

正方形詰込み問題の制約モデルと SAT 符号化を用いた解法

Constraint Models of the Square Packing Problem and their SAT Encodings

佐古田 淳史*¹ 宋 剛秀*² 番原 陸則*² 田村 直之*²
 Atsushi Sakoda Takehide Soh Mutsunori Banbara Naoyuki Tamura

*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

Recent advances of SAT solvers result in the success of its application such as Constraint Satisfaction Problems (CSPs). As is shown by the fact that a SAT-based CSP solver *Sugar* won CSP solver competitions, constraint solving using SAT encoding is successful. In this paper, we consider the square packing problem, which is a problem of packing a given set of squares into an enclosing square without overlapping. We present two constraint models to represent non-overlapping constraints: basic constraint model and additional constraint model. In particular, the additional constraint model is designed to concisely represent constraints that effectively reduce the search space. We then show SAT encodings of two constraint models by using order encoding. In experiments, our SAT-based method using the additional constraint model succeed in improving computation time compared with the one using the basic constraint model.

1. はじめに

近年、大規模な命題論理の充足可能性判定 (SAT) 問題を高速に解くことが可能な SAT ソルバーが実現され、プランニング、スケジューリング、制約充足問題 (Constraint Satisfaction Problem; CSP) など、様々な分野への応用が進んでいる [井上 10, Koshimura 10].

CSP を SAT 問題に符号化する方法は SAT 符号化法と呼ばれ、これまでに数多くの方法が提案されている [田村 10b]. 特に、順序符号化法 [Tamura 09] は 2008-2009 年国際制約ソルバー競技会の GLOBAL 部門において 2 年連続優勝した SAT 型制約ソルバー *Sugar* [田村 10a] によって有効性が示されており、様々な問題への応用が期待される。

正方形詰込み問題 [Croft 91, Korf 10] は、 $1 \times 1, \dots, N \times N$ の連続した N 個の正方形をより大きな正方形の枠内に重なりなく配置する問題である。 N が 40 以下でも未解決問題が存在するなど求解困難であるが、電子回路における素子配置などへの応用も存在し重要な問題である。

本稿では、正方形詰込み問題に対して基本制約モデルと追加制約モデルの 2 つの制約モデルを考える。基本制約モデルは、正方形詰込み問題を定義通り制約表現したものであり [Soh 10, Korf 10]、正方形の上下左右の位置関係を用いるモデルである。提案する追加制約モデルは、垂直水平の位置関係を用いた制約表現であり、探索の枝刈りを効果的に行うような制約を簡潔に表現できる点が特長である。本稿では正方形詰込み問題を上記 2 つの方法で制約モデル化した後、順序符号化と SAT ソルバーを用いて求解を行う。

性能評価のために正方形詰込み問題 52 問を用いて実験を行った結果、追加制約モデルは基本制約モデルと比較して多くの問題において求解速度を向上することができた。また、2008 年当時に未解決だった $N = 26, 27, 29, 30, 31, 35$ の 6 つの最適解を初めて報告した文献 [Simonis 08] に記載されている計算時間とも比較を行った。結果として、提案手法は $N = 29$ 以外の 6 問中 5 問においてより高速に求解できることが分かった。

連絡先: 佐古田淳史, 神戸大学 大学院システム情報学研究科, 兵庫県神戸市灘区六甲台町 1-1 神戸大学情報基盤センター田村研究室, 078-803-5364, a-sakoda@stu.kobe-u.ac.jp

2. 正方形詰込み問題と関連研究

正方形詰込み問題 [Korf 10] の定義を以下で与える。

定義 1 正方形詰込み問題

入力 • 正方形の集合 $R = \{1, 2, \dots, N\}$
 (但し $i \in R$ は一辺の長さが i の正方形とする.)

• 1 辺の長さが L の正方形の枠

制約 • すべての $i \in R$ は枠内に配置される
 • すべての $i, j \in R$ ($i \neq j$) は互いに重ならない (非重複制約)

目的 すべての制約を満たすようにすべての正方形 $i \in R$ の配置を決定する。

正方形の集合 R は N によって定まるので、正方形詰込み問題は N と L が与えられれば一意に決定できることが分かる。図 1 に、 $N = 40, L = 150$ の解の例を示す。

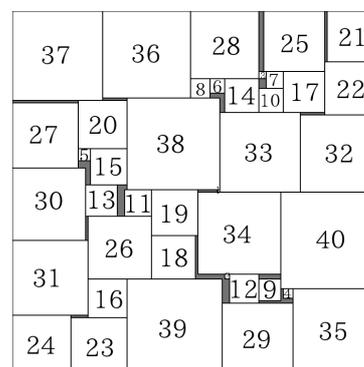


図 1: 正方形詰込み問題の解の例: $N=40, L=150$

正方形詰込み問題では、 L が入力として与えられるのに対し、与えられた N に対して詰込み可能な L の最小値を求める問題を **正方形詰込み最適化問題** と呼ぶ。正方形詰込み最適化問

題は OEIS の A005842 : Integer Sequences *1 に $N = 56$ までの、現在までに求まっている最適値が示されている。このうち $N = 38, N = 40$ については最適値が未知であり、解くことが非常に難しい問題であることが分かる。

3. 正方形詰込み問題の制約表現

本稿で用いる制約モデルで共通に使用する変数について説明を行う。枠の左下の頂点を原点とする座標系を考え、横に x 軸、縦に y 軸をとる。各入力正方形 $i \in R$ に対して、 i の左下の頂点座標を表す 2 つの整数変数 (x_i, y_i) を導入する。 (x_i, y_i) のドメインは、どちらも $\{a \in \mathbb{N} | 0 \leq a \leq L - i\}$ になる。このドメインはすべての $i \in R$ は枠内に配置されるという制約を表している。

3.1 基本制約モデル

基本制約モデルでは、非重複制約を整数変数 (x_i, y_i) を用いて以下のように表す ($i, j \in R, i < j$)。

$$x_i + i \leq x_j \vee x_j + j \leq x_i \vee y_i + i \leq y_j \vee y_j + j \leq y_i$$

この式は任意の二つの正方形が上下左右それぞれの方向に重ならないことを表しており、例えば、 $x_i + i \leq x_j$ は i が j の左側にあるという制約を表している。

3.2 追加制約モデル

追加制約モデルでは、求解速度向上のため明らかに制約を充足しない正方形の配置を禁止する制約を追加する。

例えば $N = 7, L = 15$ のとき、一辺の長さが 3, 6, 7 の正方形を水平方向に重ならないように同時に配置すると枠の長さ 15 を超えてしまうため、明らかにこの配置は不可能である。

一般的には、入力正方形の集合 R の部分集合 R' に対し、 $w(R') = \sum_{i \in R'} i$ とすると、 $w(R') > L$ を満たすような正方形の集合 R' は並べる順番に依らず垂直 (または水平) に同時に配置できない。

このような配置を禁止する制約を記述するために、 N, L が与えられたとき、要素数が $k (k \geq 1)$ 個以下の R' に対し $w(R') > L$ となるようなものの集合 S_k を考える。ここで任意の要素 q に対して $w(R') - q > L$ となる場合、 $R' \setminus \{q\}$ は既に S_{k-1} に含まれていることに注意されたい。 $R' \setminus \{q\}$ を用いる方がより強い枝刈りを行えるため、この R' は S_k には加えない。 S_k を定式化すると以下ようになる。

$$S_k = S_{k-1} \cup \{R' \subseteq R \mid |R'| = k, w(R') > L, \forall q \in R'. w(R') - q \leq L\}$$

以降、 $S_k \neq \emptyset$ となるような最小の k を k_{min} とする。以下に $N = 7, L = 15$ のときの S_k の例を示す。

$$\begin{aligned} S_3 &= \{\{3, 6, 7\}, \{4, 5, 7\}, \{4, 6, 7\}, \{5, 6, 7\}\} \\ S_4 &= S_3 \cup \{\{1, 2, 6, 7\}, \{1, 3, 5, 7\}, \{1, 4, 5, 6\}, \{2, 3, 4, 7\}, \\ &\quad \{2, 3, 5, 6\}, \{2, 3, 5, 7\}, \{2, 4, 5, 6\}, \{3, 4, 5, 6\}\} \\ S_5 &= S_4 \cup \{\{1, 2, 3, 4, 6\}\} \end{aligned}$$

S_1, S_2 は空集合となる。また、このときの k_{min} は 3 である。

この S_k を用いて、明らかに制約を充足しない正方形の配置を禁止する追加制約を記述する。この追加制約の命題論理式 ψ_k は以下ようになる。ここで $h_{i,j}, v_{i,j}$ とは、追加制約を簡

潔に表すために導入された命題変数であり、 $h_{i,j}$ は $i, j \in R$ が水平方向に重ならず配置されるとき真、また $v_{i,j}$ は $i, j \in R$ が垂直方向に重ならず配置されるとき真となる。

$$\psi_k = \bigwedge_{B \in S_k} \left(\bigvee_{a,b \in B, a < b} \neg h_{a,b} \right) \wedge \left(\bigvee_{a,b \in B, a < b} \neg v_{a,b} \right)$$

非重複制約については、命題変数 $h_{i,j}, v_{i,j}$ を用いて以下のように表される ($i, j \in R, i < j$)。

$$\begin{aligned} &h_{i,j} \vee v_{i,j} \\ &\neg h_{i,j} \vee x_i + i \leq x_j \vee x_j + j \leq x_i \\ &\neg v_{i,j} \vee y_i + i \leq y_j \vee y_j + j \leq y_i \end{aligned}$$

この式は任意の二つの正方形が重ならないことを表しており、例えば、 $h_{i,j} \vee x_i + i \leq x_j \vee x_j + j \leq x_i$ は i が j の左側または右側にあることを表している。 $R = \{1, 2, 3\}$ のときの例を以下に示す。

$$\begin{aligned} &h_{1,2} \vee v_{1,2} \\ &\neg h_{1,2} \vee x_1 + 1 \leq x_2 \vee x_2 + 2 \leq x_1 \\ &\neg v_{1,2} \vee y_1 + 1 \leq y_2 \vee y_2 + 2 \leq y_1 \\ &h_{1,3} \vee v_{1,3} \\ &\neg h_{1,3} \vee x_1 + 1 \leq x_3 \vee x_3 + 3 \leq x_1 \\ &\neg v_{1,3} \vee y_1 + 1 \leq y_3 \vee y_3 + 3 \leq y_1 \\ &h_{2,3} \vee v_{2,3} \\ &\neg h_{2,3} \vee x_2 + 2 \leq x_3 \vee x_3 + 3 \leq x_2 \\ &\neg v_{2,3} \vee y_2 + 2 \leq y_3 \vee y_3 + 3 \leq y_2 \end{aligned}$$

3.3 追加制約モデルのリテラル置換

追加制約 ψ_k の節長は最大で $k(k-1)/2$ になる。提案手法では、追加制約のリテラルについて共通部分に新しい命題変数を導入することで、節長を削減している。

具体的には、2 つ以上の $|R'|$ 個の要素を持つ集合に対し、 $|R'| - 1$ 個が同じ要素である場合、その要素間の命題変数をまとめて一つの新しい命題変数として定義する。例えば、 S_k 内に $\{1, 2, 3, 4, 5\}$ と $\{1, 2, 3, 4, 6\}$ があつた場合、新しい命題変数 $h_{1,2,3,4}$ を以下のように定義する (以下、垂直方向にも同様)。

$$(\neg h_{1,2} \vee \neg h_{1,3} \vee \neg h_{1,4} \vee \neg h_{2,3} \vee \neg h_{2,4} \vee \neg h_{3,4}) \vee h_{1,2,3,4}$$

これにより追加制約は以下のように記述できる。

$$\begin{aligned} &\neg h_{1,2,3,4} \vee \neg h_{1,5} \vee \neg h_{2,5} \vee \neg h_{3,5} \vee \neg h_{4,5} \\ &\neg h_{1,2,3,4} \vee \neg h_{1,6} \vee \neg h_{2,6} \vee \neg h_{3,6} \vee \neg h_{4,6} \end{aligned}$$

元の節長が $\{5 \times (5-1)\}/2 = 10$ であるのに対し、リテラル置換を行ったものは節長が 5 になっていることが分かる。

4. SAT 符号化

4.1 順序符号化法の概要

順序符号化法 [Tamura 09] では、各整数変数 z について、そのドメインが $\{a_1, a_2, \dots, a_n\}$ の時 (ただし $a_1 < a_2 < \dots < a_n$)、 $z \leq a_i$ を表す $n-1$ 個の命題変数 $p(z \leq a_1), p(z \leq a_2), \dots, p(z \leq a_{n-1})$ を用いる。なお、 $z \leq a_n$ は常に真であるため、命題変数 $p(z \leq a_n)$ は不要である。また、これらの命題変数間の関係を表す以下の節を用いる。

$$\neg p(z \leq a_i) \vee p(z \leq a_{i+1}) \quad (1 \leq i \leq n-2)$$

例えば、変数 z のドメインが $\{0, 1, 2\}$ の場合、2 つの命題変数 $p(z \leq 0), p(z \leq 1)$ を用いて $\neg p(z \leq 0) \vee p(z \leq 1)$ を追加する。

*1 <http://oeis.org/A005842>

この時、上記の節を充足可能にする真理値割り当ては3通りあり、それぞれ $z = 0, z = 1, z = 2$ に対応する。

$p(z \leq 0)$	$p(z \leq 1)$	解釈
1	1	$z = 0$
0	1	$z = 1$
0	0	$z = 2$

制約については、制約に違反する範囲を符号化する。例えば、整数変数 w, z のドメインが $\{0, 1, 2\}$ の場合、算術比較 $w \leq z$ は以下の2つの節 $p(w \leq 0) \vee \neg p(z \leq 0), p(w \leq 1) \vee \neg p(z \leq 1)$ に符号化される。

線形式を用いた線形制約については、より簡潔な符号化が可能である。今、 a_i を非零の整数定数、 c を整数定数、 z_i を互いに異なる整数変数とする。この時、制約 $\sum_{i=1}^n a_i z_i \leq c$ は以下のように符号化できる。

$$\bigwedge_{b_i} \bigvee_i (a_i z_i \leq b_i)^\#$$

ここで b_i は、 $\sum_{i=1}^n b_i = c - n + 1$ を満たすように動くとし、変換 $()^\#$ は以下のように定義する。

$$(a z \leq b)^\# \equiv \begin{cases} p(z \leq \lfloor b/a \rfloor) & (a > 0) \\ \neg p(z \leq \lceil b/a \rceil - 1) & (a < 0) \end{cases}$$

ただし、 z の取り得る最小値未満の a については $p(z \leq a)$ を偽に変換し、最大値以上については真に変換する。

例えば、整数変数 x_1, x_2 のドメインがそれぞれ $\{0, 1, 2, 3, 4\}, \{0, 1, 2, 3\}$ の時、制約 $x_1 + 1 \leq x_2$ は $\neg p(x_2 \leq 0), \neg p(x_2 \leq 1) \vee p(x_1 \leq 0), \neg p(x_2 \leq 2) \vee p(x_1 \leq 1), \neg p(x_1 \leq 2)$ の4つの節に符号化される。

ここで、 $\neg p(x_2 \leq 1) \vee p(x_1 \leq 0)$ は、「 $x_2 > 1$ または $x_1 \leq 0$ 」であること、すなわち「 $x_2 \leq 1$ かつ $x_1 \geq 1$ 」が制約に違反する領域であることを表している。

4.2 基本制約モデルの SAT 符号化

基本制約モデルを制約充足問題から SAT 問題に符号化することを考える。 $p(x_i, e)$ を x_i が e 以下のとき真になるブール変数、 $p(y_i, f)$ を y_i が f 以下のとき真になるブール変数とする。 e, f はそれぞれ $0 \leq e, f \leq L - i$ を満たす。これらの命題変数間の関係を以下のように表すことでドメインを表現できる。

$$\neg p(x_i, e) \vee p(x_i, e + 1) \quad \neg p(y_i, f) \vee p(y_i, f + 1)$$

命題変数 $lr_{i,j}, lr_{j,i}, ud_{i,j}, ud_{j,i} (i < j)$ により非重複制約は以下のように符号化される。

$$\begin{aligned} & lr_{i,j} \vee lr_{j,i} \vee ud_{i,j} \vee ud_{j,i} \\ & \neg lr_{i,j} \vee p(x_i, e) \vee \neg p(x_j, e + i) \\ & \neg lr_{j,i} \vee p(x_j, e) \vee \neg p(x_i, e + j) \\ & \neg ud_{i,j} \vee p(y_i, f) \vee \neg p(y_j, f + i) \\ & \neg ud_{j,i} \vee p(y_j, f) \vee \neg p(y_i, f + j) \end{aligned}$$

4.3 追加制約モデルの SAT 符号化

追加制約モデルを制約充足問題から SAT 問題に符号化することを考える。ドメインは基本制約モデルと同様である。

命題変数 $v_{i,j}, h_{i,j} (i < j)$ に加え、新たな命題変数 p, q, p', q' を導入することで、非重複制約は以下のように符号化される。

$$\begin{aligned} & v_{i,j} \vee h_{i,j} \\ & \neg p \vee p(x_i, e) \vee \neg p(x_j, e + i) \\ & \neg q \vee p(x_j, e) \vee \neg p(x_i, e + j) \\ & \neg h_{i,j} \vee p \vee q \\ & \neg p' \vee p(y_i, f) \vee \neg p(y_j, f + i) \\ & \neg q' \vee p(y_j, f) \vee \neg p(y_i, f + j) \\ & \neg v_{i,j} \vee p' \vee q' \end{aligned}$$

5. 性能評価

提案手法の有効性を評価するために、 $N = 15, \dots, 40$ のそれぞれの N に対する L の下限値とその $+1$ の値の問題 (全 52 問) を用いて実験を行った。ここで L の下限値とは、入力正方形の面積の和から求まる値であり、 $\lceil \sqrt{N(N+1)(2N+1)/6} \rceil$ で計算できる。比較に用いたモデルは以下の通りである。

- 基本制約モデル
- 追加制約モデル ($\psi_{k_{min}}, \psi_{k_{min}+1}, \psi_{k_{min}+2}$)

順序符号化を行うプログラムには、SAT 型制約ソルバー Sugar*2 [田村 10a] を用いた。Sugar で動作する SAT ソルバーとしては、Minisat [Eén 03], GlueMiniSat [鍋島 12], Glucose [Audemard 09] などが利用可能である。今回は予備実験で好成績を取めた Glucose を採用した。実験環境は、Linux マシン (Intel Xeon 3.16GHz) である。Sugar の求解時間には、CSP から SAT への符号化の時間と Glucose の求解時間が含まれている。制限時間は 10000 秒に設定した。

5.1 節数の比較

各モデルと符号化法の有効性については、解く問題の性質および SAT ソルバーの特性によって異なるため明確な基準はないが、SAT 符号化の研究者の間では節数が 1 つの基準としてよく用いられる。図 2 に各モデルの節数を示す。

追加制約モデルは、符号化後の節数の増加が懸念されるが $\psi_{k_{min}}, \psi_{k_{min}+1}$ を追加する場合は $N = 40$ のときでも基本制約モデルと比較して 10% 以下の増加に抑えられている。一方、 $\psi_{k_{min}+2}$ を追加する場合は $N = 40$ のときで 2 倍以上多くなっていることが分かる。以降では $\psi_{k_{min}}, \psi_{k_{min}+1}$ を追加する場合の追加制約モデルと基本制約モデルの性能評価を行う。

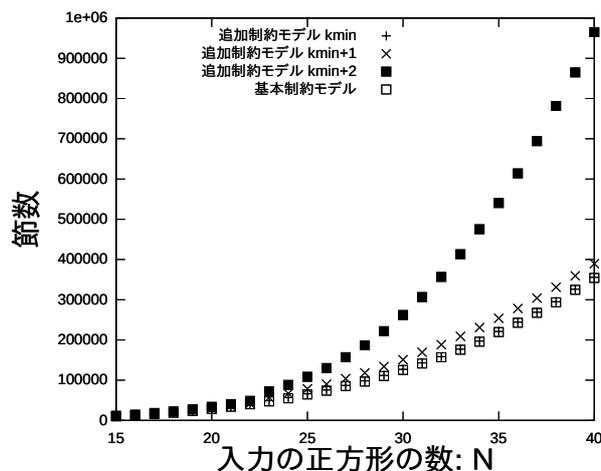


図 2: モデルの節数の比較

*2 <http://bach.istc.kobe-u.ac.jp/sugar/>

表 1: モデルの計算時間の比較

N	L	基本制約 モデル	追加制約モデル			
			リテラル置換なし		リテラル置換あり	
			k_{min}	$k_{min} + 1$	k_{min}	$k_{min} + 1$
15	36	<0.00	<0.00	0.01	<0.00	0.01
15	37	<0.00	<0.00	0.01	<0.00	<0.00
16	39	0.02	0.03	0.24	0.02	0.05
16	40	0.01	0.01	<0.00	0.01	0.01
17	43	0.01	0.03	0.01	0.02	0.03
17	44	<0.00	0.01	0.01	<0.00	0.01
18	46	821.53	322.05	145.23	321.11	90.45
18	47	0.07	0.02	0.03	0.03	0.04
19	50	111.60	10.31	59.22	10.37	42.81
19	51	0.01	0.01	0.01	0.01	0.01
20	54	3.28	0.20	0.23	0.18	0.31
20	55	0.01	0.03	<0.00	0.03	0.01
21	58	0.88	0.14	1.32	0.12	0.14
21	59	<0.00	0.02	0.02	0.01	0.02
22	62	20.20	64.62	1798.17	64.49	176.21
22	63	0.02	0.02	0.06	0.12	0.03
23	66	5467.57	8471.27	480.89	953.89	5418.32
23	67	0.09	0.05	0.05	0.03	0.02
24	71	0.88	0.62	0.34	0.46	4.08
25	75	21.35	27.41	305.89	28.76	454.06
25	76	0.04	0.04	0.15	0.03	0.06
26	80	0.26	0.32	0.49	0.28	0.07
27	84	17.02	2.20	0.80	0.48	22.45
27	85	0.06	0.04	0.06	0.07	0.03
28	89	0.96	1.13	2.82	0.21	0.16
29	93	6441.47	8800.69	1892.49	-	3358.83
29	94	0.15	0.44	0.61	1.07	0.11
30	98	1891.80	10.15	705.04	68.31	529.91
30	99	1.28	1.13	1.66	0.12	0.17
31	103	53.61	13.74	206.15	55.30	278.16
31	104	0.44	0.16	1.33	1.76	0.10
32	108	20.91	141.66	116.50	35.63	108.01
33	113	221.69	92.26	70.43	58.43	323.53
34	118	20.66	672.37	2618.36	618.43	241.28
35	123	-	313.76	-	6073.76	3076.28
35	124	2.99	6.51	3.87	11.81	0.46
36	129	6.32	8.57	19.35	1.44	38.23
37	134	52.14	36.87	188.96	211.16	63.14
38	139	2395.56	-	-	5734.95	749.45
39	145	165.53	24.77	35.92	42.72	2306.18
40	150	1611.86	5585.32	-	5587.29	749.49

5.2 計算時間の比較

表 1 に各問題に対するそれぞれのモデルの計算時間を示す。左から順に N , L , 計算時間を表している。ハイフンは制限時間内に解けなかったことを表す。また、各問題に対して最も良い CPU 時間をボード体で記している。表 2 に各モデルの解けた問題数、共通に解けた問題に対する幾何平均を示す。

表 1 及び表 2 より、リテラル置換によって、追加制約モデルの解けた問題数、計算時間は共に改善されていることが分かる。結果として、追加制約モデル ($\psi_{k_{min}}$) に対してリテラル置換を行ったモデルは計算時間の幾何平均が最も小さくなり、追加制約モデル ($\psi_{k_{min}+1}$) に対してリテラル置換を行ったモデルは解けた問題数が最も多くなった。

5.3 従来手法との比較

2008 年当時に未解決だった $N = 26, 27, 29, 30, 31, 35$ の 6 つの最適解を初めて報告した文献 [Simonis 08] に記載されている計算時間とも比較を行った (表 3)。使用 CPU に違いがあるため (Intel Xeon 3GHz), 提案手法の計算時間は表 1 の値を 1.053 倍している。結果として、提案手法は $N = 29$ 以外の 6 問中 5 問においてより高速に求解できることが分かった。

表 2: 解けた問題数と計算時間の幾何平均

	基本制約 モデル	追加制約モデル			
		リテラル置換なし		リテラル置換あり	
		k_{min}	$k_{min} + 1$	k_{min}	$k_{min} + 1$
解けた問題数	40	40	38	40	41
幾何平均	3.49	2.33	5.02	2.22	3.54

表 3: 従来手法 [Simonis 08] との比較

N L	従来手法 [Simonis 08]	基本制約 モデル	追加制約モデル			
			リテラル置換なし		リテラル置換あり	
			k_{min}	$k_{min} + 1$	k_{min}	$k_{min} + 1$
26 80	746	<0	<0	1	<0	<0
27 84	4	18	2	1	1	24
29 93	666	6782	9267	1993	-	3536
30 98	127	1992	11	742	72	558
31 103	18	57	14	217	58	293
35 123	4207	-	330	-	6396	3239

6. まとめ

本稿では、正方形詰込み問題の制約モデルとその符号化方法を説明した。特に、提案する追加制約モデルは探索の枝刈りを効果的に行うような制約を簡潔に表現することができる点が特長である。性能評価の結果、追加制約モデルが正方形詰込み問題において効果的であることが分かった。また、[Simonis 08] で報告されている 6 問の計算時間との比較の結果、追加制約モデルは 5 問においてより高速に求解できた。

参考文献

- [Audemard 09] Audemard, G. and Simon, L.: Predicting Learnt Clauses Quality in Modern SAT Solvers, in *Proceedings of IJCAI-2009*, pp. 399–404 (2009)
- [Croft 91] Croft, H. T., Falconer, K. J., and Guy, R. K.: *Unsolved Problems in Geometry*, Vol. 2, pp. 107–130 (1991)
- [Eén 03] Eén, N. and Sörensson, N.: An Extensible SAT-solver, in *Proceedings SAT 2003, LNCS 2919*, pp. 502–518 (2003)
- [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 (2010)
- [Korf 10] Korf, R. E., Moffitt, M. D., and Pollack, M. E.: Optimal rectangle packing, *Annals OR*, Vol. 179, No. 1, pp. 261–295 (2010)
- [Koshimura 10] Koshimura, M., Nabeshima, H., Fujita, H., and Hasegawa, R.: Solving Open Job-Shop Scheduling Problems by SAT Encoding, *IEICE Transactions*, Vol. 93-D, No. 8, pp. 2316–2318 (2010)
- [鍋島 12] 鍋島 英知, 岩沼 宏治, 井上 克巳: GlueMiniSat2.2.5: 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. 146–160 (2012)
- [Simonis 08] Simonis, H. and O’Sullivan, B.: Search Strategies for Rectangle Packing, in Stuckey, P. J. ed., *CP*, Vol. 5202 of *LNCS*, pp. 52–66, Springer (2008)
- [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 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 10a] 田村 直之, 丹生 智也, 番原 睦則: SAT 変換に基づく制約ソルバーとその性能評価, コンピュータソフトウェア, Vol. 27, No. 4, pp. 183–196 (2010)
- [田村 10b] 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 (2010)