

## SAT 問題における Community Rank の提案

## Proposal of Community Rank about SAT problems

戸ヶ崎 仁宣<sup>\*1</sup> 手塚 宏史<sup>\*1</sup> 稲葉 真理<sup>\*1</sup>  
 Hitoshi Togasaki Hiroshi Tezuka Mari Inaba

<sup>\*1</sup> 東京大学大学院 情報理工学研究所 創造情報学専攻

University of Tokyo Graduate School of Information Science and Technology Department of Creative Informatics

Satisfiability(SAT) Problem determines whether exists an assignment of variables that satisfies a given formula. A SAT Solver is a software that solves SAT problems. In order to solve SAT problems efficiently, we propose a new method using community structures of graphs induced from application SAT problems. By experiments, we show that our proposal method solves some problems which existing solvers can't solve.

## 1. はじめに

充足可能性問題 (SAT) とは与えられた命題論理式が充足可能 (SATISFIABLE) かまたは不可能 (UNSATISFIABLE) かを証明する問題であり, 初めて NP 完全性が証明された問題である。SAT 問題を高速に解くソフトウェアとして SAT ソルバが存在し, ソルバの性能を競うコンテストも開催されている。現在の SAT ソルバの多くには VSIDS[1], Clause 学習 [2], backjump[3] のような集中的な探索を行う機能が実装されている。また, 不毛な探索空間に陥るのを防ぐため restart のような探索のリセットを行うような機能が実装されている。並列に SAT 問題を解く研究も行われており, 現在では問題を分割する Divide and Conquer 方式ではなく, 各ソルバが並列に同じ問題を協力的かつ競争的に解く portfolio[4] と呼ばれる手法が有効とされており, その場合各ソルバの更なる多様性が重要となっている。

本稿では, ソルバの集中と多様性を獲得するために SAT 問題から導出される Variable Incidence Graph(VIG) のコミュニティに着目した。コミュニティ探索切り替え手法を提案し, また更なる多様性の確保のため PageRank を用いたコミュニティのランキング付けを新たに提案をした。実験検証を行い既存の手法で解くことができなかつた問題を解くことを確認した。

## 2. 関連研究

### 2.1 Dynamic Restart Strategy

Dynamic Restart Strategy とは gluecose[7] が採用している動的 restart 戦略である。gluecose では獲得した学習 Clause の Literal Block Distance(LBD)[7] スコアが 2 以下である学習 Clause を glueclause と呼ぶ。glueclause は重要であるとされ学習 Clause の削除の対象とならない。gluecose は LBD スコアの平均が大きくなるのを防ぐために (1) の式を満たした場合 restart を行う。

直近の LBD スコアの平均 (50 件) $\times 0.8 >$  全体の LBD スコア平均  
 (1)

### 2.2 Variable Incidence Graph

Variable Incidence Graph(VIG)[5] とは SAT 問題の変数を頂点とし, 同一 Clause 内の変数ペアをを重み付きのエッジで結ぶグラフである。Clause の集合を  $\Gamma$  とし, ある Clause を  $c \in \Gamma$  とすると, 任意の頂点のペア  $x, y \in c$  間のエッジの重みは  $w(x, y) = \sum_{x, y \in c} \frac{1}{\|c\|C_2}$  と定義される。任意の頂点ペア間のエッジの重みは, より多くの Clause にそのペアが含まれる場合, そしてその Clause が短ければ大きくなる。Application 問題の VIG に対してコミュニティ抽出アルゴリズムの Louvain 法 [6] を行うと, 高い modularity[10] が観測される事が判明している [5]。また近年, 学習 Clause の LBD スコアとコミュニティの数に強い相関があるということも判明している [9]。

## 3. Community Switching の提案

コミュニティと LBD の強い相関を利用して, さらなる探索の集中を実現することを目標として, 我々は VIG のコミュニティごとに優先度を上げ集中的な探索を行い, 不毛な探索空間に陥った場合, 優先するコミュニティを切り替える事により新たな探索空間の実現を目的とする Community Switching を提案する。

Community Switching の擬似コードを以下に示す。

```
1 loop {
2   if (restart % restart_interval == 0){
3     vig = create_vig();
4     communities = detect_community(vig);
5   }
6   target_community = next_community(communities);
7   bump_vsids(target_community, bump_ratio);
8
9   search();
10  restart++;
11 }
```

本稿では, 擬似コード内のパラメータを以下に設定した。

```
restart_interval = 3000
bump_ratio = 1000
```

コミュニティ抽出アルゴリズムは Louvain 法を用いた。またグラフの規模が大きくなるのを防ぐために Clause の長さが 30 以上のものは考慮しない。

連絡先: 戸ヶ崎 仁宣, 東京大学大学院 情報理工学研究所 創造情報学専攻, 〒113-8656 東京都文京区本郷 7-3-1 iref 棟 稲葉真理研究室 togasakitogato@gmail.com

## 4. Community Rank の提案

PageRank[8] とは, web ページの重要度の指標であり, PageRank が高い web ページとは他の web ページなどから多く参照されている。

Community Switching を行う際に, どのコミュニティに優先度を与えるかを決定するために PageRank を応用する事を検討した。PageRank の高いコミュニティはコミュニティ内は密にリンクし, また他のコミュニティと密にリンクしているため Propagation の対象に選ばれやすく, PageRank の低いコミュニティほど探索空間の対象にならないと考えた。探索の対象となりにくい探索空間の集中的な探索を可能とすれば, ソルバの新たな多様性を獲得できると考え, そこで我々は VIG からコミュニティ抽出を行い, それぞれのコミュニティを web ページと見立てコミュニティの関係を PageRank に適応して, PageRank の低いコミュニティほど Community Switching で優先的に探索を行う手法を提案する。本稿で用いるグラフは重み付き無向グラフであり, 重みを考慮した PageRank を以下のように拡張した。定数  $0 \leq d \leq 1$  とし, 対象とするグラフの頂点数を  $N$  とし, ある頂点  $A$  とリンクしている頂点の集合を  $V$  とした場合, 頂点を  $T \in V$  とすると,  $T$  から出ているエッジの総和を  $C(T)$  とする。  $T$  から  $A$  のエッジの重みを  $L(T)$  とする。頂点  $T$  の PageRank の値を  $PR(T)$  とした時に頂点  $A$  の PageRank の値  $PR(A)$  は以下と定義する。

$$PR(A) = \frac{(1-d)}{N} + d \left( \sum_{T \in V} \frac{L(T)}{C(T)} PR(T) \right) \quad (2)$$

Community Rank の擬似コードを以下に示す。

```

1 loop {
2   if (restart % restart_interval == 0){
3     vig = create_vig();
4     communities = detect_community(vig);
5     sort_by_community_rank(communities)
6   }
7   target_community = next_community(communities);
8   bump_vsids(target_community, bump_ratio);
9
10  search();
11  restart++;
12 }
```

本稿では, ページランクのパラメータを以下に設定した。

$d = 0.8$

## 5. 評価実験

### 5.1 実験内容

minisat2.20[11] にいくつかの既存手法と提案手法を実装し, 評価するために, 以下のソルバに対して SAT competition 2014 Application 部門の 300 問 (SAT 150 問 UNSAT 150 問) を下記の実験環境で実験を行った。

M:  
minisat2.20

M+L:  
minisat2.20+LBD (Literal Block Distance)

M+L+D:  
minisat2.20+LBD+DRS (Dynamic Restart Strategy)

M+L+D+CS:  
minisat2.20+LBD+DRS+CS (Community Switching)

M+L+D+CS+CR:  
minisat2.20+LBD+DRS+CS+CR (Community Rank)

### 実験環境

- CPU Intel(R) Core(TM) i7-3770K CPU @ 3.50GHz
- メモリ 32GB
- タイムアウト 5000 秒

### 5.2 実験結果

ソルバが制限時間内に解けた問題数の内訳を表 1 に示す。各ソルバが解けた問題の実行時間を昇順に sort し, 実行時間と解けた問題数との関係をプロットしたものをそれぞれ 図 1(SAT + UNSAT), 図 2(SAT), 図 3(UNSAT) に示す。M または M+L+D が解くことができなかった問題に対して M+L+D, M+L+D+CS+CR で解けた問題数の内訳を表 2 に示す。M+L+D+CS, M+L+D+CS+CR がそれぞれ解けなかった問題に対してそれぞれ M+L+D+CS, M+L+D+CS+CR が解けた問題数の表 3 に示す

表 3 より, M+L+D+CS と M+L+D+CS+CR とは UNSAT に関して解けた問題の大きな差は見られなかった。表 1, 図 1, 図 2, 図 3 より新たに提案した M+L+D+CS, M+L+D+CS+CR はソルバ単体としての性能は良くなかったが, 表 2 より M が解けなかった問題に対して M+L+D と M+L+D+CS+CR が解けた問題数は SAT 問題と UNSAT 問題でそれぞれ 13/49(SAT) と 35/78(UNSAT), 11/49(SAT) と 35/78(UNSAT) であり, M+L+D と M+L+D+CS+CR がそれぞれ新たに解けた問題数はほぼ同じであったが, M+L+D が解けなかった問題に対して, M+L+D+CS+CR が解いた問題数は SAT 問題と UNSAT 問題それぞれ 3/49(SAT), 13/48(UNSAT) であった。よって本手法は UNSAT に関しては, 既存のソルバで解くことができなかった問題に対して性能を發揮した。また本手法と M+L+D をポートフォリオに組み込めば, ポートフォリオ型ソルバの性能向上につながると考えられる。

ソルバ	SAT	UNSAT	TOTAL
M	101	72	173
M+L	102	84	186
M+L+D	101	102	203
M+L+D+CS	87	97	184
M+L+D+CS+CR	90	98	188

表 1: SAT Competition 2014 result(SAT + UNSAT)

ソルバ	SAT		UNSAT	
	M	M+L+D	M	M+L+D
M+L+D	13/49	-	35/78	-
M+L+D+CS+CR	11/49	3/49	35/78	13/48

表 2: M, M+L+D+CS+CR diff

ソルバ	SAT		UNSAT	
	M+L+D+CS	M+L+D+CS+CR	M+L+D+CS	M+L+D+CS+CR
M+L+D+CS	—	4/60	—	1/52
M+L+D+CS+CR	7/63	—	2/53	—

表 3: M+L+D+CS , M+L+D+CS+CR diff

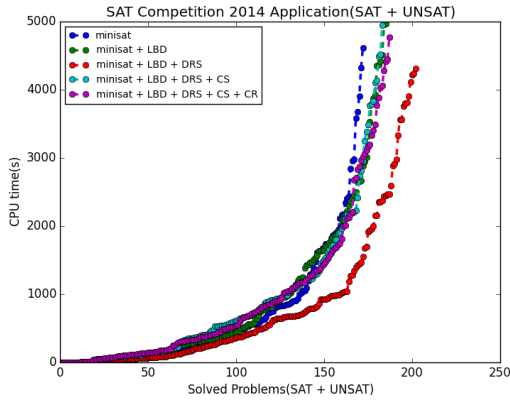


図 1: SAT Competition 2014(SAT + UNSAT)

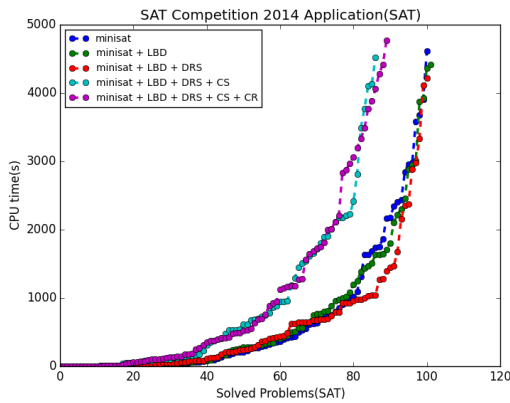


図 2: SAT Competition 2014(SAT)

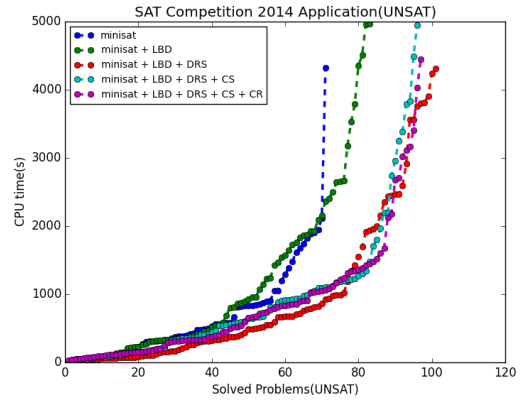


図 3: SAT Competition 2014(UNSAT)

## 6. まとめ

本稿では、ソルバの集中的な探索の実現と多様性を獲得するために SAT 問題のコミュニティ構造を利用した Community Switching と Community Rank を新たに提案した。ソルバ単体として解の向上には繋がらなかったが、既存手法では解くことができなかった問題をいくつか解けることを示した。今後は UNSAT の更なる強化を図るために UNSAT を得意とするソルバに本手法を組み合わせたリして、改良を行いたい。また本手法を既存のポートフォリオ型ソルバに追加することで性能向上に繋がるのではないかと考え、ポートフォリオ型ソルバに組み込み実験を行いたい。

## 参考文献

- [1] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: engineering an efficient SAT solver. In Proceedings of the 38th annual Design Automation Conference (DAC '01). ACM. p530-535. 2001.
- [2] Marques Silva, J.P. Sakallah, K.A. Conflict analysis in search algorithms for satisfiability. Tools with Artificial Intelligence, 1996., Proceedings Eighth IEEE International Conference on. IEEE. p467-469. 1996.
- [3] Roberto J. Bayardo, Jr. and Robert C. Schrag. Using CSP look-back techniques to solve real-world SAT instances. In Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence (AAAI'97/IAAI'97). AAAI Press. p203-208. 1997.
- [4] Carla P. Gomes and Bart Selman. Algorithm portfolio design: theory vs. practice. Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc. p190-197. 1997.
- [5] Cimatti, Alessandro, Sebastiani, Roberto. The Community Structure of SAT Formulas. Theory and Applications of Satisfiability Testing SAT 2012. Springer Berlin Heidelberg. p410-423. 2012.

- [6] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.* p10008 2008.
- [7] Gilles Audemard, Laurent Simon. Predicting Learnt Clauses Quality in Modern SAT Solvers. *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc. p399-404. 2009.
- [8] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report. Stanford InfoLab. 1999.
- [9] Newsham, Zack and Ganesh, Vijay and Fischmeister, Sebastian and Audemard, Gilles and Simon, Laurent. Impact of Community Structure on SAT Solver Performance. *Theory and Applications of Satisfiability Testing SAT 2014*. Springer International Publishing. p252-268. 2014.
- [10] Newman, M. E. J. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*. p8577-8582. 2006.
- [11] En, Niklas and Srensson, Niklas. An Extensible SAT-solver. *Theory and Applications of Satisfiability Testing*. Springer Berlin Heidelberg. p502-518. 2004.