Cutting rectangles in equal area pieces

Prosenjit Bose* Jurek Czyzowicz† Dominic Lessard*

1 Introduction

Problem 1 *Let P be an axis parallel rectangle. We want to cut P in k equal area pieces such that the total length of the cuts is minimum.*

The problem depends on how the cuts are made. Although cuts may take a general form, in this paper we restrict our attention to straight line orthogonal cuts. We assume that each cut is complete in that it divides a rectangle into two pieces.

Several variants of this problem have appeared in the literature. Overmars and Welzl [4] studied the problem of cutting a polygon drawn on a piece of paper in the cheapest possible way. Croft, Falconer and Guy [2] studied problems related to tiling and dissection of circles and squares. Frederickson [3] discusses several interesting problems related to geometric dissection. Bose et al.[1] proved many results about cutting squares and circles in equal area pieces. In section 2, we propose solutions to problem 1. In section 3 we present some results from computer simulations and discuss their consequences.

2 Algorithms

A recursive algorithm can be used to solve problem 1. The inputs are the initial rectangle and the number of equal area pieces to be cut. The output is the optimal set of straight line orthogonal cuts. Note that the first (horizontal or vertical) cut can only be made in $2\lfloor \frac{k}{2} \rfloor$ ways. After the first cut has been made, the algorithm recursively solves the two sub-problems. In this way, every possibility is verified and the optimal solution is returned.

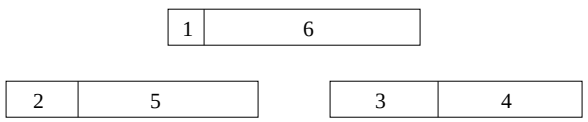

Figure 1: 3 possibilities for a first vertical cut, $k = 7$.

*Research supported in part by NSERC. School of Computer Science, Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada, K1S 5B6. E-mail: (jit,dominic)@scs.carleton.ca

†Research supported in part by NSERC. Département d'Informatique, Université du Québec à Hull, Hull, Quebec. Email:czyzowicz@uqah.uquebec.ca

Below we summarize the algorithm:
Optimize(R, k)
Inputs: A rectangle R ($a \times b$) and k. Without loss of generality, assume that $a \leq b$.
Outputs: Set of straight line cuts (or k rectangles).

BEGIN
    $T \leftarrow \phi$
    $L \leftarrow \infty$
    FOR $i = 1$ to $\lfloor \frac{k}{2} \rfloor$
        $R_1 \leftarrow \text{Rectangle}\left(a, \frac{ib}{k}\right)$
        $R_2 \leftarrow \text{Rectangle}\left(a, \frac{(k-i)b}{k}\right)$
        $T_1 \leftarrow \text{Optimize}(R_1, i)$
        $T_2 \leftarrow \text{Optimize}(R_2, k-i)$
        $l \leftarrow$ total length of the cuts from $T_1 \cup T_2$
        If $l < L$ then
            $T \leftarrow T_1 \cup T_2$
            $L \leftarrow l$
    END of the loop.
    Repeat the loop for horizontal cuts.
    Return T.
END

Note: *Rectangle*(a, b) creates a rectangle with dimensions $a \times b$.

Theorem 1 *Optimize(R, k) solves problem 1 and runs in $O(6^k)$ time.*

Proof: The number of elementary operations is given by

$$T(k) = 2 \sum_{i=1}^{\lfloor k/2 \rfloor} (T(i) + T(k-i) + c_1) \text{ where } c_1 \text{ is a small constant}$$

From this we deduce the following:

$$\begin{aligned} T(k) - T(k-1) &\leq 2T(\lfloor k/2 \rfloor) + 2T(k-1) + 2c_1 \\ T(k) &\leq 3T(k-1) + 2T(\lfloor k/2 \rfloor) + 2c_1 \\ &\leq 6T(k-1) \end{aligned}$$

This implies that the algorithm above is $O(6^k)$. ■

The time complexity of the algorithm can be slightly improved by retaining optimal solutions of subproblems in a table in order to avoid a recomputation of the same problem multiple times. However, even with this table the algorithm remains exponential.

Theorem 2 *There exist instances where the algorithm described above runs in* $\Omega(2^{k^{1-\epsilon}})$ *time for any* $\epsilon > 0$.

Proof: Note that a problem instance is specified by three parameters: the number of pieces, the length of the rectangle and the height of the rectangle. To prove the lower bound, we show that the algorithm needs to verify many unique problem instances.

Let $m = c\lfloor \frac{k}{\log k} \rfloor$ be the number of prime numbers smaller than or equal to k for some constant c (prime numbers theorem [5]). Let $P_1 > P_2 > ... > P_m$ represent these primes. There are 2^{m-1} different subsets containing an even number of such primes. For each such subset, we show how to construct a unique problem instance.

Without loss of generality, let $P_{s1}, P_{s2}, \ldots, P_{s2i}$ represent the 2i primes in decreasing order of one such subset. Figure 2 shows an example of how these primes are used to construct a unique instance. We apply alternatively apply horizontal and vertical cuts until we cut out a a rectangle with area $1/k$. This rectangle will be a unique problem instance and the prime numbers determine where the cut occurs. The first vertical cut occurs such that we have P_{s1} pieces to the right and $k - P_{s1}$ pieces to the left. Notice that the rectangle to the right has dimensions $(P_{s1}/k \times 1)$. The next cut is a horizontal cut of the latter rectangle with P_{s2} pieces above. This rectangle has dimensions $(P_{s1}/k \times P_{s2}/P_{s1})$. The next rectangle has P_{s3} pieces to the right and dimensions $(P_{s1}P_{s3}/P_{s2}k \times P_{s2}/P_{s1})$. This process continues until we have one rectangle in the top right corner. This rectangle will have the following dimensions:
$\left(\frac{P_{s1}P_{s3}...P_{s(2j-1)}}{P_{s2}P_{s4}...P_{2i}k} \times \frac{P_{s2}P_{s4}...P_{s2i}}{P_{s1}P_{s3}...P_{s(2j-1)}}\right)$.

Since the dimensions of the rectangle are determined by the primes, each instance is unique. The number of subsets is 2^{m-1} which implies the result since

$$\lim_{k\to\infty} \frac{k^\epsilon}{\log k} = \lim_{k\to\infty} \ln(10)\epsilon k^\epsilon = \infty$$

for any $\epsilon > 0$. ■

Proposition 1 *[1] There exists an approximation algorithm that finds a solution that is no more than* $1 + \frac{1}{2(\sqrt{k}-1)}$ *times the optimal solution in* $O(1)$ *time*[1]. $\Omega(k)$ *time is required to report the cuts.*

3 Results

Since the square is the minimum perimeter rectangle enclosing a fixed area, we have a lower bound on the solution of problem 1.

Theorem 3 *The total length of the straight line cuts is greater or equal to* $2\sqrt{k} - 2$ *for the unit square and* $2\sqrt{kab} - a - b$ *for a rectangle.*

In order to gain a better understanding of the structure of optimal solutions, simulations of algorithm Optimize(R, k) were made[2]. From the simulations, we noticed some recurring patterns permitting us to make the following conjectures.

[1] Assuming that square root of numbers are computed in constant time.

[2] The reader will find many results of these simulations on the following web site:
`http://www.scs.carleton.ca/~dominic/`.

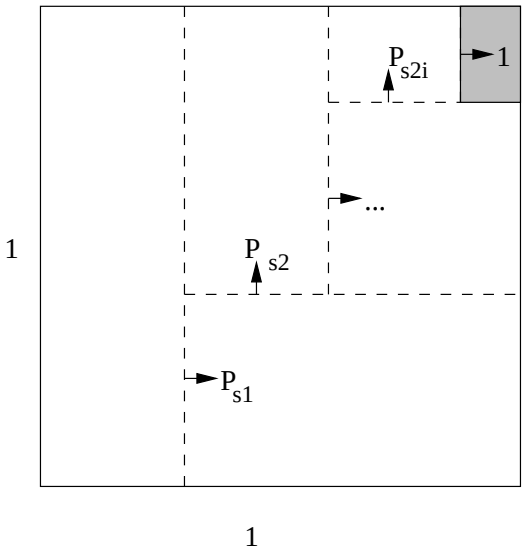

Figure 2: Cuts of a unit square

Conjecture 1 *The approximation algorithm presented in [1] is optimal.*

Conjecture 2 *The optimal solution of problem 1 contains at most two types of rectangles.*

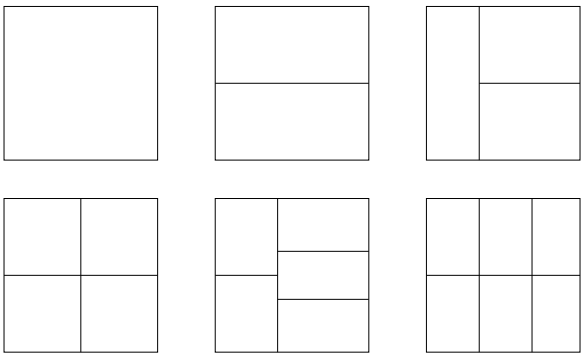

Figure 3: Squares cut in 2, 3, 4, 5 and 6 pieces optimally.

Two examples of this are shown in figures 3 and 4. Note that in figure 4 we have an example showing that the optimal solution is not unique.

Theorem 4 *If Conjecture 2 is true then there exists an algorithm solving problem 1 in $O(k^{3/2})$ time.*

Proof: The algorithm is similar to Optimize(R, k). (same inputs and same outputs.) Conjecture 2 allows us to remove the two recursive calls. Once the first cut has been made, each of the two resulting rectangles must be cut into pieces that are the same, otherwise we will have more than two types of rectangles in the final solution.

Figure 4: Two rectangles (1×6) cut in 13 pieces optimally.

To cut a rectangle into k pieces of the same type, h horizontal cuts and v vertical cuts are made such that $h \times v = k$. Therefore, we need to compute all factors of k. There are $O(\sqrt{k})$ such factors. The time complexity is therefore

$$\begin{aligned} T(k) &= 2\sum_{i=1}^{\lfloor \frac{k}{2} \rfloor} \left(c_2 + 2c_1\sqrt{k} \right) + 3 \\ &\leq k\left(c_2 + 2c_1\sqrt{k} \right) + 3 \\ &\leq c_2 k + 2c_1 k^{3/2} + 3 \end{aligned}$$

which implies that the algorithm runs in $O(k^{3/2})$ time. ■

The next table shows the optimal length of a unit square and a rectangle (7×9) cut into k equal area pieces by the two algorithms. The results seem to support the two conjectures.

	Unit square			Rectangle (7×19)	
k	Theorem 1	$O(6^k)$ algo.	$O(k^{3/2})$ algo.	$O(6^k)$ algo.	$O(k^{3/2})$ algo.
2	0.828427	1.0	1.0	7.0	7.0
3	1.4641	1.6667	1.6667	14.0	14.0
4	2.0	2.0	2.0	21.0	21.0
5	2.472136	2.6	2.6	28.0	28.0
10	4.324556	4.4	4.4	47.0	47.0
15	5.745967	5.8	5.8	64.733	64.733
20	6.944272	7.0	7.0		77.9
25	8.0	8.0	8.0		89.52

References

[1] Bose P.K., Czyzowicz, J., Kranakis, E., Krizanc, D. and Maheshwari, A. (1998), *Cutting circles and squares in equal area pieces* to appear in FUN98.

[2] Croft, H.T., Falconer K.J. and Guy. R.K. (1991), *Unsolved problems in geometry*, Springer-Verlag.

[3] Frederickson, G.N. (1997), *Dissections: Plane and Fancy*, Cambridge University Press.

[4] Overmars, M.H. and Welzl E. (1985), *The complexity of cutting paper*, Proc. 1st ACM Symposium on Computational Geometry, Baltimore, Maryland, USA, pp. 316-321.

[5] Parshin, A.N. and Shafarevich, I.R. (1995), *Number Theory I*, Encyclopaedia of Mathematical Science, Vol 49. Springer-Verlag, Berlin.