Optimal Packing of High-Precision Rectangles

Eric Huang
Palo Alto Research Center
3333 Coyote Hill Rd.
Palo Alto, CA 94304
ehuang@parc.com

Richard E. Korf
Computer Science Department
University of California, Los Angeles
Los Angeles, CA 90095
korf@cs.ucla.edu

Abstract

The rectangle-packing problem consists of finding an enclosing rectangle of smallest area that can contain a given set of rectangles without overlap. Our new benchmark includes rectangles of successively higher precision, a problem for the previous state-of-the-art, which enumerates all locations for placing rectangles. We instead limit these locations and bounding box dimensions to the set of subset sums of the rectangles' dimensions, allowing us to test 4,500 times fewer bounding boxes and solve N=9 over two orders of magnitude faster. Finally, on the open problem of the feasibility of packing a specific infinite series of rectangles into the unit square, we pack the first 50,000 such rectangles and conjecture that the entire infinite series can fit. Our solver is available at `http://code.google.com/p/rectpack`.

Introduction and Previous Work

Given a set of rectangles, our problem is to find all enclosing rectangles of minimum area that will contain them without overlap. We refer to an enclosing rectangle as a *bounding box*. The optimization problem is NP-hard, while the problem of deciding whether a set of rectangles can be packed in a given bounding box is NP-complete (Korf 2003). We introduce the *unoriented high-precision rectangle-packing benchmark*, where the given rectangles are those in the set $\frac{1}{1}$x$\frac{1}{2}$, $\frac{1}{2}$x$\frac{1}{3}$, ..., up to $\frac{1}{N}$x$\frac{1}{N+1}$ and the rectangles may be rotated by 90 degrees. For example, for N=4 one must pack rectangles of sizes $\frac{1}{1}$x$\frac{1}{2}$, $\frac{1}{2}$x$\frac{1}{3}$, $\frac{1}{3}$x$\frac{1}{4}$, and $\frac{1}{4}$x$\frac{1}{5}$. Alternatively we can scale the instance by a factor of 60, the least common multiple of the rectangle dimensions, for rectangles of sizes 60x30, 30x20, 20x15, and 15x12.

Korf (2003) divided the rectangle-packing problem into the *containment problem* and the *minimal bounding box problem*. The former tries to pack the given rectangles in a given bounding box by trying all possible locations for each rectangle, while the latter finds a bounding box of least area that can contain a given set of rectangles and calls the former algorithm as a subroutine. Simonis and O'Sullivan (2008) and our previous work (2009) further improved performance but still enumerated the rectangles' locations. This fails in our new benchmark. For example, the minimum bounding box of N=15 has over 400 billion such locations.

Rectangle packing has many applications from pallet loading to cutting stock to layout problems. It also models scheduling problems where tasks require resources allocated in contiguous chunks, such as scheduling and allocating memory address ranges to programs. Here, the width and height of a rectangle represents the amount of time and contiguous memory a program needs, respectively.

Overall Strategy

Given an instance from our high-precision benchmark described in rational numbers, we multiply all values by the least common multiple to get an instance of integer dimensions. We then solve the problem as we did before (2010), but with improvements we will subsequently explain.

We limit all coordinate values to subset sums of the rectangle dimensions, because if a solution exists, then there is always one where all rectangles are in contact with other rectangles on the left and below. Similarly, the bounding box dimensions can be limited to the same subset sums too.

Perfect Packing Transformation

Alongside the strategy of asking where a given rectangle can go, we previously (2009) also asked for a given corner, which rectangle may be placed there. This second search space required creating a number of 1x1 rectangles equal to the amount of empty space. The transformation yielded a new instance, without empty space, and consisting of the original rectangles plus the new 1x1 rectangles.

In our high-precision benchmark, this creates over 1.5 billion rectangles for N=15 since we scaled the problem up by the least common multiple. Here our old technique breaks down. We now avoid this problem by inferring when many of these 1x1 rectangles must be adjacent to one another and instead create one large rectangle to represent them.

Experimental Results

Table 1 compares the number of bounding boxes and CPU time that two solvers require to find all optimal solutions on our new benchmark. The first column gives the problem size. Column HK10 Boxes is the number of bounding boxes tested by our previous solver (Huang and Korf

Size N	HK10 Boxes	HK11 Boxes	HK10 Time	HK11 Time
6	1,979	29	:00	:00
7	4,033	46	:02	:00
8	39,357	124	1:11	:00
9	13,571	192	1:51	:00
10	2,682,948	585		:01
11		1,641		:18
12		2,366		:33
13		5,027		16:41
14		9,548		46:56
15		15,334		4:28:20

Table 1: Number of bounding boxes tested and CPU time required to find the minimal bounding boxes containing an instance of our new benchmark.

2010) when simply scaling up the problem to an integer instance. By comparison, HK11 Boxes corresponds to that of our new solver, which by N=10 tests 4,500 times fewer bounding boxes. Here HK10 Boxes ran out of memory on the last bounding box. The presence of the prime number 11 in the problem instance made N=10 much more difficult than N=9. Column HK10 Time is the time required by our old containment solver, using our improved minimal bounding box algorithm. HK11 Time is that of our new solver. By N=10, HK10 needed to create 6,597,361 1x1 rectangles and ran out of memory. This data was collected using a Linux eight core 3GHz Intel Xeon X5460 using one process, one thread, and one core.

Packing an Infinite Series of Rectangles

Meir and Moser (1968) asked what was the smallest square that can contain an infinite series of unoriented rectangles of sizes $\frac{1}{1}$x$\frac{1}{2}$, $\frac{1}{2}$x$\frac{1}{3}$, $\frac{1}{3}$x$\frac{1}{4}$, ..., etc. The total area of the infinite series equals that of the unit square and it is unknown whether a feasible packing exists for the series in the unit square.

Though there has been analytical work on this problem, more recently there have been computational attempts (Cantrell 2010) that reported packing 10,000 rectangles. Cantrell's greedy heuristic placed rectangles in the first feasible bottom-most left-most position (Chazelle 1983), but his methods were not described in sufficient detail. We use the same heuristic, and index all placed rectangles with a four-dimensional kd-tree (Bentley 1975) for overlap testing.

Figure 1 shows the first 50,000 rectangles of the infinite series. It seems that successive rectangles shrink faster than they are able to break up the empty space and so we conjecture that packing the infinite series is feasible.

Conclusion

We have proposed a new benchmark consisting of instances with rectangles of high-precision dimensions, presented techniques for using subset sums to limit the number of rectangle positions, and presented ways to reduce the number

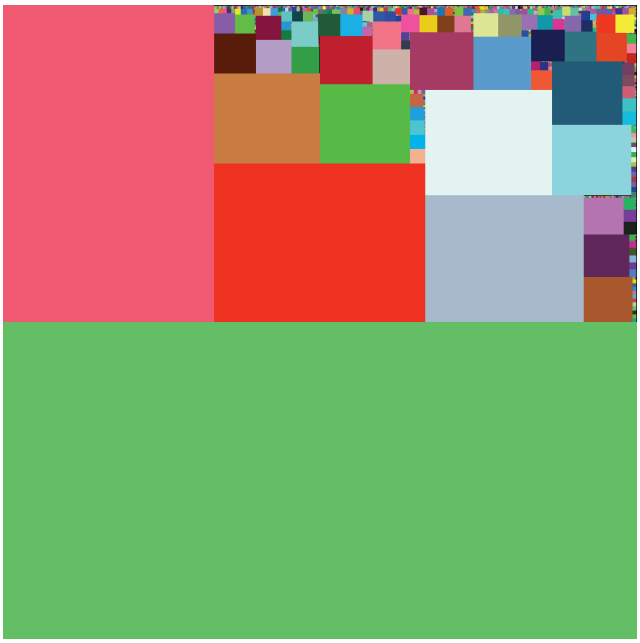

Figure 1: A packing solution for unoriented rectangles of sizes $\frac{1}{1}$x$\frac{1}{2}$, $\frac{1}{2}$x$\frac{1}{3}$, $\frac{1}{3}$x$\frac{1}{4}$, ..., $\frac{1}{50,000}$x$\frac{1}{50,001}$ in the unit square.

of rectangles created during the perfect packing transformation. Our solver is over two orders of magnitude faster at N=9 than the previous state-of-the-art and tests 4,500 times fewer bounding boxes. Finally, we report our computational results on the problem of packing an infinite series of rectangles in the unit square, and conjecture that a packing exists.

References

Bentley, J. L. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18(9):509–517.

Cantrell, D. W. 2010. Personal communication.

Chazelle, B. 1983. The bottomn-left bin-packing heuristic: An efficient implementation. *IEEE Transactions on Computers* C-32(8):697–707.

Huang, E., and Korf, R. E. 2009. New improvements in optimal rectangle packing. In Boutilier, C., ed., *IJCAI*, 511–516.

Huang, E., and Korf, R. E. 2010. Optimal rectangle packing on non-square benchmarks. In *AAAI'10: Proceedings of the 24th National Conference on Artificial intelligence*, 317–324. AAAI Press.

Korf, R. E. 2003. Optimal rectangle packing: Initial results. In Giunchiglia, E.; Muscettola, N.; and Nau, D. S., eds., *ICAPS*, 287–295. AAAI.

Meir, A., and Moser, L. 1968. On packing of squares and cubes. *Journal of Combinatorial Theory* 5(2):126–134.

Simonis, H., and O'Sullivan, B. 2008. Search strategies for rectangle packing. In Stuckey, P. J., ed., *CP*, volume 5202 of *Lecture Notes in Computer Science*, 52–66. Springer.