



ブール基数制約を経由した擬似ブール制約のSAT符号化手法

南, 雄之
宋, 剛秀
番原, 睦則
田村, 直之

(Citation)

コンピュータ ソフトウェア, 35(3):65-78

(Issue Date)

2018

(Resource Type)

journal article

(Version)

Version of Record

(Rights)

All Rights Reserved, Copyright (C) Information Processing Society of Japan. Notice for the use of this material: The copyright of this material is retained by the Japan Society for Software Science and Technology(JSSST). This material is published on this web site with the agreement of the JSSST. Please comply with Copyright Law of Japan ...

(URL)

<https://hdl.handle.net/20.500.14094/90006799>



ブール基数制約を経由した擬似ブール制約の SAT 符号化手法

南 雄之 宋 剛秀 番原 睦則 田村 直之

本論文では、擬似ブール (Pseudo-Boolean; PB) 制約の集合を命題論理式の充足可能性判定 (SAT) 問題へ符号化する新しい手法として、ブール基数 (Boolean Cardinality; BC) 制約を経由する方法を提案する。提案手法は、次の 3 つの特徴を持つ。1 つ目は、SAT ソルバーの単位伝播により一般化アーク整合性の維持が可能な点である。2 つ目は、同じ解を持つ同値な PB 制約であれば係数や右辺の値が異なっても、同一の中間表現および SAT 問題に符号化可能な点である。3 つ目は、項数に対して係数の種類が少ない PB 制約に対しては、中間表現が簡潔になり少ない節数で SAT 符号化可能な点である。このような PB 制約は、国際 PB ソルバー競技会のベンチマーク問題にも頻出している。計算機実験では、代表的な既存手法で一般化アーク整合性維持が可能な BDD 法、およびそれより弱い整合性検査が可能な Sorter 法と符号化後の節数と求解性能を比較した。結果として、異なる係数の種類が 10% 以下であるような PB 制約について、提案手法が節数と求解性能に関して比較した 2 手法よりも良いことを確認した。

In this paper, we propose a new encoding method for a set of PB (Pseudo-Boolean) constraints into propositional satisfiability (SAT) problems, in which Boolean Cardinality (BC) constraints are used as an intermediate form. The proposed method has the following three features. First, it can maintain general arc consistency by unit propagation of a SAT solver. Second, it can encode equivalent PB constraints with the same solutions—even their coefficients and right hand side value are different—into the same intermediate form and SAT instance. Third, for PB constraints whose number of kinds of coefficients is relatively small compared with the number of terms, the intermediate form becomes simpler and they can be encoded with a small number of clauses. Such PB constraints often appear in international PB solver competition benchmarks. In experiments, we compared the proposed encoding method with existing methods, BDD and Sorter. The former maintains general arc consistency by unit propagation, while the later maintains consistency checking that is weaker than general arc consistency. As the result, for PB constraints in which the number of different coefficients is not more than 10%, we confirmed that the proposed method is better than those two methods in terms of the number of encoded clauses and the efficiency in solver performance.

1 はじめに

SAT (Satisfiability) とは、与えられた命題論理式の充足可能性を判定する問題である [8]。本論文で対

象とする擬似ブール (Pseudo-Boolean; PB) 制約は命題変数上の線形制約であり、PB 制約の集合の充足可能性を判定する PB 問題は SAT 問題の拡張になっている。

SAT 問題を解くプログラムである SAT ソルバー [10] の性能が飛躍的に向上していることを背景として、PB 問題を SAT 問題に符号化 (変換) する研究が活発に行われ、数多くの手法が提案されている [12][7][16][11][14]。これらの SAT 符号化手法の良し悪しを考える代表的な基準には、整合性検査 (Consistency Checking) あるいは一般化アーク整合性 (Generalized Arc Consistency; GAC) の維持が SAT ソルバーの単位伝播により可能か否かという点がある [2]。

A SAT Encoding of Pseudo-Boolean Constraints via Boolean Cardinality Constraints

Yushi Minami, 神戸大学大学院システム情報学研究科, Graduate School of System Informatics, Kobe University.

Takehide Soh, Mutsunori Banbara, Naoyuki Tamura, 神戸大学情報基盤センター, Information Science and Technology Center, Kobe University.

コンピュータソフトウェア, Vol.35, No.3 (2018), pp.65–78.

[研究論文] 2017 年 9 月 4 日受付。

大会同時投稿論文。

整合性検査は、命題変数への部分的な値割当てにより PB 制約が矛盾した場合、その矛盾を SAT ソルバーの単位伝播により検出できるかどうかという基準である。一般化アーク整合性は、現時点で PB 制約は矛盾していないが、さらに $x = 0$ (あるいは $x = 1$) とすると矛盾するとき、 $x = 1$ (あるいは $x = 0$) を単位伝播により検出できるかどうかという基準であり、一般化アーク整合性のほうがより強く、その維持のほうが望ましい。また、これらの基準で同等の場合には、符号化後の節数が少ない手法ほど良いとされるのが一般的である。

国際 PB ソルバー競技会における優勝ソルバーの 1 つである MiniSat+^{†1} は BDD (Binary Decision Diagram) 法, Sorter 法, Adder 法の 3 種類を利用している [7]。BDD 法では、節数のオーダーは最悪の場合に指数関数的となるが、一般化アーク整合性が維持される。Sorter 法はソーティング回路を用いた手法で、節数のオーダーは $O(n \log^2 n)$ だが、一般化アーク整合性が維持されず整合性検査のみが維持される。Adder 法は、加算回路を用いた手法で節数のオーダーは $O(n)$ で最も小さいが、一般化アーク整合性も整合性検査も維持されない。このように符号化手法には、それぞれ特徴があり得意な PB 制約は異なる。既存手法にない特徴を持った符号化を考案することは、重要な研究課題といえる。

本論文では、符号化の中間表現に **ブール基数** (Boolean Cardinality; BC) 制約を用いる新しい手法を提案する。BC 制約は係数が 1 の PB 制約であり、効果的な SAT 符号化手法が知られている [15] [3]。提案手法は、次の 3 つの特徴を持つ。

- SAT ソルバーの単位伝播により一般化アーク整合性の維持が可能。
- 同じ解を持つ同値な PB 制約であれば、係数や右辺の値が異なっても、同一の中間表現および SAT 問題に符号化可能。
- 異なった係数の種類が比較的少ない PB 制約に対しては、中間表現が簡潔になり少ない節数で SAT 符号化可能。このような PB 制約は、国際

PB ソルバー競技会のベンチマーク問題にも頻出している。

提案手法の有効性を評価するため、2 種類の計算機実験を行う。1 つ目では、異なった係数の種類が一定の割合より少ない PB 制約をランダムに生成し、BDD 法, Sorter 法と符号化後の節数を比較する。2 つ目では、2016 年国際 PB ソルバー競技会^{†2} の DEC-SMALLINT-LIN 部門の 1783 問を用い、同じ 2 手法と求解性能を比較する。これら 2 つの実験結果を通し、異なった係数の種類が比較的少ない PB 制約、およびそれらを含む PB 問題について提案手法の有効性を示す。

以降、第 2 節では SAT, PB 制約, BC 制約, BC 制約の連言標準形の式である BC-CNF 式を導入する。第 3 節では提案する符号化手法の各ステップについて説明する。第 4 節で提案手法の特徴と限界について述べ、第 5 節で計算機実験の結果を考察する。最後に第 6 節で、本論文の結論を述べる。

2 SAT, PB 制約, BC 制約

命題変数 p およびその否定 $\neg p$ をリテラル (literal)、複数のリテラルの選言 (OR) を節 (clause)、複数の節の連言 (AND) を **CNF 式** (Conjunctive Normal Form formula) と呼ぶ。

SAT 問題とは、与えられた CNF 式を真にする命題変数への値割当てが存在するか否かを判定する問題である [8]。値割当てが存在する場合、CNF 式は充足可能 (satisfiable; SAT) であると言い、存在しない場合、充足不能 (unsatisfiable; UNSAT) であると言う。SAT ソルバーは、与えられた CNF 式が SAT か UNSAT かを判定し、SAT ならその解となる値割当てを返すプログラムである [10] [6] [1] [9]。また、PB 問題などの与えられた問題を CNF 式に変換することを **SAT 符号化** (SAT encoding) という。

擬似ブール制約 (PB 制約) は命題変数上の線形制約であり、以下のように定義される [13]。

定義 1 (PB 制約). $X_n = (x_1, x_2, \dots, x_n)$ は同一の変数を含まないリテラルの列、 a_i および c は整数定

^{†1} <https://github.com/niklasso/minisatp>

^{†2} <http://www.cril.univ-artois.fr/PB16/>

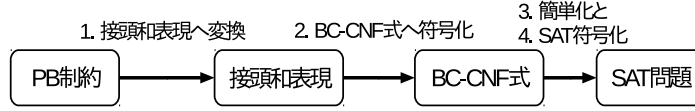


図 1 提案手法の概要

数, $\triangleright$ は関係演算子 ($=, \leq, \geq$) とする. このとき, 以下の式を X_n 上の擬似ブール制約あるいは PB 制約と呼ぶ (真を 1, 偽を 0 と表現する).

$$\sum_{i=1}^n a_i x_i \triangleright c$$

項 $a_i x_i$ で係数 a_i が負の場合, その項を $-a_i \neg x_i + a_i$ に置き換えることで, 係数を正にできる. また, 項の順序を入れ換えることで, PB 制約の左辺は係数の降順に項が並んでいるとしても一般性を失わない. すなわち, 以下では $a_1 \geq a_2 \geq \dots \geq a_n > 0$ を仮定し, 各 PB 制約ごとに定めるリテラル列 $X_n = (x_1, x_2, \dots, x_n)$ も対応する順序になっていると仮定する. さらに, $\sum_{i=1}^n a_i x_i \leq c$ は $\sum_{i=1}^n a_i \neg x_i \geq -c + \sum_{i=1}^n a_i$ に変換でき, $\sum_{i=1}^n a_i x_i = c$ は $\sum_{i=1}^n a_i x_i \geq c$ かつ $\sum_{i=1}^n a_i x_i \leq c$ に変換できる. したがって, 以下では特に必要がない限り関係演算子 $\triangleright$ が $\geq$ の場合について議論し, $\leq$ や $=$ の場合の説明は省略する.

関数 $\alpha: X_n \rightarrow \{0, 1\}$ を X_n 上の値割当てと呼び, これを n 桁の 2 進列 $(\alpha(x_1), \alpha(x_2), \dots, \alpha(x_n))$ で表す. すなわち, X_n 上の値割当て全体は $\{0, 1\}^n$ であり, これを $\mathbf{X}_n$ と表記する. 値割当て α が PB 制約などの式 ϕ を満たすとき, α を ϕ の解と呼び, ϕ の解の集合を $Sol(\phi)$ で表す.

ブール基数制約 (BC 制約) とは係数が全て 1 の PB 制約 $\sum_{i=1}^n x_i \geq c$ のことを言う [13]. X_n 上の複数の BC 制約の選言を BC 節と呼び, BC 制約の集合で表す. 複数の BC 節の連言を BC-CNF 式と呼び, BC 節の集合で表す. また, BC 節に含まれる BC 制約を BC リテラルと呼ぶことがある.

3 提案する SAT 符号化手法

提案手法により, PB 制約を BC-CNF 式を経由して命題論理の CNF 式へと符号化する各ステップは, 以下の通りである (図 1 参照).

1. PB 制約を接頭和表現と呼ぶ形式に変換する (3.1 節).
2. 接頭和表現を, 順序符号化アルゴリズム [17][16] を改良した方法により BC-CNF 式に符号化する (3.2 節).
3. BC リテラル間および BC 節の関係を利用して不要な BC リテラルと BC 節の除去を行い, BC-CNF 式を単純化する (3.3 節).
4. BC 制約の SAT 符号化の 1 つであるシーケンシャルカウンタ法 [15] を用いて, BC-CNF 式に現れるブール基数制約を SAT 問題に符号化し, さらに各 BC リテラルを命題変数のリテラルで置き換える (3.4 節).

以下では, 各ステップの詳細を述べる.

3.1 PB 制約の接頭和表現への変換

ある X_4 上の PB 制約 $\phi: \sum_{i=1}^4 a_i x_i \geq c$ について考える. もし $(1, 0, 1, 0)$ が ϕ の解ならば, $a_2 \geq a_3$ より $(1, 1, 0, 0)$ も必ず解である. 同様に $(1, 0, 1, 1)$, $(1, 1, 0, 1)$, $(1, 1, 1, 0)$, $(1, 1, 1, 1)$ も解となる. すなわち, PB 制約 $\sum_{i=1}^n a_i x_i \geq c$ の解の集合は, $\mathbf{X}_n$ 上のなんらかの順序 $\preceq$ の上方集合 (upper set) になっていると考えられる. すなわち, 解の集合を $S \subset \mathbf{X}_n$ とすると, α が S の要素ならば, $\alpha \preceq \alpha'$ を満たす任意の α' も S に含まれる.

以下では, PB 制約の接頭和表現を定義し, そのような順序構造を導入する.

定義 2 (接頭和表現). $X_n = (x_1, x_2, \dots, x_n)$ 上の PB 制約 $\sum_{i=1}^n a_i x_i \geq c$ に対し, 以下の式をその接頭和表現という (ただし $a_{n+1} = 0$ とみなす).

$$\sum_{i=1}^n (a_i - a_{i+1}) (x_1 + x_2 + \dots + x_i) \geq c$$

以下では, 各 $x_1 + x_2 + \dots + x_i$ を s_i で表し, X_n に対する i 番目の接頭和と呼ぶ. したがって, 接頭和表

現を

$$\sum_{i=1}^n b_i s_i \geq c$$

のように記述する。ただし $b_i = a_i - a_{i+1}$ であり、 $a_1 \geq a_2 \geq \dots \geq a_n > 0$ より $\forall i(b_i \geq 0)$ である。

例 1. PB 制約

$$5x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5 + x_6 \geq 9$$

の接頭和表現は、

$$2s_1 + 2s_5 + s_6 \geq 9$$

である。

PB 制約と接頭和表現は 1 対 1 に対応する。また X_n 上の PB 制約に対する可能な値割当ては n 桁の 2 進列、すなわち $\mathbf{X}_n$ の要素で表され、一方、接頭和表現中の $(s_1, s_2, \dots, s_n)$ に対する可能な値割当ては、初期値が 0 か 1 で増分が 1 以下の昇順の整数列で表される。そこで、初期値が 0 か 1 で増分が 1 以下である長さ n の昇順の整数列 (以下、昇順列と呼ぶ) の集合を $\mathbf{S}_n$ で表す。このとき、 $\mathbf{X}_n$ の要素 $(u_1, u_2, \dots, u_n)$ は $\mathbf{S}_n$ の要素 $(u_1, u_1 + u_2, \dots, u_1 + u_2 + \dots + u_n)$ と 1 対 1 に対応する。例えば、2 進列 $(1, 1, 0, 0, 0, 1)$ には昇順列 $(1, 2, 2, 2, 2, 3)$ が対応する。2 進列が PB 制約の解であることと、対応する昇順列がその接頭和表現の解であることは明らかに同値である。例えば、2 進列 $(1, 1, 0, 0, 0, 1)$ は例 1 の PB 制約の解になっており、対応する昇順列 $(1, 2, 2, 2, 2, 3)$ はその接頭和表現の解になっている。また、2 進列 $(1, 0, 1, 0, 0, 0)$ と対応する昇順列 $(1, 1, 2, 2, 2, 2)$ のどちらも、それぞれの解ではない。

次に、 $\mathbf{S}_n$ に直積順序を導入する。すなわち $\alpha = (v_1, v_2, \dots, v_n) \in \mathbf{S}_n$, $\alpha' = (v'_1, v'_2, \dots, v'_n) \in \mathbf{S}_n$ のとき、 $\alpha \preceq \alpha' \iff \forall i(v_i \leq v'_i)$ により $\mathbf{S}_n$ 上の順序 $\preceq$ を定義する。このとき、 $(\mathbf{S}_n, \preceq)$ は分配束を成す。結び $\alpha \sqcup \alpha'$ は $(\max(v_1, v'_1), \dots, \max(v_n, v'_n))$ となり、交わり $\alpha \sqcap \alpha'$ は $(\min(v_1, v'_1), \dots, \min(v_n, v'_n))$ となる。また、 $\preceq$ の定義および $b_i \geq 0$ から明らかに、 $\alpha \in \mathbf{S}_n$ が接頭和表現 $\sum_{i=1}^n b_i s_i \geq c$ の解のとき、 $\alpha \preceq \alpha'$ を満たすすべての $\alpha' \in \mathbf{S}_n$ も解である。

次に、 $\mathbf{X}_n$ 上の順序 $\preceq$ を対応する昇順列の順序 $\preceq$ として定義する。

定義 3 ($\mathbf{X}_n$ 上の順序 $\preceq$). $\alpha = (u_1, u_2, \dots, u_n) \in \mathbf{X}_n$,

$\alpha' = (u'_1, u'_2, \dots, u'_n) \in \mathbf{X}_n$ とする。 $\mathbf{X}_n$ 上の順序 $\preceq$ を以下のように定義する。

$$\alpha \preceq \alpha' \iff (u_1, u_1 + u_2, \dots, u_1 + \dots + u_n) \preceq (u'_1, u'_1 + u'_2, \dots, u'_1 + \dots + u'_n)$$

ただし右辺の $\preceq$ は上記の直積順序である。

$(\mathbf{X}_n, \preceq)$ は分配束を成し、結び $\alpha \sqcup \alpha'$ と交わり $\alpha \sqcap \alpha'$ も対応する $\mathbf{S}_n$ の要素の結びと交わりとして定義できる。そして、 $\alpha \in \mathbf{X}_n$ が PB 制約 $\sum_{i=1}^n a_i x_i \geq c$ の解のとき、 $\alpha \preceq \alpha'$ を満たすすべての $\alpha' \in \mathbf{X}_n$ も解である。また、以下の命題が成り立つ。

命題 1. $\alpha, \alpha' \in \mathbf{X}_n$ とする。 $\alpha \preceq \alpha'$ ならば、 X_n 上の任意の PB 制約 ϕ について $\alpha \in \text{Sol}(\phi) \Rightarrow \alpha' \in \text{Sol}(\phi)$ である。逆もまた成り立つ。

Proof. ($\Rightarrow$) 明らか。($\Leftarrow$) 対偶を示す。 $\alpha \not\preceq \alpha'$ とする。 α, α' の接頭和表現をそれぞれ $(v_1, \dots, v_n)$, $(v'_1, \dots, v'_n)$ とすると、 $v_i > v'_i$ となる i が存在する。そこで ϕ を $x_1 + \dots + x_i \geq v_i$ と置くと、 $\alpha \in \text{Sol}(\phi)$ かつ $\alpha' \notin \text{Sol}(\phi)$ である。□

命題 1 より、PB 制約 ϕ の解の集合 $\text{Sol}(\phi)$ は $(\mathbf{X}_n, \preceq)$ の上方集合になっている。すなわち $\alpha \in \text{Sol}(\phi)$ ならば、 $\alpha \preceq \alpha'$ を満たす任意の $\alpha' \in \mathbf{X}_n$ について $\alpha' \in \text{Sol}(\phi)$ である。したがって、 $\text{Sol}(\phi)$ の極小元の集合を $\min(\text{Sol}(\phi))$ で表すと、 $\text{Sol}(\phi) = \{\alpha' \in \mathbf{X}_n \mid \alpha \in \min(\text{Sol}(\phi)), \alpha \preceq \alpha'\}$ が成り立つ。

また、補集合 $\overline{\text{Sol}(\phi)} = \mathbf{X}_n \setminus \text{Sol}(\phi)$ は、 $(\mathbf{X}_n, \preceq)$ の下方集合 (lower set) になっている。したがって、 $\overline{\text{Sol}(\phi)}$ の極大元の集合を $\max(\overline{\text{Sol}(\phi)})$ で表すと、 $\overline{\text{Sol}(\phi)} = \{\alpha' \in \mathbf{X}_n \mid \alpha \in \max(\overline{\text{Sol}(\phi)}), \alpha' \preceq \alpha\}$ である。

解の補集合 $\overline{\text{Sol}(\phi)}$ について、さらに考察する。1 の PB 制約 $5x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5 + x_6 \geq 9$ を ϕ で表したとき、 $\max(\overline{\text{Sol}(\phi)})$ は $\{(0, 1, 1, 0, 0, 1), (1, 1, 0, 0, 0, 0)\}$ である。すなわち $\overline{\text{Sol}(\phi)}$ は、 $\alpha \preceq (0, 1, 1, 0, 0, 1) \vee \alpha \preceq (1, 1, 0, 0, 0, 0)$ を満たす $\alpha \in \mathbf{X}_4$ の集合に一致する。 $(0, 1, 1, 0, 0, 1)$ に対応する $\mathbf{S}_4$ の要素は $(0, 1, 2, 2, 2, 3)$ だから、 $\alpha \preceq (0, 1, 1, 0, 0, 1)$ の条件は、接頭和を用いて $s_1 \leq 0 \wedge s_2 \leq 1 \wedge s_3 \leq 2 \wedge s_4 \leq 2 \wedge s_5 \leq 2 \wedge s_6 \leq 3$

と表せる. 同様に $\alpha \preceq (1, 1, 0, 0, 0, 0)$ の条件は $s_1 \leq 1 \wedge s_2 \leq 2 \wedge s_3 \leq 2 \wedge s_4 \leq 2 \wedge s_5 \leq 2 \wedge s_6 \leq 2$ と表せる. $\overline{Sol(\phi)}$ の要素である条件は, これら 2 つの選言となり, 接頭和を用いた選言標準形 (Disjunctive Normal Form; DNF) で表せることがわかる.

$Sol(\phi)$ の要素である条件は, $\overline{Sol(\phi)}$ の条件の否定だから, 以下の命題を得る.

命題 2. X_n 上の任意の PB 制約 ϕ に対し, X_n 上の BC-CNF 式 ψ が存在し $Sol(\phi) = Sol(\psi)$ である.

Proof. $\overline{Sol(\phi)}$ の極大元の集合を $\{\alpha_1, \dots, \alpha_m\}$ とし, 各 α_j の接頭和表現を $(v_{1j}, \dots, v_{nj})$ とする. BC-CNF 式 $\bigwedge_{j=1}^m \bigvee_{i=1}^n (s_i \geq v_{ij} + 1)$ を ψ と置くと, $Sol(\phi) = Sol(\psi)$ である. $\square$

なお, 例 1 の PB 制約 $5x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5 + x_6 \geq 9$ の場合, 後述する余分な BC リテラルの削除の処理を行うと BC-CNF 式 $(s_1 \geq 1 \vee s_5 \geq 3) \wedge (s_6 \geq 3)$ が得られる.

3.2 接頭和表現の BC-CNF 式への符号化

接頭和表現中の各 s_i を整数変数と見なし, 論文[16]の順序符号化アルゴリズムを用いれば, 接頭和表現を BC-CNF 式に符号化することが可能である. そのアルゴリズムを Algorithm 1 に示す. このアルゴリズムの引数は PB 制約の接頭和表現 $b_1s_1 + \dots + b_ns_n \geq c$ である. 例えば, 例 1 の PB 制約を符号化する際の呼び出しは $\text{Encode}(2s_1 + 0s_2 + 0s_3 + 0s_4 + 2s_5 + s_6 \geq 9)$ となる. また, アルゴリズム中で $\text{Dom}(s_i)$ は接頭和 s_i のドメイン $\{0, 1, \dots, i\}$ を返す関数である.

Algorithm 1 の 7 行目では $b_{i+1}s_{i+1} + \dots + b_ns_n \geq c - b_id$ が充足可能となる値 d の集合を D として求めており, 8 行目では充足不能となる値の集合を D' として求めている. 11 行目では, D' が空集合でない場合における節を追加している.

しかし, Algorithm 1 の場合, 各 s_i のドメイン $\{0, 1, \dots, i\}$ はそれぞれ独立していると見なされ, D と D' の計算において接頭和間の関係 $s_i \leq s_{i+1} \leq s_i + 1$ が考慮されておらず, 多数の無駄な BC 節が生成される. 例えば, 例 1 に示した接頭和表現 $2s_1 + 2s_5 + s_6 \geq 9$ の場合, 以下のような 8 個の BC 節となり余分な BC

節が含まれている.

$$\begin{aligned} C_1 &: (s_1 \geq 1) \vee (s_5 \geq 2) \\ C_2 &: (s_1 \geq 1) \vee (s_5 \geq 3) \vee (s_6 \geq 5) \\ C_3 &: (s_1 \geq 1) \vee (s_5 \geq 4) \vee (s_6 \geq 3) \\ C_4 &: (s_1 \geq 1) \vee (s_5 \geq 5) \vee (s_6 \geq 1) \\ C_5 &: (s_5 \geq 1) \\ C_6 &: (s_5 \geq 2) \vee (s_6 \geq 5) \\ C_7 &: (s_5 \geq 3) \vee (s_6 \geq 3) \\ C_8 &: (s_5 \geq 4) \vee (s_6 \geq 1) \end{aligned}$$

そこで, 接頭和間の関係 $s_i \leq s_{i+1} \leq s_i + 1$ を符号化アルゴリズムに組み込んだ改良を行った.

改良した順序符号化アルゴリズムを Algorithm 2 に示す. このアルゴリズムの第 1 引数は PB 制約の接頭和表現 $b_1s_1 + \dots + b_ns_n \geq c$, 第 2 引数は直前の係数が 0 でない接頭和のインデックス i_0 , 第 3 引数はその接頭和に対する値 d_0 である. 最初の呼び出しでは, 接頭和表現を $b_1s_1 + \dots + b_ns_n \geq c$, $i_0 = 0$, $d_0 = 0$ とする. 例えば, 例 1 の PB 制約を符号化する際の呼び出しは $\text{Encode}(2s_1 + 0s_2 + 0s_3 + 0s_4 + 2s_5 + s_6 \geq 9, 0, 0)$ となる.

Algorithm 2 では, 7, 8 行目において, $s_i \leq s_{i+1} \leq s_i + 1$ の関係を用い, ありえない値を D と D' から省いている. これにより, 多くの無駄な BC 節の生成を避けることが可能になった. 例えば, 上記の接頭和表現は以下の例 2 に示すように 3 個の BC 節に符号化され, 大幅な BC 節数の削減を実現できている.

例 2. Algorithm 2 を例 1 に示した接頭和表現 $2s_1 + 2s_5 + s_6 \geq 9$ に適用した場合, 以下の 3 個の BC 節に符号化される.

$$\begin{aligned} C_1 &: (s_1 \geq 1) \vee (s_5 \geq 3) \\ C_2 &: (s_5 \geq 2) \\ C_3 &: (s_5 \geq 3) \vee (s_6 \geq 3) \end{aligned}$$

3.3 BC-CNF 式の簡単化

前節では, 論文[16]の順序符号化アルゴリズムの改良方法について述べ, 大幅な BC 節数の削減が実現できることを示した. しかし, まだ無駄な BC リテラルや BC 節が存在する.

本節では, より一般的に BC 節から不要な BC リテラルを除去する方法, および BC-CNF 式から不要

Algorithm 1: Encode($b_i s_i + \dots + b_n s_n \geq c$)

```

1: if  $b_i = 0$  then
2:    $\psi \leftarrow \text{Encode}(b_{i+1}s_{i+1} + \dots + b_n s_n \geq c)$ 
3: else
4:   if  $i = n$  then
5:      $\psi \leftarrow \{(s_i \geq \lceil c/b_i \rceil)\}$ 
6:   else
7:      $D \leftarrow \{d \in \text{Dom}(s_i) \mid c - \sum_{j=i+1}^n b_j j \leq b_i d < c\}$ 
8:      $D' \leftarrow \{d \in \text{Dom}(s_i) \mid b_i d < c - \sum_{j=i+1}^n b_j j\}$ 
9:      $\psi \leftarrow \{(s_i \geq d+1)\} \cup \theta \mid d \in D, \theta \in \text{Encode}(b_{i+1}s_{i+1} + \dots + b_n s_n \geq c - b_i d)\}$ 
10:    if  $D' \neq \emptyset$  then
11:       $\psi \leftarrow \{(s_i \geq \max(D') + 1)\} \cup \psi$ 
12:    end if
13:  end if
14: end if
15: return  $\psi$ 

```

Algorithm 2: Encode($b_i s_i + \dots + b_n s_n \geq c, i_0, d_0$)

```

1: if  $b_i = 0$  then
2:    $\psi \leftarrow \text{Encode}(b_{i+1}s_{i+1} + \dots + b_n s_n \geq c, i_0, d_0)$ 
3: else
4:   if  $i = n$  then
5:      $\psi \leftarrow \{(s_i \geq \lceil c/b_i \rceil)\}$ 
6:   else
7:      $D \leftarrow \{d \in \text{Dom}(s_i) \mid d_0 \leq d \leq i - i_0 + d_0, c - \sum_{j=i+1}^n b_j(j-i) - \sum_{j=i+1}^n b_j d \leq b_i d < c - \sum_{j=i+1}^n b_j d\}$ 
8:      $D' \leftarrow \{d \in \text{Dom}(s_i) \mid b_i d < c - \sum_{j=i+1}^n b_j(j-i) - \sum_{j=i+1}^n b_j d\}$ 
9:      $\psi \leftarrow \{(s_i \geq d+1)\} \cup \theta \mid d \in D, \theta \in \text{Encode}(b_{i+1}s_{i+1} + \dots + b_n s_n \geq c - b_i d, i, d)\}$ 
10:    if  $D' \neq \emptyset$  then
11:       $\psi \leftarrow \{(s_i \geq \max(\max(D') + 1, d_0))\} \cup \psi$ 
12:    end if
13:  end if
14: end if
15: return  $\psi$ 

```

な BC 節を除去する方法について述べる.

定義 4. $0 \leq a \leq i, 0 \leq b \leq j$ のとき, X_n 上の BC リテラル $s_i \geq a, s_j \geq b$ 間の含意関係を以下のように定義する.

$$s_i \geq a \Rightarrow s_j \geq b \iff (i \leq j \wedge a \geq b) \vee (i \geq j \wedge i - a \leq j - b)$$

BC リテラルについて以下が成り立つ.

命題 3. X_n 上の BC リテラル l, l' について以下が成り立つ.

$$l \Rightarrow l' \iff \text{Sol}(l) \subseteq \text{Sol}(l')$$

Proof. ($\Rightarrow$) 明らか. ($\Leftarrow$) 対偶を示す. l と l' をそれぞれ $s_i \geq a$ と $s_j \geq b$ と置き, $l \not\Rightarrow l'$ とす

る. このとき, $i \leq j$ ならば $a < b$ が成り立ち, $i > j$ ならば $i - a > j - b$ が成り立つ. また, $\alpha = (u_1, \dots, u_n) \in \mathbf{X}_n$ とする. $i \leq j$ の場合, $u_1 = u_2 = \dots = u_a = 1, u_{a+1} = \dots = u_n = 0$ と置くと $\alpha \in \text{Sol}(s_i \geq a)$ であるが $\alpha \notin \text{Sol}(s_j \geq b)$ である. $i > j$ の場合, $u_1 = u_2 = \dots = u_{i-a} = 0, u_{i-a+1} = \dots = u_i = 1, u_{i+1} = \dots = u_n = 0$ と置くと $\alpha \in \text{Sol}(s_i \geq a)$ であるが $\alpha \notin \text{Sol}(s_j \geq b)$ である. $\square$

この命題を用いることで不要な BC リテラルの除去を実現できる. すなわち, BC 節中の異なる BC リテラル l, l' について $l \Rightarrow l'$ が成り立つとき, その BC

節から BC リテラル l を除去して良い。

例 3. 例 2 の BC-CNF 式において, $s_5 \geq 3 \Rightarrow s_6 \geq 3$ より C_3 中の BC リテラル $s_5 \geq 3$ を除去できる。

定義 5. X_n 上の BC 節 C, C' 間の含意関係を以下のように定義する。

$$C \Rightarrow C' \iff \forall l \in C \exists l' \in C' (l \Rightarrow l')$$

以下の補題と命題が成り立つ。

補題 1. X_n 上の BC リテラル l, l_1, l_2 について, $Sol(l) \subseteq Sol(l_1 \vee l_2)$ ならば, $Sol(l) \subseteq Sol(l_1)$ または $Sol(l) \subseteq Sol(l_2)$ が成り立つ。

Proof. $Sol(l)$ が空の場合は明らか。空でなければ最小元 α を持つ。 α は $Sol(l_1)$ または $Sol(l_2)$ に含まれるから, $Sol(l) \subseteq Sol(l_1)$ または $Sol(l) \subseteq Sol(l_2)$ が成り立つ。 $\square$

命題 4. X_n 上の BC 節 C, C' について以下が成り立つ。

$$C \Rightarrow C' \iff Sol(C) \subseteq Sol(C')$$

Proof. ($\Rightarrow$) 明らか。 ($\Leftarrow$) $\exists l \in C \forall l' \in C' (l \not\Rightarrow l')$ と仮定し矛盾を導く。仮定を満たす BC リテラルを $l \in C$ とする。 $Sol(C) \subseteq Sol(C')$ より $Sol(l) \subseteq Sol(C')$ であり, 補題 1 より, ある $l' \in C'$ について $Sol(l) \subseteq Sol(l')$ である。したがって, 命題 3 より $l \Rightarrow l'$ だが, これは仮定に矛盾する。 $\square$

この命題を用いることで**不要な BC 節の除去**を実現できる。すなわち, BC-CNF 式中の異なる BC 節 C, C' について $C \Rightarrow C'$ が成り立つとき, その BC-CNF 式から BC 節 C' を除去して良い。

例 4. 例 3 の BC-CNF 式において, $C_3 \Rightarrow C_2$ より BC 節 C_2 を除去でき, 以下を得る。

$$C_1 : (s_1 \geq 1) \vee (s_5 \geq 3)$$

$$C_3 : (s_6 \geq 3)$$

後述の 4.2 節で示すように, 上記の簡単化を繰り返すことで, 同値な PB 制約からは同一の BC-CNF 式が得られる。そこで, 不要な BC リテラルをすべて除去した BC 節を**既約な BC 節**と呼び, さらに不要な BC 節をすべて除去した BC-CNF 式を**既約な BC-CNF 式**と呼ぶ。

3.4 BC-CNF 式の SAT 符号化

BC 制約を SAT に符号化する良い手法の 1 つとして知られているシーケンシャルカウンタ法[15]を用いれば, BC-CNF 式を容易に SAT 符号化できる。

$\sum_{i=1}^n b_i s_i \geq c$ を PB 制約の接頭和表現とする。 $s_i \geq a$ を意味するブール変数 $s_{i,a}$ を導入し, シーケンシャルカウンタ法により以下のような節を生成する。

$$s_{i-1,a-1} \vee \neg s_{i,a} \quad (1 \leq a \leq n, 1 \leq i \leq n)$$

$$s_{i-1,a} \vee x_i \vee \neg s_{i,a} \quad (1 \leq a \leq n, 1 \leq i \leq n)$$

ただし, リテラル $s_{i,a}$ ($i < a$) を含む節から $s_{i,a}$ を消去し, $s_i \geq 0$ を含む節を削除する。このとき, $\sum_{i=1}^n b_i s_i \geq c$ を符号化した BC-CNF 式において, 各 BC リテラル $s_i \geq a$ を命題変数 $s_{i,a}$ で置き換えれば良い。

例 5. 例 4 の BC-CNF 式を SAT 符号化した結果は, 以下の節で与えられる。

$$C_1 : s_{1,1} \vee s_{5,3}$$

$$C_3 : s_{6,3}$$

さらに, 各命題変数 $s_{i,a}$ が $x_1 + \dots + x_i \geq a$ を表すための節を加える。この例の場合, $x_1 + x_2 + \dots + x_6 \geq 3$ をシーケンシャルカウンタ法で SAT 符号化した節を加えれば良い。

また, シーケンシャルカウンタ法では, SAT ソルバーの単位伝播により一般化アーク整合性維持が可能であることが知られている [15]。

4 提案手法の特徴

本節では以下に示す提案手法の 3 つの特徴について説明する。

- SAT ソルバーの単位伝播により一般化アーク整合性の維持が可能。
- 同じ解を持つ同値な PB 制約であれば, 係数や右辺の値が異なっても, 同一の中間表現および SAT 問題に符号化可能。
- 異なった係数の種類が比較的少ない PB 制約に対しては, 中間表現が簡潔になり少ない節数で SAT 符号化可能。

さらに, 提案手法の節数が指数関数的に増加するような場合についても考察を行う。

4.1 単位伝播による一般化アーク整合性の維持

提案手法は、シーケンシャルカウンタと同じく一般化アーク整合性の維持が可能であり、したがって整合性検査も維持される。

提案手法で変換された SAT 問題は、シーケンシャルカウンタを表す部分と BC-CNF 式を表す部分に分かれている。リテラル列 X_n に対する部分的な値割当てを α とする。シーケンシャルカウンタ部分により、 α により成立する BC リテラル $s_i \geq a$ に対応する変数 $s_{i,a}$ は単位伝播で 1 に割当てられ、矛盾する BC リテラル $s_i \geq a$ に対応する変数 $s_{i,a}$ は 0 に割当てられる。したがって、 α で元の PB 制約が矛盾する場合、BC-CNF 式中のある BC 節が矛盾することが単位伝播で検出でき、整合性検査が維持される。また、 α で元の PB 制約は矛盾せず、 α に $x_i = 0$ を追加した α' で矛盾する場合、BC-CNF 式中のある BC 節 C について、それは α では矛盾せず α' で矛盾する。すなわち C 中のある BC リテラル $s_{j,a}$ は、 α で未定義であるが、 α' で 0 になる。したがって、 α のもとで、 $s_{j,a} = 1$ が単位伝播され、さらにシーケンシャルカウンタ部分に対する単位伝播で $x_i = 1$ が得られる。

例 6. PB 制約 $3x_1 + 2x_2 + 2x_3 + x_4 + x_5 \geq 5$ を例に取る。部分値割当て $x_2 = x_3 = 0$ の状況から、 $x_1 = x_4 = x_5 = 1$ が SAT ソルバーの単位伝播で導出されなければならない。提案手法で得られている CNF 式は以下の通りである。

$$\begin{array}{ll}
 C_1 : x_1 \vee \neg s_{1,1} & C_2 : s_{1,1} \vee x_2 \vee \neg s_{2,1} \\
 C_3 : s_{1,1} \vee \neg s_{2,2} & C_4 : x_2 \vee \neg s_{2,2} \\
 C_5 : s_{2,1} \vee x_3 \vee \neg s_{3,1} & C_6 : s_{2,1} \vee \neg s_{3,2} \\
 C_7 : s_{2,2} \vee x_3 \vee \neg s_{3,2} & C_8 : s_{2,2} \vee \neg s_{3,3} \\
 C_9 : x_3 \vee \neg s_{3,3} & C_{10} : s_{3,1} \vee x_4 \vee \neg s_{4,1} \\
 C_{11} : s_{3,1} \vee \neg s_{4,2} & C_{12} : s_{3,2} \vee x_4 \vee \neg s_{4,2} \\
 C_{13} : s_{3,2} \vee \neg s_{4,3} & C_{14} : s_{3,3} \vee x_4 \vee \neg s_{4,3} \\
 C_{15} : s_{4,1} \vee x_5 \vee \neg s_{5,1} & C_{16} : s_{4,1} \vee \neg s_{5,2} \\
 C_{17} : s_{4,2} \vee x_5 \vee \neg s_{5,2} & C_{18} : s_{4,2} \vee \neg s_{5,3} \\
 C_{19} : s_{4,3} \vee x_5 \vee \neg s_{5,3} & C_{20} : s_{3,2} \vee s_{5,3} \\
 C_{21} : s_{1,1} \vee s_{3,2} & C_{22} : s_{1,1} \vee s_{5,3}
 \end{array}$$

C_1 から C_{19} はシーケンシャルカウンタの部分であり、 C_{20} から C_{22} が BC-CNF 式に対応した部分である。 $x_2 = x_3 = 0$ のとき、 C_4, C_7, C_9, C_{13} から、単

位伝播により $s_{2,2} = s_{3,2} = s_{3,3} = s_{4,3} = 0$ が導かれる。さらに、BC-CNF 式に対応した節 C_{20}, C_{21} から $s_{1,1} = s_{5,3} = 1$ 、すなわち $s_1 \geq 1$ と $s_5 \geq 3$ が単位伝播で得られる。最後に、再度シーケンシャルカウンタ部分での単位伝播により $x_1 = x_4 = x_5 = 1$ となる。

このように提案手法では SAT ソルバーの単位伝播により一般化アーク整合性の維持が可能であり、符号化後の SAT 問題を SAT ソルバーで解く際に、効率的な枝刈りが実現され、無駄な探索を避けることができる。

4.2 同値な PB 制約に対する符号化

以下では、同値な PB 制約から得られる既約な BC-CNF 式は同一になることの証明を示す。まず、既約な BC 節の同値性について、以下の補題と命題が成り立つ。

補題 2. X_n 上の既約な BC 節 $C = \{l_1, \dots, l_m\}$ について、 C 中の BC リテラル l_i は $Sol(C)$ の極小元 α_i と 1 対 1 に対応し、 $\{\alpha_i\} = \min(Sol(l_i))$ である。

Proof. 恒偽でない BC リテラル l について、 $Sol(l)$ はただ 1 つの極小元 (最小元) を持つ。そこで、各 $l_i \in C$ について $Sol(l_i)$ の最小元を α_i と置く。このとき、 $i \neq j$ ならば $\alpha_i \not\leq \alpha_j$ かつ $\alpha_j \not\leq \alpha_i$ が成り立つ。なぜなら、 $\alpha_i \leq \alpha_j$ とすると $Sol(l_j) \subseteq Sol(l_i)$ であり、命題 3 より $l_j \Rightarrow l_i$ が成り立ち、 C が既約であることに反する。 $\alpha_j \leq \alpha_i$ の場合も同様である。したがって、各 α_i は $Sol(C)$ の極小元になっており、 $i \neq j$ ならば $\alpha_i \neq \alpha_j$ である。

逆に、 α を $Sol(C)$ の任意の極小元とする。 α はある BC リテラル l_i の最小解になっている。□

命題 5. X_n 上の既約な BC 節 C, C' について以下が成り立つ。

$$C = C' \iff Sol(C) = Sol(C')$$

Proof. ($\implies$) 明らか。($\impliedby$) C と C' の極小解を $\{\alpha_1, \dots, \alpha_m\}$ とする。補題 2 より、 C は $\{\alpha_i\} = \min(Sol(l_i))$ を満たす BC リテラル l_i の集合であり、 C' もまたそうだから、 $C = C'$ である。□

既約な BC-CNF 式の同等性について、以下の補題と定理が成り立つ。

補題 3. $\psi = \{C_1, \dots, C_m\}$ を X_n 上の既約な BC-CNF 式とする。 ψ の BC 節 C_i は ψ の解の補集合の極大元 $\alpha_i \in \max(\overline{\text{Sol}(\psi)})$ と 1 対 1 に対応し、 $\{\alpha_i\} = \max(\text{Sol}(\neg C_i))$ である。

Proof. 恒真でない BC リテラル $l = (s_i \geq a)$ について、 $\neg l = (s_i \leq a - 1)$ の解 $\text{Sol}(\neg l)$ はただ 1 つの極大元を持つ。したがって、恒真でない BC 節 C について、 $\text{Sol}(\neg C) = \bigcap_{l \in C} \text{Sol}(\neg l)$ はただ 1 つの極大元 (最大元) を持つ。

そこで、各 $C_i \in \psi$ について $\text{Sol}(\neg C_i)$ の最大元を α_i と置く。すなわち、 $\text{Sol}(\neg C_i) = \{\alpha \in \mathbf{X}_n \mid \alpha \preceq \alpha_i\}$ である。このとき、 $i \neq j$ ならば $\alpha_i \not\preceq \alpha_j$ かつ $\alpha_j \not\preceq \alpha_i$ が成り立つ。なぜなら、 $\alpha_i \preceq \alpha_j$ とすると $\text{Sol}(\neg C_i) \subseteq \text{Sol}(\neg C_j)$ であり、命題 4 より $C_j \Rightarrow C_i$ が成り立ち、 ψ が既約であることに反する。 $\alpha_j \preceq \alpha_i$ の場合も同様である。したがって、各 α_i は $\overline{\text{Sol}(\psi)}$ の極大元になっており、 $i \neq j$ ならば $\alpha_i \neq \alpha_j$ である。

逆に、 α を $\overline{\text{Sol}(\psi)}$ の任意の極大元とする。 $\text{Sol}(\psi) = \bigcup_{C \in \psi} \text{Sol}(\neg C_i)$ より、 α が極大元になっている $\text{Sol}(\neg C_i)$ が存在する。 $\square$

定理 1. X_n 上の既約な BC-CNF 式 ψ, ψ' について以下が成り立つ。

$$\psi = \psi' \iff \text{Sol}(\psi) = \text{Sol}(\psi')$$

Proof. ($\implies$) 明らか。 ($\impliedby$) 補題 3 より、 ψ の BC 節 C と ψ' の BC 節 C' は 1 対 1 に対応し、 $\text{Sol}(\neg C) = \text{Sol}(\neg C')$ 、すなわち $\text{Sol}(C) = \text{Sol}(C')$ である。さらに、命題 5 より $C = C'$ である。 $\square$

以上により、同値な PB 制約に対する既約な BC-CNF 式は常に同一になることが示せた。

2 つの PB 制約が同値であるかどうかの判定問題は co-NP 完全なので、co-NP $\neq$ P を仮定すると、既約な BC-CNF 式を求める多項式時間の手続きは存在しない。

4.3 PB 制約の項数に対して係数の種類が少ないときの符号化節数

提案手法には、PB 制約の項数に対して係数の種類が少ないときに、PB 制約を効果的に SAT 符号化できる、別の言い方をすると、PB 制約の同じ係数の変数の数が増えても符号化節数が増えにくいという長所がある。

3.2 節で述べた PB 制約 $5x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5 + x_6 \geq 9$ 、および、その接頭和表現 $2s_1 + 2s_5 + s_6 \geq 9$ について考える。提案手法で符号化すると以下の BC-CNF 式が得られる。

$$C_1 : (s_1 \geq 1) \vee (s_5 \geq 3)$$

$$C_2 : (s_5 \geq 2)$$

$$C_3 : (s_5 \geq 3) \vee (s_6 \geq 3)$$

この 6 項の PB 制約に対し、係数が 3 である変数を 4 つ増やした 10 項の PB 制約 $5x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5 + 3x_6 + 3x_7 + 3x_8 + 3x_9 + x_{10} \geq 9$ 、および、その接頭和表現 $2s_1 + 2s_9 + s_{10} \geq 9$ について考える。この PB 制約を BC-CNF 式に符号化すると、以下の BC-CNF 式が得られ、これは上述の 6 項の場合の BC-CNF 式における接頭和 s_5, s_6 が、 s_9, s_{10} に置き換わったものとなっている。

$$C_1 : (s_1 \geq 1) \vee (s_9 \geq 3)$$

$$C_2 : (s_9 \geq 2)$$

$$C_3 : (s_9 \geq 3) \vee (s_{10} \geq 3)$$

このように、提案手法では、接頭和表現により同じ係数の変数の値の和の範囲によって場合分けされるから、PB 制約の項数が増えても、同じ係数であれば、接頭和表現でまとめて表すことができる。よって BC-CNF 式の BC 節数は増えない。

この場合、SAT 符号化後の SAT 問題におけるシーケンシャルカウンタを表す部分のみ節数が増える。4.4 節で述べるように、シーケンシャルカウンタで生成される節数は多項式的である。この例だと、最終的に提案手法で生成される SAT 問題の節数は、前者の PB 制約が 20 節、後者の PB 制約が 40 節となる。

一方、BDD 法 [7] の場合を考えると、変数を取りうる値の場合分けを行うから、上述の例で 6 項から 10 項に増えたときに、前者の PB 制約が 28 節、後者の PB 制約が 72 節となり、提案手法に比べ節数の増え

方が大きくなる。

また、後述の、2016 年国際 PB ソルバー競技会 DEC-SMALLINT-LIN 部門での問題 1783 問では、PB 制約の項数に対する係数の種類の数の割合が最大で 10% 以下の問題が 856 問あり、項数に対して係数の種類が少ない PB 制約は多く存在する。そのような PB 制約に対し、提案手法は、同じく単位伝播により一般化アーク整合性を維持できる BDD 法と比べ、節数が小さな SAT 問題に符号化することができる。

4.4 BC-CNF 式が指数関数的に増大する場合

提案手法で PB 制約を SAT 符号化した場合、シーケンシャルカウンタ $x_1 + \dots + x_n \geq k$ の部分から生成される節数は $O(nk)$ で多項式的であるが[15]、BC-CNF 式の部分は指数的に増大する場合が存在する。

例えば、PB 制約 $\sum_{i=1}^n (n-i+1)x_i \geq \lfloor n(n+1)/4 \rfloor + 1$ の場合、BC-CNF 式中の BC 節数は、集合 $\{1, 2, \dots, n\}$ の部分集合でその和が $\lfloor n(n+1)/4 \rfloor$ に一致する個数となり、その値は $\sqrt{6/\pi} 2^{n-3/2}$ で近似される^{†3}。したがって、指数オーダーの BC 節数が必要となる。

5 計算機実験

本節では、提案手法と既存手法である BDD 法および Sorter 法[7] に対し、比較のために行った計算機実験とその結果について説明し、提案手法の有効性とその特徴を評価する。なお Adder 法で生成される節数のオーダーは $O(n)$ であるが、SAT ソルバーの単位伝播では一般化アーク整合性も整合性検査も維持されないため、本論文では比較の対象としない。

最初に、異なった係数の種類が比較的少ない PB 制約について、ランダムに生成した PB 制約を用い、提案手法 (BC-CNF 式の簡単化あり、BC-CNF 式の簡単化なし) と既存手法で生成される節数について比較する。このような問題は提案手法が有効に働く場合であり、既存手法との違いを明らかにする。

次に、2016 年国際 PB ソルバー競技会の DEC-

SMALLINT-LIN 部門の 1783 問を用い、求解性能 (時間制限内に解けた問題数) を比較する。また、これらの問題を異なった係数の種類が比較的少ない問題とそうでない問題に分類し、既存手法との違いについて評価する。

なお、提案手法の実装は、次の 4 つのプログラムから構成されている: i) PB 制約を接頭和にするプログラム, ii) 接頭和を BC-CNF 式に符号化するプログラム, iii) BC-CNF 式を簡単化するプログラム, iv) BC-CNF 式を SAT 問題に符号化するプログラム。また、プログラムはすべて Scala 言語を用いて記述されている。

比較には、論文 [7] で提案され、代表的な PB 制約の SAT 符号化手法となっている以下の 2 つを用いた。

- BDD 法: 二分決定木を用いた手法。
- Sorter 法: ソーティング回路を用いた手法。

実験では、これらの SAT 符号化手法の実装として、国際 PB ソルバー競技会における優勝ソルバーの 1 つである MiniSat+^{†4} を用いた。

5.1 ランダムに生成した PB 制約での比較

本節では、最初に、異なった係数の種類が比較的少ない PB 制約について、ランダムに生成した制約を用い、提案手法と既存手法で生成される節数について比較する。

まず、係数種類の割合を以下のように定義する。

$$\text{係数種類の割合} = \frac{\text{係数の種類の個数}}{\text{PB 制約の項数 } n}$$

例えば、50 項の PB 制約で、異なった係数が 5 種類現れている場合、係数種類の割合は 10% である。

提案手法は、異なった係数の種類が少なれば少ないほど生成される節数が減少するが、ここでは係数種類の割合が 10% 以下の場合について評価する。なお、2016 年国際 PB ソルバー競技会の DEC-SMALLINT-LIN 部門の 1783 問中、問題中に現れている全 PB 制約の係数種類の割合が 10% 以下になっているものは 856 問で、半数近くがそのような問題である。

最初に、係数種類の割合を 10% とした上で、以下の

^{†3} <https://oeis.org/A025591>

^{†4} <https://github.com/niklasso/minisatp>

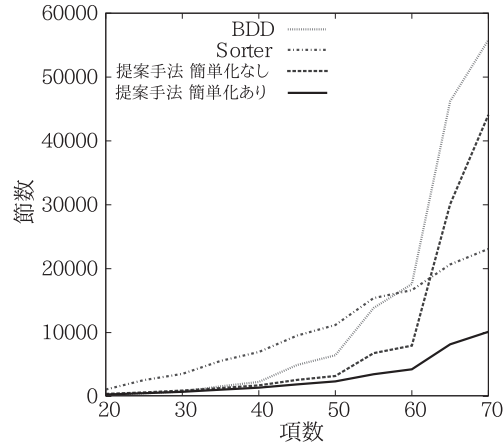


図 2 係数種類の割合が 10%である PB 制約に対する提案手法と既存手法の SAT 節数の比較

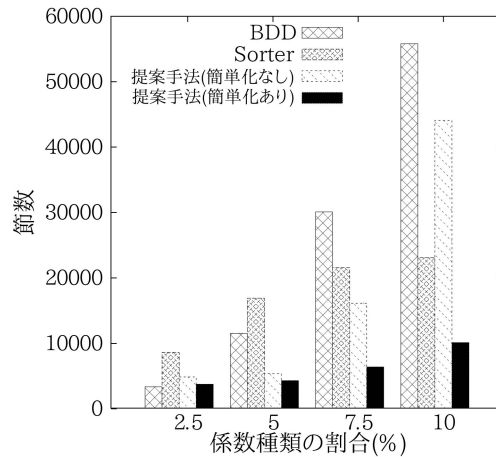


図 3 項数が 70 のときの係数種類の割合に対する提案手法と既存手法の SAT 節数の比較

条件で PB 制約 $\sum_{i=1}^n a_i x_i \geq c$ をランダムに生成した.

- 項数: $n \in \{20, 25, \dots, 70\}$
- 係数: $1 \leq a_i \leq 5000$
- 右辺: $c = \lceil \sum_{i=1}^n a_i / 2 \rceil$

この条件のもとで、各項数に対し 100 個の PB 制約を生成し、計 1100 個の PB 制約を生成した。なお、2016 年の国際 PB ソルバー競技会で使われた DEC-SMALLINT-LIN 部門の問題 1783 問の中、項数が 70 以下で係数が 5000 以下の問題は 1287 問である。

図 2 は、各項数に対する平均節数をプロットしたものである。提案手法 (BC-CNF 式の簡単化あり) が最も節数が小さくなっており、また、項数の増加に伴う

節数の増加量も他に比べ小さくなっている。

次に、項数が 70 について、係数種類の割合が 2.5%、5%、7.5%、10% となる PB 制約を上記と同様にランダムに各 100 問ずつ計 400 問生成した。図 3 にそれらの平均節数を示す。係数種類の割合が大きくなればなるほど提案手法の簡単化ありと簡単化なしでの差が大きくなり開いており、簡単化を行うことによって無駄な節を除くことができていることが確認できる。また、係数種類の割合が小さくなるにつれて提案手法の節数も小さくなっている。

これは、4.3 節で述べたように、提案手法は、同じ係数の変数が増えても節数が増えにくいという性質

表 1 DEC-SMALLINT-LIN 部門の問題 1783 間における提案手法と既存手法の解けた問題数と平均節数の比較

係数種類の割合の最大値 R	問題数	提案手法		BDD		Sorter	
		解けた 問題数	節数の 平均	解けた 問題数	節数の 平均	解けた 問題数	節数の 平均
$R \leq 2.5\%$	512	388	13972	381	18747	370	84683
$2.5\% < R \leq 5\%$	145	120	4526	118	4936	117	6657
$5\% < R \leq 7.5\%$	43	42	1848	42	2256	42	3627
$7.5\% < R \leq 10\%$	156	143	82305	144	116436	144	327687
$10\% < R$	927	458	160692	602	119741	570	188336
合計	1783	1151		1287		1243	

表 2 係数種類の割合が 10% 以下の問題 856 間における提案手法と既存手法の指定時間以内に解けた問題数

時間 (秒)	提案手法	BDD	Sorter
≤ 100	618	611	616
≤ 200	647	634	642
≤ 300	662	654	664
≤ 400	673	670	673
≤ 500	686	675	673
≤ 600	693	685	673

を持っている特徴が活かされたからだと考えられる。

なお、提案手法および BDD の場合、一般化アーク整合性が維持されるため SAT ソルバーの単位伝播だけで解が求まり、これらの問題に対する実際の求解時間はほぼ 0 となる。

5.2 国際 PB ソルバー競技会のベンチマーク問題での比較

本節では、2016 年国際 PB ソルバー競技会の DEC-SMALLINT-LIN 部門の 1783 間を用い、提案手法 (BC-CNF 式の簡単化あり) と既存手法の求解性能 (時間制限内に解けた問題数) を比較する。比較する既存手法は、前節と同じく BDD 法と Sorter 法の 2 種類である。また、これらの問題を異なった係数の種類が比較的少ない問題とそうでない問題に分類し、既存手法との違いについて評価する。

実験では、まず各手法によって PB 問題を制限時間 600 秒で符号化して SAT 問題を生成した。その後、SAT ソルバー MiniSat version 2.2 [5] を用いて生成された SAT 問題を解き、制限時間 600 秒以内で解け

た問題数を比較した。実験には CPU に Intel Xeon 3.00GHz、メモリ 16GB を搭載したマシンを用いた。

表 1 に各手法で解けた問題数と節数の平均値を示す。一番左の欄は係数種類の割合の最大値 R の範囲を表している。例えば、表内の 1 行目は R が 2.5% 以下の問題に対する値であり、2 行目は R が 2.5% より大きく 5% 以下の問題に対する値である。

係数種類の割合の最大値 R が 2.5%, 5%, 7.5% 以下である PB 問題に対して、提案手法が最も求解数が多く、平均節数も少なくなっており、7.5% より大きく 10% 以下の場合においては、求解数はわずかに少ないものの、節数は他手法より小さくなっている。

また、表 2 は係数種類の割合が 10% 以下の PB 制約だけが含まれている 856 間について、各手法で指定時間以内に解けた問題数を示している。提案手法を BDD 法と比較すると、制限時間をどの秒数で区切ったとしても解けた問題数は提案手法の方が多くなっている。Sorter 法と比較した場合、求解数、節数の差は BDD 法の場合よりも大きくなっており、制限時間が増えるにつれて提案手法との求解数の差も大きくなっ

ている。

以上から、異なった係数の種類が比較的少ない問題について、提案手法は既存手法である BDD 法、Sorter 法より優れた手法といえる。

しかし、係数種類の割合の最大値が 10% より大きい問題について見ると、節数の差はそれほど大きくないが、制限時間 600 秒以内に解けた問題数では大きく差がついた。これは、SAT 符号化自体が制限時間以内にできなかったことが 1 つの原因である。係数種類の割合が 10% 以下のもので符号化できなかった問題は 856 問中 10 問だったが、10% より大きい問題では 927 問中 158 問だった。例えば、係数種類の割合が 100% の PB 制約を含む **hw128** の問題では、BC-CNF 式のサイズが膨大となり、制限時間内での BC-CNF 式の生成に失敗していた。

符号化に失敗した原因をより詳細に調べたところ、要した時間のほとんどは 3.3 節で述べた BC-CNF 式の単純化の処理が占めていた。現在の実装では、BC 節の除去において、2 つの BC 節を選択するすべての組合せに対し、包摂の条件を調べている。このため、処理時間が増大したと考えられる。すべての組合せを調べないで BC 節の除去を実現する手法の考案は、今後の研究課題である。

6 結論

本論文では、BC 制約を経由した PB 制約の新しい SAT 符号化手法を提案した。提案手法では、与えられた PB 制約の接頭と表現を求め、順序符号化を改良したアルゴリズムを用いて BC-CNF 式に符号化し、無駄な BC リテラルと BC 節を除去した後、シーケンシャルカウンタ法を用いて SAT 符号化を行っている。提案手法は、次の 3 つの特徴をもつ。

- SAT ソルバーの単位伝播により一般化アーク整合性の維持が可能。
- 同じ解を持つ同値な PB 制約であれば、係数や右辺の値が異なっても、同一の中間表現および SAT 問題に符号化可能。
- 異なった係数の種類が比較的少ない PB 制約に対しては、中間表現が簡潔になり少ない節数で SAT 符号化可能。このような PB 制約は、国際

PB ソルバー競技会のベンチマーク問題にも頻出している。

計算機実験の結果、PB 制約の項数に対して係数の種類が 10% 以下であるようなランダムに生成した PB 制約に対しては、代表的な既存手法である BDD 法、Sorter 法よりも少ない節数で符号化できることが分かった。また 2016 年の国際 PB ソルバー競技会で使われた 1783 問の問題に対しては、やはり PB 制約の項数に対して係数の種類の割合の最大が 10% 以下であるような PB 問題に対して、他の既存手法よりも多くの問題を解くことができ有効であることが分かった。

今後の課題として、基数制約を直接扱うことができる SAT ソルバーの利用や [4]、計算機実験の節で議論したように、高速な符号化・単純化アルゴリズムを研究することが挙げられる。

謝辞 本研究は JSPS 科研費 16H02803 の助成を受けたものである。

参考文献

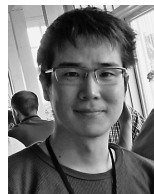
- [1] Gilles Audemard and Laurent Simon. Predicting learnt clauses quality in modern SAT solvers. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pp. 399–404, 2009.
- [2] Fahiem Bacchus. GAC via unit propagation. In *Proceedings of the 13th International Joint Conference on Principles and Practice of Constraint Programming (CP 2007)*, LNCS 4741, pp. 133–147, 2007.
- [3] Olivier Bailleux and Yacine Boufkhad. Efficient CNF encoding of Boolean cardinality constraints. In *Proceedings of the 9th International Joint Conference on Principles and Practice of Constraint Programming (CP 2003)*, LNCS 2833, pp. 108–122, 2003.
- [4] Daniel Le Berre and Anne Parrain. The Sat4j library, release 2.2. *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 7, No. 2–3, pp. 59–64, 2010.
- [5] Niklas Eén and Niklas Sörensson. An extensible SAT-solver. In *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT 2003)*, LNCS 2919, pp. 502–518, 2003.
- [6] Niklas Eén and Niklas Sörensson. MiniSat: A

- SAT solver with conflict-clause minimization. In *Proceedings of the 8th International Conference on Theory and Applications of Satisfiability Testing (SAT 2005)*, LNCS 3569, pp. 502–518, 2005.
- [7] Niklas Eén and Niklas Sörensson. Translating pseudo-Boolean constraints into SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, Vol. 2, No. 1–4, pp. 1–26, 2006.
- [8] 井上克巳, 田村直之. SAT ソルバーの基礎. 人工知能学会誌, Vol. 25, No. 1, pp. 57–67, 2010.
- [9] Hidetomo Nabeshima, Koji Iwanuma and Katsumi Inoue. GlueMiniSat 2.2.8. In *Proceedings of SAT Competition 2014*, pp. 35–36, 2014.
- [10] 鍋島英知, 宋剛秀. 高速 SAT ソルバーの原理. 人工知能学会誌, Vol. 25, No. 1, pp. 68–76, 2010.
- [11] Toru Ogawa, Yangyang Liu, Ryuzo Hasegawa, Miyuki Koshimura and Hiroshi Fujita. Modulo based CNF encoding of cardinality constraints and its application to MaxSAT solvers. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence, Herndon, VA, USA, November 4–6, 2013*, pp. 9–17, 2013.
- [12] Tobias Philipp and Peter Steinke. PBLib – A library for encoding pseudo-Boolean constraints into CNF. In *Proceedings of the International Conference on Theory and Applications of Satisfiability Testing (SAT 2015)*, pp. 9–16, 2015.
- [13] Olivier Roussel and Vasco M. Manquinho. Pseudo-Boolean and cardinality constraints. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Vol. 185 of *Frontiers in Artificial Intelligence and Applications*, pp. 695–733. IOS Press, 2009.
- [14] Masahiko Sakai and Hidetomo Nabeshima. Construction of an ROBDD for a PB-constraint in band form and related techniques for PB-solvers. *IEICE Transactions*, Vol. 98-D, No. 6, pp. 1121–1127, 2015.
- [15] Carsten Sinz. Towards an optimal CNF encoding of Boolean cardinality constraints. In *Proceedings of the 11th International Joint Conference on Principles and Practice of Constraint Programming (CP 2005)*, LNCS 3709, pp. 827–831, 2005.
- [16] Naoyuki Tamura, Mutsunori Banbara and Takehide Soh. PBSugar: Compiling pseudo-boolean constraints to SAT with order encoding. In *Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2013)*, IEEE, pp. 1020–1027, November 2013.
- [17] Naoyuki Tamura, Akiko Taga, Satoshi Kitagawa and Mutsunori Banbara. Compiling finite linear CSP into SAT. *Constraints*, Vol. 14, No. 2, pp. 254–272, 2009.



南 雄之

2016 年神戸大学工学部情報知能工学科卒業。同年神戸大学大学院システム情報学研究科情報科学専攻博士前期課程入学。2018 年同大学大学院修了。システム情報学修士。



宋 剛秀

2004 年神戸大学工学部電気電子工学科卒業。2006 年同大学大学院自然科学研究科電気電子工学専攻修了。サントリー (株) を経て、2011 年総合研究大学院大学複合科学研究科情報学専攻博士後期課程修了。博士 (情報学)。2012 年より神戸大学情報基盤センター助教。制約プログラミング, SAT 技術およびそれらの応用に興味をもつ。日本ソフトウェア科学会, 情報処理学会, 人工知能学会会員。



番原 睦則

1994 年神戸大学理学部数学科卒業。1996 年同大学大学院自然科学研究科博士課程前期数学専攻修了。1996 年国立奈良工業高等専門学校助手。1998 年同校講師。2003 年より神戸大学情報基盤センター講師。2007 年同校准教授。博士 (工学)。論理プログラミング, 制約プログラミング, SAT 技術等に興味をもつ。日本ソフトウェア科学会, 情報処理学会, 人工知能学会会員。



田村 直之

1980 年神戸大学理学部物理学科卒業。1985 年同大学大学院自然科学研究科博士課程システム科学専攻修了。学術博士。1985 年日本 IBM 東京基礎研究所入社。1988 年神戸大学工学部勤務。2003 年より神戸大学情報基盤センター教授。論理プログラミング, 制約プログラミング, SAT 技術等に興味をもつ。日本ソフトウェア科学会, 情報処理学会, 人工知能学会会員。