A-023

正方形詰込み問題の数理最適化ソルバーによる解法と比較実験

Solving the Square Packing Problem with Mathematical Optimization Solvers

越村 三幸[1] 信岡 大輝[1] 岡本 和也[2]
Miyuki Koshimura Hiroki Nobuoka Kazuya Okamoto
木村 慧[1] 横尾 真[1]
Kei Kimura Makoto Yokoo

1 はじめに

正方形詰込み問題は、一辺の長さが $1, 2, \ldots, N$ 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 [4, 7] など求解困難であるが、電子回路における素子配置、倉庫内での商品や資材の効率的な配置、印刷物のレイアウト設計にける異なる図形のページ内での効率的な配置、など様々な応用がある問題である。

本稿では、正方形詰込み問題に対して基本制約と対称性除去制約の二つの制約を考える。前者は N 個の正方形をお互いに重ならないように配置する制約であり、後者は一辺の長さが N である最大正方形の配置位置を制限し、対称性に起因する無駄な探索を除去する制約である。

性能評価のため、$N = 15, 16, \ldots, 40$ の問題を用いて実験を行った。評価には、Z3[5] と Gurobi[1] の二つのソルバーを用いた。Z3 は SMT(Satisfiability Modulo Theories) ソルバー、Gurobi は代表的な数理最適化ソルバーである。正方形詰込み問題 52 問を用いた比較実験では、Z3 が Gurobi に比べて 6 割ほど多くの問題を解くことができた。

2 正方形詰込み問題

一辺の長さが i の正方形を単に正方形 i と呼ぶことにする。正方形詰込み問題は、次のような決定問題である。

正方形詰込み問題：連続した N 個の正方形 $1, 2, \ldots, N$ を正方形 L の枠内に互いに重ならないように配置することができるか否かを判定せよ。また、できるのであればその配置を示せ。

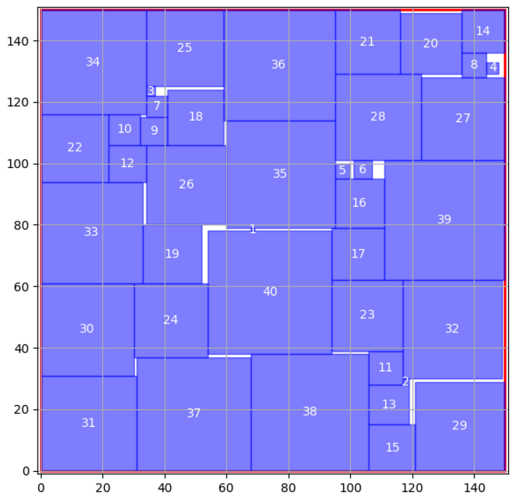

図 1: $N = 40, L = 150$ の解

正方形詰込み問題は N と L が与えられれば一意に決定する。図 1 は、$N = 40, L = 150$ の解の例である。

正方形詰込み問題は、N と L が入力として与えられる決定問題であるが、与えれた N に対して詰込み可能な L の最小値を求める問題は、**正方形詰込み最適化問題**と呼ばれる。

3 基本制約と対称性除去制約

デカルト座標を用いて問題を定式化する。正方形 L の左下の座標を原点 $(0, 0)$ にとり、他の 3 点の座標を $(L, 0), (L, L), (0, L)$ とする。また、正方形 i の左下の座標を (x_i, y_i) とする。本稿では、正方形を斜めに置くことは考慮しないので、他の 3 点の座標は $(x_i + i, y_i), (x_i + i, y_i + i), (x_i, y_i + i)$ となる。

[1]九州大学 Kyushu University
[2]株式会社 豊田自動織機 TOYOTA INDUSTRIES CORPORATION

3.1 基本制約

正方形 i が正方形 L 内に配置される制約は次のように表す（$1 \le i \le N$）。

$$0 \le x_i \le L - i$$
$$0 \le y_i \le L - i$$
$$x_i, y_i \in \mathbb{Z}$$

二つの正方形 i と j が互いに重ならない制約は次のように表す（$1 \le i < j \le N$）。

$$x_i + i \le x_j \lor x_j + j \le x_i \lor y_i + i \le y_j \lor y_j + j \le y_i$$

この制約は正方形 i と j が上下左右それぞれの方向に重ならないことを表している。Z3 にはこの制約をそのまま入力することができるが、Gurobi ではできないため、0-1 変数 $h_{i,j}$ と $v_{i,j}$ を導入して次のような制約として表す [6]。ここで、$h_{i,j} = 1$ は i が j の左にあることを表し、$h_{i,j} = 0$ はそうでないことを表す。また、$v_{i,j} = 1$ は i が j の下にあることを表し、$v_{i,j} = 0$ はそうでないことを表す（$1 \le i, j \le N,\ i \ne j$）。

$$x_i + i \le x_j + L(1 - h_{i,j})$$
$$y_i + i \le y_j + L(1 - v_{i,j})$$
$$h_{i,j} + h_{j,i} + v_{i,j} + v_{j,i} \ge 1$$
$$h_{i,j}, v_{i,j} \in \{0, 1\}$$

3.2 対称性除去制約

詰込み配置には対称性がある。例えば、一つの配置を 90° 回転しても見た目は違う詰込み配置が得られるが実質的には同じ配置である。このような対称性を除去するために最大正方形 N に着目した次のような制約を導入する [3]。

$$x_N \le \lfloor (L - N)/2 \rfloor$$
$$y_N \le x_N$$

4 性能評価

基本制約と対称性除去制約を評価するために、$N = 15, 16, \ldots, 40$ のそれぞれの N に対して、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、詰め込む N 個の正方形の面積の和から求まる値、具体的には $\sqrt{\sum_{i=1}^{N} i^2}$ 以上の最小の整数である。

実験環境は、Linux マシン（Intel Xeon W-1270Px16、メモリ 64GB）である。1 問当たり制限時間は 10000 秒とした。計算時間はソルバーの前処理時間を含む。表 1 に結果を示す。T.O. は制限時間内では解けなかったことを表す。また、各問題に対して最も短時間の CPU 時間をボールド体で記している。配置の欄の○は配置があることを、×は配置がないこと、つまりどうやっても正方形を詰め込むことができないことを、？は（今回の実験では）不明であることを表す。

Z3 と Gurobi を比べると、基本制約のみ、基本制約+対称性除去制約、いずれも Z3 の性能が Gurobi に勝っていることが分かる。解けた問題数で比べると 1.6 倍程度の開きがある。また、Gurobi は N が 30 を超えるとほとんど解けなかった。

平均的には対称性除去制約の効果はあるようにみえるが、問題によっては性能低下をもたらしているものもある。対称性除去制約により探索空間は削減されるが、探索空間における解の密度は変わらないことが原因であると思われる。ただし、密度が 0 である配置がない問題では、探索空間の削減効果が現れている（$N = 18, L = 46$ と $N = 24, L = 70$）。

佐古田ら [7] は、基本制約といくつかの追加制約の SAT 符号化を用いた解法を提案し評価している。使用 CPU に違いがあり、直接の比較はできないが、Z3 の性能と概ね似たような傾向を示している。すなわち、一方ですぐ解ける問題は他方でもすぐに解け、一方で時間を要する問題は他方でも時間を要する。ただ $N = 24, L = 70$ の 1 問だけ、Z3（基本+対称性）で解けているのに対し、佐古田らは解けていない。

正方形詰込み最適化問題は OEIS の A005842: Integer Sequences[3]に、現在までに判明している最適値が示されている。今回の実験で扱った問題（$N \le 40$）で、この最適値に対応する配置が得られなかったのは、$(N, L) = (36, 128), (37, 133), (39, 144)$ の 3 問であった。

5 まとめ

本稿では、正方形詰込み問題の基本制約と対称性除去制約を用いた解法実験を行った。よく知られている数理最適化ソルバー Z3 と Gurobi の性能比較を行ったところ Z3 の優位性が示された。また、対称性除去による計算時間短縮効果も確認できた。今後はこの知見を踏まえて、長方形詰込み問題、直方体詰込み問題にチャレンジしていきたい。

[3] http://oeis.org/A005842

表 1: 計算時間の比較（単位：秒）

N	L	配置	Z3		Gurobi	
			基本	+対称	基本	+対称
15	36	○	0.06	0.05	**0.04**	0.11
	37	○	0.06	0.05	**0.04**	0.05
16	39	○	0.99	**0.12**	0.82	1.32
	40	○	0.08	0.06	**0.05**	0.06
17	43	○	0.48	**0.09**	0.23	0.95
	44	○	0.14	**0.06**	0.14	0.11
18	46	×	196.14	**25.47**	T.O.	T.O.
	47	○	0.51	**0.10**	0.79	0.14
19	50	○	69.04	**0.45**	635.58	1356.16
	51	○	0.28	**0.08**	0.55	0.77
20	54	○	4.24	**1.57**	100.70	25.93
	55	○	0.34	**0.08**	0.35	0.22
21	58	○	22.78	**4.93**	115.65	8.59
	59	○	0.75	**0.10**	0.96	0.37
22	62	○	**188.13**	365.09	T.O.	T.O.
	63	○	1.14	**0.29**	2.23	1.78
23	66	○	**404.61**	728.63	T.O.	T.O.
	67	○	2.73	**1.23**	2.99	4.83
24	70	×	T.O.	**7420.44**	T.O.	T.O.
	71	○	27.46	**11.16**	86.17	146.34
25	75	○	**134.75**	145.93	6690.79	T.O.
	76	○	3.30	**0.24**	2.15	2.66
26	79	?	T.O.	T.O.	T.O.	T.O.
	80	○	2.46	**1.85**	25.11	111.77
27	84	○	26.73	**13.10**	85.27	8454.95
	85	○	13.64	**0.49**	2.09	0.91
28	88	?	T.O.	T.O.	T.O.	T.O.
	89	○	29.38	**3.67**	122.39	3.91
29	93	○	**1137.05**	2334.99	T.O.	T.O.
	94	○	4.72	**1.16**	18.72	5.60
30	98	○	623.58	**26.37**	T.O.	T.O.
	99	○	25.44	3.74	**2.32**	22.64
31	103	○	243.82	**146.38**	T.O.	T.O.
	104	○	25.79	**1.64**	38.76	4.17
32	107	?	T.O.	T.O.	T.O.	T.O.
	108	○	169.96	**56.54**	T.O.	T.O.
33	112	?	T.O.	T.O.	T.O.	T.O.
	113	○	**53.83**	350.39	T.O.	T.O.
34	117	?	T.O.	T.O.	T.O.	T.O.
	118	○	2004.25	**699.94**	T.O.	T.O.
35	123	○	563.10	**408.75**	T.O.	T.O.
	124	○	42.84	**28.02**	T.O.	493.51
36	128	?	T.O.	T.O.	T.O.	T.O.
	129	○	**119.89**	150.85	T.O.	T.O.
37	133	?	T.O.	T.O.	T.O.	T.O.
	134	○	2002.31	**763.15**	T.O.	T.O.
38	138	?	T.O.	T.O.	T.O.	T.O.
	139	○	T.O.	**9916.58**	T.O.	T.O.
39	144	?	T.O.	T.O.	T.O.	T.O.
	145	○	684.92	**186.47**	T.O.	T.O.
40	149	?	T.O.	T.O.	T.O.	T.O.
	150	○	T.O.	**9324.43**	T.O.	T.O.
解けた問題数			39	42	25	25

謝辞

本研究は、豊田自動織機との共同研究である。議論頂いたメンバーの方々に感謝致します。

参考文献

[1] Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual, `https://www.gurobi.com` (2024)

[2] Silvano Martello, Michele Monaci, and Daniele Vigo: An Exact Approach to the Strip-Packing Problem, *INFORMS J. on Computing*, Vol. 15, No. 3, pp. 310-319, (2003)

[3] Helmut Simonis and Barry O'Sullivan: Search Strategies for Rectangle Packing, In Proc. of CP 2008, LNCS 5202, pp.52-66, (2008)

[4] Takehide Soh: Packing Consequtive Squares into a Sqaure, XCSP3 Competition, (2019)

[5] Leonardo Mendonça de Moura and Nikolaj Bjørner: Z3: An Efficient SMT Solver, In Proc. of TACAS 2008, LNCS 4963, pp.337–340, (2008)

[6] 今堀慎治、胡艶楠、橋本英樹、柳浦睦憲：Python における図形詰込みアルゴリズム入門、オペレーションズ・リサーチ、2018 年 12 月号、pp.24-31 (2018)

[7] 佐古田淳史、宋剛秀、番原睦則、田村直之：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、第 27 回人工知能学会全国大会、2E5-OS-09b-1、(2013)