

基于加权平均值的一种分支启发式方法

胡忠雪 徐扬 胡容

(西南交通大学数学学院 成都 610031)

摘要 在可满足性(Satisfiability, SAT)问题算法中提出有效的分支策略可以提高求解器的效率,文中主要从冲突分析的角度出发,依据变量是否发生冲突和冲突的次数,提出一种基于加权平均值的分支启发式方法。该方法首先采用一组序列来记录变量是否参与冲突;其次赋予一个加权平均函数,依据变量的序列和决策层求出函数值;最后选择具有最大的函数值变量赋值,执行实例分析比较。由于该方法是对控制编码方法的改进,因此在进行例子分析时,采用了比较法和分析法,同时分析比较了所提方法、SUM(Sum in experiment)策略和 ACIDS(Average Conflict-index Decision Score)策略。对 SATLIB(SAT Little Information Bank)中的实例进行分析,结果表明所提方法能够实现更多子句被满足或最新冲突子句优先满足。

关键词 序列,加权平均,决策层,冲突次数,分支启发式

中图分类号 TP181 **文献标识码** A

Branching Heuristic Method Based on Added Weight Average Value

HU Zhong-xue XU Yang HU Rong

(School of Mathematics, Southwest Jiaotong University, Chengdu 610031, China)

Abstract The effective branching strategy was proposed in the satisfiability (SAT) problem algorithm, which can improve the efficiency of the solver. In consideration of the conflict, according to whether the variables join in conflict and conflict times, a branching heuristic method based on added weight average value was proposed. Firstly, a sequence is used to record whether the variables are involved in the conflict. Secondly, a weighted average function is given, the function value is calculated through variable sequences and decision levels. Finally, variable with the largest value is assigned, example analysis and comparison are conducted. The new method is an improved method according to the control encoding method, so in the example analysis, the comparison and analysis ways were used, and SUM (Sum in experiment) strategy and ACIDS(Average Conflict-index Decision Score) strategy were compared with the new method at the same time. Through analysis of examples in SATLIB (SAT little information bank), the results show that more clauses are satisfied or the latest conflict clause can get the priority by the new method.

Keywords Sequence, Weighted average, Decision layer, Times of conflict, Branching heuristic

1 引言

命题可满足性(Satisfiability SAT)问题^[1]为:判断是否存在一组变量赋值使得命题演算中的公式是可满足的。SAT问题被广泛地应用在人工智能以及正式认证等多方面, SAT算法在这些领域上取得了卓越成就,解决了诸多问题,在科学领域应用上也占据优势,对实际生活问题的解决起到突破性作用^[2]。SAT算法主要分为两大类:完备算法和不完备算法。倘若问题是可满足的,能保证找到一组可满足真值赋值,倘若命题是不可满足,却可以证明不可满足,那么这个算法被称为完备算法;倘若命题是可满足的,那么就能快速找到解,然而不可满足命题无法证明,那么称这个算法是不完备算法。JW(Jeroslow-Wang)算法、DP(Davis-Putnam)算法、DPLL(Davis-Putnam-Logemann-Loveland)算法^[3]、CDCL(Conflict-Driven-Clause-Learning)算法等是基本的完备算法^[4]。在解

决有关非确定性多项式(Non-deterministic Polynomial, NP)完全问题之一的 SAT 问题时,这些算法提高了效率。

现在主流的 SAT 算法是 CDCL 算法,CDCL 算法的主要过程有:初始化、变量决策、布尔约束传播、非时序性回溯、随机重启。变量决策作为 CDCL 算法中的基本技术,有助于变量赋值时有依据地赋值,并且在遇到冲突时使冲突分析后的变量赋值能避免相同的搜索空间。具有突破卓越贡献的变量决策策略是 VSIDS (Variable State Independent Decaying Score)策略,本文方法就是受到 VSIDS 的相关策略——SUM (Sum in experiment)策略和 ACIDS(Average Conflict-index Decision Score)策略而提出的有关变量选择的方法。

在每次布尔约束传播前,必须进行变量选择,依据分支启发式策略对选择的变量赋予相应的值,因此变量选择是主要因素,如何选择下一个变量的分支启发式(或者分支规则)在计算方面将产生重大影响,在一个基本相同的算法中采用不

本文受国家自然科学基金项目:基于矛盾体分离的动态自动演绎推理研究(61673320)资助。

胡忠雪(1991—),女,硕士生,主要研究方向为人工智能和自动推理, E-mail: 1690213008@qq.com;徐扬(1956—),男,博士,教授,主要研究方向为一阶逻辑、人工智能和自动推理;胡容(1992—),女,硕士生,主要研究方向为人工智能和自动推理。

同的分支策略在计算方面也可能产生不同的结果。早前的一些分支启发式策略主要是为了满足绝大多数的子句。在进行布尔约束传播过程中遇到冲突时,就必须重新选择变量赋值,所以为了避免发生同样的冲突,冲突分析就显得尤为重要^[5-6]。目前,从成功的现实工业例子来看,分支启发式的方法取得了突破性的进展,尤其是有关驱使冲突子句的研究。在 Chaff 中首次提出的 VSIDS 策略是一个突破性的策略^[7-8],在这个策略基础上学者们不断探究,继而基于该策略提出了许多新策略,如 NVSIDS(Normalized VSIDS)和 ACIDS 等,这些策略不断地改进,使得许多问题得以解决。

本文第 2 节介绍与方法相关的基础知识;第 3 节介绍几个相关策略,主要有 VSIDS 策略、SUM 策略和 ACIDS 策略;第 4 节介绍加权平均值,以及主要方法的具体步骤;第 5 节介绍实际例子与分析;最后总结全文。

2 相关基础知识

以下给出命题逻辑和分支启发式^[9]的相关知识。

目前主要有两大类形式的 SAT 问题,分别是公式型和集合型。在多项式时间内公式型与集合型的 SAT 问题可以相互转换。这里用 T 表示有限个变量的集合,故 $T = \{p_1, p_2, p_3, p_4, \dots, p_n\}$, $\forall p_i \in T$ 是命题逻辑中的变量,也是需要被赋值的对象。每个变量在 SAT 问题中存在两种文字形式。对于 $\forall p_i$,存在 p_i 与 $\neg p_i$ 两种文字, p_i 是变量的正文字, $\neg p_i$ 是变量的负文字,这两种文字是相反的,即 $x = p_i$ 当 $\neg x = \neg p_i$; $x = \neg p_i$ 当 $\neg x = p_i$ 。 $\forall p_i$ 可以赋予值 0 或者 1,也就是 $p_i = 0$ 或者 $p_i = 1$,然而正负文字的赋值情况是相反的,当 $p_i = 0$ 时, $\neg p_i = 1$ 。

把若干个变量的析取称为子句,子句一般用拉丁文字的大写字母来表示,如 $C = p_1 \vee \neg p_2 \vee p_3 \vee p_4 \vee \dots \vee p_m$,有时也可以写成集合形式 $C = \{p_1, \neg p_2, p_3, \dots, p_m\}$ 。子句中的变量以它的文字出现,这里任意的文字为 0,那么 $C = 0$,倘若存在某个文字的值 1,那么 $C = 1$ 。 $C = 0$ 说明这个子句不可满足,相反, $C = 1$ 说明这个子句可满足。若一个子句中有一个变量的正负文字都出现,则这个子句一定满足,好比 $A = x \vee y \vee \neg x \vee z$,则 $A = 1$ 。在子句中没有任何变量的子句被称为空子句,用 \square 表示,在命题逻辑中空子句是不可满足的,但是空集被规定为可满足。

有限个子句的合取就是命题逻辑中所说的公式,公式的形式为: $f = A_1 \wedge A_2 \wedge A_3 \wedge \dots \wedge A_n$,对于公式中的子句至少有一个不可满足,整个公式也就不可满足,存在 $A_i = 0$,那么 $f = 0$ 。故要使公式满足,只有 $\forall A_i = 1$ 时, $f = 1$ 。将公式中每个变量的赋值过程视为一个映射的过程, $T \rightarrow \{0, 1\}^n$,每个变量都有两种赋值的可能性,所以一个含有 n 个变量的公式有 2^n 组赋值,判断是否可满足的过程就是在这些赋值中搜索是否有一组值得使得公式满足。

因为从 $(0, 0, 0, \dots, 0)$ 到 $(1, 1, 1, \dots, 1)$ 的 2^n 种情况中探索答案是一个巨大的工程,所以变量决策方法就显得十分重要。良好的方法不仅可以在比较小的空间中搜索答案,而且能够迅速地找到答案,因此变量决策方法在整个算法中具有重要作用。

在命题逻辑中还有另一种公式表示型,即只显示出数字,每组数字代表一个子句,例如:

```
F cnf 8 8
-1 2 3 0
4 -6 8 0
1 2 0
1 7 8 0
-1 7 5 4 0
1 -4 7 0
-3 6 7 8 0
%
0
```

公式用 F 代表, cnf 指代合取范式,第一个 8 指代 8 个变量,第 2 个 8 就是 8 个子句,结束用 % 和 0 来代表。后面例子即采用这种表示法。

把决策的深度称为决策层,一般将一个变量的决策层和赋值结合表示,中间用一个连接符号,譬如: $x = a@d$, a 表示变量的赋值, $@$ 就是连接赋值, d 是决策层。一组变量的决策层图用二叉树来表示,下面用一个简单的图示(见图 1)来解释决策层。

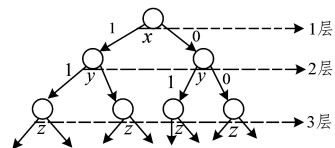


图 1 决策层的展示

第一层也可以定为 0 或者负数,但树越深,层数就越大。

3 相关策略

自基于 2001 年的 Chaff 算法提出具有卓越贡献的分支选择策略——VSIDS 后,改进的策略就被不断提出,如 NVSIDS 策略、ACIDS 策略等^[10],这些策略大大提高了求解器的运算速度,并且还能解决更多的问题。下面介绍几个策略的具体过程。

3.1 VSIDS 策略

解决 SAT 问题的分支策略——VSIDS 策略首次出现在 2001 年由 Malik 等提出的 Chaff 算法中,该策略在解决 SAT 问题时具有巨大贡献,具体步骤如下:

(1) 馈赠给每个变量一个计分器,记下变量的分值,初始值设为 0。

(2) 当任意的一个子句添加到子句库时,子句中所含变量的得分值就增加 1;选取具有最高分值却没有赋值的变量赋值,对于分值相同且都没赋值的变量,随机选取一个。

(3) 当进行布尔约束传播发生冲突时,进行冲突分析,把分析学习得到的子句添加到子句库中,对与冲突(冲突也称为碰撞)相关的变量分值增加 1,而没发生碰撞的变量的得分值保持不变。

(4) 以 256 为周期,定期将每个变得分值除以 2,即衰退过程。

3.2 SUM 策略

SUM 策略是提出得比较早的策略,这是一个关于冲突后的变量决策。在 SUM 策略中,对于每个发生了冲突碰撞的变量,在原来的分数基础之上增加冲突次数分值。如果没有发生任何碰撞,变量的分值保持不变,即每次发生碰撞后变量

得分的更新规则如下:

- (1) $s' = s + i$ (i 为冲突次数,参与冲突的变量分值更新);
- (2) $s' = s$ (不参与冲突)。

3.3 ACIDS 策略

依据 SUM 策略延展,在冲突后给予一定的分值权重,因此 Armin 等于 2016 年提出 ACIDS 策略,这个策略是把冲突之后的变量的分值更新为:

- (1) $s' = \frac{s+i}{2}$ (冲突相关变量的分值更新);
- (2) $s' = s$ (与冲突无关变量)。

以上这些策略不仅在解决 SAT 问题中起到了重大作用,而且为后面一系列的改进策略提供了帮助。

4 新策略

陈进超等于 2014 年提出了 Bit-encoding 策略,这个策略是计算参与冲突的每一个变量在决策层上的权重^[11-13],计算公式为:

$$n = b_k 2^k + b_{k-1} 2^{k-1} + \dots + b_1 2 + b_0$$

这个公式计算了变量 x 在第 n 次搜索时的分值,其中 b_k 是第 k 个决策层。若把这个多项式函数看作是一个权重函数,则这个计算分值的方法就与冲突无关。当发生冲突时,这个新策略依据变量决策层的不同改变其分值。

用函数 $f(x)$ 表示某组序列中 x 的权重,那么整组序列的权重平均值为:

$$W = \frac{f(x_1) + f(x_2) + \dots + f(x_n)}{n} = \frac{\sum_{i=1}^n f(x_i)}{n}$$

称 W 为加权平均值,具体计算方法如下。

设某组序列为 (1, 5, 2, 4, 3, 7), 权重函数为 $f(a) = 1 + a^2$, 那么这组序列的加权平均值为:

$$\begin{aligned} W &= \frac{f(1) + f(5) + f(2) + f(4) + f(3) + f(7)}{6} \\ &= \frac{2 + 26 + 5 + 17 + 10 + 50}{6} \\ &\approx 18.3 \end{aligned}$$

在新策略中用 1 表示发生了碰撞, 0 表示没有发生碰撞, 继而用一组序列数记录变量是否发生碰撞, 例如以下这组子句:

```
F cnf m n
1 - 2 0
- 1 - 3 0
...
```

假设在进行布尔约束传播的过程中所发生的 6 次冲突中, 变量 x_1, x_2 和 x_3 是否参与冲突的情况可以根据记录的序列得知, x_1 的序列为 (1, 0, 1, 0, 0, 1); x_2 的序列为 (0, 1, 0, 1, 1, 0); x_3 的序列为 (1, 1, 0, 0, 1, 1)。

在进行决策变量赋值时, 依据变量所得到的分值依次赋值, 不同的策略有不同的选择方法, 这里依据子句库中每个变量所对应的一组序列值的一个权重函数值对每个序列按照该权重函数计算相应的权重值, 此函数为:

$$f(r_i) = r_i d_i$$

其中, r_i 是序列中的第 i 个值; d 是这组序列相应的变量在没发生任何冲突前的冲突层, i 是冲突次数, 那么变量的权重就是它所对应的序列中每个值的权重之和^[14-16]。在上面的例

子中变量 x_1 (设变量 x_1 的决策层为 1) 的权重为:

$$\begin{aligned} W(x_1) &= 1 \times 1 \times 1 + 0 \times 1 \times 2 + 1 \times 1 \times 3 + 0 \times 1 \times 4 + 0 \times 1 \times 5 + 1 \times 1 \times 6 \\ &= 1 + 3 + 6 \\ &= 10 \end{aligned}$$

新策略的具体步骤如下。

(1) 对于整个子句集, 在没有发生任何冲突时, 赠予每个变量奖励值, 其方法与 VSIDS 策略相同。

(2) 当发生冲突时, 让每个变量都得到一个序列组, 记录每次发生冲突时变量是否与冲突相关的情况, 变量 $R(y_m)$ 的序列为:

$$R(y_m) = (r_1, r_2, \dots, r_i, \dots, r_n), r_i \in (0, 1)$$

其中, r_i 是第 i 次变量是否发生冲突的情况, 0 代表变量 y_m 没有参与冲突, 1 代表参与冲突。

一旦有冲突发生, 每个变量被赋予一个权重:

$$\overline{W}(y_m) = \frac{\sum_{i=1}^n f(r_i)}{n} = \frac{\sum_{i=1}^n r_i d_i}{n}$$

其中, d 是布尔约束传播在没有发生任何冲突之前变量所在的决策层, i 是冲突次数。

(3) 冲突首次发生后每个变量的得分值更新为:

$$s' = \frac{s_0}{2} + \overline{W}(y_m)$$

即

$$s' = \frac{s_0}{2} + \frac{\sum_{i=1}^n r_i d_i}{n}$$

其中, s_0 是 SAT 问题的子句集首次冲突发生之前变量的分值。

(4) 其他步骤与 VSIDS 一致。

新策略只改变冲突分析过程的变量得分值, 将这种加权法命名为加权平均法。权重函数不是直接赋在变量上, 而是通过序列值间接产生一个权重函数。在 ACIDS 策略中用到的平均值是分值与冲突次数的平均值, 而该方法采用的是函数平均值。

5 例子分析

在每次冲突分析之后, 依据变量是否参与最新冲突来更新变量的分值。这个方法主要是为了能优先满足最新学习子句和尽量降低随机赋值的可能性。采用 SATLIB(SAT Little Information Bank) 的例子进行分析, 基于 SUM 策略、ACIDS 策略和新方法得到以下结果。

```
F cnf m n
- 1 - 2 0
- 1 - 3 0
- 2 - 3 0
1 2 3 0
- 4 - 5 0
...
```

在这个子句集中变量 y_1 的决策层 $d=3$, 变量 y_2 的决策层 $d=2$, 变量 y_3 的决策层 $d=2$, 变量 y_4 的决策层 $d=2$, 变量 y_5 的决策层 $d=4$ 。在冲突发生之前, 各个变量的积分为: $s_0(y_1) = 10$; $s_0(y_2) = 16$; $s_0(y_3) = 16$; $s_0(y_4) = 12$; $s_0(y_5) = 10$ 。

经过6次冲突后,变量各自的序列为: $R^1=(1,0,1,0,0,1)$; $R^2=(0,1,0,1,1,0)$; $R^3=(1,1,0,0,1,1)$; $R^4=(1,1,0,0,1,1)$; $R^5=(1,0,1,0,0,1)$ 。

采用新方法将这5个变量首次发生冲突后的6次得分的更新情况列入表1—表3。这里将 y_1 与 y_5 、 y_2 与 y_3 、 y_3 与 y_4 放在一起列成表格,以便比较分析结果。

在表1中,变量 y_1 与变量 y_5 的参与冲突的情况一致,但是 y_5 的冲突层大于 y_1 的冲突层,有冲突参与时, y_5 的分值就会大于 y_1 的分值,这样在重新选择变量赋值时依据分值大者优先选则的原则, y_5 就容易被选择,因此初始值(首次冲突发生前的分值)和冲突相同的情况下最深一层的变量容易被选择。

表1 变量 y_1 与变量 y_5 用新方法计算的结果

y_1 是否参与冲突	y_1 的得分	y_5 是否参与冲突	y_5 的得分
是(1)	8.0	是(1)	9.00
否(0)	6.5	否(0)	7.00
是(1)	9.0	是(1)	10.33
否(0)	8.0	否(0)	9.00
否(0)	7.4	否(0)	8.20
是(1)	10.0	是(1)	11.67

在表2中,变量 y_3 与变量 y_4 除首次冲突前的分值 s_0 不一样,其他均一样, y_3 的 s_0 大于 y_4 ,所以它更易于先被赋值,冲突前的分值就是变量在子句中出现的个数之和,其使得更多子句满足。

表2 变量 y_3 与变量 y_4 用新方法计算的结果

y_3 是否参与冲突	y_3 的得分	y_4 是否参与冲突	y_4 的得分
是(1)	10.00	是(1)	8.00
是(1)	11.00	是(1)	9.00
否(0)	10.00	否(0)	8.00
否(0)	9.50	否(0)	7.50
是(1)	11.2	是(1)	9.20
是(1)	12.67	是(1)	10.67

在表3中,两个变量的冲突情况不一样,相当于同一变量在不同冲突情况时的分值,6次冲突过程中错位看冲突情况,每次最新冲突的分值会高于之前冲突的分值,这样就降低了前面冲突对变量碰撞次数的影响,可优先满足最新冲突。

表3 变量 y_3 与变量 y_2 用新方法计算的结果

y_3 是否参与冲突	y_3 的得分	y_2 是否参与冲突	y_2 的得分
是(1)	<u>10.00</u>	否(0)	8.00
是(1)	11.00	是(1)	<u>10.00</u>
否(0)	10.00	否(0)	9.33
否(0)	9.50	是(1)	<u>11.00</u>
是(1)	<u>11.20</u>	是(1)	<u>12.40</u>
是(1)	<u>12.67</u>	否(0)	13.67

SUM策略与新方法一样,主要是为了能满足最新冲突和降低赋值随机性,优先满足最新学习子句和缩小搜索空间,其计算结果如表4—表6所列。

表4 变量 y_1 与变量 y_5 用SUM策略计算的结果

y_1 是否参与冲突	y_1 的得分	y_5 是否参与冲突	y_5 的得分
是	11	是	11
否	11	否	11
是	14	是	14
否	14	否	14
否	14	否	14
是	20	是	20

表5 变量 y_3 与变量 y_4 用SUM策略计算的结果

y_3 是否参与冲突	y_3 的得分	y_4 是否参与冲突	y_4 的得分
是	17	是	13
是	19	是	15
否	19	否	15
否	19	否	15
是	24	是	20
是	30	是	26

表6 变量 y_3 与变量 y_2 用SUM策略计算的结果

y_3 是否参与冲突	y_3 的得分	y_2 是否参与冲突	y_2 的得分
是	<u>17</u>	否	16
是	<u>19</u>	是	<u>18</u>
否	19	否	18
否	19	是	<u>22</u>
是	<u>24</u>	是	<u>27</u>
是	<u>30</u>	否	27

ACIDS策略也为最新冲突赋予更大权重,以降低较早冲突对变量碰撞次数的影响。用上面变量 y_2 与变量 y_3 的数据,得到表7的计算结果。

表7 变量 y_3 与变量 y_2 用ACIDS策略计算的结果

y_3 是否参与冲突	y_3 的得分	y_2 是否参与冲突	y_2 的得分
是	<u>8.500</u>	否	16.000
是	<u>5.250</u>	是	9.000
否	5.250	否	9.000
否	5.250	是	<u>6.500</u>
是	<u>4.625</u>	是	<u>5.625</u>
是	<u>5.813</u>	否	5.625

结束语 采用本文提出的新方法可以使更多子句被满足或最新冲突子句被优先满足。本文提出了一个分值策略的方法。该方法主要是通过平均权重函数来计算得分,但是此方法还没有嵌入到CDCL的计算程序中实现操作运算。因此,后期计划设计程序并将这个新方法嵌入到CDCL算法中,以实现计算机操作。

参考文献

- [1] 田奕,刘涛.求解可满足性问题的一种高效遗传算法[J].模式识别与人工智能,1996,9(3):209-212.
- [2] WU G F, XU Y, CHANG W, et al. Parallel Genetic Algorithm for SAT Problem Based on the Coarse-grained Model [C]// Proceedings of the 12th International FLINS Conference on Uncertainty Modelling in Knowledge Engineering and Decision Making. City Council of Roubaix; World Scientific Publishing Co. Pte Ltd, 2016: 489-495.
- [3] 陈稳.基于DPLL的SAT算法的研究及应用[D].成都:电子科技大学,2011.
- [4] JIANG C, ZHANG T. Partial Backtracking in CDCL Solvers [C]// Proceedings of the 19th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning. Berlin Heidelberg; Springer Verlag, 2013: 490-502.
- [5] MARQUESSILVA J P, SAKALLAH K A. GRASP: A Search Algorithm for Propositional Satisfiability[J]. IEEE Transactions on Computers, 1999, 48(5): 506-521.
- [6] 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. Digest of Technical Papers; IEEE Press, 2003: 73-89.

结束语 通过对船舶航行轨迹预测进行研究,结合船舶 AIS 信息的特点,构建基于 AIS 信息的船舶航行轨迹表达模型,提出结合 AIS 信息和 RNN-LSTM 模型的船舶航行轨迹预测方法。运用提出的方法对目标水域内上百条船舶航行轨迹数据进行实验,利用训练好的 RNN-LSTM 模型对后续船舶轨迹特征进行预测。通过对预测值和原有的真实航行数据进行误差对比分析,得到随时间发展变化,该 RNN-LSTM 模型能对船舶的经度、纬度、航速及航向等船舶轨迹特征值进行有效的学习和辨识,从而对未来船舶航行轨迹进行精确、实时的预测。目前在船舶轨迹的预测上,基于时间序列的深度学习 RNN 模型在实际使用中还存在诸多不足,例如连续预测多个时刻的轨迹点还需要递归形式上的多次预测,因此下一步工作将深入研究序列预测(sequence to sequence),即可以在此基础上考虑使用 RNN 的自动编码解码器(AutoEncoder)等模型。

参 考 文 献

- [1] 徐铁,蔡春君,胡勤友,等. 基于卡尔曼滤波算法船舶 AIS 轨迹估计研究[J]. 现代电子技术,2014(5):97-100.
- [2] 赵帅兵,唐诚,梁山,等. 基于改进卡尔曼滤波的控制河段船舶航迹预测[J]. 计算机应用,2012,32(11):3247-3250.
- [3] PERERA L P, OLIVEIRA P, SOARES C G. Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction[J]. IEEE Transactions on Intelligent Transportation Systems,2012,13(3):1188-1200.
- [4] 王艳锋,李红祥. 桥区水域失控船舶的航迹预测[J]. 武汉船舶职业技术学院学报,2011,10(4):36-38.
- [5] 刘锡铃,阮群生,龚子强. 船舶航行 GPS 定位轨迹的新预测模型[J]. 江南大学学报(自然科学版),2014,13(6):686-692.
- [6] 茅晨昊,潘晨,尹波,等. 基于高斯过程回归的船舶航行轨迹预测[J]. 科技创新与应用,2017(31):28-29.
- [7] TONG X, CHEN X, SANG L, et al. Vessel trajectory prediction in curving channel of inland river[C]//International Conference on Transportation Information and Safety. IEEE, 2015: 706-714.
- [8] 徐婷婷,柳晓鸣,杨鑫. 基于 BP 神经网络的船舶航迹实时预测[J]. 大连海事大学学报,2012,38(1):9-11.
- [9] 甄荣,金永兴,胡勤友,等. 基于 AIS 信息和 BP 神经网络的船舶航行行为预测[J]. 中国航海,2017,40(2):6-10.
- [10] 孙志远,鲁成祥,史忠植,等. 深度学习研究与进展[J]. 计算机科学,2016,43(2):1-8.
- [11] GRAVES A, MOHAMED A R, HINTON G. Speech recognition with deep recurrent neural networks[C]//IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013:6645-6649.
- [12] GRAVES A. Supervised Sequence Labelling with Recurrent Neural Networks[M]. Springer Berlin Heidelberg,2012.
- [13] 张庆庆,贺兴时. BP 神经网络结点数选取的改进方法及其应用[J]. 西安工程大学学报,2008,22(4):502-505.
- [14] 谭伟,陆百川,黄美灵. 神经网络结合遗传算法用于航迹预测[J]. 重庆交通大学学报(自然科学版),2010,29(1):147-150.
- [15] CHEN J. A Bit-Encoding Phase Selection Strategy for Satisfiability Solvers[C]//Proceedings of the 11th Annual Conference on Theory and Applications of Models of Computation. New York, Springer Verlag,2014:158-167.
- [16] PIPATSRISAWAT K, DARWICHE A. A Lightweight Component Caching Scheme for Satisfiability Solvers[C]//Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing. Lisbon Portugal; Springer Verlag,2007:294-299.
- [17] 刘志明,吴明芬,许勇. 一种基于遗传算法的权重的确定方法[J]. 五邑大学学报(自然科学版),2006,20(3):45-48.
- [18] LIANG J H, GANESH V, POUPART P, et al. Exponential Recency Weighted Average Branching Heuristic for SAT Solvers[C]//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Infosys; AAAI Press,2016:3434-3440.
- [19] SILVA J, MARQUES O P. The Impact of Branching Heuristics in Propositional Satisfiability Algorithms[C]//Proceedings of the 9th Portuguese Conference on Artificial Intelligence. Berlin Heidelberg; Springer Verlag,1999:62-74.
- [20] LIU Y, WANG J, XU C, et al. An Effective Branching Strategy Based on Structural Relationship among Multiple Forbidden Induced Subgraphs [J]. Journal of Combinatorial Optimization, 2015,29(1):257-275.

(上接第 120 页)

- [7] MOSKEWICZ M W, MADIAN C F, ZHAO Y, et al. Chaff: Engineering an Efficient SAT Solver[C]//Proceedings of the 38th Design Automation Conference. United States: IEEE Press, 2005:530-535.
- [8] LIANG J H, GANESH V, ZULKOSKI E, et al. Understanding VSIDS Branching Heuristics in Conflict-Driven Clause-Learning SAT Solvers[C]//Proceedings of the 11th International Haifa Verification Conference on Verification and Testing of Hardware and Software. Haifa (Israel); Springer International Publishing,2015:225-241.
- [9] DERSHOWITZ N, HANNA Z, NADEL A. A Clause-based Heuristic for SAT Solvers[C]//Proceedings of the 8th International Conference on Theory and Applications of Satisfiability Testing. Intel Corporation; Springer-Verlag,2005:46-60.
- [10] BIÈRE A, FROHLICH A. Evaluating CDCL Variable Scoring Schemes[C]//Proceedings of the 18th International Conference on Theory and Applications of Satisfiability Testing. Bioinformatics; Springer International Publishing,2015:405-422.
- [11] CHEN J. A Dynamic Phase Selection Strategy for Satisfiability Solvers [J]. Computer Science,2012,13(5):208-213.
- [12] CHEN J. Phase Selection Heuristics for Satisfiability Solvers [J]. Computer Science,2011,36(6):312-320.