Understand Restart of SAT Solver Using Search Similarity Index  
(Student Abstract)

Yoichiro Iida\textsuperscript{1}, Tomohiro Sonobe\textsuperscript{2}, Mary Inaba\textsuperscript{1}
\begin{itemize}
\item \textsuperscript{1} Information Science and Technology, The University of Tokyo
\item \textsuperscript{2} National Institute of Informatics, Japan
\end{itemize}
yoichiro-iida@g.ecc.u-tokyo.ac.jp, tomohiro.sonobe@nii.ac.jp, mary@is.s.u-tokyo.ac.jp

Abstract

SAT solvers are widely used to solve many industrial problems because of their high performance, which is achieved by various heuristic methods. Understanding why these methods are effective is essential to improving them. One approach to this is analyzing them using qualitative measurements. In our previous study, we proposed \textit{search similarity index} (SSI), a metric to quantify the similarity between searches. SSI significantly improved the performance of the parallel SAT solver. Here, we apply SSI to analyze the effect of \textit{restart}, a key SAT solver technique. Experiments using SSI reveal the correlation between the difficulty of instances and the search change effect by restart, and the reason behind the effectiveness of the state-of-the-art restart method is also explained.

Introduction

SAT solvers, applications for solving SAT problems, particularly conflict-driven clause learning solvers, are widely used for industrial and academic purposes. SAT solvers are supported by various heuristic methods. Understanding the reasons for the effectiveness of these methods is essential to improve them. One approach is analyzing their behavior and characteristics with qualitative measurements. In our previous study, we proposed \textit{search similarity index} (SSI) (Iida, Sonobe, and Inaba 2022), a metric that quantifies the degree of similarity between searches. Using SSI, similar parallel searches are avoided, and the portfolio solver’s performance is significantly improved.

Here, we apply SSI to analyze the effect of \textit{restart} (Gomes, Selman, and Kautz 1998) —one of SAT solvers’ key techniques. The search consists of \textit{decisions}, \textit{propagation}, and \textit{backtrack}. Restart is the ultimate backtrack that deletes the entire decision history and resumes the search. Various restart methods have been proposed such as Luby (Luby, Sinclair, and Zuckerman 1993) and exponential moving average (EMA) (Biere and Fröhlich 2019). Restart is said to be effective as it changes the search by attempting different assignments. SSI quantifies the effect of restart, which is represented by differences in variable assignments.

Through experiments, we analyzed the effect of restart based on the difficulty of the instances and in comparison with existing restart methods. This study contributes to a better understanding of SAT problems and restart methods.

Methods

SSI quantifies the similarity between searches. A search is represented by the value-assignment plan in the \textit{decision}. The plan’s information set is the \textit{current search direction} \textit{(CSD)}. \textit{CSD} \textit{k} denotes the information at the \textit{k}th step while searching (herein the \textit{k}th restart). With two similar \textit{CSD}s, the searches are regarded as similar and vice versa. Thus, the SSI is calculated by a comparison between \textit{CSD} \textit{k} and \textit{CSD} \textit{j} of arbitrary \textit{i} and \textit{j}. The value assignment in the \textit{decision} can be decomposed to \textit{polarity} and \textit{priority} —"polarity" is the Boolean value and \textit{priority} is the variables’ assignment order. \textit{CSD} \textit{k} consists of \textit{polarity} \textit{v} and \textit{priority} \textit{v} \textit{s} t. \textit{v} \textit{∈} \textit{V} (all variables). \textit{polarity} \textit{v} is defined as \{\textit{True}, \textit{False}\}, and \textit{priority} \textit{v} is defined as \textit{(order of variable \textit{v})}/|\textit{V}|, where |\textit{V}| is the set of effective variables at the \textit{k}th restart.

SSI is defined as the weighted sum of the variables’ similarity, \textit{similarity} \textit{i,j} (\textit{v}), and it is defined as the multiplication of the similarity of polarity and priority of variable \textit{v}. The similarity of \textit{polarity} (\textit{v}) is defined as 1 if \textit{polarity} \textit{v} = \textit{polarity} \textit{v} \textit{a}; otherwise, it is 0. The similarity of \textit{priority} (\textit{v}) is defined as \textit{1 − |priority} \textit{v} \textit{a} − priority} \textit{v} \textit{j}|. \textit{Importance} is a weight factor. Higher \textit{importance} scores are assigned to higher \textit{importance} variables. \textit{importance} \textit{i,j} (\textit{v}) is defined as \textit{2 − priority} \textit{v} \textit{a} × C + 2 − priority} \textit{v} \textit{a} × C, where constant \textit{C}’s value is set at 0.1. To use SSI as a metric, its value is normalized to between 0 and 1 —0 represents zero similarity, whereas 1 represents identical searches. We normalize SSI by dividing the sum of \textit{importance}. \textit{SSI} \textit{k} is the average value of the set of \textit{SSI} \textit{a,a+k}, where \textit{a} denotes all intra-search restarts. As \textit{SSI} \textit{k} is the similarity between two searches \textit{k} restarts apart, it represents the average effect of \textit{k} restarts to change the search.

\begin{equation}
\text{SSI}_{i,j} := \frac{\sum \text{similarity}_{i,j}(v) \times \text{importance}_{i,j}(v)}{\sum \text{importance}_{i,j}(v)}
\end{equation}
Experiments

We used CaDiCaL (Biere et al. 2020) as the base solver; 400 industrial instances of SAT competition 2021 main track were used as the benchmark. The solver finished its search when it found the solution or it reached 5000s runtime.

Effect of Restart by Instances

The restart effectiveness is known to differ depending on the problem. We examined it using SSI. Figure 1 shows the variance of $SSI_{100}$ of each instance. The vertical axis represents $SSI_{100}$. The dots depict instances and are categorized based on the search results. SAT and UNSAT denote instances solved as satisfiable and unsatisfiable, respectively. Unsolved denotes instances for which the solver could not find a solution. The horizontal axis of the scatter plot represents the search runtime. The right-hand boxplot indicates the variance of SSI for each category.

Next, we analyzed the differences of the effect of three famous restart methods, uniform, Luby, and EMA. Uniform invokes a restart at a specific interval; in this experiment, it was after every 256 conflicts. Luby uses a sequence generated by the Luby algorithm as intervals, with an initial interval at 32 conflicts. EMA is a state-of-the-art restart method that invokes restart when the evaluation of learning is worse than the criteria. The more recently learning have more weight on the evaluation based on the concept of exponential moving average. We used the default setting of EMA on CaDiCaL.

Figure 2 shows the variance of $SSI_{k}$ of instances as a boxplot, where $k = \{1, 10, 100, 1000\}$, when comparing the three restart methods. The mean of $SSI_{1}$ of EMA is larger and its variance smaller than the others. However, the mean and variance of $SSI_{1000}$ of EMA are mostly equivalent to the others. Moreover, EMA achieves the effect of search change of 1000 restarts faster than others. As EMA performed more frequent restarts than the others (only nine conflicts on average were observed between two consecutive EMA restarts), EMA’s 1000 restarts required fewer conflicts than others (i.e., generally with less runtime). This explains at least one reason for the effectiveness of EMA.

Conclusion

We analyzed the effect of restart of SAT solvers using SSI. Experiments revealed that intensive search is essential to solving difficult instances. We also compared the effect of restart methods and revealed one of the reasons of the effectiveness of EMA restart, indicating the advantages of using SSI. SSI can also be used to analyze other SAT techniques such as clause sharing and decision heuristics in the future. Improving SAT solvers performance based on these insights using SSI is also a future project.

References


