チャネリング制約を用いた alldifferent 制約の SAT 符号化

小菅 脩司 † 宋 剛秀 ‡ 田村 直之 § 番原 睦則 ¶
名古屋大学 † 神戸大学 ‡ 神戸大学 § 名古屋大学 ¶

alldifferent 制約は，制約プログラミングにおける代表的なグローバル制約の一つである．alldifferent$(x_1, x_2, \ldots, x_n)$ は，整数変数 x_1, x_2, $\ldots$, x_n の値が互いに異なることを表す制約である．すなわち

$$\text{alldifferent}(x_1, x_2, \ldots, x_n) \Longleftrightarrow \bigwedge_{1 \le i < j \le n} x_i \ne x_j$$

である．各 x_i が 1 以上 d 以下の値を取る場合，alldifferent 制約の解は d 個から n 個を取り出す順列に対応する．alldifferent 制約は，人工知能分野の諸問題に頻繁に現れる．そのため，alldifferent 制約の効率的な実装は重要な研究課題であり，古くから研究がなされている．

一方，2000 年以降，命題論理の充足可能性判定問題 (Boolean SATisfiability; SAT) を解く SAT ソルバーの性能が飛躍的に向上し，alldifferent 制約を含む制約充足問題 (制約プログラミングの言語) を SAT に符号化して解く手法の研究が進められた．**順序符号化法** (Order Encoding; OE) は，各整数変数 x と各整数定数 $a \in Dom(x)$ に対して，$x \le a$ を意味する命題変数 $p(x \le a)$ を用いる [2]．この順序符号化法に基づいた SAT 型制約ソルバー Sugar [*1] は，2008 年国際制約ソルバー競技会のグローバル制約部門で第 1 位になるなど，alldifferent 制約に対して優れた性能を示している．

本発表では，alldifferent 制約の SAT 符号化として，順序符号化法と直接符号化法をチャネリング制約を用いて融合させた手法を提案する．また，クイーングラフ彩色問題を用いた評価結果について述べる．この問題は alldifferent 制約だけで記述でき，D. E. Knuth の著書 [1] でも種々な SAT 符号化を比較するためのベンチマークとして用いられている．

提案手法の基本的アイデアは，各整数変数 x と各整数定数 $a \in Dom(x)$ に対して，順序符号化法と直接符号化法の両方の命題変数を導入し，以下のようなチャネリング制約を追加する点である [*2]．

$$p(x = a) \Longleftrightarrow \neg p(x \le a-1) \wedge p(x \le a)$$

$p(x = a)$ は直接符号化法 (Direct Encoding; DE) の命題変数であり $x = a$ を意味する．これにより，提案手法は，alldifferent$(x_1, x_2, \ldots, x_n)$ を $\bigwedge x_i \ne x_j$ の形に分解し，各々の $x_i \ne x_j$ を順序符号化法 (OE) と直接符号化法 (DE) のいずれかを用いて SAT に符号化できる．また，順序符号化法で有効性が確認されている鳩の巣原理 (Pigeon Hole Principle; PHP) に基づくヒント制約，$n = d$ の場合に直接符号化法で有効性が確認されている at-least-one 制約 (ALT1) を組み合わせて利用できる．さらに，alldifferent 制約を $\bigwedge x_i \ne x_j$ の形に分解する代わりに，擬似ブール (Pseudo-Boolean; PB) 制約に符号化 [3] し，その後，SAT に符号化することもできる．

提案手法の有効性を評価するために，クイーングラフ彩色問題 ($8 \le N \le 12$) を用いた実行

SAT Encoding of Alldifferent Constraints with Channeling Constraints
† Shuji Kosuge, Nagoya University
‡ Takehide Soh, Kobe University
§ Naoyuki Tamura, Kobe University
¶ Mutsunori Banbara, Nagoya University
*1 https://cspsat.gitlab.io/sugar/

*2 $a - 1 \notin Dom(x)$ の場合は，$\neg p(x \le a-1)$ は省略

表 1　比較に用いた alldifferent 制約の符号化一覧

符号化	整数変数の符号化法	alldifferent 制約の表現				PHP	ALT1
		≠ 分解	基本 PB	PB3 [3]	PB4 [3]		
0	OE	OE					
1	OE	OE				OE	
2	OE⇔DE	DE					
3	OE⇔DE	DE					DE
4	OE⇔DE	DE				OE	
5	OE⇔DE	DE				OE	DE
6	OE⇔DE		OE				
7	OE⇔DE		OE			OE	
8	OE⇔DE			OE			
9	OE⇔DE			OE		OE	
10	OE⇔DE				OE		
11	OE⇔DE				OE	OE	

表 2　クイーングラフ彩色問題の実験結果: 一問あたりの制限時間は 2 時間．ただし，$N = 12$ については，$N = 11$ で性能の良かった上位 3 モデルについて制限時間を 72 時間に延ばして得られた結果である．

符号化	$N = 8$ UNSAT	$N = 9$ UNSAT	$N = 10$ UNSAT	$N = 11$ SAT	$N = 12$ SAT
0	20.120	1296.917	3017.705	TO	-
1	0.118	479.376	TO	TO	-
2	11.532	1333.950	TO	TO	-
3	0.005	1.600	25.872	758.905	-
4	0.047	364.726	TO	TO	-
5	0.006	1.571	24.978	311.325	TO
6	0.005	1.605	27.360	761.812	-
7	0.005	1.525	25.105	610.408	-
8	0.006	0.884	21.967	446.034	-
9	0.006	1.195	22.950	81.861	142694.686
10	0.007	1.069	21.644	954.395	-
11	0.006	1.147	26.128	332.564	TO

実験を行なった．比較に用いた符号化を表 1 に，結果を表 2 に示す．表 1 の符号化 1 は，高速制約ソルバー Sugar のデフォルト設定と同じである．符号化 2–11 はチャネリング制約を用いた提案手法である．例えば，符号化 9 は，alldifferent 制約を PB 符号化 [3] したのち，鳩の巣原理に基づくヒント制約と併せて，順序符号化法を用いて SAT に符号化する．

表 2 より，提案手法は既存の順序符号化法を単体で用いるよりも優れた結果を示している．特に，符号化 9 は，クイーングラフ彩色問題の SAT 解法について様々な手法を比較した論文 [4] でも成功していない $N = 12$ の発見に成功しており，提案手法の有効性を示している．

参考文献

[1] Donald E. Knuth. *Satisfiability*, Vol. 4, Fascicle 6 of *The Art of Computer Programming*. Addison-Wesley Professional, 2015.

[2] Naoyuki Tamura, Akiko Taga, Satoshi Kitagawa, and Mutsunori Banbara. Compiling finite linear CSP into SAT. *Constraints*, Vol. 14, No. 2, pp. 254–272, 2009.

[3] 大野周亮, 番原睦則, 宋剛秀, 田村直之. alldifferent 制約のブール基数制約への符号化手法の提案とクイーングラフ彩色問題への応用. 人工知能基本問題研究会, Vol. 109, pp. 6–11, 2019.

[4] 田村直之, 宋剛秀, 番原睦則. SAT ソルバーの使い方 —問題を SAT に符号化する方法—. 第 58 回プログラミング・シンポジウム予稿集, Vol. 109, pp. 165–172, 2017.