3J1-OS7-1in

整数有限領域上の制約充足問題のコンパクトかつ効率的なSAT符号化

A Compact and Efficient SAT encoding of finite CSP on integers

丹生 智也[*1] 田村 直之[*2] 番原 睦則[*2]
Tomoya Tanjo Naoyuki Tamura Mutsunori Banbara

[*1]神戸大学 大学院工学研究科
Graduate School of Engineering, Kobe University

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

This paper describes a new SAT encoding method, named *compact order encoding*, applicable to finite domain CSP. It is a generalization of log encoding (compact encoding) and order encoding which is adopted by an award-winning SAT-based CSP solver. The basic idea of the compact order encoding is the use of a numeral system of some base. Each integer variable is divided into some digits and each digit is encoded by using the order encoding. It generates much smaller SAT instances than the order encoding and more efficient SAT instances than the log encoding in general because it requires fewer carry propagations when using larger base than two. We confirmed these observations through the experimental results on Open-Shop Scheduling problems. Especially for very large instances, our solver with MiniSat engine outperformed other encodings and the state-of-the-art CSP solvers choco and Mistral.

1. はじめに

近年の SAT 技術の発展により，プランニング，スケジューリング，ハードウェア・ソフトウェア検証，制約充足等の問題を SAT に符号化 (変換) し，MiniSat [Eén 03] 等の高速な SAT ソルバーを用いて求解を行う **SAT 符号化**の研究が活発に行われている [井上 10]．また，**制約充足問題 (CSP)** は，与えられた有限領域上の制約を充足するような変数への値割り当てを探す組み合わせ問題である [Rossi 06]．SAT 型制約ソルバーは，CSP を SAT に符号化し [Prestwich 09, 田村 10]， SAT ソルバーにより解を探索するプログラムである．

CSP から SAT への符号化法は，直接符号化法 [de Kleer 89, Walsh 00]，支持符号化法 [Kasif 90, Gent 02]，対数符号化法 [Iwama 94, Gelder 08]，対数支持符号化法 [Gavanelli 07]，順序符号化法 [Tamura 06, Tamura 09] 等，数多く提案されている．特に順序符号化法 [Tamura 09] は，オープンショップ・スケジューリング問題 [Tamura 09]，2 次元ストリップパッキング問題 [Soh 10] で未知だった最適値の決定に成功する等，広い範囲の問題で良い性能を示している [Inoue 06, Nabeshima 06, Banbara 10]．

しかし順序符号化法には、元の CSP のドメインサイズが大きい時に巨大な SAT 問題が生成され，問題が解けない場合がある．例えば, 整数変数のドメインサイズを d とすると，n-項制約 $\sum_{i=1}^{n} a_i x_i \le b$ は，$O(d^{n-1})$ 個の節に符号化される．一方で，対数符号化法 (コンパクト符号化法) は整数の 2 進数表記に着目した符号化法であり，生成される SAT 問題は非常に小さくなる ($\log d$ に比例する)．しかし，桁上がりの伝播に多くの推論ステップが必要なため，一般に性能は良くない．

1.1 コンパクト順序符号化法の提案

そこで本論文では，コンパクト (生成される SAT 問題のサイズが小さい) かつ効率的であることを目指した新しい符号化法である **コンパクト順序符号化法**を提案する．コンパクト順序符号化法では，ドメインサイズを小さくするために整数の B 進表現を用いる ($B \ge 2$) [Tanjo 10]．すなわち，各整数変数 x を $\sum_{i=0}^{m-1} B^i x^{(i)}$ で表す．ここで，$m = \lceil \log_B d \rceil$ であり，すべての $x^{(i)}$ に対して $0 \le x^{(i)} < B$ である．また，コンパクト順序符号化法は，$B \ge d$ の場合には順序符号化法と等価となり，$B = 2$ の場合には対数符号化法と等価となる．よって，コンパクト順序符号化法は両方の符号化法の一般化であるとみなせる．

しかし単純に整数変数 x を $\sum_{i=0}^{m-1} B^i x^{(i)}$ で置き換えるだけでは，順序符号化法よりも符号化後の節数が増えてしまう．そのため，変数のドメインサイズだけでなく，各算術制約に含まれる変数の数も制限することが重要である．

コンパクト順序符号化法を用いた整数有限領域上の CSP の符号化は次のように行われる．まず，CSP を三項 CSP に還元する．次に，還元された三項 CSP をコンパクト三項 CSP に還元する．そして，還元されたコンパクト三項 CSP を順序符号化法を用いて SAT に符号化する．各 CSP については第 2. 節と第 3. 節で述べる．

本論文では，第 2. 節で三項 CSP を定義し，整数有限領域上の任意の CSP が三項 CSP に還元できる事を示す．第 3. 節でコンパクト三項 CSP を定義し，第 4. 節でいくつかの制約の還元例を示す．第 5. 節で他の符号化法と節数に関して比較し，第 6. 節で性能評価を行った後，第 7. 節で結論を述べる．性能評価では，実際的な応用としてオープンショップ・スケジューリング問題を用い，コンパクト順序符号化法と順序符号化法，対数符号化法の性能を比較する．また，最先端の CSP ソルバーである choco と Mistral との比較も行った．

2. 三項 CSP

最初にいくつかの記法を導入する．$\mathbb{Z}$ と $\mathbb{N}$ はそれぞれ，整数の集合と非負整数の集合を表す．

整数有限領域上の**三項 CSP (3ary-CSP)** は，各算術制約に含まれる整数変数が高々3 つである CSP であり，以下のように定義される．

定義 1 (三項 CSP)．(整数有限領域上の) 三項 CSP は，以下を満たす組 (X, u, P, C) である．

連絡先: 丹生 智也，神戸大学大学院 工学研究科，〒 657-8501 神戸市灘区六甲台町 1-1，Tel: 078-8035-364，Mail: tanjo@stu.kobe-u.ac.jp

- X は整数変数の有限集合である．
- u は X から $\mathbb{N}$ への写像であり，整数変数の上限を表す(下限は 0 である)．
- P は命題変数の有限集合である．
- C は満たすべき制約を表す論理式であり，以下の文法で表される．

$$\begin{aligned} C \quad ::= \quad & p \ | \ \neg p \ | \ \sum_{i=1}^{n} a_i x_i \triangleright b \ | \ z = xy \\ & | \ C \wedge C \ | \ C \vee C \end{aligned}$$

ただし $p \in P$，$n \in \{1, 2, 3\}$，$x_i, x, y, z \in X$，$a_i \in \mathbb{Z}$，$b \in \mathbb{N}$，$\triangleright \in \{\leq, \geq, =\}$ である．

有限領域上の任意の CSP は，以下の手順で容易に三項 CSP へ還元できる．

- 外延的制約を，上で定義した算術制約で表す．
- 整数変数 x の下限が 0 でない場合には，x を新しい整数変数 x' に置き換える．ただし，$x' = x - l$ である．
- 制約 $\sum_i a_i \prod_j x_{ij} \triangleright b$ を，制約 $\sum_i a_i z_i \triangleright b$ と $z_i = \prod_j x_{ij}$ に還元する．ただし，各 z_i は新しい整数変数である．
- 4 変数以上を含む算術制約を，部分和・部分積をそれぞれ新しい変数で置き換えることで三項制約に還元する．

以下では，整数有限領域上の三項 CSP を単に CSP と呼ぶ．

3. コンパクト三項 CSP

この節では，**コンパクト三項 CSP (Compact 3ary-CSP)** を定義する．コンパクト三項 CSP は，各整数変数の上限をある整数 $B - 1$ に固定した CSP であり，B は**基底**と呼ばれる ($B \geq 2$)．また，コンパクト三項 CSP に含まれる算術制約はすべて線形制約である．

定義 2 (コンパクト三項 CSP)．コンパクト三項 CSP $(B; X, P, C)$ は，以下を満たす CSP (X, u, P, C) である．

- $B \geq 2$ は基底を表す整数である．
- $u(x) = B - 1$ $(\forall x \in X)$．
- 論理式 C は以下の文法で表される．

$$\begin{aligned} C \quad ::= \quad & p \ | \ \neg p \ | \ \sum_{i=1}^{n} \pm x_i \triangleright b \ | \ By + z = ax \\ & | \ C \wedge C \ | \ C \vee C \end{aligned}$$

ただし，$p \in P$，$n \in \{1, 2, 3\}$，$x_i, x, y, z \in X$，$0 \leq a < B$，$0 \leq b \leq 3(B - 1)$，$\triangleright \in \{\leq, \geq, =\}$ である．

4. CSP からコンパクト三項 CSP への還元

ページ数の都合により，以下では制約 $x \leq 26$，$z = x + 34$，$z = 9x$ の還元のみを示す．CSP 全体の還元については [Tanjo ar] を参照すること．

以下では，CSP の整数変数 x の i 桁目を表すコンパクト三項 CSP の整数変数を $x^{(i)}$ と表記する．

$x^{(1)}$	$x^{(0)}$	充足可能
0–1	0–9	Yes
2	0–6	Yes
2	7–9	No
3–9	0–9	No

図 1: $(x^{(1)} \leq 2) \wedge (x^{(1)} \leq 1 \vee x^{(0)} \leq 6)$ の充足可能性

		$x^{(1)}$	$x^{(0)}$
	$\times$		9
		v_{01}	v_{00}
	v_{11}	v_{10}	
	$z^{(2)}$	$z^{(1)}$	$z^{(0)}$

図 2: $100z^{(2)} + 10z^{(1)} + z^{(0)} = 9(10x^{(1)} + x^{(0)})$ の筆算

4.1 $x \leq 26$ の還元例

CSP の論理式 $x \leq 26$ ($u(x) = 99$) は，基底 $B = 10$ のコンパクト三項 CSP の変数を用いて $10x^{(1)} + x^{(0)} \leq 26$ と表される．各桁での比較を考えることで，これは以下の論理式に還元できる．

$$(x^{(1)} \leq 2) \wedge (x^{(1)} \leq 1 \vee x^{(0)} \leq 6)$$

図 1 に還元された論理式の充足可能性を示す．この図から，還元された論理式は $10x^{(1)} + x^{(0)} \leq 26$ と同値であることがわかる．また，この論理式に含まれる算術制約は，すべて単項制約である．

4.2 $z = x + 34$ の還元例

CSP の論理式 $z = x + 34$ ($u(z) = u(x) = 99$) は，基底 $B = 10$ のコンパクト三項 CSP の変数を用いて $10z^{(1)} + z^{(0)} = 10x^{(1)} + x^{(0)} + 34$ と表される．各桁での加算を考えることで，これは以下の充足同値な論理式に還元できる．

$$\begin{aligned} & (c \vee z^{(1)} = x^{(1)} + 3) \wedge (\neg c \vee z^{(1)} = x^{(1)} + 4) \\ \wedge \quad & (c \vee z^{(0)} = x^{(0)} + 4) \wedge (\neg c \vee z^{(0)} = x^{(0)} - 6) \end{aligned}$$

ここで c は $x^{(0)} + 4$ からの桁上がりを表す新しい命題変数である．例えば，$c = false$ の場合には $(z^{(1)} = x^{(1)} + 3) \wedge (z^{(0)} = x^{(0)} + 4)$ となり，$10z^{(1)} + z^{(0)} = 10x^{(1)} + x^{(0)} + 34$ を導出できる．

4.3 $z = 9x$ の還元例

CSP の論理式 $z = 9x$ ($u(z) = 999$, $u(x) = 99$) は，基底 $B = 10$ のコンパクト三項 CSP の変数を用いて $100z^{(2)} + 10z^{(1)} + z^{(0)} = 9(10x^{(1)} + x^{(0)})$ と表される．乗算の筆算を考えることで，これは以下の充足同値な論理式に還元できる (図 2 参照)．ここで，v_{ij} $(i, j \in \{0, 1\})$ は部分積 $ax^{(i)}$ の j 桁目を表す新しい整数変数であり，c は $v_{01} + v_{10}$ からの桁上がりを表す命題変数である．

$$\begin{aligned} & (10v_{11} + v_{10} = 9x^{(1)}) \wedge (10v_{01} + v_{00} = 9x^{(0)}) \\ \wedge \quad & (c \vee z^{(2)} = v_{11}) \wedge (\neg c \vee z^{(2)} = v_{11} + 1) \\ \wedge \quad & (c \vee z^{(1)} = v_{01} + v_{10}) \wedge (\neg c \vee z^{(1)} = v_{01} + v_{10} - 10) \\ \wedge \quad & (z^{(0)} = v_{00}) \end{aligned}$$

5. 各符号化法の比較

コンパクト順序符号化法では，還元されたコンパクト三項 CSP は順序符号化法を用いて SAT に符号化される．変数のドメインサイズを d とすると，順序符号化法では線形制約 $\sum_{i=1}^{n} a_i x_i \triangleright b$ ($\triangleright \in \{\leq, \geq, =\}$) は $O(d^{n-1})$ 個の節に符号化される．

制約	順序 $(B \geq d)$	コンパクト順序	対数 $(B = 2)$
$x \leq a$	$O(1)$	$O(m)$	$O(\log_2 d)$
$x \leq y$	$O(d)$	$O(mB)$	$O(\log_2 d)$
$z = x + a$	$O(d)$	$O(mB)$	$O(\log_2 d)$
$z = x + y$	$O(d^2)$	$O(mB^2)$	$O(\log_2 d)$
$z = ay$ $(0 \leq a < B)$	$O(d)$	$O(mB^2)$	$O(\log_2 d)$
$z = xy$	$O(d^2)$	$O(mB^3 + m^2B^2)$	$O(\log_2^2 d)$

図 3: 各符号化法を用いた SAT 符号化後の節数の比較

主な制約を符号化するのに必要な節数を図 3 に示す．d は変数のドメインサイズであり，$m = \lceil \log_B d \rceil$ は変数の桁数を表す．例えば，コンパクト順序符号化法では制約 $z = x + y$ は m 個の三項制約に還元され，また各三項制約は $O(B^2)$ 個の節に符号化されるため，$O(mB^2)$ 個の節に符号化される．これは，順序符号化法の $O(d^2)$ 個よりもはるかに小さい．

また制約 $z = ay$ は，各 $ay^{(i)}$ を計算する m 個の三項制約に還元されるため，$O(mB^2)$ 個の節に符号化される．特殊な場合として，$B \geq d$ の場合には $z = ay$ は $O(d)$ 個の節に符号化される．

非線形制約 $z = xy$ はコンパクト順序符号化法を用いて $O(mB^3 + m^2B^2)$ 個の節に符号化される．特殊な場合として，$B \geq d$ の場合には，$O(d^2)$ 個の節に符号化される．

6. 性能評価

コンパクト順序符号化法の実行効率及び大規模な問題での性能を評価するため，オープンショップ･スケジューリング (OSS) 問題を用いて性能評価を行った．OSS 問題には，Brucker らのベンチマークの中で最も大規模である “j7” と “j8” を用いた [Brucker 97]．これは，2006 年まで最適値が未解決であった 3 問を含む難易度の高いベンチマークである．また，よりドメインサイズの大きい問題を得るために，各問題の処理時間を c 倍して生成した問題も用いた ($c \in \{1, 10, 20, 100, 200, 1000\}$)．例えば，$c = 1000$ の場合には，整数変数の最大ドメインサイズは約 10^6 になる．

比較にはコンパクト順序符号化法 ($B = \sqrt{d}$)，順序符号化法，対数符号化法を用いた．各問題の総処理時間には最適値-1 を使用し，各符号化法を用いて SAT に符号化した．符号化後の SAT 問題は全て充足不能である．バックエンドの SAT ソルバーとして MiniSat [Eén 03] を用いた．また，最先端の CSP ソルバーである choco 2.11 [The choco team 08] と Mistral 1.550 [Hebrard 08] とも比較を行った．

図 4 に各係数ごとの求解問題数を示す．“ドメインサイズ d” は，整数変数のドメインサイズの平均を表す．各係数と全問題中で最も求解問題数が多いものをそれぞれ太字で示している．ベンチマークは Xeon 3.0 GHz，メモリ 14GB の PC 上で実行し，制限時間は 3600 秒とした．

結果として，全ての係数でコンパクト順序符号化法が最も多くの問題を解いた．また順序符号化法や Mistral が，c が大きくなると求解問題数が減っていくのに対し，コンパクト順序符号化法は $c = 1000$ の場合であっても，多くの問題を求解している．

また，図 5 に各ソルバーでの求解問題数と CPU 時間を示す．横軸が求解問題数，縦軸が符号化の時間も含めた CPU 時間を表している．コンパクト順序符号化法は，ほとんどの制限時間に関して最も多くの問題を解くことがわかる．

係数 c	ドメインサイズ d	問題数	順序	コンパクト順序	対数	choco	Mistral
1	$d \approx 10^3$	17	**14**	**14**	13	6	**14**
10	$d \approx 10^4$	17	12	**13**	**13**	6	12
20		17	12	**13**	**13**	6	12
100	$d \approx 10^5$	17	8	**13**	12	7	7
200		17	5	**13**	12	6	6
1000	$d \approx 10^6$	17	0	**14**	12	7	4
	全問題数	102	51	**80**	75	38	55

図 4: 各符号化法と choco，Mistral の求解問題数の比較

7. 結論

本論文では，順序符号化法と対数符号化法を一般化したコンパクトかつ効率的な新しい符号化法であるコンパクト順序符号化法を提案した．コンパクト順序符号化法は，各整数変数を B 進法で表現し，各桁を順序符号化法を用いて符号化する ($B \geq 2$)．

オープンショップ・スケジューリング問題を用いた性能評価の結果，コンパクト順序符号化法は順序符号化法，対数符号化法および最先端の CSP ソルバーである choco と Mistral よりも多くの問題を求解した．特に，他のソルバーが解けないドメインサイズが非常に巨大な問題に対しても，コンパクト順序符号化法が有効であることを示した．

今後の課題として，より大規模な問題を用いた性能評価と問題に対して適切な基底を選ぶ方法が挙げられる．

参考文献

[Banbara 10] Banbara, M., Matsunaka, H., Tamura, N., and Inoue, K.: Generating Combinatorial Test Cases by Efficient SAT Encodings Suitable for CDCL SAT Solvers, in *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-17), LNCS 6397*, pp. 112–126 (2010)

[Brucker 97] Brucker, P., Hurink, J., Jurisch, B., and Wöstmann, B.: A Branch & Bound Algorithm for the Open-shop Problem, *Discrete Applied Mathematics*, Vol. 76, No. 1–3, pp. 43–59 (1997)

[de Kleer 89] de Kleer, J.: A Comparison of ATMS and CSP Techniques, in *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI 1989)*, pp. 290–296 (1989)

[Eén 03] Eén, N. and Sörensson, N.: 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)

[Gavanelli 07] Gavanelli, M.: The Log-Support Encoding of CSP into SAT, in *Proceedings of the 13th International Joint Conference on Principles and Practice of Constraint Programming (CP 2007), LNCS 4741*, pp. 815–822 (2007)

[Gelder 08] Gelder, A. V.: Another Look at Graph Coloring via Propositional Satisfiability, *Discrete Applied Mathematics*, Vol. 156, No. 2, pp. 230–243 (2008)

[Gent 02] Gent, I. P.: Arc Consistency in SAT, in *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI 2002)*, pp. 121–125 (2002)

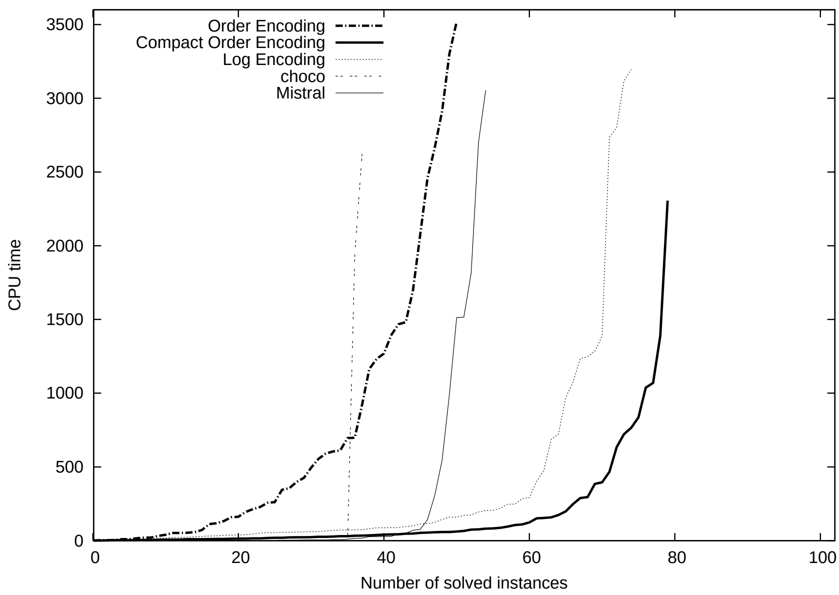

図 5: 各符号化法と choco，Mistral の求解問題数と求解 CPU 時間の比較

[Hebrard 08] Hebrard, E.: Mistral, a Constraint Satisfaction Library, in *Proceedings of the 3rd International CSP Solver Competition*, pp. 31–39 (2008)

[Inoue 06] Inoue, K., Soh, T., Ueda, S., Sasaura, Y., Banbara, M., and Tamura, N.: A Competitive and Cooperative Approach to Propositional Satisfiability, *Discrete Applied Mathematics*, Vol. 154, No. 16, pp. 2291–2306 (2006)

[井上 10] 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)

[Iwama 94] Iwama, K. and Miyazaki, S.: SAT-Variable Complexity of Hard Combinatorial Problems, in *Proceedings of the IFIP 13th World Computer Congress*, pp. 253–258 (1994)

[Kasif 90] Kasif, S.: On the Parallel Complexity of Discrete Relaxation in Constraint Satisfaction Networks, *Artificial Intelligence*, Vol. 45, No. 3, pp. 275–286 (1990)

[Nabeshima 06] Nabeshima, H., Soh, T., Inoue, K., and Iwanuma, K.: Lemma Reusing for SAT based Planning and Scheduling, in *Proceedings of the International Conference on Automated Planning and Scheduling 2006 (ICAPS 2006)*, pp. 103–112 (2006)

[Prestwich 09] Prestwich, S. D.: CNF Encodings, in *Handbook of Satisfiability*, pp. 75–97, IOS Press (2009)

[Rossi 06] Rossi, F., Beek, van P., and Walsh, T.: *Handbook of Constraint Programming*, Elsevier Science Inc., New York, NY, USA (2006)

[Soh 10] Soh, T., Inoue, K., Tamura, N., Banbara, M., and Nabeshima, H.: A SAT-based Method for Solving the Two-dimensional Strip Packing Problem, *Fundamenta Informaticae*, Vol. 102, No. 3-4, pp. 467–487 (2010)

[Tamura 06] Tamura, N., Taga, A., Kitagawa, S., and Banbara, M.: Compiling Finite Linear CSP into SAT, in *Proceedings of the 12th International Joint Conference on Principles and Practice of Constraint Programming (CP 2006), LNCS 4204*, pp. 590–603 (2006)

[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 10] 田村 直之, 丹生 智也, 番原 睦則 : 制約最適化問題と SAT 符号化, 人工知能学会誌, Vol. 25, No. 1, pp. 77–85 (2010)

[Tanjo 10] Tanjo, T., Tamura, N., and Banbara, M.: Towards a Compact and Efficient SAT-Encoding of Finite Linear CSP, in *Proceedings of the 9th International Workshop on Constraint Modelling and Reformulation (ModRef 2010)* (2010)

[Tanjo ar] Tanjo, T., Tamura, N., and Banbara, M.: Proposal of a compact and efficient SAT encoding using a numeral system of any base, in *First International Workshop on the Cross-Fertilization Between CSP and SAT (CSPSAT 2011)* (to appear)

[The choco team 08] The choco team, : choco: an Open Source Java Constraint Programming Library, in *Proceedings of the 3rd International CSP Solver Competition*, pp. 7–13 (2008)

[Walsh 00] Walsh, T.: SAT v CSP, in *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming (CP 2000)*, pp. 441–456 (2000)