CER 的重构方法,下面先给出 CER 的重构方法和相关证明,随后给出计算极大项个数和子句互补判定的增量求解方法.

3.1 公式重构

定义 5. 对于两个子句,称它们是互补子句当且仅当它们至少含有一对互补文字.

定义 6. 称一个包含多个子句的子句集是广义互补的当且仅当子句集中至少含有一对互补子句.

显然,定义 5 是定义 6 的一种特殊情况,定义 6 是定义 5 的扩展形式.由定理 4 可得:对于一对互补子句,它们所对应的极大项集合交集为空.下面给出定理 4 的扩展定理:

定理 6. 给定子句集,若它是广义互补的,则其包含的所有子句扩展出的极大项集合交集为空集.

证明:由定义 6 可知:假设给定子句集包含子句 $C_1,C_2,\ldots,C_n$,若它是广义互补的,那么所有子句中至少含有一对互补子句.假设为 C_i,C_j,则由定理 4 可知,P_i 与 P_j 的交集为空.而由集合的性质可知:对于 $P_1,P_2,\ldots,P_n$,它们的交集必是其中任意两个集合交集的子集;特殊地,也必是 P_i 与 P_j 的交集的子集,故 $P_1,P_2,\ldots,P_n$ 的交集也为空集. □

显然,当给定子句集中含有的互补子句较多时,公式(1)中值为 0 的计算项较多,从而此时这种基于扩展规则的方法效率一般较高.为了衡量给定子句集中互补子句的多少,下面给出互补因子的概念.

定义 7[20]. 给定一个子句集$\Phi=\{C_1,C_2,\ldots,C_n\}$,互补因子(complementary factor)是子句集中互补子句对个数与所有子句对个数之比,即 $T/(n*(n-1)/2)$,其中,T 表示互补子句对的个数.

CER 方法在求解时未能考虑公式(1)中各计算项之间的扩展关系,下面给出重构 CER 方法中关于各计算项间扩展关系的相关定义.

定义 8. 称公式(1)中形如 $P_{i_1}\cap P_{i_2}\cap\ldots\cap P_{i_h}$ 的公式为极大项相交集,用字母 Q 表示,$|P_{i_1}\cap P_{i_2}\cap\ldots\cap P_{i_h}|$ 表示极大项相交集中极大项的个数,称形如 $P_{i_1}\cap P_{i_2}\cap\ldots\cap P_{i_h}\cap P_k$ 的公式为其扩展极大项相交集.用 $ET(Q)$表示极大项相交集 Q 的所有扩展极大项相交集;相应地,称子句集$\{C_{i1},C_{i2},\ldots,C_{ih}\}$为 $P_{i_1}\cap P_{i_2}\cap\ldots\cap P_{i_h}$ 的基础子句集,并用 $BS(Q)$表示极大项相交集 Q 的基础子句集.其中,$h=1,2,\ldots,n,k\neq i_1,i_2,\ldots,i_h$ 且 $i_1<i_2<\ldots<i_h<k$.特殊地,称 $h=1$ 时的极大项相交集为单相交集.

由定义 8 可知:一个极大项相交集的基础子句集是其扩展极大项相交集基础子句集的子集,且只相差一个子句元素.基于它们之间这种扩展关系,下面给出已知前者的互补性判断后者是否广义互补的定理.

定理 7. 给定极大项相交集 Q 和 Q',且 $Q'\in ET(Q),BS(Q')=BS(Q)\cup\{C\}$:

1) 若已知基础子句集 $BS(Q)$广义互补,那么 $BS(Q')$也广义互补;
2) 若已知基础子句集 $BS(Q)$不是广义互补的,那么 $BS(Q')$是广义互补当且仅当 $BS(Q)$中至少存在一个子句与 C 互补.

证明:

1) 由定义 6 可知:若 $BS(Q)$是广义互补的,那么它至少包含一对互补子句;又因为 $BS(Q)$是 $BS(Q')$的子集,所以 $BS(Q')$中也含有互补子句对,故 $BS(Q')$也是广义互补的;
2) 当 $BS(Q)$不是广义互补的,即,$BS(Q)$中不含有互补子句对时:
   - 充分性:若 $BS(Q')$是广义互补的,在 $BS(Q)\cup\{C\}$中必有互补子句对;又由于 $BS(Q)$中子句两两之间不互补,故其中必然至少存在一个子句与 C 互补;
   - 必要性:反之,当 $BS(Q)$中至少存在一个子句与子句 C 互补时,显然,子句集 $BS(Q')$是广义互补的.

□

下面给出已知极大项相交集的值,计算其扩展极大项相交集的定理:

定理 8. 对于给定的极大项相交集 $P_i\cap P_j\cap\ldots\cap P_l$ 和它的一个扩展极大项相交集 $P_i\cap P_j\cap\ldots\cap P_l\cap P_k$:

1) 若子句集$\{C_i,C_j,\ldots,C_l,C_k\}$是广义互补的,则有$|P_i\cap P_j\cap\ldots\cap P_l\cap P_k|=0$;
2) 若子句集$\{C_i,C_j,\ldots,C_l,C_k\}$不是广义互补的,则有$|P_i\cap P_j\cap\ldots\cap P_l\cap P_k|=|P_i\cap P_j\cap\ldots\cap P_l|/2^{|Z|}$,其中,$Z=\{x|x\in C_k$ 且 $x\notin C_h,h=i,j,\ldots,l\}$.

证明:

1) 若子句集$\{C_i,C_j,\dots,C_l,C_k\}$为广义互补时,由定理 6 可知,$|P_i\cap P_j\cap\dots\cap P_l\cap P_k|=0$;

2) 若子句集$\{C_i,C_j,\dots,C_l,C_k\}$不是广义互补的,显然,子句集$\{C_i,C_j,\dots,C_l\}$也不是广义互补的,则有$|P_i\cap P_j\cap\dots\cap P_l|=2^{|X|-|C_i\cup C_j\cup\dots\cup C_l|}$,而且$|P_i\cap P_j\cap\dots\cap P_l\cap P_k|=2^{|X|-|C_i\cup C_j\cup\dots\cup C_l\cup C_k|}$,由此可得如下公式:

$$|P_i\cap P_j\cap\dots\cap P_l\cap P_k|=|P_i\cap P_j\cap\dots\cap P_l|/2^{|C_i\cup C_j\cup\dots\cup C_l\cup C_k|-|C_i\cup C_j\cup\dots\cup C_l|}.$$

其中,$|C_i\cup C_j\cup\dots\cup C_l\cup C_k|-|C_i\cup C_j\cup\dots\cup C_l|$即表示包含在子句 C_k 中同时不被子句 $C_i,C_j,\dots,C_l$ 包含的文字个数,即$|Z|$. □

例 5:对于一个包含子句 C_1,C_2 的子句集Φ,计算其所能扩展出的极大项个数时,假设 C_1,C_2 不互补,已知$|P_1|=2^{|X|-|C_1|}$,所以计算$|P_1|$时只需要遍历子句 C_1,计算其长度即可.而在计算$|P_1\cap P_2|$时要计算出$|C_1\cup C_2|$,即,要遍历 C_1 和 C_2 计算它们总共包含多少个文字.而 $P_1\cap P_2$ 是 P_1 的扩展极大项相交集,根据定理 8,计算$|P_1\cap P_2|$恰恰可以利用计算$|P_1|$时的结果,只计算 C_2 与 C_1 不同的文字数即可.

CER 方法在利用公式(1)分层求解给定子句集所能扩展出的极大项个数时,未能考虑极大项相交集和扩展极大项相交集之间的扩展关系.为在计算扩展极大项相交集时能充分利用对应的极大项相交集计算后的结果,对 CER 方法中的公式(1)进行重构,其重构后的公式如下:

$$S=\sum_{i=1}^{n}\left(|P_i|-\sum_{j=i+1}^{n}\left(|P_i\cap P_j|-\sum_{l=j+1}^{n}(|P_i\cap P_j\cap P_i|-\dots-(|P_i\cap P_j\cap\dots\cap P_n|)\dots)\right)\right) \tag{2}$$

定理 9. 公式(2)与公式(1)等价.

证明:

$$\begin{aligned}
S&=\sum_{i=1}^{n}\left(|P_i|-\sum_{j=i+1}^{n}\left(|P_i\cap P_j|-\sum_{l=j+1}^{n}(|P_i\cap P_j\cap P_l|-\dots-(|P_i\cap P_j\cap\cap\dots P_n|)\dots)\right)\right)\\
&=\sum_{i=1}^{n}|P_i|-\sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(|P_i\cap P_j|-\sum_{l=j+1}^{n}(|P_i\cap P_j\cap P_l|-\dots-(|P_i\cap P_j\cap\cap\dots P_n|)\dots)\right)\\
&=\sum_{i=1}^{n}|P_i|-\sum_{1\le i<j\le n}|P_i\cap P_j|+\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{l=j+1}^{n}(|P_i\cap P_j\cap P_l|-\dots-(|P_i\cap P_j\cap\cap\dots P_n|)\dots)\\
&\dots\\
&=\sum_{i=1}^{n}|P_i|-\sum_{1\le i<j\le n}|P_i\cap P_j|+\sum_{1\le i<j<l\le n}|P_i\cap P_j\cap P_l|-\dots+(-1)^{n-1}|P_i\cap P_j\cap\cap\dots P_n|.
\end{aligned}$$

由以上推导过程可知,公式(1)与公式(2)是等价的. □

公式(2)是公式(1)等价的重构表达形式,下面给出公式(2)的展开过程:

$$\begin{aligned}
S&=\sum_{i=1}^{n}\left(|P_i|-\sum_{j=i+1}^{n}\left(|P_i\cap P_j|-\sum_{l=j+1}^{n}(|P_i\cap P_j\cap P_l|-\dots-(|P_i\cap P_j\cap\dots\cap P_n|)\dots)\right)\right)\\
&=|P_1|-\sum_{j=2}^{n}\left(|P_1\cap P_j|-\sum_{l=j+1}^{n}(|P_1\cap P_j\cap P_l|-\dots-(|P_1\cap P_j\cap\dots\cap P_n|)\dots)\right)\\
&\quad+\sum_{i=2}^{n}\left(|P_i|-\sum_{j=i+1}^{n}\left(|P_i\cap P_j|-\sum_{l=j+1}^{n}(|P_i\cap P_j\cap P_l|-\dots-(|P_i\cap P_j\cap\dots\cap P_n|)\dots)\right)\right)\\
&=|P_1|-|P_1\cap P_2|+\sum_{l=3}^{n}(|P_1\cap P_2\cap P_l|-\dots-(|P_1\cap P_2\cap\dots\cap P_n|)\dots)\\
&\quad-\sum_{j=3}^{n}\left(|P_1\cap P_j|-\sum_{l=j+1}^{n}(|P_1\cap P_j\cap P_l|-\dots-(|P_1\cap P_j\cap\dots\cap P_n|)\dots)\right)
\end{aligned}$$

$$+\sum_{i=2}^{n}\left(|P_i|-\sum_{j=i+1}^{n}\left(|P_i\cap P_j|-\sum_{l=j+1}^{n}(|P_i\cap P_j\cap P_l|-...-(|P_i\cap P_j\cap...\cap P_n|)...)\right)\right)$$

...

RCER 方法按照公式(2)以上展开形式顺序计算极大项相交集和其扩展极大项相交集的值,下面给出结合公式(2)对极大项相交集对应的扩展极大项相交集进行剪枝的相关推论.

由定义 6 和定理 6 易得如下推论:

推论 1. 已知一个广义互补的子句集Φ,则子句集Φ的任意超集Φ'中所有子句扩展出的极大项集合交集必为空.

证明:由定义 6 可知,子句集Φ中必有互补子句对,所以包含这个子句集的任意子句集Φ'也必然含有互补子句对,是广义互补的;再根据定理 6 可知,子句集Φ'中所有子句扩展出的极大项集合交集必为空. □

例 6:给定一个子句集$\Phi=\{C_1,C_2,C_3,C_4\}$,若子句 C_1 和 C_2 互补,那么不仅$|P_1\cap P_2|=0$,只要以同时包含 C_1,C_2 的子句集为基础子句集的极大项相交集,其值也必为 0,如,$\{C_1,C_2,C_3\}$,$\{C_1,C_2,C_4\}$,$\{C_1,C_2,C_3,C_4\}$对应的极大项相交集 $P_1\cap P_2\cap P_3$,$P_1\cap P_2\cap P_4$,$P_1\cap P_2\cap P_3\cap P_4$,的值都等于 0.

由推论 1 可知:在公式(2)的计算过程中,已知一个极大项相交集的值为 0,便可得其所有扩展极大项相交集的值也为 0,从而可将其对应的冗余计算项提前从公式(2)中删除,减少不必要的计算.

下面给出求解扩展极大项相交集时,充分利用极大项相交集计算结果的增量求解方法.

3.2 极大项相交集增量计算方法

事实上,公式(2)中所有极大项相交集的基础子句集是需求解子句集幂集中的元素,只要对相应的基础子句集中子句进行文字计数,便可求出该极大项相交集的值.按照公式(2)展开形式的计算顺序,一般情况下,下一个即将计算的极大项相交集是上一个已计算的极大项相交集的扩展极大项相交集,可在上一个已计算的极大项相交集结果的基础上增量计算,即,只需对每次求解时多出的一个子句进行文字计数,避免了对子句集中所有子句逐一遍历求解.

按照公式(2)展开形式进行极大项个数求解,充分利用了极大项相交集和扩展极大项相交集之间的扩展关系:(a) 根据定理 7,判断当前极大项相交集 Q'的基础子句集 $BS(Q')$是否广义互补时,$BS(Q')$除了包含之前已计算极大项相交集 Q 的基础子句集 $BS(Q)$中所有子句外,还包含一个不在 $BS(Q)$中的子句,只需要判断此子句是否与 $BS(Q)$中的子句互补即可.因 $BS(Q)$中子句对的互补情况已经判断过,无需重复计算;(b) 根据定理 8,在计算当前极大项相交集 Q'的值时,上一个已计算的极大项相交集 Q 的基础子句集比 Q'的基础子句集少包含一个子句,只需要计算此子句中没有被 $BS(Q)$覆盖的文字个数即可.

下面给出基于扩展规则重构的#SAT 增量求解方法 RCER.

算法. RCER ($\Phi=\{C_1,C_2\ldots,C_n\}$,$X=\{x_1,x_2,\ldots,x_m\}$).

```
01.  初始化:BS=∅, inte_V=0, Prev_com=0, Sum=0, Prev_VarCover[m]={0};
02.  Count_Max_terms(BS,inte_V,Prev_com,Prev_VarCover)
03.    BEGIN
04.      For (i=inte_V; i<n; i++)
05.        IF (BS∪{C_i}中有互补子句对) THEN
06.          Continue;
07.        ENDIF
08.        Fol_VarCover[m]←Prev_VarCover[m];
09.        Fol_com←Prev_com;
10.        For (C_i 中每个变量 x_j)
11.          IF (Fol_VarCover[j]==0) THEN
12.            Fol_VarCover[j]=1;
```

```
13.            Fol_com++;
14.          ENDIF
15.        ENDFor
16.        IF (BS∪{Ci}共包含奇数个子句) THEN
17.          Sum+=2^m–Fol_com;
18.        ELSE
19.          Sum–=2^m–Fol_com;
20.        ENDIF
21.        Count_Max_terms(BS∪{Ci},i+1,Fol_com,Fol_VarCover);
22.      ENDFor
23.    END
24.  Num_models=2^m–Sum;
25.  Return Num_models;
```

算法中,*BS* 代表极大项相交集的基础子句集,即,子句集*Φ*的子集,初始时为∅.用 *Sum* 表示子句集*Φ*能扩展出的极大项个数,*Prev_com* 表示前一个基础子句集中变量个数,并且用数组 *Prev_VarCover*[*m*]记录都有哪些变量:若有变量 x_i,则 *Prev_VarCover*[*i*]的值为 1;否则为 0.而 *Fol_com* 则表示当前的基础子句集中变量个数,对应的 *Fol_VarCover*[*m*]则记录当前基础子句集的变量覆盖情况.*Count_Max_terms* 是求解 *Sum* 值的递归函数,每调用一次函数就进入下一层的计算,即,开始计算当前极大项相交集的扩展极大项相交集.若当前要计算极大项相交集的基础子句集是广义互补的,那么直接进行下一次循环.通过这种方式,较好地避免了对其值必为 0 的极大项相交集的冗余计算.

RCER 算法的主要工作是计算给定子句集所能扩展出的极大项个数,根据公式(2)的展开形式计算可以充分利用计算项之间的扩展关系,不仅在上一次计算结果上进行增量计算,还可避免多数冗余项的计算,进而较好地提高了计算效率.在 RCER 算法中,每生成一个极大项相交集的基础子句集,都要判断其中是否含有互补子句对,若每次都对相应基础子句集中的所有子句对进行遍历判断是否含有互补文字,则降低了判断效率.因此,提出一种基于互补表的增量判定方法来提高效率.

3.3 互补表增量判定方法

该方法中构建一个互补表,对子句间的互补关系进行记录.考虑到空间复杂性,互补表由一个下三角矩阵构成.互补表中对角线位置记录相应子句与其他子句互补情况,互补表中其他位置记录相应子句对的互补关系.互补表方法有效避免了互补判定中的重复计算,尤其是相应子句与其他所有子句都不互补时可直接得到判定结果.包含 4 个子句的子句集*Φ*={C_1,C_2,C_3,C_4}的互补表如图 2 所示.

$a_{1,1}$			
$a_{2,1}$	$a_{2,2}$		
$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	
$a_{4,1}$	$a_{4,2}$	$a_{4,3}$	$a_{4,4}$

Fig.2 Example of complementary table

图 2 互补表示例

若子句 C_i 与子句 C_j 是互补的,则表中元素 $a_{i,j}$=1;否则,$a_{i,j}$=0.通过这种方式,互补表可以记录所有子句间的互补关系.当多次判断同一子句对是否互补时,除第 1 次判断外,仅需要查找互补表中相应元素的值即可.表中 $a_{i,i}$

的值等于第 i 行中除其之外所有元素的和,即,$a_{i,i}$ 表示子句 C_i 与 $C_1,\ldots,C_{i-1}$ 中互补的子句个数.初始时,可将表中除对角线外所有的元素设为较大的值,如 n(给定子句集中子句个数),且 $a_{i,i}$ 初始值为$(i-1)n$.

在RCER方法中,根据定理7,可以在极大项相交集的基础上增量计算扩展极大项相交集基础子句集是否广义互补,进而提高子句集广义互补的判定效率.当前需判定的极大项相交集基础子句集是前一个已判断的极大项相交集基础子句集的超集,且只多包含一个子句,只需判断这个子句与基础子句集中其他子句是否互补即可.根据 RCER 算法中极大项相交集基础子句集生成顺序,新生成的基础子句集中多包含的子句在当前子句集中标号最大,这里用 C_l 表示,按照如下步骤判断基础子句集是否广义互补:

1) 在互补表中,先查找对角线元素 $a_{l,l}$ 的值是否为 0,即,是否与标号小于它的其他所有子句都不互补.如果都不互补,则直接得知此基础子句集不是广义互补子句集;
2) 若 $l-a_{l,l}$ 的值小于当前子句集中子句个数,即 $C_1,C_2,\ldots,C_{l-1}$ 中与 C_l 不互补的子句个数小于当前子句集中除 C_l 外的子句个数,则子句集中必然存在与 C_l 互补的子句,故基础子句集是广义互补的;
3) 否则,继续在表中分别查找相应位置,判断子句 C_l 与基础子句集中其他子句是否互补.

互补表增量判定方法较大程度地减少 CER 算法在判断互补时所花费的时间,提高了算法效率.下面给出判断子句集中是否含有互补子句对的增量判定方法.

EST_Complementry(Φ).

/*Φ为当前基础子句集,假设Φ中 C_l 为标号最大子句,且Φ中子句个数为 k*/

```
01.  IF a_{l,l}==0 THEN Return False;
02.  IF l-k<a_{l,l}<l THEN Return True;
03.  For (Φ中除 C_l 外的其他每个子句 C_i)
04.    IF a_{l,i}==1 THEN Return True;
05.    IF a_{l,i}==n THEN
06.      遍历 C_l,C_i;
07.      IF C_l 与 C_i 含互补文字 THEN
08.        a_{l,i}=1;
09.        a_{l,l}=a_{l,l}-(n-1);
10.        Return True;
11.      ELSE
12.        a_{l,i}=0;
13.        a_{l,l}=a_{l,l}-n;
14.      ENDIF
15.    ENDIF
16.  ENDFOR
17.  Return False;
```

4 实验结果

本节将RCER方法与CER方法进行比较,并给出两种方法在随机问题实验测试用例的结果.实验平台如下:Windows XP 操作系统,CPU AMD Athlon(tm) 64 X2 Dual Core Processor 3600+ 1.9GHz,2.00GB RAM.

实验用例由随机生成器产生,其输入参数有变量个数 m、子句个数 n 以及变量在每个子句中的出现概率 p,每个子句都按一定概率 p 从 m 个变量中选取变量,因变量的正负文字相对于子句集来说是对称的,所以这里只控制变量为正的概率 p',p'范围是(0.1,0.5).本文的实验数据采用变量个数为 30,子句个数为 100 的随机样例进行实验测试.通过限定变量在子句中出现的概率,并结合其正文字出现的概率来限定子句集中子句的平均子句长度以及互补因子的大小.实验测试互补因子范围为(0.1~0.9),实验结果是 10 次实验的平均结果.

从图 3 的对比图中可以看到:在大部分测试用例中,RCER 方法的求解效率要比 CER 方法有明显提高,并且图中有些散点已经达到 2 倍对比线甚至 3 倍对比线.但是注意到:在耗时相对较少的例子中,效率提高并不明显,甚至对于一些测试用例两种方法求解时间相近.

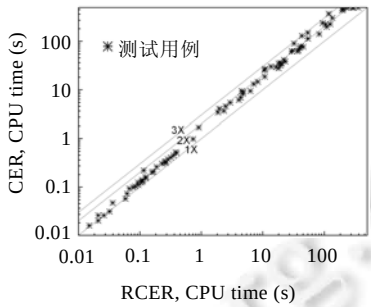


Fig.3 Experimental comparisons of RCER and CER

图 3 RCER 方法与 CER 方法实验结果比较

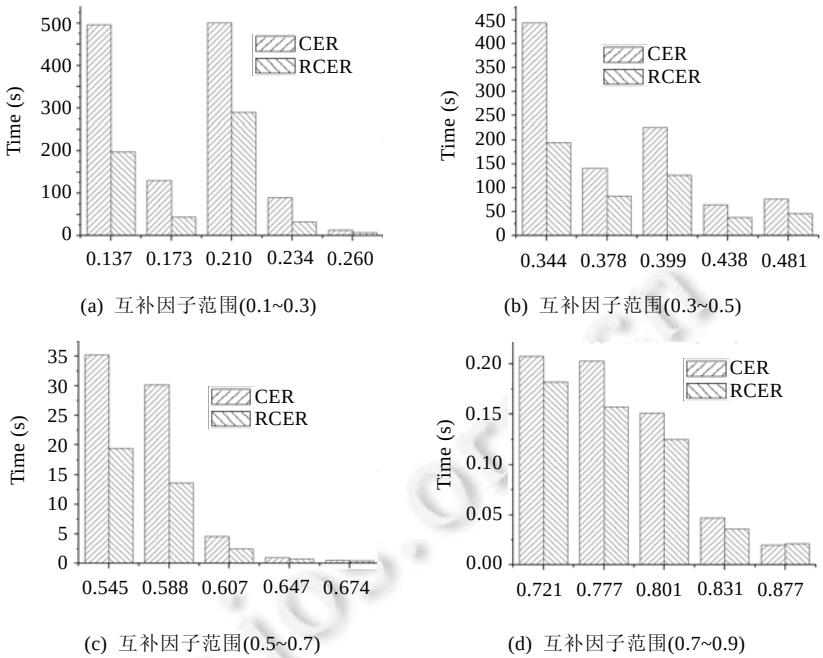
RCER 方法与 CER 方法的效率受互补因子影响较大,当互补因子较高时,公式(1)和公式(2)中存在较多的计算项值为 0,需要计算的项就相对较少,从而两种方法的效率就越高.为了更好地测试这两种方法的性能与互补因子的关系,下面给出互补因子由低到高变化时,两种方法对应实验的对比结果,如图 4 所示.

从图 4 可以看出:

- 随着互补因子的提高,RCER 方法与 CER 方法的求解时间都急速下降;并且在互补因子较低时,RCER 方法求解效率明显高于 CER 方法;
- 随着互补因子的增大,两方法的求解时间差逐渐缩小;且当互补因子大于 0.6 时,两个方法的求解时间差开始明显缩小;
- 互补因子更大时,RCER 方法的求解时间还要略高于 CER 方法.

当互补因子较高时,子句集中互补的子句对相对就较多,从而公式(1)和公式(2)中值为 0 的计算项就较多,需要计算的项就较少.这样,在 RCER 方法中可以利用的极大项相交集之间的扩展关系较少,所以此时两种方法的求解时间相差不大.由于需要对当前子句集中的文字覆盖情况进行记录,所以有时效率会略低于 CER 方法.值得注意的是:虽然互补因子较高时,相对于 CER 方法,RCER 方法并没有显著提高,但是在这种情况下,CER 方法的求解时间已经很少,算法效率的提升空间已经很小;并且在这种情况下,RCER 方法的求解效率也依然较高.

从图 4(a)中容易发现:对于互补因子较低的测试用例,算法的求解时间相对并不是很多.原因在于:算法实现中加入了单元子句规则,这里用变量在子句中出现的概率和其正负文字出现的概率来控制互补因子大小,因此,当一个子句集互补因子较小时,其中可能包含较多的单元子句,再通过单元子句规则对子句集进行处理,就可以减少子句集规模,从而使得求解时间减少.



(a) 互补因子范围(0.1~0.3) (b) 互补因子范围(0.3~0.5)

(c) 互补因子范围(0.5~0.7) (d) 互补因子范围(0.7~0.9)

Fig.4 Experimental comparisons of the two methods based on diverse complementary factors

图 4 不同互补因子下两种方法实验结果比较

下面给出两个算法对图 4 中各测试用例的求解时间比,见表 1.

Table 1 Time ratio (s/s)

表 1 时间比(s/s)

测试用例	互补因子	时间比(CER/RCER)	测试用例	互补因子	时间比(CER/RCER)
1	0.137	2.506	11	0.545	1.822
2	0.173	2.991	12	0.588	2.214
3	0.210	1.729	13	0.607	1.824
4	0.234	2.768	14	0.647	1.334
5	0.260	1.829	15	0.674	1.278
6	0.344	2.279	16	0.721	1.143
7	0.378	1.695	17	0.777	1.293
8	0.399	1.792	18	0.801	1.208
9	0.438	1.696	19	0.831	1.306
10	0.481	1.677	20	0.877	0.952

从表 1 可以看出:CER 方法求解时间与 RCER 方法求解时间的比值随着互补因子的增大在逐渐减小,并在互补因子 0.6 之后减小趋势更加明显.即:随着互补因子的增大,RCER 方法比 CER 方法的效率提升越来越不明显.可以想象:在极端情况下,当子句集互补因子为 1 时,即,子句集中任意两个子句都是互补的情况,公式(1)和公式(2)中需要计算的项仅有 n 个单相交集组成的计算项,此时的计算项间都不存在扩展关系,所以 RCER 方法与 CER 方法的计算过程是一样的.然而,RCER 方法还需对每个项中文字覆盖情况进行记录,因此,CER 方法的效率要高于 RCER 方法.虽然这种情况非常少,但可以依据这种趋势来理解随着互补因子的增大,RCER 方法的优势逐渐减小的原因.

5 结束语

#SAT 问题计算 CNF 公式的模型个数,是 SAT 问题的一种扩展问题,在人工智能领域具有广泛应用,如可以在多项式时间内与贝叶斯推理问题进行转化.在对扩展规则在 SAT 及#SAT 求解问题中的应用进行深入研究和

分析后,对已有的基于扩展规则的模型计数求解方法 CER 进行改进.CER 方法在求解时没有充分利用极大项相交集和其扩展极大项相交集之间的扩展关系来避免冗余计算.针对此问题,本文给出适合增量求解的 RCER 方法.此方法优先计算扩展极大项相交集对应极大项个数,并重用极大项相交集计算结果;对广义互补子句集对应的所有扩展极大项相交集进行剪枝,有效避免了计算所有极大项相交集对应极大项个数时的冗余求解;优先判定扩展极大项相交集互补关系,并重用极大项相交集互补结果的互补判定增量方法,较好地避免了判断子句间和各极大项相交集的基础子句集互补关系时的重复计算.实验结果表明:两种方法在互补因子较高的问题中具有较快的求解速度;并且在互补因子较低的问题中,RCER 算法的求解效率明显优于 CER 算法,效率提升 2~3 倍.

致谢 本文作者对所有的匿名审稿人的辛勤工作和给予本文宝贵的意见表示真诚的感谢.

References:

[1] Monasson R, Zecchina R, Kirkpatrick S, Selman B, Troyansky L. Determining computational complexity from characteristic 'phase transitions'. Nature, 1999,400(6740):133−137. [doi: 10.1038/22055]

[2] Mezard M, Parisi G, Zecchina R. Analytic and algorithmic solution of random satisfiability problems. Science, 2002,297(5582): 812−815. [doi: 10.1126/science.1073287]

[3] Zhang LT, Madigan CF, Moskewicz MH, Malik S. Efficient conflict driven learning in a Boolean satisfiability solver. In: Pileggi LT, Kuehlmann A, eds. Proc. of the 2001 IEEE/ACM Int'l Conf. on Computer-Aided Design (ICCAD 2001). Piscataway: IEEE Press, 2001. 279−285. [doi: 10.1109/ICCAD.2001.968634]

[4] Pan XY, Jiao LC, Liu F. A multi-agent social evolutionary algorithm for SAT problem. Chinese Journal of Computers, 2014,37(9): 2011−2020 (in Chinese with English abstract).

[5] Kautz H, Selman B. Planning as satisfiability. In: Neumann B, ed. Proc. of the 10th European Conf. on Artificial Intelligence (ECAI'92). Vienna: John Wiley and Sons, 1992. 359−363.

[6] Kautz H, Selman B. Unifying SAT-based and graph-based planning. In: Dean T, ed. Proc. of the 16th Int'l Joint Conf. on Artificial Intelligence (IJCAI'99). Stockholm: Morgan Kaufmann Publishers, 1999. 318−325.

[7] Kautz H. Deconstructing planning as satisfiability. In: Veloso M, Kambhampati S, eds. Proc. of the 21st National Conf. on Artificial Intelligence (AAAI 2006). Menlo Park: AAAI Press, 2006. 1524−1526.

[8] Grastien A, Anbulagan A. Diagnosis of discrete event systems using satisfiability algorithms: A theoretical and empirical study. IEEE Trans. on Automatic Control, 2013,58(12):3070−3083. [doi: 10.1109/TAC.2013.2275892]

[9] Cook SA. The complexity of theorem-proving procedures. In: Harrison MA, Banerji RB, Ullman JD, eds. Proc. of the 3rd Annual ACM Symp. on Theory of Computing. New York: ACM Press, 1971. 151−158. [doi: 10.1145/800157.805047]

[10] Valiant LG. The complexity of computing the permanent. Theoretical Computer Science, 1979,8(2):189−201. [doi: 10.1016/0304-3975(79)90044-6]

[11] Roth D. On the hardness of approximate reasoning. Artificial Intelligence, 1996,82(1):273−302. [doi: 10.1016/0004-3702(94)00092-1]

[12] Majercik SM, Littman ML. Contingent planning under uncertainty via stochastic satisfiability. Artificial Intelligence, 2003, 147(1-2):119−162. [doi: 10.1016/S0004-3702(02)00379-X]

[13] Dubois O. Counting the number of solutions for instances of satisfiability. Theoretical Computer Science, 1991,81(1):49−64. [doi: 10.1016/0304-3975(91)90315-S]

[14] Zhang WH. Number of models and satisfiability of sets of clauses. Theoretical Computer Science, 1996,155(1):277−288. [doi: 10.1016/0304-3975(95)00144-1]

[15] Birnbaum E, Lozinskii EL. The good old Davis-Putnam procedure helps counting models. Journal of Artificial Intelligence Research, 1999,10(1):457−477. [doi: 10.1613/jair.601]

[16] Bayardo RJ, Pehoushek JD. Counting models using connected components. In: Kautz H, Porter B, eds. Proc. of the 17th National Conf. on Artificial Intelligence (AAAI 2000). Menlo Park: AAAI Press, 2000. 157−162.

[17] Sang T, Bacchus F, Beame P, Kautz H, Pitassi T. Combining component caching and clause learning for effective model counting. In: Hans KB, Zhao XS, eds. Proc. of the SAT 2004. Berlin, Heidelberg: Spinger-Verlag, 2004. 20−28.

[18] Sang T, Beame P, Kautz H. Heuristics for fast exact model counting. In: Bacchus F, Walsh T, eds. Proc. of the SAT 2005. Berlin, Heidelberg: Springer-Verlag, 2005. 226−240. [doi: 10.1007/11499107_17]

[19] Thurley M. SharpSAT-Counting models with advanced component caching and implicit BCP. In: Biere A, Gomes CP, eds. Proc. of the SAT 2006. Berlin, Heidelberg: Springer-Verlag, 2006. 424−429. [doi: 10.1007/11814948_38]

[20] Lin H, Sun JG, Zhang YM. Theorem proving based on the extension rule. Journal of Automated Reasoning, 2003,31(1):11−21. [doi: 10.1023/A:1027339205632]

[21] Wu X, Sun JG, Lin H, Feng SS. Modal extension rule. Progress in Natural Science, 2005,15(6):550−558. [doi: 10.1080/10020070512331342540]

[22] Lai Y, Ouyang DT, Cai DB, Lü S. Model counting and planning using extension rule model counting and planning using extension rule. Journal of Computer Research and Development, 2009,46(3):459−469 (in Chinese with English abstract).

[23] Lin H, Sun JG. Knowledge compilation using the extension rule. Journal of Automated Reasoning, 2004,32(2):93−102. [doi: 10.1023/B:JARS.0000029959.45572.44]

[24] Sun JG, Li Y, Zhu XJ, Lü S. A novel theorem proving algorithm based on extension rule. Journal of Computer Research and Development, 2009,46(1):9−14 (in Chinese with English abstract).

[25] Zhang LM, Ouyang DT, Bai HT. Theorem proving algorithm based on semi-extension rule. Journal of Computer Research and Development, 2010,47(9):1522−1529 (in Chinese with English abstract).

[26] Li Y, Sun JG, Wu X, Zhu XJ. Extension rule algorithms based on IMOM and IBOHM heuristics strategies. Ruan Jian Xue Bao/Journal of Software, 2009,20(6):1521−1527 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3420.htm [doi: 10.3724/SP.J.1001.2009.03420]

[27] Yin MH, Lin H, Sun JG. Solving #SAT using extension rules. Ruan Jian Xue Bao/Journal of Software, 2009,20(7):1714−1725 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3320.htm [doi: 10.3724/SP.J.1001.2009.03320]

[28] Biere A, Heule M, van Maaren H, Walsh T. Handbook of Satisfiability. Amsterdam: IOS Press, 2009,3-4:633−634.

[29] Sun JG, Yang FJ, Ouyang DT, Li ZS. Discrete Math. Beijing: Higher Education Press, 2002. 4−5 (in Chinese).

附中文参考文献:

[4] 潘晓英,焦李成,刘芳.求解 SAT 问题的多智能体社会进化算法.计算机学报,2014,37(9):2011−2020. [doi: 10.3724/SP.J.1016.2014.02011]

[22] 赖永,欧阳丹彤,蔡敦波,吕帅.基于扩展规则的模型计数与智能规划方法.计算机研究与发展,2009,46(3):459−469.

[24] 孙吉贵,李莹,朱兴军,吕帅.一种新的基于扩展规则的定理证明算法.计算机研究与发展,2009,46(1):9−14.

[25] 张立明,欧阳丹彤,白洪涛.基于半扩展规则的定理证明方法.计算机研究与发展,2010,47(9):1522−1529.

[26] 李莹,孙吉贵,吴瑕,朱兴军.基于 IMOM 和 IBOHM 启发式策略的扩展规则算法.软件学报,2009,20(6):1521−1527. http://www.jos.org.cn/1000-9825/3420.htm [doi: 10.3724/SP.J.1001.2009.03420]

[27] 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. http://www.jos.org.cn/1000-9825/3320.htm [doi: 10.3724/SP.J.1001.2009.03320]

[29] 孙吉贵,杨凤杰,欧阳丹彤,李占山.离散数学.北京:高等教育出版社,2002.4−5.

贾凤雨(1991−),男,北京人,硕士生,主要研究领域为基于模型诊断.

张立明(1980−),男,博士,主要研究领域为基于模型诊断,模型验证,自动定理证明.

欧阳丹彤(1968−),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为基于模型诊断,模型验证,自动定理证明.

刘思光(1988−),男,硕士生,主要研究领域为基于模型诊断.