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

Yoichiro Iida[1], **Tomohiro Sonobe**[2], **Mary Inaba**[1]

[1] Information Science and Technology, The University of Tokyo
[2] National Institute of Informatics, Japan
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 *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 *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 the *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 *restart* (Gomes, Selman, and Kautz 1998) —one of SAT solvers' key techniques. The search consists of *decisions*, *propagation*, and *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 *decision*. The plan's information set is the *current search direction* (CSD). CSD_i denotes the information at the ith step while searching (herein the ith restart). With two similar CSDs, the searches are regarded as similar and vice versa. Thus, the SSI is calculated by a comparison between CSD_i and CSD_j of arbitrary i and j. The value assignment in the *decision* can be decomposed to *polarity* and *priority* —*"polarity"* is the Boolean value and *priority* is the variables' assignment order. CSD_i consists of $polarity_i(v)$ and $priority_i(v)$ s.t. $v \in V$ (all variables). $polarity_i(v)$ is defined as $\{True, False\}$, and $priority_i(v)$ is defined as *(order of variable v)*$/|V_i|$, where $|V_i|$ is the set of effective variables at the ith restart.

SSI is defined as the weighted sum of the variables' similarity, $similarity_{i,j}(v)$, and it is defined as the multiplication of the similarity of polarity and priority of variable v. The similarity of $polarity(v)$ is defined as 1 if $polarity_i(v)$ = $polarity_j(v)$; otherwise, it is 0. The similarity of $priority(v)$ is defined as $1-|priority_i(v) - priority_j(v)|$. *Importance* is a weight factor. Higher *importance* scores are assigned to higher *priority* variables. $importance_{i,j}(v)$ is defined as $2^{-priority_i \times C} + 2^{-priority_j \times C}$, where constant 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 *importance*. $SSI_{|k|}$ is the average value of the set of $SSI_{a,a+k}$, where a denotes all intra-search restarts. As $SSI_{|k|}$ is the similarity between two searches k restarts apart, it represents the average effect of k restarts to change the search.

$$SSI_{i,j} := \frac{\sum_v \{similarity_{i,j}(v) \times importance_{i,j}(v)\}}{\sum_v (importance_{i,j}(v)} \quad (1)$$

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.

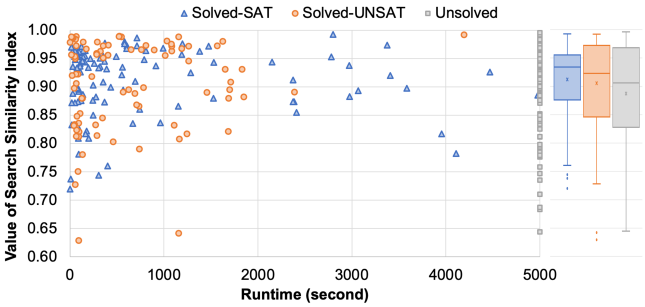

Figure 1: Variance of $SSI_{|100|}$ of each instance by instances

The scatter plot indicates widely dispersed SSI values from 0.7 to 1.0 for easy instances, which were rapidly solved. Conversely, as runtime increased, mostly higher SSI values were observed in difficult instances. The boxplot indicates that unsolved instances have the smallest means and largest variances. These results imply that a high similarity search —intensive search —is essential to solving difficult instances while an extensive search results in unsolved situations. This explains why gradually reducing the frequency of restarts over time in the search, the de-facto standard setting of SAT solvers, contributes to their performance. Furthermore, it suggests a new idea of a restart method that intensifies its search to solve more instances.

Effect of Restart by Various Methods

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

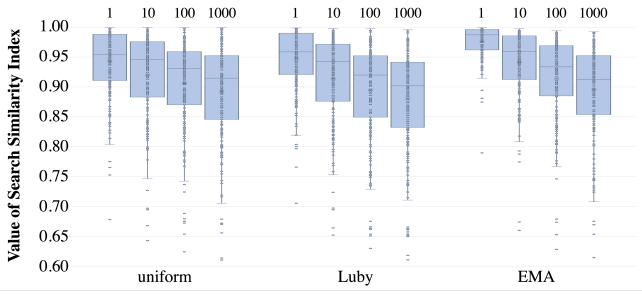

Figure 2: Variance of $SSI_{|k|}$ of instances by method

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

Biere, A.; Fazekas, K.; Fleury, M.; and Heisinger, M. 2020. CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling Entering the SAT Competition 2020. In *SAT Competition 2020 – Solver and Benchmark Descriptions*, 51–53. University of Helsinki.

Biere, A.; and Fröhlich, A. 2019. Evaluating CDCL Restart Schemes. In *Proceedings of Pragmatics of SAT 2015 and 2018*, 1–17. EasyChair.

Gomes, C. P.; Selman, B.; and Kautz, H. 1998. Boosting combinatorial search through randomization. In *15th national conference on artificial intelligence and 10th conference on Innovative applications of artificial intelligence*, 431–437. AAAI press.

Iida, Y.; Sonobe, T.; and Inaba, M. 2022. Diversification of Parallel Search of Portfolio SAT solver by Search Similarity Index. In *19th Pacific Rim International Conferences on Artificial Intelligence*. Springer International Publishing.

Luby, M.; Sinclair, A.; and Zuckerman, D. 1993. Optimal speedup of Las Vegas algorithms. *Information processing letters*, 47(4): 173–180.