

# 基于子句权重学习的求解 SAT 问题的遗传算法

凌应标 吴向军 姜云飞

(中山大学信息科学与技术学院 广州 510275)

**摘 要** 该文提出了一种求解 SAT 问题的改进遗传算法(SAT-WAGA), SAT-WAGA 算法有多个改进性特点:将 SAT 问题的结构信息量化为子句权重,增加了学习算子和判定早熟参数,学习算子能根据求解过程中的动态信息对子句权重进行调整,以便防止遗传进程的早熟,同时,算法还采用了最优染色体保存策略,防止进化过程的发散. 该文最后描述了实现包括 SAT-WAGA 等多个算法的实验系统,对选择最佳早熟判定参数值给出了一些有效的建议. 实验结果表明:与一般遗传算法相比, SAT-WAGA 算法在求解速度、成功率和求解问题的规模等方面都有明显的改善.

**关键词** SAT 问题;遗传算法;子句权重;早熟

中图法分类号 TP301

## Genetic Algorithm for Solving SAT Problems Based on Learning Clause Weights

LING Ying-Biao WU Xiang-Jun JIANG Yun-Fei

(School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275)

**Abstract** A novel genetic algorithm, SAT-WAGA, is proposed for solving SAT problems based on learning clause weights in this paper. Several new characteristics of the algorithm are innovative. The new algorithm makes use of the heuristic information from the structure of SAT problems by clause weights. A new operator of learning clause weights is designed to prevent precocity in the process of solving problems. This operator adapts the weights of clauses according to a criteria condition. A criteria parameter for detecting precocity is defined. The strategy of keeping the best chromosomes guarantees the property of convergence in the evolution iteration. To demonstrate the feasibility of the new algorithm, an experiment system of several famous algorithms is implemented. The experiment works focus on comparing the total span of all plateaus in evolution iteration, the success rates and the total time of the new algorithm to a classical genetic algorithm by solving several groups of various scales of random generated SAT problem instances. The appropriate values of the precocity criteria parameter of the new algorithm are also tested and presented. The experimental results show that the SAT-WAGA performs remarkably better than a classical genetic algorithm in the aspects of speed, the success rate and the solvable problem scales.

**Keywords** SAT problem; genetic algorithm; clause weight; precocity

收稿日期:2003-06-12;修改稿收到日期:2005-02-28. 本课题得到国家教育部博士点基金项目“智能规划及其应用研究”和中山大学重点建设高水平大学专项资金资助. 凌应标,男,1965年生,博士研究生,主要研究方向为智能规划、演化计算和智能优化. E-mail: isslyb@zsu.edu.cn. 吴向军,男,1965年生,副教授,主要研究方向为人工智能、算法设计和网络应用. 姜云飞,男,1945年生,教授,博士生导师,主要研究方向为自动推理、智能规划和基于模型诊断等.

## 1 引言

可满足性(SAT)问题是命题逻辑中的一个经典问题,也是计算机科学理论与应用的一个核心问题.研究解决 SAT 问题的有效算法不仅具有重大的理论意义,而且在智能规划、智能决策、定理证明、电路诊断和优化计算等诸多领域有着实际的意义.

在 SAT 问题的算法方面有完全算法和不完全算法两大类.前者的主要成果有 DP 算法<sup>[1]</sup>、归结法等.完全算法理论上能保证找到解,但是,由于 SAT 问题是一个 NP 完全问题,这些算法在效率上还很难满足需要.

20 世纪 90 年代以来,有关 SAT 问题的算法研究热点转向了不完全算法研究.不完全算法虽然不能保证一定能找到解,但多数情况下其求解速度比完全算法快,实用性高.近年来,不完全算法的研究成果非常丰富,如 Selman 和 Kautz 的 GSAT 算法<sup>[2]</sup>和 WALKSAT 算法<sup>[3]</sup>、顾均的 SAT1.3 算法<sup>[4]</sup>、李未等的数学物理方法<sup>[5]</sup>、梁东敏等的 WALKSAT 改进算法<sup>[6]</sup>和张德富等的拟人退火算法<sup>[7]</sup>等等.

SAT 问题的遗传算法是一类重要的不完全算法,从 1989 年提出<sup>[8]</sup>以来一直有这方面的研究<sup>[9~14]</sup>,但总的来说,算法性能不尽人意.遗传算法的研究和大量的实验表明:利用启发信息是提高算法性能的有效途径,因此,发现和利用 SAT 问题自身的结构信息和求解过程中的动态信息是非常关键的.本文提出了一种求解 SAT 问题的改进性遗传算法,它把 SAT 问题的结构信息量化为子句权重,并根据求解过程中的动态信息进行子句权重学习操作,从而较好地解决遗传算法中的早熟问题.我们对使用该机制的改进性遗传算法进行了多方面的实验.实验结果表明:该机制明显改善算法在求解速度、成功率和求解问题规模等方面的性能.

## 2 SAT 问题的有关定义

SAT 问题可分为子句集型和公式型,后者可在多项时间内转化为前者,本文所说的 SAT 问题是指子句集型.下面给出文中所用的概念和符号.

**定义 1.** 用符号  $V$  表示一个命题变元的集合.若  $V$  由  $n$  个命题变元  $x_1, x_2, \dots, x_n$  组成,那么,  $V = \{x_1, x_2, \dots, x_n\}$ ,用  $|V|$  表示集合  $V$  的元素个数.

**定义 2.**  $V$  的一个真值指派  $\mathcal{A}$  是一个映射:  $V \rightarrow \{0, 1\}$ ,它可用一个  $n$  维向量  $(v_1, v_2, \dots, v_n)$  表示,其中  $v_i \in \{0, 1\}$ ,0 和 1 分别代表命题常元 False 和 True.

对任意一个命题变元  $x_i$ ,如果其在真值指派  $\mathcal{A}$  下取真,则  $\mathcal{A}(x_i) = 1$ ;否则  $\mathcal{A}(x_i) = 0$ .在  $V$  上,存在  $2^n$  个不同的真值指派,所有真值指派构成  $n$  维向量空间.

**定义 3.** 对任意一个命题变元  $x$ ,称符号  $x$  和  $\neg x$  是其文字,称  $x$  是正文字,  $\neg x$  是负文字.

**定义 4.** 子句是  $V$  上若干文字的析取,用集合  $c$  表示,子句  $c$  中的文字数称为子句长度,记为  $|c|$ .

**定义 5.** 子句集  $C$  是由  $V$  上的子句组成的集合.它在真值指派  $\mathcal{A}$  下是可满足的,当且仅当  $C$  中所有子句在真值指派  $\mathcal{A}$  下都是真的.

**定义 6.** SAT 问题定义为:给定命题变元的集合  $V$  和其子句集  $C$ ,问:是否存在一个关于  $V$  的真值指派  $\mathcal{A}$ ,使得  $C$  是可满足的.

## 3 SAT 问题的一般遗传算法

用遗传算法求解 SAT 问题,主要包括问题转换、染色体编码、遗传操作设计等三个方面.

问题转换的核心是定义适应值函数  $f$ ,将 SAT 问题转换为对适应值函数求极值的最优化问题.定义适应值函数  $f$  的方法很多,一种简单的定义是真值指派使公式取得的真值,其值域为  $\{0, 1\}$ ;还有定义为真值指派所不能满足的子句数<sup>[8]</sup>或所不能满足的子句权重之和<sup>[15]</sup>.这样,SAT 问题转换为适应值函数求极值的最优化问题.由于求解 SAT 问题的性能与适应值函数的空间形态密切相关,这个函数的空间形态不能提供真值指派中局部变更对应的梯度信息,不能引导寻优的选择操作.

用二进制串表示真值指派是最直观的染色体编码方法,这种编码方案充分利用了 SAT 本身的特点,便于计算适应值函数和设计各种遗传操作.对于  $n$  个变元的 SAT 问题,染色体用  $n$  位二进制串表示,它与由真值指派中变元的取值直接对应,可直接解释成真值指派.

遗传操作有三种:选择操作(select)、交叉操作(crossover)和变异操作(mutation).选择操作的作用是从当代个体中选择出群体,为繁殖下一代作准备.select 操作的步骤如下:

1. 对群体中的每个个体  $\mathcal{A}$ ,计算其适应值  $f(\mathcal{A})$ ,并计

算当代个体的总适应值  $F$ ;

2. 对群体中的每个个体  $A$ , 计算其选择概率  $p(A) = f(A)/F$ ;

3. 按选择概率大小(模拟滚花轮的方式)从当代群体中选择  $M$  个个体作为父代, 适应值大的个体被选中机会就多。

交叉操作的作用是模拟生物繁殖过程, 通过两个个体染色体的部分交换, 产生出两个新个体。由于染色体采用二进制编码, 交叉操作可通过二进制串的截断和拼接来实现。单点交叉操作可分以下 3 步:

1. 从群体中选择两个个体  $fa$  和  $ma$  作双亲, 设双亲  $fa$  和  $ma$  的染色体串分别为  $f_1 f_2 \dots f_n$  和  $m_1 m_2 \dots m_n$ , 其中  $f_i, m_i \in \{0, 1\} (1 \leq i \leq n)$ ;

2. 产生随机数  $r$  作为交叉点  $(1 \leq r \leq n)$ ;

3. 产生两个新个体, 它们的染色体串分别为  $f_1 f_2 \dots f_r m_{r+1} m_{r+2} \dots m_n$  和  $m_1 m_2 \dots m_r f_{r+1} f_{r+2} \dots f_n$ ;

变异操作是模拟生物进化时染色体基因的突变现象, 该种操作是按一定的概率对染色体的每位进行翻转, 即  $0 \rightarrow 1, 1 \rightarrow 0$ 。

下面给出求解 SAT 问题的一般遗传算法 (SAT-GA) 描述。

1.  $popu(0) \leftarrow$  随机生成  $m$  个定长二进制位串  $\{a_1, a_2, \dots, a_m\}$  为第 0 代的群体,  $n \leftarrow 0$ ;
2. while (进化终止条件不满足) do
  - 2.1.  $popux \leftarrow$  select( $popu(n)$ ); /\* 选择操作, 产生父代  $m$  个体存放在  $popux$  \*/
  - 2.2.  $popuy \leftarrow$  crossover( $popux$ ); /\* 交叉操作, 由父代  $popux$  产生  $m$  个新个体临时存放在  $popuy$  \*/
  - 2.3.  $popu(n+1) \leftarrow$  mutation( $popuy$ ); /\* 变异操作, 对  $m$  个新个体进行基因变异, 产生下一代  $m$  个体存放在  $popu(n+1)$  \*/
  - 2.4.  $n \leftarrow n+1$ ; /\* 累计进化代数 \*/
3. endwhile

说明:  $popu(n)$  为第  $n$  代的群体,  $m$  为群体规模, 进化终止条件是  $n > MG$  (即最大进化代数), 或找到满足子句集的真值指派。

## 4 基于子句权重学习的遗传算法

### 4.1 启发信息及其量化

在 SAT 问题的研究中, 一些研究者提出了如何利用 SAT 问题的结构信息来构造求解过程中的启发信息, 如文献[16]用子句加权改进 GSAT 算法, 文献[5]用子句动态加权改进 GSAT, 文献[15]用子句动态加权作为约束强弱的量化改进 GA 算法, 用于求解 3-SAT 问题, 文献[5]用学习临时权重的方法改进 GSAT, 文献[17]也提出了基于单元传播的

启发式信息的方法。

求解 SAT 问题的启发信息有两类: 源于子句结构的静态信息和源于求解过程的动态信息。

静态信息存在于子句长短不一的 SAT 问题中。子句越短, 满足该子句的候选真值指派就越少, 所以, 在一组真值指派中, 满足短子句多者有利于进化为最终解。由于子句长度是 SAT 问题自身存在的, 是引导进化搜索的一种重要的启发信息, 我们把它量化为子句权重的初值。

另外, 还有两种静态信息: 出现在单文字子句中的变元和纯文字形式出现的变元, 其取值是可简单地确定下来的, 然后在子句集中传播, 这样可确定其它一些变元的取值, 并将其表示为染色体的特定的模式, 在遗传过程中, 对这些特定的模式加以保护, 以提高求解性能。

动态启发信息是在求解过程中动态发现的。在求解过程中, 如果搜索陷入局部极值区(最高适应值连续多代不增加), 那么, 在连续多代的最高适应值的真值指派中不能满足的子句通常是同一组子句, 这时, 需要改变一些变元的取值, 才能跳出局部极值区。这种现象可通过子句权重的学习操作来克服。

### 4.2 子句权重设置与学习机制

静态信息量化为子句权重的初值。静态权重定义为子句长度的非线性函数, 此函数满足: 短子句的权重数值大, 长子句的权重数值小。相应地, 适应值函数定义为真值指派所满足的子句的权重之和。

在求解 CSP 问题时, 根据变元可能取值的个数从少到多进行排序, 按此顺序对变元指派值可提高求解的效率。本质上来说, 遗传算法也是通过搜索来求解。SAT 问题是一种 CSP 问题, 这种做法对 SAT 问题的求解同样有效。但是由于 SAT 问题的特殊性, 我们用子句的权重来构成了变元指派值的搜索顺序, 子句权重的分布就是这种搜索顺序的一种反映, 子句权重的学习机制就是获得有利于搜索的子句权重分布。

为了便于理解, 下面用一个简单的例子说明子句权重分布如何构成变元指派值的搜索顺序。设  $V = \{x_1, x_2, x_3\}$ , 子句集见表 1 的第一列。对任意一个子句  $c$ , 其子句权重  $W(c) = (MAXLEN - |c|)^2$  (其中  $MAXLEN$  是最长子句的长度 + 1)。

从表 1 可以看出, 不同的变元指派可满足不同的子句集, 如  $x_2 = 1$  可满足 4 个子句, 该 4 个子句权重之和为 30,  $x_4 = 1$  可满足 2 个子句, 这些子句权重合计为 2。根据适应值函数的定义和选择操作的作

用, 包含  $x_2 = 1$  的真值指派在搜索过程中, 比其它变元的指派先固定下来的可能性要大, 如此类推. 这样, 子句权重的分布构成的搜索顺序为  $x_2 = 1, x_1 = 0, x_3 = 0, x_4 = 0$ . 这组变元指派正好是一个解. 一般情况下, 遗传搜索过程具有一定的随机性, 这种搜索顺序非确定性地引导搜索过程.

子句权重构成的变元指派的搜索顺序不一定是惟一的, 前面的初值设置方法也不能保证正确, 随着

搜索的进行, 其偏差可能误导搜索, 因此, 需根据搜索过程进行调整, 这就是增加子句权重学习方法的出发点. 我们研究发现: 算法陷于局部极值区域的原因是当时未满足的子句权重偏小, 误导了搜索过程, 因此, 应适度增加相应子句的权重, 学习机制应增加难满足的子句的权重, 以提高其被满足的可能性, 这样可重新构造变元指派的搜索顺序, 从而引导搜索在新的区域进行.

表 1 子句权重分布与变元指派的搜索顺序的关系

| 子句                                 | $W(c)$ | $x_1=0$ | $x_1=1$ | $x_2=0$ | $x_2=1$ | $x_3=0$ | $x_3=1$ | $x_4=0$ | $x_4=1$ |
|------------------------------------|--------|---------|---------|---------|---------|---------|---------|---------|---------|
| $\{x_2\}$                          | 16     | 0       | 0       | 0       | 16      | 0       | 0       | 0       | 0       |
| $\{\neg x_1, \neg x_2\}$           | 9      | 9       | 0       | 9       | 0       | 0       | 0       | 0       | 0       |
| $\{\neg x_1, x_2\}$                | 9      | 9       | 0       | 0       | 9       | 0       | 0       | 0       | 0       |
| $\{x_1, \neg x_3\}$                | 9      | 0       | 9       | 0       | 0       | 9       | 0       | 0       | 0       |
| $\{\neg x_1, x_2, x_3\}$           | 4      | 4       | 0       | 0       | 4       | 0       | 4       | 0       | 0       |
| $\{x_1, \neg x_2, \neg x_3\}$      | 4      | 0       | 4       | 4       | 0       | 4       | 0       | 0       | 0       |
| $\{x_1, \neg x_3, \neg x_4\}$      | 4      | 0       | 4       | 0       | 0       | 4       | 0       | 4       | 0       |
| $\{\neg x_2, x_3, \neg x_4\}$      | 4      | 0       | 0       | 4       | 0       | 0       | 4       | 4       | 0       |
| $\{x_1, x_2, x_3, x_4\}$           | 1      | 0       | 1       | 0       | 1       | 0       | 1       | 0       | 1       |
| $\{\neg x_1, \neg x_2, x_3, x_4\}$ | 1      | 1       | 0       | 1       | 0       | 0       | 1       | 0       | 1       |
| 合计                                 | 61     | 23      | 18      | 18      | 30      | 17      | 10      | 8       | 2       |
| 搜索顺序                               |        | 2       |         |         | 1       | 3       |         | 4       |         |

学习机制的触发时机是算法陷于局部极值区域时刻, 即当搜索出现平台后, 子句权重学习一般会持续若干代, 随着这些子句的权重增加, 满足它们的真值指派将变为最高适应值的真值指派, 并被选择操作选择出来, 从而引导搜索跳出局部极值区域并进入新的区域.

子句权重学习机制  $WeightAdapting()$  的描述如下:

#### 1. Repeat

对任意一个子句  $c \in$  子句集, 做:

如果  $NotTrueByTheBestChromosome(c)$ , 即当代最优真值指派不能使子句  $c$  满足, 则  $w_{n+1}(c) \leftarrow w_n(c) + \Delta w$ , 即以  $\Delta w$  增加子句  $c$  的权重,  $\Delta w = k \cdot w_n(c)$ , 它是子句  $c$  当前权重  $w_n(c)$  的线性函数;

#### 2. Until 处理了所有子句.

### 4.3 早熟与发散问题的解决

早熟现象是遗传算法中常见的问题之一, 是影响算法性能的一大因素. 遗传算法中的早熟现象(平台现象)是指搜索陷于一个局部极值区域进化, 表现为在连续若干代的最优个体适应值保持不增, 虽然有时能跳出, 但耗费大量的时间, 甚至导致算法失败结束. 在算法中当发现早熟现象时, 启动子句权重学习机制, 加快搜索过程跳出局部极值区域.

遗传算法中还存在发散问题, 发散是群体失去了进化方向, 表现在最优个体的适应值上下波动, 其

对应的解的质量时好时坏. 导致发散原因是多方面的, 如适应值函数的空间形态不好、变异概率偏大, 遗传操作的设计不当等等. 我们通过改进选择操作, 较好地克服了发散问题. 方法是保存最优个体染色体, 即仅当新一代最优个体  $d$  的适应值大于当前一代中的最优个体  $D$  的适应值时, 用由  $d$  取代  $D$  作为下一代的最优个体, 否则, 仍以  $D$  作为下一代的最优个体. 这样可保证最优个体的适应值是非递降的.

### 4.4 SAT-WAGA 算法的描述

采用子句权重学习机制后, 求解 SAT 问题的遗传算法(SAT-WAGA)描述如下:

- $popu(0) \leftarrow$  随机生成  $m$  个定长二进制位串  $\{a_1, a_2, \dots, a_m\}$  为第 0 代的群体;
- while(进化终止条件不满足) do
  - $popux \leftarrow select(popu(n));$  /\* 最优个体染色体保存的选择操作, 产生父代  $m$  个体存放在  $popux$ , 其中第一个是最优个体的染色体 \*/
  - 如果早熟判定条件为真, 则调用  $WeightAdapting()$  进行子句权重学习;
  - $popuy \leftarrow crossover(popux);$  /\* 交叉操作, 由父代  $popux$  产生  $m$  个新个体临时存放在  $popuy$ , 其中第一个是父代  $popux$  的最优个体的染色体 \*/
  - $popu(n+1) \leftarrow mutation(popuy);$  /\* 变异操作, 对  $M-1$  个新个体进行基因变异, 产生下一代  $M-1$  个体存放在  $popu(n+1)$ , 其中第一个是父代

$popu$  的最优个体的染色体 \*/

2.5.  $n \leftarrow n+1$ ; /\* 进化代数累计 \*/

3. endwhile

## 5 实验结果与比较

目前,一般的实验方法是求解随机生成的可满足测试样本,获得不同算法的成功率和运行时间,再比较算法的性能.根据 Mitchell 的研究<sup>[18]</sup>,随机生成的最难求解的 3-SAT 测试样本是子句数目与变元数目之比为 4.3,为此,我们编制了测试样本生成程序,生成了可满足的 3-SAT 测试样本和结构化测试样本.

每 10 个测试样本存放在一个文件中,测试样本文件名可以与测试样本的规格对应,例如,文件名 Ex128\_3 表示样本是 3-SAT 型,变元数目为 128,子句数目是 4.3X128,而文件名 X1024\_10 表示样本是结构化型,子句文字最多 10 个,变元数目 1024,子句数目是 2.1X1024,其它如此类推.

SAT-GA 是一般遗传算法,SAT-WAGA 是基于子句权重学习机制的遗传算法.算法采用默认参数(除非特别注明):群体数目  $M$  为 10,交叉概率  $P_c$  为 55%,变异概率  $P_m$  为 0.1%,早熟平台参数  $Q_3$  为 100,最大进化代数  $MG$  为 10000,SAT-GA 没有参数  $Q_3$ .

实验环境:联想昭阳 PIII750/256MBRAM/

WIN98/VC++ 6.0.

每组求解 3 次,时间是 3 次各组总时间的平均值,单位:秒,成功率是成功求解例数/测试样本总例数 $\times 100$ .通过实验数据,我们得出的结果如下:

### (1) 学习机制克服早熟问题

表 2 统计了算法 SAT-WAGA 与 SAT-GA 求解相同三组测试样本的总代数和平台总代数,实验结果表明:前者明显比后者小,说明子句权重学习机制能有效地克服早熟问题.

表 2 解决早熟的效果对比

| 测试样本    | 总代数      |        | 平台总代数    |        |
|---------|----------|--------|----------|--------|
|         | SAT-WAGA | SAT-GA | SAT-WAGA | SAT-GA |
| Ex64_3  | 10844    | 77401  | 10086    | 76870  |
| Ex128_3 | 24060    | 181899 | 22589    | 172030 |
| Ex192_3 | 43559    | 269374 | 41166    | 265873 |

### (2) SAT-WAGA 算法参数 $Q_3$ 与性能的关系

SAT-WAGA 算法根据参数  $Q_3$  来判定搜索是否早熟,如果连续  $Q_3$  代,最优适应值保持不变,则算法认为进化出现早熟现象.它启动子句权重的学习机制,因此,专门做了一组实验,观察参数  $Q_3$  与算法性能的关系,结果如表 3 所示.

由表 3 不难看出,参数  $Q_3$  对算法 SAT-WAGA 的成功率和求解速度有明显的影响,对规模不同的问题,使算法性能最佳的参数  $Q_3$  值是不同的(表中加粗线部分为最佳性能),基本规律是:随着问题规模的增大,合适的  $Q_3$  值也相应地增大.

表 3 SAT-WAGA 算法参数  $Q_3$  与性能关系

| 测试样本    | 成功率        |            |           |            |           | 时间(s)     |            |           |             |           |
|---------|------------|------------|-----------|------------|-----------|-----------|------------|-----------|-------------|-----------|
|         | $Q_3=10$   | $Q_3=50$   | $Q_3=100$ | $Q_3=200$  | $Q_3=400$ | $Q_3=10$  | $Q_3=50$   | $Q_3=100$ | $Q_3=200$   | $Q_3=400$ |
| Ex128_3 | <b>100</b> | 100        | 100       | 100        | 100       | <b>65</b> | 99         | 97        | 133         | 175       |
| Ex256_3 | 100        | <b>100</b> | 100       | 100        | 100       | 772       | <b>583</b> | 822       | 849         | 1108      |
| Ex384_3 | 93.3       | 93.3       | 93.3      | <b>100</b> | 100       | 5566      | 3527       | 3497      | <b>3834</b> | 4054      |

(3) 算法 SAT-GA 与 SAT-WAGA 的性能比较  
在实验中,有一项内容是用一般遗传算法和 SAT-WAGA 求解多组测试样本,以比较两者的性能,部分实验结果见表 4.

由表 4 可以看出,当 SAT 问题的变元较多但有解时,一般遗传算法 SAT-GA 获得解的能力大为下降,甚至有时找不到解,但 SAT-WAGA 却一般来说总可以找到解,而且所用的时间也少了很多.

因为作者在文献<sup>[19]</sup>中介绍了 SAT-GA 与著名的 GSAT 算法和 WALKSAT 算法的比较结果,在求解多于 100 个变元的测试样本时,SAT-GA 算

法明显优于后两个算法,所以,间接可推知 SAT-WAGA 性能较 GSAT 和 WALKSAT 更好.

表 4 算法 SAT-GA 与 SAT-WAGA 的性能比较

| 测试样本     | 成功率    |          | 总时间(s) |          |
|----------|--------|----------|--------|----------|
|          | SAT-GA | SAT-WAGA | SAT-GA | SAT-WAGA |
| Ex64_3   | 93.3   | 100      | 122    | 17       |
| Ex128_3  | 80.0   | 100      | 783    | 101      |
| Ex256_3  | 0      | 100      | 1387   | 806      |
| Ex384_3  | 0      | 93.3*    | 4649   | 3497     |
| X1024_10 | -      | 100      | -      | 2207     |
| X2048_20 | -      | 100      | -      | 3466     |
| X4096_40 | -      | 100      | -      | 15735    |

\*注:此组在调整算法参数后仍可 100% 求解.

## 6 结束语

SAT-WAGA 算法能利用问题求解的启发信息,其学习机制较好地解决了早熟问题,具有优良求解速度和求解成功率,可实际用于求解大规模的 SAT 问题. 与其它方法相比, SAT-WAGA 算法具有实现简单、求解速度快、求解成功率高和能解决问题规模大的特点.

### 参 考 文 献

- 1 Davis M., Putnam H.. A computing procedure for quantification theory. *Journal of Association for Computing Machinery*, 1960, 7(3): 201~215
- 2 Selman B., Levesque H. J., Mitchell D.. A new method for solving hard satisfiability problem. In: *Proceedings of the AAAI-92*, Menlo Park, 1992, 440~446
- 3 Selman B., Kautz H., Cohen B.. Noise strategies for improving local search. In: *Proceedings of the AAAI-94*, Seattle, Washington, 1994, 337~343
- 4 Gu J.. Local search for satisfiability (SAT) problem. *IEEE Transactions on Systems, Man and Cybernetics*, 1993, 23(4): 1108~1128
- 5 Li Wei, Huang Wen-Qi. A mathematic-physical approach to the satisfiability problem. *Science in China, Series A*, 1994, 24(11): 1208~1217(in Chinese)  
(李 未, 黄文奇. 一种求解合取范式可满足性问题的数学物理方法. *中国科学, A 辑*, 1994, 24(11): 1208~1217)
- 6 Liang Dong-Min, Wu Hua, Ma Shao-Han. An efficient local search algorithm for solving structured SAT problems. *Chinese Journal of Computers*, 1998, 21(Supplement): 92~97(in Chinese)  
(梁东敏, 吴 晔, 马绍汉. 一个求解结构 SAT 问题的高效局部搜索算法. *计算机学报*, 1998, 21(增刊): 92~97)
- 7 Zhang De-Fu, Wang Wen-Qi, Wang Hou-Xiang. Personification annealing algorithm for solving SAT problems. *Chinese Journal of Computers*, 2002, 25(2): 148~152(in Chinese)  
(张德富, 黄文奇, 汪厚祥. 求解 SAT 问题的拟人退火算法. *计算机学报*, 2002, 25(2): 148~152)
- 8 de Jong K. A., Spears W. M.. Using genetic algorithms to solve NP-complete problems. In: *Proceedings of the 3rd International Conference on Genetic Algorithms*, 1989, 124~132

- 9 Zhang Bo, Zhang Ling. Good point set based genetic algorithm. *Chinese Journal of Computers*, 2001, 24(9): 917~1922(in Chinese)  
(张 钺, 张 铃. 佳点集遗传算法. *计算机学报*, 2001, 24(9): 917~1922)
- 10 Hao J. K., Dorne R.. An empirical comparison of two evolutionary methods for satisfiability problems. In: *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, 1994, 1: 450~455
- 11 Spears W. M.. Simulated annealing for hard satisfiability problems. NCARAI, Washington D. C., Technical Report #AIC-93-015, 1993
- 12 Young A. R., Reel A.. A hybrid genetic algorithm for a logic problem. In: *Proceedings of the 9th European Conference on Artificial Intelligence*, Stockholm, Sweden, 1990, 744~746
- 13 Folino G., Pizzuti C., Spezzano G.. Parallel hybrid method for SAT that couples genetic algorithms and local search. *IEEE Transactions on Evolutionary Computation*, 2001, 5(4): 323~334
- 14 Pedrycz W., Succi G., Shai O.. Genetic-fuzzy approach to the Boolean satisfiability problem. *IEEE Transactions on Evolutionary Computation*, 2002, 6(5): 519~525
- 15 Eiben A. E., van der Hauw J. K.. Solving 3-SAT by GAs adapting constraint weights. In: *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1997, 13~16, 81~86
- 16 Selman B., Kautz H.. Domain independent extensions to GSAT: Solving large satisfiability problem. In: *Proceedings of the IJCAI-93*, Chambéry, France, 1993, 290~295
- 17 Li Chu Min, Anbulagan. Heuristics based on unit propagation for satisfiability problem. In: *Proceedings of the IJCAI-97*, Nagoya, Japan, 1997, 366~371
- 18 Mitchell D., Selman B., Levesque H. J.. Hard and easy distributions of SAT problems. In: *Proceedings of the AAAI-92*, San Jose, California, 1992, 459~465
- 19 Ling Ying-Biao, Wu Xiang-Jun. An efficient genetic algorithm for SAT problems. In: *Proceedings of the National Conference on Theoretical Computer Science*, Qingdao, 2003, 167~171 (in Chinese)  
(凌应标, 吴向军. 可满足性问题的遗传算法. 见: 2003 年全国理论计算机科学学术年会论文集, 青岛, 2003, 167~171)
- 20 Selman B., Kautz H., Cohen B.. Local search strategies for satisfiability testing. In: *Proceedings of the DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 1996, 26: 521~536



**LING Ying-Biao**, born in 1965, Ph. D. candidate. His research interests include AI planning, evolution computation and intelligence optimization.

**WU Xiang-Jun**, born in 1965, associate professor. His research interests include artificial intelligence, algorithm design and network design.

**JIANG Yun-Fei**, born in 1945, professor, Ph. D. supervisor. His research interests include knowledge engineering AI planning and intelligence diagnosis.

## Background

This project is supported by the National Research Foundation for the Doctoral Program of Higher Education of China. The project title: The Research on AI Planning and Its Applications. The project includes three main tasks: (1) Propose efficient algorithms for AI planning, (2) Study strategies for domain knowledge utilization in AI planners and, (3) Integrate algorithms of scheduling in AI planning systems for practical domain applications.

Some related achievements of the research group are reached by now: A plan recognition algorithm based on plan

knowledge graph, a method of solving planning recognition problems based on constraints, a method of graphplan extension, a causality-based approach to indirect effect of planning action, a method of handling conditional effects based on components, and a formalism of planning with domain constraint based on model checking are proposed by Jiang Yun-fei, Ma Ning, Wu Kanghen and Zhu Man-Fei *et al.*.

The SAT planners are promising. The SAT solver is an engine of a SAT planner. The algorithm proposed in this paper will be used in a new SAT planner.