Almost squares in almost squares: solving the final instance

van den Berg, D.; Braam, F.; Moes, M.; Suilen, E.; Bhulai, S.

Publication date
2016

Document Version
Final published version

Published in
Data Analytics 2016

Almost Squares in Almost Squares: Solving the Final Instance

Daan van den Berg

University of Amsterdam,
Faculty of Science,
Amsterdam, The Netherlands
Email: D.vandenBerg@uva.nl

Florian Braam, Mark Moes, Emiel Suilen, and Sandjai Bhulai

Vrije Universiteit Amsterdam,
Faculty of Sciences,
Amsterdam, The Netherlands
Email: florianbraam@gmail.com, mark_moes@hotmail.com,
emiel.suilen@gmail.com, s.bhulai@vu.nl

***Abstract*—The "almost-squares in almost-squares" (Asqas) problem is a rectangle packing problem in which a series of almost-squares (rectangles of dimensions $n \times (n+1)$) needs to be placed inside an almost-square frame without open areas or overlaps. Asqas-34, consisting of almost-squares $1 \times 2, 2 \times 3, \ldots, 34 \times 35$, remains unsolved. This paper shows Asqas-34 is the only remaining unsolved instance of Asqas, and describes several solutions to Asqas-34, and the methods used to find them.**

Keywords–Asqas; almost-squares in almost-squares; rectangle packing problem.

I. INTRODUCTION

Rectangle Packing Problems come in a broad variety and have quite a practical appeal besides their theoretical interest. Minimum-waste fabric cutting in clothes manufacture, maximum storage in warehouses, and optimal arrangement of text and advertisements in newspapers all involve finding the best arrangement of a set of rectangles [1]. While its industrial relevance has been recognized for some time (see, e.g., [2]–[4]), closely related bin packing problems also show a quite remarkable and very interesting extension to scheduling issues [5], where they provide a practical foothold for optimization problems in logistics and planning.

Almost-square packing problems are a special class of rectangle packing problems. A sequence of n almost-square tiles $(1 \times 2, 2 \times 3, \ldots, n \times (n+1))$ must be placed inside a small as possible rectangular frame, with no overlap and as little possible unused space. Most notable is the work of [6], who optimally solved these problems up to $n = 13$ by hand, and up to $n = 26$ by computer, meaning they found exact-fit configurations within frames of appropriate dimensions. Fig. 1 displays the Asqas-8 problem, which is the third of five instances of almost-squares-in-almost-squares, having 8 consecutive almost-square tiles of $1 \times 2, \ldots, 8 \times 9$ to be placed in an almost-square frame, in this case 15×16.

In this paper, we present a solution to the $n = 34$ problem instance, which is not only a relatively large instance of the almost-square packing problem, but also belongs to the slightly more exclusive class of almost-squares-in-almost-squares (Asqas) as well. This is a set of exactly five problem instances for which it is known that a frame of exact fit could have almost-square dimensions as well. The $n = 34$ instance of almost-squares has exactly 13 exact-fit frames, of which 35×408 is the most eccentric and 119×120, the almost-square one, is the most concentric. Erich's packing center, an extensive collection of open and solved packing problems maintained by prof. Erich Friedmann of Stetson University (see [7]) contains solutions to the first four instances of Asqas, but leaves the fifth open. It is this instance that we solve, but it is also the last open instance of Asqas, which we will firstly show.

The structure of the paper is as follows. We discuss the five instances of the Asqas problem in Section II. In Section III, we describe how border sets can help to reduce the complexity of the problem. This is used in Section IV to create borders of the tiles. Finally, in Section V we solve the interior of the border, which leads to the solution of the Asqas-34 problem.

II. THE FIVE INSTANCES OF ALMOST-SQUARES-IN-ALMOST-SQUARES

There are exactly five instances of Asqas and its proof relies on the observation that there is an intriguing bijective function from Asqas to geometry and triangular numbers in number theory closely reminiscent of homeomorphic topological conjugacies used in chaotic discrete dynamical systems [8]. Since a sequence of n almost-square tiles $(1\times2, 2\times3, \ldots, n\times(n+1))$ must be placed inside an almost-square frame, the first check should be whether such a frame actually exists for a given n, or equivalently, whether the summed area of all the separate tiles is equal to the area of an almost-square frame (of any size). More formally put: an Asqas-instance of size n exists only if there is a natural number p such that Equation (1) holds:

$$\sum_{i=1}^{n} i(i+1) = p(p+1). \tag{1}$$

As it turns out, there are exactly five values of n for which the equation holds, and the proof is underpinned by the existence of a near-trivial relation between the sequence of almost-square tiles $1, \ldots, n$ and the triangular numbers $1, \ldots, n$. Fig. 2 shows that summing the first n triangular numbers (bottom row) leads to the nth tetrahedral number. If this number happens to be a triangular number as well, then the existence of an Asqas-instance with the same n is guaranteed by a simple relation between summing triangular numbers (TR) and summing almost squares (AS) – the kth triangle is exactly half the area of the kth almost-square. Note that the relation critically depends on the existence of a triangular number for the nth tetrahedral number (in this case TR_4 for TH_3) and there are only five tetrahedral numbers (TH_1, TH_3,

 ISBN: 978-1-61208-510-4

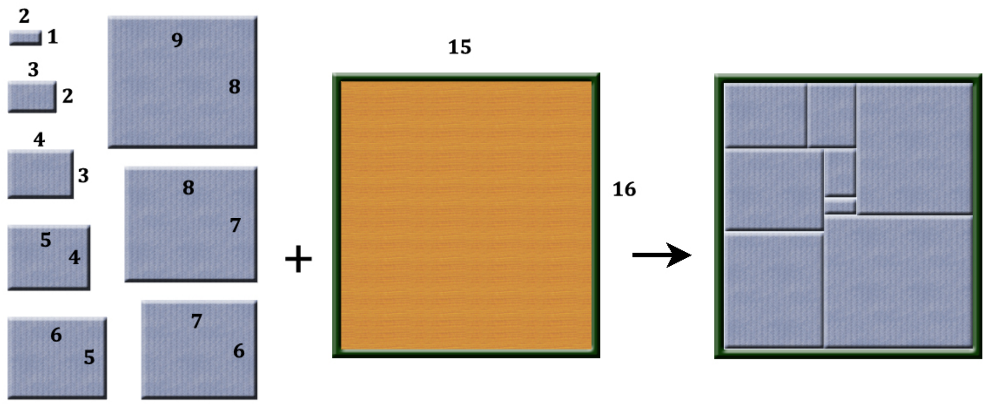

Figure 1. Illustration of the Asqas-8 problem.

TH_8, TH_{20} and TH_{34}) which have a corresponding triangular number. For this reason, there are exactly five Asqas-instances.

The area of any almost-square k is exactly twice the area of the corresponding triangular number k. Since the sum of the areas of the first n almost-squares should be expressible in $p(p+1)$ for the problem instance to exist, the question easily translates to whether the sum of the first n triangular numbers yields a new triangular number. However, the sum of the first n triangular numbers is most commonly known as a tetrahedral number, since 'stacking' subsequent triangles yields a tetrahedron, and as such the question becomes "which tetrahedral numbers are also triangular numbers?". The answer is given by [9]: only the 1st, 3rd, 8th, 20th, and 34th tetrahedral numbers have corresponding triangular numbers (the 1st, 4th, 15th, 55th, and 119th respectively). This ensures us that Asqas-34 is both the largest and the only remaining unsolved instance of Asqas.

A point worth noting is that a similar approach has been adopted for assessing instances of consecutive squares-in-squares; the sum of the series $(1 \times 1, 2 \times 2, \ldots, n \times n)$ gives a square pyramidal number P_n. Only two numbers are both square and pyramidal: $P_1 = S_1 = 1$ and $P_{24} = S_{70} =$ 4,900 [10], the first being trivial and the second, consisting of the first 24 consecutive squares having no solution at all [11]. This instance is a special cases of 'perfectly squared squares' (see [12], [13]) and a nice visual overview can be found at [14].

III. Border sets and eligible border sets

When considering any tight configuration of tiles in a frame as a sequence, the number of possible arrangements is equal to $n!$. In the almost-square case, tiles can be put either horizontally or vertically, doubling the possibilities for each individual tile, and as such increasing the total number of configurations by a factor 2^n. For the case of Asqas-34, the number of possible arrangements is therefore equal to Expression (2):

$$34! \cdot 2^{34} = 507{,}206{,}086{,}632{,}656 \cdot 10^{34} \approx 5 \cdot 10^{48}. \quad (2)$$

Roughly speaking, any number of states exceeding 10^{20} becomes too cumbersome for calculation on a single computer within reasonable time.

Our heuristic approach is mainly fed by the observation that for Asqas-20, larger tiles are situated in the border of the frame. Asqas-20 has 54,992 solutions having a total of 9,812 different borders. Fig. 3 shows that the number of tiles in these borders follows a narrow distribution (left), with larger tiles more prevalent than smaller tiles (right), and some of the larger tiles (14×15, 17×18, 18×19 and 19×20) being present in the border of almost every solution. For this reason, it makes sense to start looking for solutions to Asqas-34 with larger tiles in the border.

For this purpose, without loss of generality, we split the set of tiles of the $n = 34$ instance into 'border sets' of b tiles, and the remaining 'interior sets' of $34 - b$ tiles. The number of any possible border sets of size b then equals $\binom{34}{b}$ and each of these has $b! \cdot 2^b$ of possible configurations, whereas the complementary interior set is left with $34 - b$ tiles, with $(34-b)! \cdot 2^{34-b}$ configurations. As such, the size of the state-space can better be viewed as in Expression (3):

$$\binom{34}{b} \cdot b! \cdot 2^b \cdot (34-b)! \cdot 2^{34-b}, \quad (3)$$

for any b in the range 1 to 34. The initial drawback is that for our approach to be complete, we have to repeat this procedure for every $b = 1, 2, \ldots, 34$ and as such multiply the gross calculations with 34. This however, turns out to be a small investment with a huge return. Firstly, it allows us to filter out 'eligible border sets' from which potentially valid borders can be constructed from non-eligible border sets with relative computational ease. The second advantage is rather practical: the compartmentation of the state space allows designated areas to be marked as 'covered', and even if no solution was to be found it would serve as progress for other teams working on the subject. But the third advantage, directly related to the second, is that by compartmenting the state space into borders of size b and interiors of size $34-b$, we can start an optimized complete search in the area of the smallest border sets – which have the largest tiles on average. This assumption however, was not thoroughly investigated and therefore starting here was a clear cut heuristic: an educated guess.

Sifting eligible border sets from discardable non-eligible border sets was done simply on the global constraint of perimeter: every set has a maximum perimeter ($p_{\max}$) and a

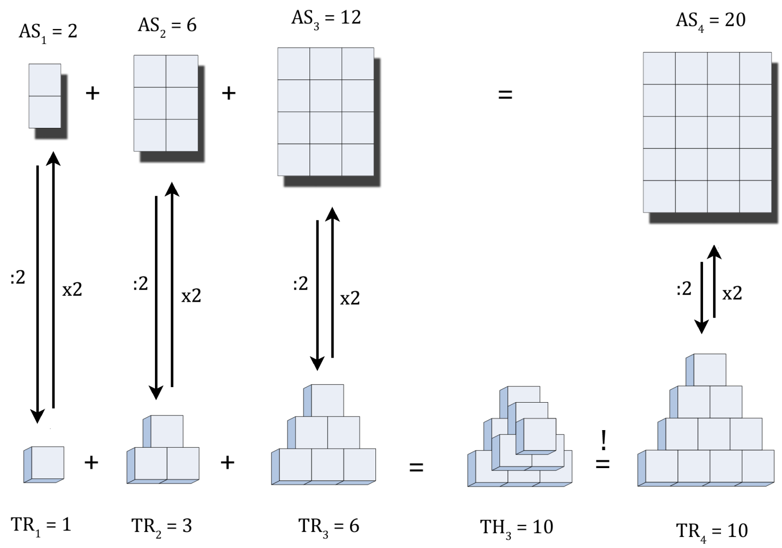

Figure 2. Relation between the sum of almost squares and triangular numbers.

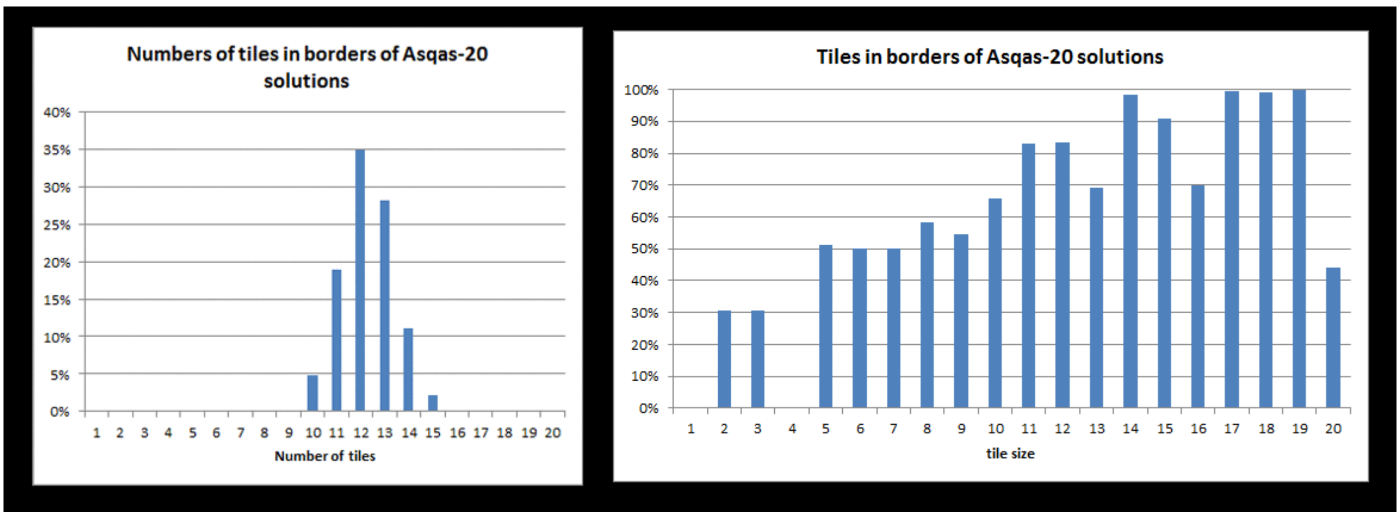

Figure 3. Distribution of the number of tiles in Asqas-20.

minimum perimeter ($p_{\min}$). The maximum perimeter is the greatest distance that any set of tiles can cover when placed along the inside of the frame in constructing a border from that set. It assumes the largest tiles to be put in the corners and all others with their long sides on the frame. As such, $p_{\max}$ is the sum of all the long sides of tiles in the set, plus the short sides of the four largest tiles in the set. The minimum perimeter assumes the opposite: placing the smallest tiles of the set in the corners of the frame and all others with their short sides on the frame, and as such is the sum of the short sides of all tiles in the set, plus the long sides of the four smallest tiles. The total length of the Asqas-34 frame is 119 + 119 + 120 + 120 = 478 and as such any set which has $p_{\min} \leq 478 \leq p_{\max}$ was to be marked *eligible*; all others were discarded. It is easy to see there are no eligible 3-border sets (too short) and no eligible 31-border sets (too long), but it should be noted that for various reasons, many of the eligible border sets have no validly constructable borders at all. The perimeter-approach has the great advantage of being computationally very effective, allowing the number of eligible border sets to be greatly reduced in a few days on a single stand-alone computer. On an abstract level, this procedure somewhat resembles the approach described by [15], using global constraints for packing rectangles as it “it leads to improvement in execution times”, which of course, in terms can make an unsolvable problem solvable.

IV. Making borders of 12 tiles

The reduction from all border sets to eligible border sets only greatly varied with the number of tiles involved in a set, but was most drastic at its extremes. Table I shows that the number of border sets with b tiles is exactly $\binom{34}{b}$, but only a small fraction of these meet the eligibility-criterion of having enough but not too much length to potentially construct

TABLE I. NUMBER OF BORDER SETS IN THE REDUCTION.

set size	#sets	#eligible sets		(cont.)	(cont.)	(cont.)
≤ 11	var.	0		21	927,983,760	788,458,963
12	548,354,040	30		22	548,354,040	502,691,341
13	927,983,760	18,014		23	286,097,760	256,326,310
14	1,391,975,640	749,552		24	131,128,140	104,965,820
15	1,855,967,520	10,072,560		25	52,451,256	33,285,209
16	2,203,961,430	64,774,105		26	18,156,204	7,530,562
17	2,333,606,220	238,039,950		27	5,379,616	1,053,473
18	2,203,961,430	552,324,976		28	1,344,904	69,036
19	1,855,967,520	863,895,027		29	278,256	933
20	1,391,975,640	958,043,855		≥ 30	var.	0

a border of 478 units long. The percentage eligible sets is smallest at its extremes, for $b = 12$ and $b = 29$.

As our heuristic assumed that a solution of the Asqas-34, if existent, was likely to have larger tiles situated in the border, and the smaller sets necessarily consist of larger tiles, these two factors led us to explore all eligible border sets consisting of 12 tiles. There are only 30 of those sets, having a total of $30 \cdot 12! \cdot 2^{12} = 58{,}859{,}716{,}608{,}000 \approx 5.9 \cdot 10^{13}$ possible borders, a number small enough to exhaustively compute on a stand-alone computer with a simple backtracking algorithm. Fig. 4 shows that the number of valid borders per eligible 12-set differs considerably, but roughly follows the fluctuations of $p_{\max}$. Sets have been numbered #1 – #30 following tile size (#1 having the largest tiles) but the number of valid borders shows no relation to this numbering (horizontal axis). Remarkably enough, the inset shows that when plotted in a log-normal scale the distribution of sets almost follows a straight line with a slope of -0.12 and intercept 6.13 (correlation coefficient: -0.994).

Of the $5.9 \cdot 10^{13}$ possible configurations, only 4,425,341 actually turned out to be valid borders (see Fig. 4). The border-construction algorithm was complete, and effectively ignored flip-isomorphic borders and partial-flip-isomorphic borders. Two borders are flip-isomorphic if a complete horizontal or vertical flip changes one into the other, and two borders are partial-flip-isomorphic if flipping two adjacent tiles in the border leaves the shape of the interior unchanged. It is worth noting that it is theoretically quite possible that other isomorphic borders were still present in our set, and the only way to ensure this is storing the exact polygon shape of the remaining interior and the accompanying set of interior tiles. We never bothered, for the storage capacity and checking algorithm needed for this mechanism to work appears to greatly exceeded its practical benefit, if any.

The total number of 4,425,341 borders constructable from any 12 tiles is fairly comprehensive, but each of them still needed its remaining 22 tiles to be arranged in at most $22! \cdot 2^{22}$ configurations, yielding a total of $4{,}425{,}341 \cdot 471{,}440{,}074{,}852{,}053 \cdot 10^{13} = 208{,}628{,}309{,}228{,}586 \cdot 10^{20} \approx 2 \cdot 10^{34}$ possible configurations to explore, still far too many to analyze on a stand-alone computer. From here, we put our faith in all the optimizations we could think of for solving the interior, the computational power of third and fourth generation Distributed ASCI Supercomputers (DAS-3 and DAS-4), and the correctness of our heuristic intuition.

V. SOLVING THE INTERIOR

A pilot run showed that the calculation time for deciding whether any given valid border had a solvable interior, e.g., contained a complete Asqas-34 solution, was extremely unevenly distributed. Whereas the vast majority of borders was decided in a few minutes at most, approximately one in thousand took nearly a day to be completely puzzled out. In a worst-case scenario, this would stall even a supercomputer of 100 calculating nodes for over a month with all the quickly decidable borders waiting behind it. To effectively manage this risk, to maximize the number of analyzed borders or, ultimately, to find a solution to Asqas-34, we set up a small server with 4,439 text files, each containing at most $1,000$ border configurations, effectively covering all 4,425,341 borders of 12 tiles. The text files' content was arranged in accordance with the 30 different sets, but the files were randomly distributed over 80 nodes of DAS-3 grid computers and up to 90 nodes of DAS-4 grid computers located throughout The Netherlands and Belgium, each node running several instances of the interior solver (see Fig. 5 for the computational setup).

Each instance of the interior solver comprised an optimized complete backtracking algorithm that, on startup, retrieved one text file with 1,000 borders from our server, read the first border from the file, placed the 12 tiles inside the frame and commenced an exhaustive backtracking routine using the remaining 22 tiles to solve the interior. The tiles were placed starting on the first empty position bottom-left, and starting off with the largest tile standing up first. It optimized in three ways: firstly, whenever the first row was not completed, the search for the border was cut off and dismissed as unsolvable. Second, if a row was completed, it was checked for "impossible gaps", mostly high and narrow spaces which could not be filled by any combination of tiles. Third, rectangles were not reversed, meaning that any two consecutively placed tiles in the interior forming a rectangle were not checked in reverse order. This has an optimizing effect but the apparent drawback of missing solutions that only differed by a partial swap of two tiles. That effect however, was nearly insignificant as the number of solutions turned out to be so small they could be hand-checked.

The text file server tracked the progress per file, but also per border to minimize the "redo" time in case results came back incomplete.

The employed DAS3- and DAS4-nodes were located at VU and UvA universities in Amsterdam, Technical University Delft, Leiden University (LIACS), the astronomical ASTRON center in Drenthe, and at the Lab for Perceptual Dynamics of the Catholic University in Leuven, Belgium. Each DAS-3

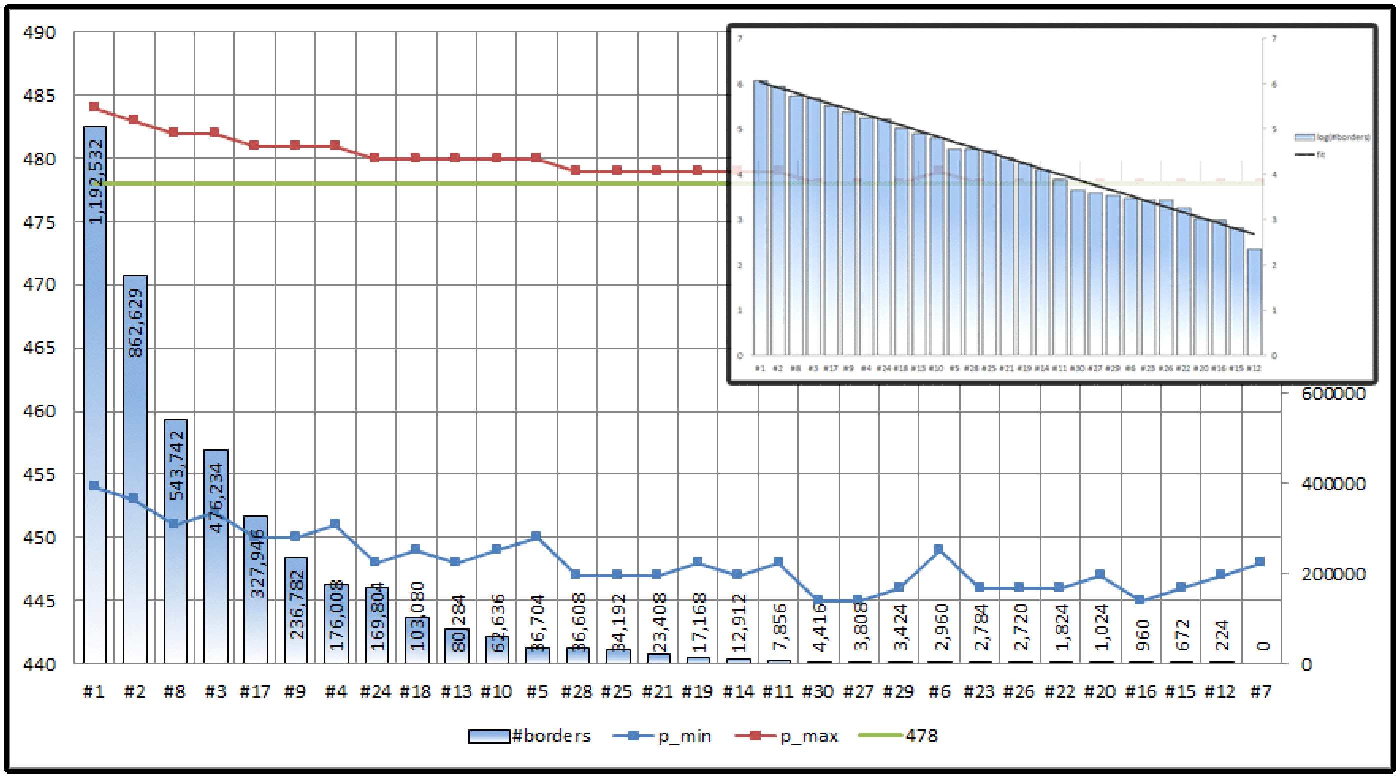

Figure 4. Number of valid borders per eligible 12-set.

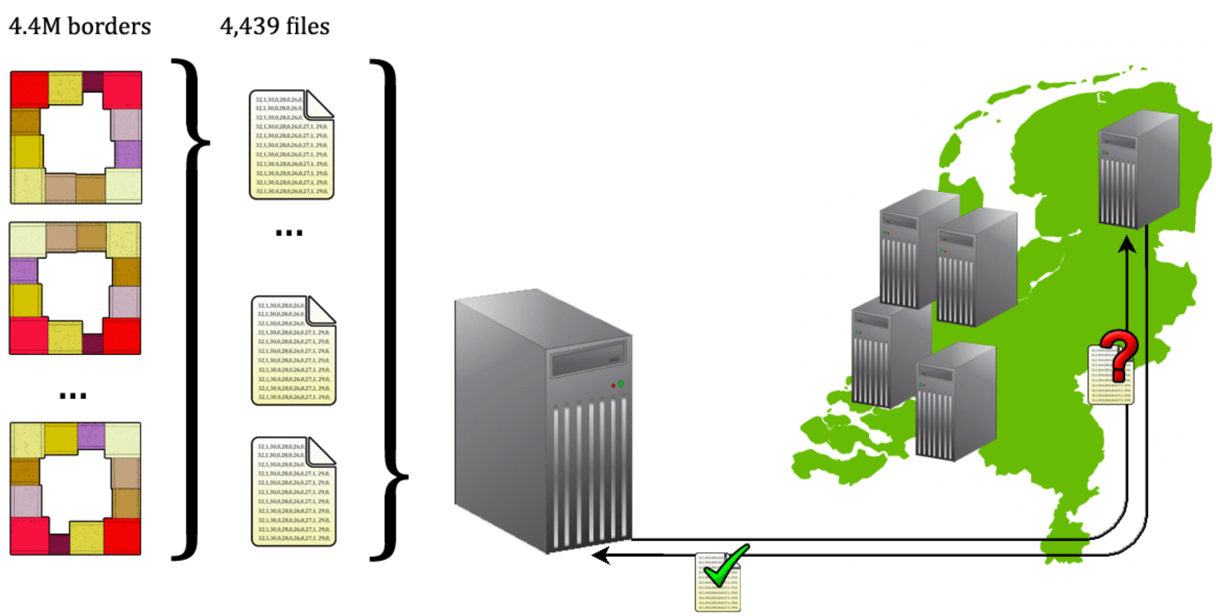

Figure 5. Setup for the solving the Asqas-34 problem.

node can simultaneously run up to four instances of an interior solver, each DAS-4 node up to eight. It is very hard to give an exact run-time estimate since each of these computers is used in a great variety of tasks, has highly volatile availability, occasionally malfunctions, was temporarily shut down for maintenance or were refused reservations for unknown reasons. So although we have completed the entire 12-set, it is quite hard to give an accurate run time estimate.

For a project of this scale, we can give a reasonable upper bound estimate: the entire set was exhaustively investigated in 80 days, running between 800 and 1,000 instances of the interior solver in parallel for between 96 and 116 hours a week. In this time, it found exactly 15 unique solutions to the Asqas-

 ISBN: 978-1-61208-510-4

Figure 6. Two solutions to Asqas-34.

34 problem, which are its only solutions with 12 tiles in the border. Nonetheless, solutions with 13 or 14 border tiles are known to exist, because they can be constructed from one of the found solutions. Fig. 6 depicts two solutions to Asqas-34. Note that as both solutions have isomorphic solutions from a complete horizontal or vertical flip, the left hand solution holds potentially three others solutions from flipping either pair of highlighted tiles. The right hand solution can be transformed into at least 255 other solutions by flipping tiles or groups of tiles (highlighted and darkened), some of which have more than 12 tiles in the border.

ACKNOWLEDGEMENTS

Our thanks go to Cees van Leeuwen and Marco Maas for letting us use their server at the Laboratory for Perceptual Dynamics at Catholic University in Leuven, and their support in doing so. Further gratitude is to Rena Bakhshi (VU), Guillaume Pierre (VU), and especially Dion Gijswijt from TU Delft for proofreading this paper so thoroughly.

REFERENCES

[1] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: A survey," *European Journal of Operational Research*, vol. 141, pp. 241–252, 2002.

[2] K. A. Dowsland and W. B. Dowsland, "Packing problems," *European Journal of Operational Research*, vol. 56, pp. 2–14, 1992.

[3] L. Kantorovich, "Mathematical methods of organising and planning production," *Management Science*, vol. 6, pp. 366–422, 1960.

[4] M. Formann and F. Wagner, "A packing problem with applications to lettering of maps," in *Proceedings of the Seventh Annual Symposium on Computational Geometry, SCG '91*, 1991, pp. 281–288.

[5] J. E.G. Coffman, M. Garey, and D. Johnson, "An application of bin-packing to multiprocessor scheduling," *SIAM Journal of Computing*, vol. 7, no. 1, 1978.

[6] H. Simonis and B. O'Sullivan, "Almost square packing," in *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Springer-Verlag Berlin, 2011, pp. 196–209.

[7] E. Friedman, "Erich's packing center," URL: http://www2.stetson.edu/ efriedma/packing.html, 2005, retrieved: August 29, 2016.

[8] R. A. Holmgren, *A first course in discrete dynamical systems*, 2nd ed. Springer, New York, 1996, chapter 9.

[9] E. T. Avanesov, "Solution of a problem on figurate numbers (russian)," *ActaArith.*, vol. 12, pp. 409–420, 1966/1967, via http://mathworld.wolfram.com/TetrahedralNumber.html.

[10] G. Watson, "The problem of the square pyramid," *Messenger. Math.*, vol. 48, pp. 1–22, 1918.

[11] E. W. Weisstein, "Perfect square dissection," URL: http://mathworld.wolfram.com/PerfectSquareDissection.html, 2012, retrieved: August 29, 2016.

[12] R. L. Brooks, C. Smith, A. Stone, and W. Tutte, "The dissection of rectangles into squares," *Duke Math. J.*, vol. 7, no. 312–340, 1940.

[13] A. Duijvestijn, "Simple perfect squared square of lowest order," *J. Combin. Theory Ser. B*, vol. 25, no. 2, pp. 240–243, 1978.

[14] Retrieved: August 29, 2016. [Online]. Available: www.squaring.net

[15] H. Simonis and B. O'Sullivan, "Using global constraints for rectangle packing," in *Proceedings of the first Workshop on Bin Packing and Placement Constraints BPPC 2008, associated to CPAIOR 2008*, 2008.