1D4-OS-02a-3

SAT 型制約ソルバーによるナンバーリンクの解法とその評価

Solving Numberlink by a SAT-based Constraint Solver and its Evaluation

迫 龍哉 [*1] Tatsuya Sako　川原 征大 [*1] Masahiro Kawahara　宋 剛秀 [*2] Takehide Soh　番原 睦則 [*2] Mutsunori Banbara　田村 直之 [*2] Naoyuki Tamura　鍋島 英知 [*3] Hidetomo Nabeshima

[*1]神戸大学大学院システム情報学研究科
Graduate School of System Informatics, Kobe University

[*2]神戸大学情報基盤センター
Information Science and Technology Center, Kobe University

[*3]山梨大学大学院医学工学総合研究部
Department of Research Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi

Remarkable improvements on SAT technology over the last decade have led the application of SAT-based constraint solvers in a wide range of fields. In this paper, we introduce a SAT-based constraint solver iSugar+GlueMiniSat which is the winner at the Algorithm Design Contest (ADC), a competition of Numberlink puzzle solvers held in 2015. iSugar+GlueMiniSat can use incremental SAT method to efficiently solve optimization problems. In the modeling of Numberlink, we introduced additional constraints and formulated as optimization problem. We show the effectiveness of the solver through the result of ADC and experiments.

1. はじめに

近年，命題論理の充足可能性判定 (SAT) 問題を解く技術が進歩しており，非常に高速に解を求めることのできる SAT ソルバーが実現されている．これにより，元の問題を一度 SAT 問題に変換してから SAT ソルバーで解を求め，それを復号化して元の問題の解を求める **SAT 型システム**が成功を収めている．特に，制約充足問題 (CSP) を解く SAT 型システムは **SAT 型制約ソルバー**と呼ばれ，活発な研究開発が行われている．

本稿では，SAT 型制約ソルバー iSugar+GlueMiniSat を紹介したのち，回路配線などに応用のあるナンバーリンク [*1] を SAT 型制約ソルバーで解く方法について説明する．iSugar+GlueMiniSat は，既存の SAT 型制約ソルバー Sugar を**インクリメンタル SAT 解法** [Whittemore 01, Eén 03] が利用できるように拡張し，高速 SAT ソルバー GlueMiniSat[鍋島 12] と結合させたものである．インクリメンタル SAT 解法は，SAT ソルバーが同様の探索失敗を防ぐために獲得する学習節を保持することで，無駄な探索を行うことなく，節を追加した SAT 問題を連続的に解く手法であり，有界モデル検査 [Biere 99] などに利用されている．この拡張により，最適化問題，解列挙問題などの高速な求解を実現している．

iSugar+GlueMiniSat を用いて，ナンバーリンクソルバーの競技会であるアルゴリズムデザインコンテスト [*2] に参加した．その結果，iSugar+GlueMiniSat は参加チーム中で最も多くの問題を解き，最優秀賞を受賞した．

ナンバーリンクの求解における主な工夫として，

- 特定の曲がり方を禁止する追加制約の導入
- 最適解探索アルゴリズムの工夫
- インクリメンタル SAT 解法の利用

を行った．これらの工夫については 4 節で詳しく説明する．

上記の改良やコンテストでの求解戦略の有効性を確認するために実験を行ったところ，新たな追加制約やインクリメンタル SAT 解法の利用によって求解性能が向上したことが分かり，コンテストでの戦略も概ね妥当であったことを確認できた．

連絡先: 迫 龍哉，神戸大学大学院システム情報学研究科，神戸市灘区六甲台町 1-1，078-803-5365，t_sako@stu.kobe-u.ac.jp

*1 「ナンバーリンク」は株式会社ニコリの登録商標．

*2 http://www.sig-sldm.org/designcontest.html

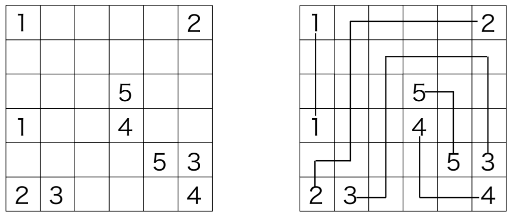

図 1: ナンバーリンクの問題例とその解

2. ナンバーリンク

本節では，ナンバーリンクの説明およびアルゴリズムデザインコンテストの紹介を行う．

2.1 ナンバーリンク

ナンバーリンクはペンシルパズルの一種であり，IC の一層平面配線問題と等価な問題として知られている．ニコリによるナンバーリンクのルールは以下の通りである．

1. 白マスに線を引いて，同じ数字どうしをつなげましょう．
2. 線は，マスの中央を通るようにタテヨコに引きます．線を交差させたり，枝分かれさせたりしてはいけません．
3. 数字の入っているマスを通過するように線を引いてはいけません．
4. 1 マスに 2 本以上の線を引いてはいけません．

図 1 に，ナンバーリンクの問題とその解の例を示す．ナンバーリンクの制約モデリングについては，4.1 節で詳述する．

2.2 アルゴリズムデザインコンテスト

アルゴリズムデザインコンテスト (以下 ADC) とは，情報処理学会が主催する DA シンポジウムで 2012 年から開催されている，ナンバーリンクソルバーの競技会である．2014 年以前の ADC では，決定問題としてナンバーリンクが出題されていた．これまでに分枝限定法，Simpath 法，ZDD[Minato 93]

など様々な手法を用いたソルバーが参加している．2014 年度は Sugar+GlueMiniSat が優勝し，SAT 型制約ソルバーを用いたアプローチ [田村 14] の有効性を示した．

しかし，応用上は配線遅延やコストの観点から，線長と線の曲がる回数は小さいほうが良い．このことから，2015 年度は (線長+曲がる回数) を最小化する最適化問題として出題され，解く問題数も増加した．そのため，効率よく解くには求解を高速化するための工夫が必要となった．

3. SAT 型制約ソルバー

本節では，SAT 型制約ソルバーについて説明した後，iSugar+GlueMiniSat の紹介を行う．

3.1 SAT 型制約ソルバー

SAT 型制約ソルバーは，制約充足問題 (CSP) を SAT 問題に変換 (SAT 符号化) し，SAT ソルバーを用いて求めた解を復号化して元の問題の解を求めるシステムである．

代表的な SAT 型制約ソルバーである Sugar は，順序符号化法 [Tamura 09] と呼ばれる SAT 符号化法を用いており，2008,2009 年の国際 CSP ソルバー競技会の GLOBAL 部門で優勝するなど，高い性能を示している．

しかしながら，Sugar は最適化問題や解列挙問題など，SAT 問題を複数解く必要のある問題に対しては各回の求解ごとに SAT ソルバーを起動する．そのため，学習節をはじめとする探索に有用な情報を次回以降に保持できず，探索に無駄が生じ，求解効率が低下するという問題点がある．

3.2 iSugar+GlueMiniSat

前節で挙げた問題点を解決する方法として，インクリメンタル **SAT** 解法の導入が考えられる．

SAT ソルバーは一般に，求解終了後も結果が SAT であれば，その探索で追加された学習節を次回以降の探索でも利用可能である．そのため，求解終了後も SAT ソルバーを起動したままにし，節を追加して続けて求解を行うことができる．これをインクリメンタル SAT 解法と呼ぶ．

iSugar+GlueMiniSat は，別途開発したインクリメンタル SAT 解法ライブラリである iSAT Library*3[迫 16] を用いて Sugar と GlueMiniSat を結合することで，最適化問題，解列挙問題などの高速な求解を実現したシステムである．

4. システムの改良と工夫

本節では，より高速にナンバーリンクを解くために行った，システムの改良と工夫について説明する．

4.1 制約モデルの改良

制約モデリングによる求解性能を向上するため，従来のモデル [田村 14] をもとに，制約の追加などを行った．以下では，従来の制約モデルと新たに追加した制約について述べる．なお，従来のモデルについては，その制約の概要のみを説明する．

4.1.1 従来の制約モデル

各マスを頂点とし，上下左右に隣接するマスとの間に辺を持つ有向グラフを考えると，ナンバーリンクの解はこの部分グラフとして表せる．以下では，問題中で数字の書かれたマスを数字マス，それ以外のマスを白マスと呼ぶ．

辺に関する制約として，隣接するマス間は高々1 本の辺をもつ．すなわち，反対向きに同時に線が引かれることはない．

*3 http://bach.istc.kobe-u.ac.jp/iSATLib/

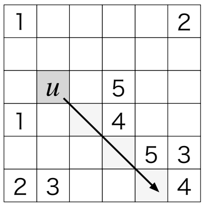

図 2: 追加制約説明図 1

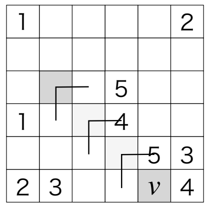

図 3: 追加制約説明図 2

次に，各マスの次数に関する制約を与える．線が引かれているマスの次数を 1，引かれていないマスの次数を 0 とすると，始点の数字マスではそこから出ていく辺が 1 本だけあり，入ってくる辺はない．終点の数字マスはこの逆で，出ていく辺がなく入ってくる辺が 1 本だけある．また白マスについては，出ていく辺と入ってくる辺の本数はそのマスの次数に等しい．

次に，各マスのラベルについて考える．数字マスの場合，ラベルはそのマスの数字に等しい．また，辺で結ばれているマスのラベルは同じでなければならない．

この他，各領域の次数に関する制約も導入する．4 分割した盤面のある領域から隣接する各領域への出次数の合計は，その領域内の始点のマス数から終点のマス数を引いた値に等しい．

以上の制約に加えて，求解性能を向上させるための追加制約として，(a) 全てのマスを通る，(b) コの字型の回り道をしない，の 2 つを考える．(a) の制約は，全マスの次数を 1 とすることで表せる．また制約 (b) は，田の字型に隣接する 4 つのマスについて，辺が高々2 本しかないことと同値である．

以上の制約モデルでは解に余分な閉路を含む可能性があるが，これを除く制約は複雑なものとなり，求解性能が低下する恐れがある．そのため閉路除去の制約は追加せず，制約モデルの解からナンバーリンクの解を抽出する際に閉路を除去する．

4.1.2 新たな追加制約

求解性能をさらに向上させるため，前節の追加制約 (a)(b) に加え，(c) 特定の曲がり方を禁止する 制約を導入した．

図 2 のように，そこから斜め右下の方向に数字マスが存在しないような白マス u に着目する．このとき，制約 (a)(b) のもとでは，u は右，下の両方には辺をもたないことを示す．

斜め右下方向に数字マスのない白マス u が，右と下に辺をもつと仮定する．制約 (a) より，全てのマスは辺をもつ．いま，u の右下のマスは制約 (b) より左と上には辺をもたないので，右と下に辺をもつ．これを繰り返すと図 3 のように盤面の外周のマス v に到達するが，このマスには制約 (b) を満たす線を引くことができない．よって，u は右と下の両方には辺をもたない．これは他の 3 方向についても同様である．

上記の制約を定式化する．マス u の上下左右のマスをそれぞれ n, s, w, e とし，u の斜め左上，右上，左下，右下方向のマスの集合をそれぞれ NW_u, NE_u, SW_u, SE_u とすると，この制約は次のように表せる．ただし，a_{uv} は u から v への辺の有無を表す 0-1 変数，B は白マスの集合を表す．

$$a_{un} + a_{nu} + a_{uw} + a_{wu} \leq 1 \quad (\forall u(NW_u \subset B)) \qquad (1)$$

$$a_{un} + a_{nu} + a_{ue} + a_{eu} \leq 1 \quad (\forall u(NE_u \subset B)) \qquad (2)$$

$$a_{us} + a_{su} + a_{uw} + a_{wu} \leq 1 \quad (\forall u(SW_u \subset B)) \qquad (3)$$

$$a_{us} + a_{su} + a_{ue} + a_{eu} \leq 1 \quad (\forall u(SE_u \subset B)) \qquad (4)$$

この追加制約 (c) は，4.1.1 節の制約と (a)(b) の制約から導出可能であり本来は不要であるが，盤面の中央付近のマスではその証明は非常に長いものになるため，証明の短縮のために追

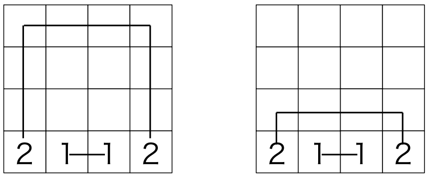

図 4: 曲がる回数が最小の解と後処理後の解

加した．また，制約 (c) を加えても (a)(b) の制約下で得られる解は全て求められるが，それ以外では解を漏らす可能性があり，制約 (a) 同様解をひとつも発見できない場合もある．

4.1.3 最適化問題としての定式化

線長の合計を L，線の曲がる回数を T とすると，$L+T$ を最小化することで解を最適化できる．線長は $L=\sum_{(u,v)} a_{uv}$ で求められる．(u,v) は隣接する 2 マス u と v の組である．

T を求めるため，白マス u で線が曲がる条件を考える．u の次数が 0 ならば，線が存在しないので曲がっていない．u が四隅のマスの場合，次数が 1 ならば線が存在するので曲がっている．u が四隅を除く外周のマスの場合，その内側のマス (例えば，u が左端のマスの場合はその右隣のマス) との間に辺があれば曲がっている．u がそれ以外のマスの場合，上下もしくは左右方向について，辺が 1 本だけ存在すれば曲がっている．

上記の条件の定式化を行う．各白マス u について，そこで線が曲がっているかどうかを表す変数 $t_u \in \{0,1\}$ を導入する．四隅，四隅以外の外周，それ以外のマスの集合をそれぞれ V_c, V_o, V_i とすると，上記の制約は次のように表せる．ただし V は全マスの集合，$e_{uv}=a_{uv}+a_{vu}$，$adj(u)$ は u に隣接するマスの集合である．

$$d_u = 0 \implies t_u = 0 \quad (\forall u \in V) \tag{5}$$

$$d_u = 1 \implies t_u = 1 \quad (\forall u \in V_c) \tag{6}$$

$$e_{uv} = 1 \implies t_u = 1 \quad (\forall u \in V_o, v \in adj(u) \cap V_i) \tag{7}$$

$$e_{un} + e_{us} = 1 \implies t_u = 1 \quad (\forall u \in V_i) \tag{8}$$

$$e_{uw} + e_{ue} = 1 \implies t_u = 1 \quad (\forall u \in V_i) \tag{9}$$

線の曲がる回数は $T=\sum_{u\in B} t_u$ で求められる．

4.2 最適解探索アルゴリズム

4.1.3 節で述べた制約をそのまま導入すると，符号化後の節数が膨大になり，求解に時間がかかってしまう．これは，限られた時間で多くの問題を解く ADC には不向きである．

ADC では，最適化問題を解く場合には曲がる回数のみを最小化するようにした．この場合，例えば図 4 の左のような解が得られる場合があるが，このような解については別で作成した後処理プログラムを実行し，図 4 の右のように解の品質を上げるようにした．線長のみの最小化も考えられるが，曲がる回数のみの最小化に比べて問題の規模が大きくなることや，線がジグザグに折れ曲がるような解が得られた場合の後処理が前述の処理より煩雑であると考えられたため採用しなかった．

この手順によって得られる解は，(線長+曲がる回数) が最小とならない可能性があるが，これによって解の品質が著しく低下することはないと考え，高速に解けることを重視した．

4.3 インクリメンタル SAT 解法の利用

4.1 節および 4.2 節の改良は，Sugar にも適用可能である．しかし，3.1 節で述べた通り，Sugar は最適化問題を解く際に十分な性能を発揮できない．そのため，今回の ADC では

表 1: アルゴリズムデザインコンテスト結果

順位	1	2	3	4	5	6	7
求解数	**34**	19	22	18	9	12	2
スコア	**155.8**	85.9	83.5	67.5	43.3	39.9	5.1

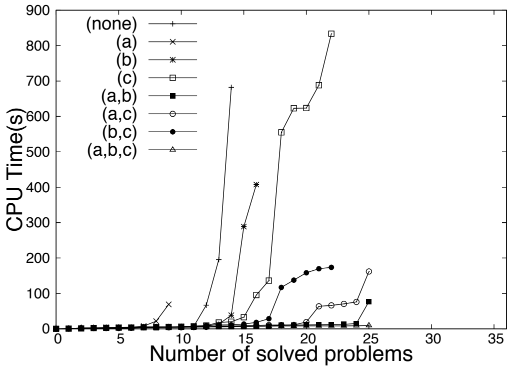

図 5: 追加制約の組み合わせによる性能比較

iSugar+GlueMiniSat を用いて，インクリメンタル SAT 解法を利用することとした．

5. ADC 結果と評価実験

本節では，2015 年度の ADC(以下単に ADC と表記) の結果を紹介する．また，ADC で採った求解戦略の妥当性と，提案した追加制約およびインクリメンタル SAT 解法の導入による有効性を検証するために行った実験結果について述べる．

5.1 ADC 結果

4 節で述べた工夫を施した iSugar+GlueMiniSat で，2015 年度 ADC に参加した．参加チームは全 7 チームで，SAT や ZDD のほか，整数計画法や機械学習を用いた手法などが見られた．

出題数は 36 問，制限時間は全体で 40 分であった．1 問あたりに取れる時間が少ないため，解を求めることを優先して以下のような戦略を採った．まず，全ての追加制約 (a)，(b)，(c) を与えたモデルで求解を行う．このモデルは非常に高速に求解可能であるが，解を失う問題もあると考えられたため，そのような問題については制約 (b)，(c) を与えたモデルで解き直し，それでも解が見つからない問題は制約 (b) のみを与えたモデルで解く．この一連の求解は全て決定問題として行う．その後，(b)，(c) および (b) で解けた問題については，最適化問題として解き直し，解の品質を向上させる．

ADC の結果を表 1 に示す．iSugar+GlueMiniSat は 36 問中 34 問の求解に成功し，最優秀賞を受賞した．このうち Q7 と Q21 の 2 問は，iSugar+GlueMiniSat だけが解けた問題であった．より詳細な結果については結果資料 *4 を参照されたい．

5.2 評価実験

評価実験では，ベンチマーク問題として 2015 年の ADC で出題されたナンバーリンク 36 問を用いた．CPU 時間は Linux Ubuntu 14.04 (Xeon 2.40GHz，16GB メモリ) 上で計測し，全実験において 1 問あたりの制限時間は 900 秒とした．

5.2.1 追加制約による性能比較

追加制約の効果を確認するため，(a)，(b)，(c) の制約の有無による 8 通りのモデルで実験を行った．ソルバーには iSugar+GlueMiniSat を用いた．図 5 にカクタスプロットを示

*4 http://www.sig-sldm.org/DC2015/ADC2015results2.pdf

表 2: (a) を含まないモデルの 11 問における性能比較 (sec)

追加制約	(none)	(b)	(c)	(b,c)
Q6	(T.O.)	(T.O.)	(T.O.)	-
Q7	(T.O.)	407.66	5.98	4.03
Q8	3.19	2.88	-	-
Q9	5.03	6.10	-	-
Q11	(T.O.)	(T.O.)	136.23	18.13
Q17	5.68	3.02	-	-
Q20	195.49	13.12	-	-
Q21	(T.O.)	288.82	17.69	10.75
Q27	1.08	1.43	0.99	1.18
Q29	4.10	3.22	-	-
Q32	(T.O.)	(T.O.)	(T.O.)	(T.O.)
求解数	6	8	4	4

す．横軸が求解問題数，縦軸が求解時間である．グラフが右下にあるほど多くの問題を高速に解いていることを示している．

実験の結果，(a,b,c) が 25 問を解き，最も高い性能を示した．次いで (a,b)，(a,c) も 25 問を解いたが，これらは全て同一の問題であった．しかし，上記の 3 つのモデルは，残りの 11 問全ての解を失っていた．そこで，制約 (a) を含まない 4 つのモデルについて，11 問の求解結果をまとめたものを表 2 に示す．(T.O.) は 900 秒以内に解けなかったこと，ハイフンは 900 秒以内に解がないと判定された (解を失った) ことを表す．

表 2 より，最も多くの問題を解いたのは (b) であったが，(b,c) が解いた 4 問のうち 3 問は (b) よりも時間が大幅に短縮したことが分かる．特に Q7 と Q21 は 100 倍以上高速化し，制約 (c) の効果が大きく現れた．そのため，(b) では時間がかかる問題を先に (b,c) で解いてから残りを (b) で解くことで，全体として短時間でより多くの問題の求解が可能になったといえる．この結果から，制約 (c) により性能が向上したことと，ADC での求解戦略手順が概ね妥当であったことが分かった．

5.2.2　インクリメンタル SAT 解法の利用による性能比較

4.1.3 節で述べた変数と制約を導入し，インクリメンタル SAT 解法を利用する iSugar+GlueMiniSat と利用しない Sugar+GlueMiniSat で最適化問題に対する求解性能を比較する．以降，これらのシステムをそれぞれ iSugar, Sugar と略記する．最小化する目的変数として，線長と曲がる回数の和 $L+T$ と，曲がる回数 T の 2 種類を考え，それぞれ実験した．

Sugar では求解の度に SAT ソルバーを起動し，目的変数の取り得る値の範囲を変えた決定問題を繰り返し解くため，学習節などの情報を各探索回ごとに破棄する．iSugar では求解結果に応じて目的変数の範囲を更新する制約を追加することで，SAT ソルバーを起動したまま，学習節を再利用して求解する．

最適化問題として解く場合，追加制約 (a) の除去と (b) の導入は本質的である．(a) は線長を取り得る最大の値にしてしまうため，最適化には適さない．一方 (b) は冗長路を含む解を排除するので，最適解は必ず発見できる．そこで，追加制約は (b) のみと (b,c) の 2 通りとし，目的変数 2 通り，ソルバー 2 通りの計 8 通りの組み合わせで実験した．表 3 に，その結果を示す．表中の最良値とは各制約のもとで解いたときの最適値であり，括弧内の数字は，(b) のもとで $L+T$ を最小化する場合に得られる最良値 (厳密な最適値) と一致した問題数を表す．

表 3 より，どのモデルでも iSugar のほうが Sugar よりも良い性能を示した．(b,c) で T を最小化した場合，解の品質が低下した問題は Q5,Q27 の 2 問だけであり，解の品質が大きく低下することはなかった．また，(b) で $L+T$ を最小化するモデルでは時間内に解けなかった 11 問について解を求めていた．以上の結果より，インクリメンタル SAT 解法が有効であることと，(b,c) で T を最小化するという ADC での戦略が良いものであったことが分かった．

表 3: インクリメンタル SAT 解法利用の有無による比較 (sec)

(b) のみを与えた場合

目的関数	$\min(L+T)$		$\min(T)$	
	Sugar	iSugar	Sugar	iSugar
最良値の求解数	8	11	11(10)	12(10)

(b,c) を与えた場合

目的関数	$\min(L+T)$		$\min(T)$	
	Sugar	iSugar	Sugar	iSugar
最良値の求解数	15(6)	16(6)	16(6)	**19(6)**

6. まとめ

本稿では，SAT 型制約ソルバー iSugar+GlueMiniSat によるナンバーリンクの求解方法を述べた．解法の工夫として，

- 追加制約 (c) の導入による制約モデルの改良
- 曲がる回数を最小化する最適解探索アルゴリズム
- 最適化問題に対するインクリメンタル SAT 解法の利用

などを行った．

iSugar+GlueMiniSat を用いて 2015 年度の ADC に参加した．ADC では上記の改良に加え，3 つの追加制約を組み合わせた複数のモデルによる求解戦略を採り，最優秀賞を受賞した．

出題された問題を用いて評価実験を行ったところ，追加制約とインクリメンタル SAT 解法の導入のそれぞれで性能が向上したこと，ADC で採った戦略が妥当であったことが分かった．

今後の課題としては，追加制約によって解が失われる問題の特徴を分析し，制約を改良することでそのような問題の数を削減することや，線長や曲がる回数の定式化方法を見直し，求解性能をさらに向上させることなどが挙げられる．

参考文献

[Biere 99] Biere, A., Cimatti, A., Clarke, E. M., and Zhu, Y.: Symbolic Model Checking without BDDs, in *Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS 1999), LNCS 1579*, pp. 193–207 (1999)

[Eén 03] Eén, N. and Sörensson, N.: Temporal Induction by Incremental SAT Solving, *Electronic Notes in Theoretical Computer Science*, Vol. 89, No. 4 (2003)

[Minato 93] Minato, S.: Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, in *DAC*, pp. 272–277 (1993)

[鍋島 12] 鍋島 英知, 岩沼 宏治, 井上 克巳：GlueMiniSat 2.2.5: 単位伝搬を促す学習節の積極的獲得戦略に基づく高速 SAT ソルバー, コンピュータソフトウェア, Vol. 29, No. 4, pp. 4_146–4_160 (2012)

[迫 16] 迫 龍哉, 宋 剛秀, 番原 睦則, 田村 直之, 鍋島 英知, 井上 克巳：インクリメンタル SAT 解法ライブラリとその応用, コンピュータソフトウェア，2016 年 11 月号掲載予定 (2016)

[Tamura 09] Tamura, N., Taga, A., Kitagawa, S., and Banbara, M.: Compiling Finite Linear CSP into SAT, *Constraints*, Vol. 14, No. 2, pp. 254–272 (2009)

[田村 14] 田村 直之, 宋 剛秀, 番原 睦則, 鍋島 英知：SAT 型制約ソルバーを用いたナンバーリンクの解法, DA シンポジウム 2014 論文集, Vol. 2014, pp. 215–220 (2014)

[Whittemore 01] Whittemore, J., Kim, J., and Sakallah, K. A.: SATIRE: A New Incremental Satisfiability Engine, in *Proceedings of the 38th Design Automation Conference, DAC 2001, Las Vegas, NV, USA, June 18-22, 2001*, pp. 542–545 (2001)