SAT solver using genetic programming

Nguyen Xuan Thang

CTU - FIT

nguyexu2@fit.cvut.cz

31. prosince 2022

1 Introduction

The goal of the work was to implement and study the options of implementing an efficient SAT solver using genetic programming. The program needs to find a binary evaluation of the variables, such that the given SAT formula is satisfied when using this evaluation. All formulas will be in CNF, meaning we have an NP-complete problem that we will try to solve using an iterative method.

2 Data

All data are taken from the SATLIB library. This library of different SAT formulas contains many hard and easy instances to test the solver. It also contains many instances that encode real-life problems and are not limited to only randomly generated data. This library is widely used to compare the strength of different SAT solvers and is often included in many SAT solving competitions.

All instances are encoded in DIMACs format, for this reason, no further preprocessing was needed on the data itself.

3 Methods

To solve the SAT instances, a genetic algorithm was implemented using C++. Each chromosome is defined as a binary bit vector, where each element represents the binary truth value of a variable. The population is then a set of such bit vectors and to select the next generation, a roulette wheel selection approach was chosen. The better individuals will have a larger probability of being chosen and the fitness of an individual is calculated as the number of clauses being solved using this evaluation.

To explore the search space, 2 operators were implemented. The binary operator takes 2 chromosomes and applies 1-point crossover. The mutation process randomly flips k-bit of the chromosome, where k is a parameter.

The main parameters that we will train are the population size and the number of generations allowed to search the search space. While having bigger values for these parameters, we can expect better results, but the time needed to find the solution will also be bigger. Another parameter that was tuned is the elitism rate and the mutation rate. A high elitism rate will enforce convergence toward the solution whereas a high mutation rate will allow more diversification.

4 Results

The parameteres were first tuned on very simple data to see the correlation between the size of population, number of generations compared to the time needed to find the solution.

As we can predict, with more allowed iterations and more diverse population, the solution gets better. Though after number of allowed generations, the improvement is not as significant.

Jakých výsledků bylo dosaženo, co na ně melo vliv. Srovnání s očekáváním, *diskuze nad výsledky* – zvláště důležitá v případě, že něco vyšlo *divně*.

5 Závěr

A variation of genetic algorithm was implemented for SAT. It yielded relatively good results, with the error being only around 5% . exact number

There is still some room for improvement regarding the implementation of the selection process. It would be interesting to see, if an island model would help with this process.

Reference