sup>.

局部搜索法中比较经典的算法有 GSAT<sup>[3]</sup>, WSAT<sup>[4]</sup>, NSAT<sup>[5]</sup>, TSAT<sup>[6,10]</sup>, SDF<sup>[7]</sup>等, 其中文献 [3] 把 GSAT 和完备方法之一 PD 方法<sup>[1]</sup>进行了比较, 在许多方面 GSAT 有明显的优势. 但 GSAT 算法也有易陷入极小点的问题. 后来的 WSAT, NSAT, TSAT, SDF 等方法在其基础上进行了有效的改进.

笔者研究发现, GSAT, WSAT 和 NSAT 等局部搜索法对变量的初始指派都是随机的且都没有任何限制. 本文对这种变量的初始指派用变量的初始概率进行适度的限制, 使得改进后的 GSAT, WSAT, NSAT, SDF 算法等的执行效率都有了明显的提高.

本文首先介绍局部搜索法这些典型的算法, 然后介绍对这些算法的变量初始指派的改进, 最后通过对不同规模、不同结构的 SAT 问题的实例进行测试实验, 对改进前后的算法的执行效率进行了比较和分析.

## 2 SAT 问题中局部搜索法

### 2.1 SAT 问题及其表示

(1) SAT 问题的基本组成要素

① 文字 (literal):  $m$  个布尔变量的集合  $U =$

$\{x_1, x_2, \dots, x_n\}$ , 每个布尔变量  $x_i \in U$ ,  $x_i$  和  $\neg x_i$  ( $x_i$  的非) 称为文字.

② 子句 (clause):  $n$  个子句的集合为  $C = \{c_1, c_2, c_3, \dots, c_n\}$ , 每个子句  $c_i$  由若干文字通过析取运算符 ( $\vee$ ) 连接而成, 子句中文字的个数称子句的长度. 如  $c_i = x_1 \vee \neg x_3 \vee x_8$  的长度为 3.

③ 真值指派: 是一个函数  $t: U \rightarrow \{T, F\}$ , 若  $U$  的布尔变量的个数为  $m$ , 每个真值指派为一  $m$  元布尔向量.

④ 合取范式 CNF:  $F = c_1 \wedge c_2 \wedge \dots \wedge c_n$ , 其中  $c_i$  为子句,  $\wedge$  为合取运算符. 范式  $F$  在一真值指派  $t$  下为真当且仅当每个子句  $c_i$  在真值指派  $t$  下为真.

SAT 问题就是确定是否存在一真值指派  $t$ , 使得布尔变量集  $U$  上的 CNF 范式  $F$  为真, 若存在一真值指派  $t$  使  $F$  为真, 则称 SAT 问题可满足.  $F$  中的每个子句的长度都为  $l$ , 则称  $l$ -SAT 问题.

### (2) 局部搜索法

局部搜索法是解决许多 NP 问题的常有方法. SAT 的局部搜索法的基本思想是: 先任意给每个变量取个值, 得到一个初始指派, 它一般使一部分子句为真, 然后反复对指派进行局部的调整, 使得被满足的子句越来越多, 最后得到满足一个所有子句的指派, 即得到一个解.

GSAT, WSAT, NSAT, TSAT, SDF 等的公共算法解释如下所示:

算法 1. 局部搜索算法.

输入: 子句  $\alpha$  的集合, 最大的翻转次数 MAX-FLIPS, 最大的尝试次数 MAX-TRIES.

输出: 使子句  $\alpha$  为真的指派 (若有).

begin

for  $i = 1$  to MAX-TRIES

$T =$  一个随机产生的初始指派; // (a)

for  $j = 1$  to MAX-FLIPS

if  $T$  满足  $\alpha$  then return  $T$ ;

$p =$  按一定策略选出一布尔变量; // (b)

$T = T$  中的  $p$  的指派被翻转;

end for

end for

return “没有发现可满足的指派”;

end

各算法的主要区别在 (b) 处, 各算法简述如下:

(1) GSAT 算法是从当前所有的变量中, 选中

一变量  $p$ , 若变量  $p$  的翻转可使可满足的子句增幅最大且增幅不能小于零. 算法简单, 但该算法极小值的问题比较明显(即可能有解, 但陷在极值点来回摆动, 而无法获得解).

(2) WSAT 对 GSAT 算法的(b)处进行了有效的改进, 在当前不可满足的子句中随机挑选子句  $\alpha$ , 在子句  $\alpha$  的变量中, 以一定的概率  $\rho$  随机在子句  $\alpha$  挑选一变量  $p$ ; 以  $1-\rho$  的概率在子句  $\alpha$  挑选一变量  $p$  可满足的子句增幅最大. 该算法有效解决了极值点的问题, 求解效率比 GSAT 高, 但还有较多的平移现象.

(3) NSAT 算法稍微复杂一些, 它包含两种挑选变量  $p$  的方法(既 NOVELITY 方法和 RNOVELITY 方法). 这两种方法除在挑选最优  $p$  时有少许差别, 其他方面基本相同. NOVELITY 方法首先随机从不可满足的子句中随机选一子句  $\alpha$ , 在子句  $\alpha$  选出最优的变量  $p_1$  和次优的变量  $p_2$ , 这两个变量的翻转是使可满足的子句增幅最大的前两个变量, 若变量  $p_1$  不是该子句中最近被翻转过的变量, 就选  $p_1$ . 否则, 以概率  $P$  选变量  $p_2$ , 以概率  $1-P$  选变量  $p_1$ . RNOVELITY 方法在选择  $p_1$  还是  $p_2$  的策略上有些不同, 详见文献[5]. 总之, NSAT 方法可避免同一个变量被反复选中, 较好地解决了极值点的问题. 但求解的成功率比 WSAT 低.

(4) TSAT 采用了先进先出的 tabu 列表, 最近翻转过的进入该列表, 如选出的变量在列表中, 则不翻它, 而再选不在列表中的变量来翻转. 在一定程度上减少重复翻转的次数. 文献[12]采用了更有效的启发策略来挑选变量, 效率有进一步的提高.

(5) SDF 是比较新的局部搜索方法. 它根据各变量的权重的关系, 设计了非常成功的跳出极值点的跳坑函数, 文献[7]对上述算法进行的多种比较测试说明, SDF 具有较明显的优势.

上述各算法在(a)处都是一样的, 即( $T$  是一个随机产生的初始指派), 笔者发现该处的初始指派的好坏对算法效率的影响是很大的. 若初始指派较好, 可大大减少变量翻转的次数, 减少陷入极值点的可能性, 就能较快求得问题的解. 但目前笔者尚未发现有论文对该处进行优化的讨论.

### 3 本文对局部搜索法的分析和改进

SAT 问题中的临界点  $r_0$  (一般为  $n \div m$ ), 当实

际的  $r < r_0$  时, 解的个数以指数速度增加, 当实际的  $r > r_0$ , 解的个数以指数的速度减少到零. 文献[12, 13]等对  $r_0$  的大小进行了讨论和实验. 各文献的讨论有所差别, 但一致的观点认为变量可产生的子句的样本空间要大大地大于可满足的子句数  $m$ . 例如在典型的 3-SAT 问题中, 一般认为  $r_0$  在 4.3 附近, 当  $m$  个变量所产生子句的样本空间的样本数为  $2^3 C_m^3$ , 而临界点  $r_0$  处子句的数量  $n \approx 4.3m$ .

由于  $n \ll$  样本空间的样本数  $2^3 C_m^3$ , 各文字在子句中的分布非常不均, 甚至有些文字不在子句中出现. 例如在子句中只有  $x_3$ , 但根本没有  $x_3$ , 通过实验表明,  $x_3$  为解的可能性要大于  $x_3$  为解的可能性. 若能对变量的初始指派进行优化, 尽量使初始指派可满足的子句多, 这样, 可大大提高收敛的速度, 大大减少跌入极值点的可能.

本文对这类局部搜索法的改进基于上述思想, 主要就是在算法 1 的(a)处(即一个随机产生的初始指派的改进). 对各变量的随机指派根据它在各子句中出现的次数加以适当地约束. 设计了各变量  $i$  的初始指派取正的概率  $P_i$ , 其概率公式为

$$P_i = \begin{cases} m_i \times \Delta / (m_i + n_i) + \gamma & (m_i \neq 0 \text{ 或 } n_i \neq 0), \\ 0.5 & (m_i = 0 \text{ 且 } n_i = 0), \end{cases} \quad (1)$$

其中,  $m_i$  为取变量  $p_i$  为正的子句的个数;  $n_i$  为包含变量  $p_i$  为负的子句的个数;  $\Delta$  为 0.5~1 之间的一个常数;  $\gamma$  为  $-(1-\Delta) \sim (1-\Delta)$  之间的随机数.

例如: 有 6 个子句分别为  $c_1, c_2, c_3, c_4, c_5, c_6$ , 6 个变量分别为  $r_1, r_2, r_3, r_4$ , 其中:

$$c_1: r_1 \vee r_2 \vee r_3,$$

$$c_2: r_2 \vee r_3 \vee r_4,$$

$$c_3: r_1 \vee r_2 \vee r_3,$$

$$c_4: r_2 \vee r_4,$$

$$c_5: r_1 \vee r_2,$$

$$c_6: r_3 \vee r_4.$$

$r_1$  出现在子句  $c_1$  中,  $r_1$  出现在子句  $c_3, c_5$  中, 子句  $c_2, c_4, c_6$  中既无  $r_1$ , 也无  $\neg r_1$ . 若取  $\Delta = 0.9$ , 则  $r_1$  作为初始指派为正的的概率  $P_1$ , 由式(1)可得  $P_1 = 2 \times 0.9 / (2+1) + \gamma = 0.6 + \gamma$ . 在进行初始指派时,  $r_1$  以概率  $P_1$  为正进行随机指派. 变量  $r_2, r_3, r_4$  其相应的  $P_2, P_3, P_4$  同理也可求得. 在进行变量的初始指派时, 各变量  $x_i$  的初始指派以概率  $P_i$

取正.

改进后的算法如下所示：

算法 2. 改进后的局部搜索算法.

输入 :子句  $\alpha$  的集合 ,最大的翻转次数  $MAX-FLIPS$  ,最大的尝试次数  $MAX-TRIES$  .

输出 :使子句  $\alpha$  为真的指派( 若有 ).

begin

  计算各变量的初始指派为真的概率  $P_i$  ;

  for  $i = 1$  to  $MAX-TRIES$

$T =$  各变量以概率  $P_i$  为正 ,产生一初始随机指派 ;

    for  $j = 1$  to  $MAX-FLIPS$

      if  $T$  满足  $\alpha$  then return  $T$  ;

$p =$  按一定策略选出一布尔变量 ;

$T = T$  中的  $p$  的指派被翻转 ;

    end for

  end for

  return “ 没有发现可满足的指派 ”;

end

4 改进前后执行效率的比较和分析

针对以上的改进 ,对各算法的改进效率进行了如下的实验比较. 运行环境 :PIV 1.7GHz ,1GB 内存的 PC 机 ,Linux 操作系统.

由于 GSAT 算法存在易陷入极值点的问题 ,在进行大样本的随机测试中 ,效率较低 ,花费时间太多 ,故本实验主要对 GSAT 的改进算法 WSAT 和 NSAT 等进行测试比较 . 用 WSAT ,NSAT-N (NOVELITY 方法) ,NSAT-RN( RNOVELITY 方法) ,SDF 表示原算法 ,改进后的算法用 WSAT' ,NSAT-N' ,NSAT-RN' ,SDF' 表示.

测试所用的 SAT 问题的实例有随机产生的和有实际问题转化过来的两大类. 3-SAT 问题是 SAT 问题中一类 NP 难解的问题 ,其他的 SAT 的问题可转化为 3-SAT 问题. 因此 ,不失一般性 ,我们随机产生多种 3-SAT 的问题实例来测试. 每一种 3-SAT 问题的实例随机产生 50 次( 每次保证有解) ,每次随机产生的实例都用各算法求解 100 次 ,求得最后的平均值. 另一类测试所用的 SAT 问题是具有一定应用背景的组合问题转化而来的 ,它具有随机产生的 SAT 问题不具备的特性 ,如子句长度不等、一子句包含其他子句等 ,这些实例取自“ 1996 北京 SAT 问题实用算法国际比赛”<sup>[14]</sup>和 SATLIB-BenchmarkProblems<sup>[15]</sup>的比赛实例 ;还有一类基于 SAT 的相变现象<sup>[13]</sup>设计的难解实例<sup>[16]</sup>等 ,有效解决这类问题有较高的理论和应用价值.

通过测试不同类型、不同规模的 SAT 问题可以较充分地测试和验证算法的有效性. 它们的实验结果由表 1、表 2 所示：

Table 1 Experimental Results for 3-SAT Problem Instances (  $MAX-TRIES = 100$  ,  $MAX-FLIPS = 400000$  )  
表 1 随机产生的 3-SAT 问题实例的实验结果(  $MAX-TRIES = 100$  ,  $MAX-FLIPS = 400000$  )

CNF( $m, n, l$ )			WSAT	WSAT'	NSAT-N	NSAT-N'	NSAT-RN	NSAT-RN'	SDF	SDF'
Variables	Clauses	Clause Length	Time/Number	Time/Number	Time/Number	Time/Number	Time/Number	Time/Number	Time/Number	Time/Number
300	600	3	0.0006/100	0.0003/100	0.0007/100	0.0004/100	0.0009/99	0.0005/100	0.0009/100	0.0007/100
300	900	3	0.0015/100	0.0008/100	0.0092/100	0.0035/100	0.0974/91	0.0388/96	0.0019/100	0.0016/100
300	1200	3	0.0365/100	0.0211/100	0.4661/93	0.1756/98	1.865/45	1.0776/51	0.0413/100	0.0378/100
300	1260	3	0.409/92	0.306/99	15.128/28	5.84/39	51.9/50	31.02/61	0.666/97	0.555/99
1000	2000	3	0.0021/100	0.0012/100	0.0018/100	0.0013/100	0.0134/98	0.0032/100	0.0076/100	0.0053/100
1000	3000	3	0.0046/100	0.0034/100	0.0112/99	0.0048/100	0.2769/79	0.1115/89	0.0170/100	0.0135/100
1000	4000	3	0.1423/100	0.1297/100	*	*	17.7543/7	8.5358/16	0.9204/100	0.8911/100
5000	10000	3	0.0135/100	0.0101/100	0.0151/100	0.0103/100	0.0473/97	0.0210/99	0.1742/100	0.1196/100
5000	15000	3	0.0542/100	0.0406/100	0.0866/99	0.0516/100	1.2593/55	0.5344/74	0.2483/100	0.2088/100
5000	20000	3	2.2235/95	1.9121/97	*	*	*	*	*	*
10000	20000	3	0.0453/100	0.0313/100	0.0517/100	0.0324/100	0.1194/94	0.0412/99	0.9344/100	0.5326/100
10000	30000	3	0.1417/100	0.1086/100	0.2129/99	0.1292/99	3.2780/42	1.0433/67	1.4495/100	1.0173/100
10000	40000	3	12.2764/58	10.0667/62	*	*	*	*	*	*

Note : An asterisk mean that we don't test it in details for the hard solution of NSAT-N , NSAT-RN.

Table 2 Experimental Results for Structured SAT Problem Instances and PhaseTransition SAT Instances  
( MAX-TRIES = 100 , MAX-FLIPS = 400000 )  
表 2 实际问题转化来的 SAT 问题实例和部分利用相变现象构造的难解实例的实验结果  
( MAX-TRIES = 100 , MAX-FLIPS = 400000 )

Sample	CNR( $m, n, d$ )			WSAT	WSAT'	NSAT-N	NSAT-N'	NSAT-RN	NSAT-RN'	SDF	SDF'
	Variables	Clauses	Clause Length	Time/Number	Time/Number	Time/Number	Time/Number	Time/Number	Time/Number	Time/Number	Time/Number
VLSI Instances	649	1562	Not Fixed	0.0047/100	0.0031/100	Fail	1.0057/60	165.3465/1	3.2163/25	2.7644/89	3.697/94
	252	766		0.0012/100	0.0008/100	4.5233/29	0.0020/100	2.2321/38	0.8854/46	0.012/100	0.0176/100
	8432	31310		0.8100/100	0.7381/100	*	*	*	*	*	*
Block Instances	370	13732	Not Fixed	26.1083/17	16.5762/26	*	*	*	*	4.7467/100	4.0456/100
	540	34199		74.4978/13	68.9679/70	*	*	*	*	*	*
	1087	13772		6.8675/34	5.9886/34	*	*	*	*	1.6776/100	1.2005/100
JobShop Instances	19500	104527	Not Fixed	Fail	373.56/3	*	*	*	*	*	*
PhaseTransition Instances	450	19084	Not Fixed	37.94/56	10.933/88	*	*	*	*	16.29/78	16.306/75

Note : An asterisk mean that we don't test it in details for the hard solution of NSAT-N , NSAT-RN and for the abortion of SDF.

实验结果分析：

由表 1、表 2 的实验数据表明 ,在各种情况下 ,改进后的算法运行的效率绝大部分都有明显的提高 ( 仅有 SDF' 算法测试的个别难解样例例外 )。效率明显提高的原因就是以一定的概率对变量进行初始指派后 ,使算法开始执行时 ,可满足的子句的个数大幅度增加 ,这样就大大减少各变量翻转的次数。同时由于可满足的子句的个数增加 ,减少了陷入极小点的机会 ,减少走回头路的机会等。

通过多种算法比较还可看到不仅求解的速度提高 ,许多情况下求解的能力也有较大幅度的提高( 成功的样例数 )

各算法改进前后 ,当变量不变 ,子句数增加时 ,速度提升的幅度有一定程度的下降 ,但求解的能力有所上升。

5 结束语

通过以上的实验可看出 ,在用局部搜索法解决 SAT 的问题时 ,对变量的初始指派的随机性进行适当的限制 ,相当于进行一个预处理 ,可大大提高本类算法的搜索效率。若能针对所求 SAT 问题的子句特点 ,对各变量的初始概率  $P_i$  的求值公式再进行优化调整 ,效率提高会更明显。

本方法不是对一个算法的改进 ,而是对一类算法的改进。本文由于时间和版面所限 ,只对局部搜索法中的几个典型的速度较快算法进行了改进和比较 ,另对 GSAT ,TSAT 等算法的改进后 ,经过少量抽样测试也有较大幅度的提高。相信本文所论的改进方法对局

部搜索法中的其他类似算法也有参考价值。

从理论上对上述改进方法的复杂性进行更准确地分析正在研究中。

参 考 文 献

1 M. Davis , H. Putnam. A computer procedure for quantification theory. Journal of the ACM , 1960 , 7( 3 ) : 202 ~ 215

2 K. Iwama. CNF satisfiability test by counting and polynomial average time. SIAM J. Comput. , 1989 , 18( 2 ) : 385 ~ 391

3 B. Selman , H. J. Levesque , D. G. Mitchell. A new method for solving hard satisfiability problems. In : Proc. of the 10th AAAI ' 92. Menlo Park , CA : AAAI Press , 1992. 440 ~ 446

4 B. Selman , H. A. Kautz , B. Cohen. Noise strategies for improving local search. In : Proc. of the 12th AAAI ' 94. Menlo Park , CA : AAAI Press , 1994. 337 ~ 343

5 D. McAllester , B. Selman , H. Kautz. Evidence for invariants in local search. In : Proc. of the 14th AAAI ' 97. Menlo Park , CA : AAAI Press , 1997. 321 ~ 326

6 B. Mazure , L. Sais , E. Gregoire. Tabu search for SAT. In : Proc. of the 14th AAAI ' 97. Menlo Park , CA : AAAI Press , 1997. 281 ~ 285

7 D. Schuurmans , F. Southey. Local search characteristics of incomplete SAT procedures. Artificial Intelligence , 2001 , 132 ( 2 ) : 121 ~ 150

8 Li Wei , Huang Wenqi. A mathematic-physical approach to the satisfiability problem. Science in China( Series A ) , 1995 , 38( 1 ) : 116 ~ 127

9 G. Jun. Local search for satisfiability( SAT ) problem. IEEE Trans. on Systems , Man and Cybernetics , 1993 , 23( 4 ) : 1108 ~ 1129

10 Huang Wenqi , Zhang Defu , Wang Houxiang. An algorithm based on tabu search for satisfiability problem. Journal of Computer

Science and Technology , 2002 , 17( 3 ) : 340 ~ 346

12 J. M. Crawford , L. D. Auton. Experimental results on the crossover point in satisfiability. In : Proc. of the 11th AAAI '93. Menlo Park , CA : AAAI Press , 1993. 21 ~ 27

11 Zhang Jian. The Satisfiability Problem of Logic Formulas—Method , Tool and Application. Beijing : Science Press , 2000 ( in Chinese )  
( 张健. 逻辑公式的可满足性判定——方法、工具及应用. 北京 : 科学出版社 , 2000 )

13 Xu Ke , Li Wei. The SAT phase transition. Science in China ( Series E ) , 1999 , 42( 5 ) : 494 ~ 501 ( in Chinese )  
( 许可 , 李未. SAT 问题的相变现象. 中国科学( E 辑 ) , 1999 , 29( 4 ) : 354 ~ 360 )

14 [http :// www. cirl. uoregon. edu/ jc/ beijing/ blocks. cnf. bitadd. cnf](http://www.cirl.uoregon.edu/jc/beijing/blocks.cnf.bitadd.cnf)

15 [http :// www. intellektik. informatik. tu- darmstadt. de/ SATLIB/ benchm. html , bw\\_ large. b. cnf](http://www.intellektik.informatik.tu-darmstadt.de/SATLIB/benchm.html ,bw_large.b.cnf)

16 [http :// www. nlsde. buaa. edu. cn/ ~ kexu/ benchmarks/ benchmarks. htm , frb30\\_15. cnf. tar. gz](http://www.nlsde.buaa.edu.cn/~kexu/benchmarks/benchmarks.htm ,frb30_15.cnf.tar.gz)



**Yang Jinji** , born in 1968. He is currently associate professor and Ph. D. candidate in the Department of Computer Science of Zhongshan University. His research interests include verification of security protocol , artificial logic , and multi-media system.

杨晋吉 , 1968 年生 , 博士研究生 , 副教授 , 主要研究方向为安全协议验证、人工智能逻辑、多媒体系统.



**Su Kaile** , born in 1964. Professor and Ph.D. supervisor in the Department of Computer Science of Zhongshan University. His research interests include artificial logic , agent , and network security.

苏开乐 , 1964 年生 , 教授 , 博士生导师 , 主要研究方向为人工智能逻辑、智能体与网络安全.

Research Background

The SAT problem is not only one of long-term issues of computer algorithm and theory , but also can be used to many important applications such as VLSI design , CAD , model checking , verification , network design , and so on. As SAT is one of the most well-studied NP-complete problems , its fast and efficient algorithms attract more attention. In recent years , a variety of local search methods have been proposed for the SAT problem. They include , for examples , GSAT , GWSAT , SDF , WalkSAT , NoveltySAT , R- NoveltySAT , and so on. These methods can outperform the traditional complete methods on many aspects. But all of the above local search methods use random initial assignment to variables without any constraint. In this paper , we present an efficient initial probability that can constrain the random initial assignment of variable in local search method. The experimental results indicate that the improved algorithms are more efficient than these old ones. This work is supported by the National Natural Science Foundation of China under the grant No. 60473004.