Near-Optimal Partitioning of Rectangles and Prisms

Prosenjit Bose* Jurek Czyzowicz† Evangelos Kranakis* Danny Krizanc*
Dominic Lessard*

1 Introduction

This paper focuses on the following problems:

Problem 1 *Given an axis parallel rectangle, how do you cut it in k equal area pieces such that the total length of the cuts is minimum? What are the properties of an optimal cut?*

Problem 2 *Given an axis parallel prism, how do you cut it in k equal volume pieces such that the total surface area of the cuts is minimum? What are the properties of an optimal cut?*

The problems depend on how the cuts are made. Although cuts may take a general form, in this paper we restrict our attention to straight line cuts and planar cuts. We assume that each cut is complete in that it divides a rectangle or a prism into two pieces (such a cut is often referred to in the literature as a *glass cut* or a *guillotine cut*).

1.1 Previous Work

Several variants of this problem have appeared in the literature. Overmars and Welzl [OW85] studied the problem of cutting a polygon drawn on a piece of paper in the cheapest possible way. Croft, Falconer and Guy [CFG91] studied problems related to tiling and dissection of circles and squares. Bose et al. [BCK+98, BCL98] studied the problem of cutting squares and circles into equal area pieces.

Kong et al. [KMW87] and [KMR88] addressed a variant of Problem 1 in the context of parallel computing, where they were concerned with partitioning a rectangle equitably among a set of processors. However, their objective was to minimize the maximum perimeter of the rectangles in their decomposition.

2 Orthogonal Cuts are Sometimes Optimal

Suppose we want to cut a square into k equal-area pieces using glass cuts. In this section, we show that if k is a perfect square, then the optimal solution consists of orthogonal cuts.

Lemma 1 *The sum of the perimeters of a unit area triangle and a unit area circle is more than twice the perimeter of a unit square.*

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

†Research supported in part by NSERC. Département d'Informatique, Université du Québec à Hull, Pavillon Lucien-Brault, 101 St-Jean-de-Bosco, case postale 1250, succursale B, Hull, Québec, J8X 3X7. E-mail: jurek@uqah.uquebec.ca

Proof: A square of fixed area a has a perimeter of $4\sqrt{a}$. The perimeter of a circle containing also area a is given by $2\sqrt{a\pi}$, since its area equals to πr^2 and its circumference equals to $2\pi r$. We can assume that T is an equilateral triangle in order to minimize its perimeter which is $2 \cdot 3^{\frac{3}{4}}\sqrt{a}$. Since $a = \frac{bh}{2}$ and $h^2 + \frac{b^2}{4} = b^2$, where b is the length on one side of T. Therefore the sum of the perimeters of T and C is

$$2 \cdot 3^{\frac{3}{4}}\sqrt{a} + 2\sqrt{a\pi} \sim 8.1039\sqrt{a}$$

which is greater than $8\sqrt{a}$. ■

Theorem 1 *If $k = p^2$ where p is an integer then the length of the cuts required to cut a square into k equal-area pieces using straight line glass cuts is minimized when each of the pieces is a square.*

Proof: Suppose that the optimal solution does not cut k squares. Let S be the sum of the number of vertices in each piece of the optimal solution. Since a glass cut can introduce at most 4 vertices, the number of vertices in the optimal solution is at most $4k$.

If the optimal solution only consists of quadrilaterals, then we obtain a contradiction since the square is the minimum perimeter quadrilateral enclosing a fixed area.

If this is not the case, then a counting argument shows that for every piece in the solution with more than 4 vertices there must exist at least one triangle. Otherwise, the upper bound on the total number of vertices is violated. Using the previous lemma, we have to match each piece with at least 5 vertices with one triangle, this is possible since we have enough triangles. Any pair has total perimeter greater than twice the perimeter of a square. Therefore we derive another contradiction (because our solution solution is worse than the one with squares), thereby proving the theorem. ■

This leads to a similar result for rectangles.

Corollary 1 *Let R be a $1 \times b$ rectangle, $b \geq 1$, if $k = r \cdot p$ and $\frac{r}{p} = b$ then the optimal solution of cutting R in k equal area pieces is given by straight line orthogonal cuts, say $r \times p$ squares.*

3 Near-Optimal Partitioning of Rectangles

If we restrict our attention to *orthogonal* straight-line cuts, then in [BCK+98], it was shown how to decompose a unit square (in $O(1)$ time) into k equal area rectangles such that the sum of the lengths of the cuts is at most $1 + \frac{1}{2(\sqrt{k}-1)}$ times the optimal solution. In this section, we generalize this decomposition to rectangles.

Theorem 2 *There exists an approximation algorithm solving problem 1 for $1 \times b$ rectangles such that the length of the cuts is no more than $1 + \frac{b+1}{2\sqrt{kb}-b-1}$ where $b \geq 1$. This algorithm runs in $O(1)$ time. $\Omega(k)$ time is required to report the cuts.*

Proof: Without loss of generality, we assume that the longest side of the rectangle R is vertical. First, the algorithm computes the ratio $\frac{k}{b}$. Then it computes the number of columns c. Let r be the ratio and $m = \lfloor\sqrt{r}\rfloor$. If $r \geq m(m+1)$ then $c = m+1$. Otherwise, $c = m$. Once this step is done, it remains to compute how many rectangles (p) will appear in each of the c columns. Three solutions are computed in $O(1)$ time and we keep the best one.

The first solution has $c-1$ complete vertical cuts in it. It can be described in the following way: If the number of pieces to cut is equal to jc where j is an integer, then the solution will contain only one type of rectangle, say j rectangles in each column. Otherwise, it will contain $k \bmod c$ columns with $\lfloor \frac{k}{c} \rfloor + 1$ rectangles and $c - k \bmod c$ columns with $\lfloor \frac{k}{c} \rfloor$ rectangles. Figure 1 shows an example with $c-1$ complete vertical cuts. This solution does not always find the optimal, thus, we compute two other solutions in which the first cut is horizontal. The construction of such solutions is similar to the current one. The difference is that there is no complete vertical cut (the first cut being horizontal). Figure 2 shows two solutions where the first cut is horizontal. The number of pieces below and above that cut is also shown.

Using the fact that a square is the minimum perimeter rectangle enclosing a fixed area, we have that $C_{opt} \geq 2\sqrt{kb} - b - 1$. Then each total length of the cuts is divided by C_{opt}. The total lengths of the cuts and their upper bounds for the three different solutions appear in the following table. The parameters of the total lengths come from the figures 1 and 2.

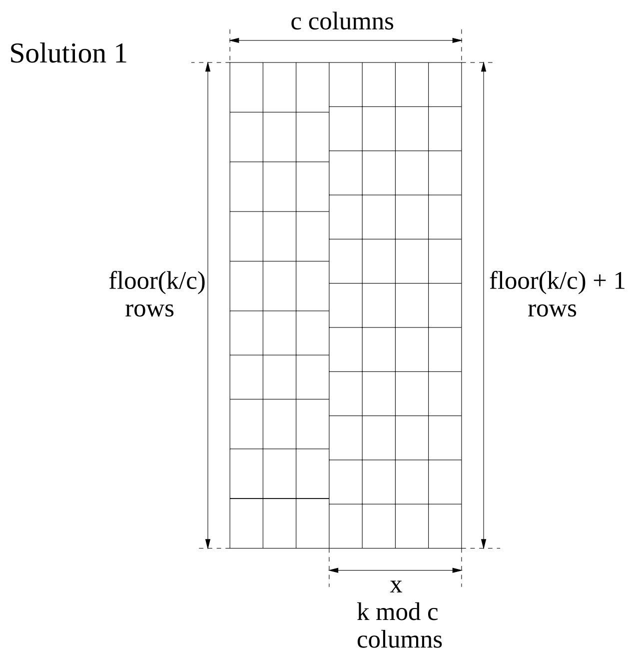

Figure 1: Cutting a rectangle with k rectangles, using c columns ($c-1$ complete vertical cuts).

The upper bounds are obtained in the following way. Each total length of the solutions is divided into two cases: 1) $c = m = \lfloor \sqrt{\frac{k}{b}} \rfloor$ and 2) $c = m + 1 = \lfloor \sqrt{\frac{k}{b}} \rfloor + 1$. In the worst case, $k \bmod c = c - 1$. For the first solution (the one with complete vertical cuts only) $x = \frac{\lfloor \frac{k}{c} + 1 \rfloor (k \bmod c)}{k} \leq 1$.

Solution	Total Length	Upper Bound
1	$(c-1)b + x - 1 + \lfloor \frac{k}{c} \rfloor$	$1 + \frac{b+1}{2\sqrt{kb}-b-1}$
2	$(c-1)b + y + \frac{k2}{c} - 1 + k \bmod c$	$1 + \frac{b+1+2\sqrt{\frac{b}{k}}}{2\sqrt{kb}-b-1}$
3	$(c-1)b - z + \frac{k2}{c} - 1 + (c - k \bmod c)$	$1 + \frac{b+2}{2\sqrt{kb}-b-1}$

Even if the upper bounds proven for solutions 2 and 3 are worse than the first solution, there are cases where one of them is the optimal solution[1]. Note that if $k < b$ then the total length of the cuts given by

[1]The complete proof of the upper bounds can be found at www.scs.carleton.ca/~dlessard/papers/.

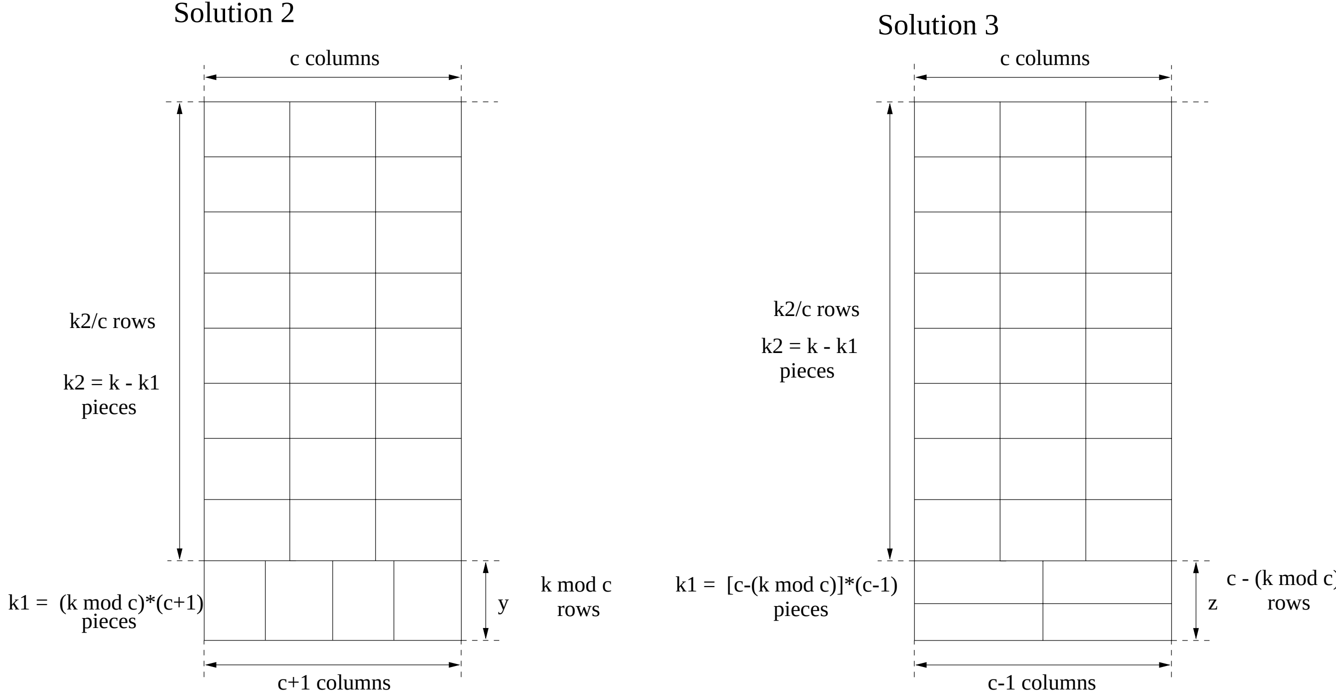

Figure 2: Computation of 2 solutions with the first cut as horizontal.

the algorithm is $k-1$. Therefore, there exists an approximation algorithm solving problem 1 for $1 \times b$ rectangles and the length of the cuts is no more than $1 + \frac{b+1}{2\sqrt{kb}-b-1}$ times the optimal solution where $b \geq 1$. ■

Conjecture 1 *The previous algorithm gives optimal solutions for cutting rectangles into equal area pieces.*

Conjecture 2 *The previous algorithm when computing only the first proposed solution gives optimal results for rectangles with dimensions $1 \times b$, where $b \geq 1$ is an integer and $k \geq 2$.*

4 Near-Optimal Partitioning of Prisms

In this section we study problem 2. We first show lower bounds on the total surface area of the cuts. We then provide an exponential time algorithm for finding optimal solutions. Finally, we show a constant time, near-optimal approximation algorithm. We restrict our attention to orthogonal cuts.

Theorem 3 *The total area of the cuts, which is a solution to problem 2, is at least equal to $3\sqrt[3]{k}-3$ for a unit cube.*

Proof: Let P_0 be a cube with volume 1. By cutting P_0 into k equal volume pieces, we get P_1, $P_2, ..., P_k$. The total surface area of these rectangular cuts is given by A. We also know that the area of the boundary of P_0 is 6.

Since each cut is a rectangle touching two pieces P_i and P_j (P_i and P_j are adjacent and they shared a boundary) where $i \neq j$ and $0 \leq i, j \geq k$, we have

$$A + 6 = \frac{1}{2} \sum_{j=0}^{k} area(P_j)$$

This implies that

$$A = \frac{1}{2} \sum_{j=0}^{k} area(P_j) - 6 = \frac{1}{2} \sum_{j=1}^{k} area(P_j) - 3$$

$$\geq \frac{1}{2} \sum_{j=1}^{k} \frac{6}{k^{2/3}} - 3 = \frac{3k}{k^{2/3}} - 3 = 3\sqrt[3]{k} - 3$$

Because each smaller cube has volume $\frac{1}{k}$ and side of length $\frac{1}{\sqrt[3]{k}}$ and each of its face has surface area of $\frac{1}{k^{2/3}}$. We used the fact that a cube is the prism enclosing a fixed volume with the smallest boundary area. Therefore, this bound may be lower than the optimal solution for a given instance. ■

This leads to a more general result:

Corollary 2 *The total area of the cuts which is a solution to problem 2, is at least equal to* $3\sqrt[3]{k}(abc)^{2/3} - (ab + ac + bc)$ *where* a, b *and* c *are the dimensions of the prism.*

We present here an algorithm solving problem 2. It visits all possible solutions and returns the optimal one.

Optimize3D(P, k)
Inputs: A prism P ($a \times b \times c$) and a positive integer k.
Outputs: Set of surface cuts (or $k-1$ rectangles). The k pieces having equal volume (volume(P)$/k$).

BEGIN
    $S \leftarrow \phi$. $L \leftarrow \infty$.
    FOR $i = 1$ to $\lfloor \frac{k}{2} \rfloor$
        $\{P_1, P_2\} \leftarrow$ CutPrism(XY, i, k).
        $S_1 \leftarrow$ Optimize3D(P_1, i).
        $S_2 \leftarrow$ Optimize3D(P_2, $k - i$).
        $l \leftarrow$ total area of the cuts from $S_1 \cup S_2$ + area of current cut.
        IF $l < L$ THEN
            $S \leftarrow S_1 \cup S_2 \cup$ current cut.
            $L \leftarrow l$.
    Repeat the loop for YZ and XZ cuts.
    Return S.
END.

The function *CutPrism* cuts P into two smaller prisms with a surface cut specified by the first parameter. The second and third parameters are the number of pieces to cut in P_1 and the total number of pieces respectively. The cut is done with respect to the proportion of the number of pieces to cut in P_1 and P_2.

Theorem 4 *Let* k *be the number of pieces to cut. Algorithm Optimize3D runs in* $O(6^k)$ *time.*

Proof: The number of elementary operations is given by

$$T(k) = 3 \sum_{i=1}^{\lfloor k/2 \rfloor} (T(i) + T(k-i) + c_1)$$

where c_1 is a small constant. From this we deduce the following:

$$\begin{aligned} T(k) - T(k-1) &= 3 \sum_{i=1}^{\lfloor \frac{k}{2} \rfloor} (T(i) + T(k-i) + c_1) \\ &- 3 \sum_{i=1}^{\lfloor \frac{k-1}{2} \rfloor} (T(i) + T(k-1-i) + c_1) \\ &\le 3T(\lfloor k/2 \rfloor) + 3T(k-1) + 3c_1 \end{aligned}$$

Therefore,

$$T(k) \le 4T(k-1) + 3T(\lfloor k/2 \rfloor) + 3c_1 \le 6T(k-1)$$

■

The *for* loop in the previous algorithm is executed three times. But if the input is a cube then applying this loop only once is sufficient. Similarly, if two sides of the prism have same dimensions, that loop can be executed two times instead of three. This is a slight improvement, since the following result states an exponential lower bound.

Theorem 5 *Let k be the number of pieces to cut. Algorithm Optimize3D runs in $\Omega(2^k)$ time.*

Proof: Since the best case (number of pieces to cut) is a cube, the total number of operations is given by

$$T(k) \ge \sum_{i=1}^{\lfloor k/2 \rfloor} (T(i) + T(k-i) + c_2) \text{ where } c_2 \text{ is a small constant.}$$

We deduce the following (whether k is even or odd):

$$\begin{aligned} T(k) - T(k-1) &= \sum_{i=1}^{\lfloor \frac{k}{2} \rfloor} (T(i) + T(k-i) + c_2) - \sum_{i=1}^{\lfloor \frac{k-1}{2} \rfloor} (T(i) + T(k-1-i) + c_2) \\ &\ge T(\lfloor k/2 \rfloor) + T(k-1) + c_2 \end{aligned}$$

In fact, the number of pieces to cut is not a cube at each level of recursion. Therefore,

$$T(k) \ge 2T(k-1).$$

Thus, *Optimize3D* runs in $\Omega(2^k)$ time. ■

The exponential algorithm does not keep track of all computed solutions. Hence, subproblems may be computed many times. Suppose now that while the algorithm is running, we put in a table all different optimal solutions to subproblems. Before computing a solution to an instance of the problem, the algorithm could look into the table in order to verify if the solution is already known. But, this is a small improvement since the number of different subproblems is still exponential, as stated in the following theorem.

Theorem 6 *Let k be the number of pieces. There exist instances where the algorithm Optimize3D runs in $\Omega\left(2^{k^{1-\epsilon}}\right)$ time for any $\epsilon > 0$.*

Proof: We will show that the algorithm needs to compute many different instances, each of which is described by the number of pieces and the dimensions of the prism. The proof is similar to the one presented in [BCL98].

Let P be an axis-parallel prism with volume of 1 and dimensions $a \times b \times c$. Let m be the number of prime numbers smaller than or equal to k, determined by the prime numbers theorem ($m = d\lfloor\frac{k}{\log k}\rfloor$, for some constant d). Let $p_1 > p_2 > \cdots > P_m$ represent those primes. There are $\theta(2^{m-1})$ subsets containing $3q$ elements, q is an integer. For each of those we will show how to build a unique problem instance, yielding to an exponential number of different instances.

Let S be a subset with $3q$ prime numbers: $p_{s_1} > p_{s_2} > \cdots p_{s_q}$. We alternatively apply XY, YZ and XZ cuts, until we get a prism of volume $\frac{1}{k}$. This prism will be a unique problem instance and the prime numbers determine where the cut occurs. The first cut C is parallel to the XY plane and p_{s_1} equal volume pieces will be cut above C. Let P_1 be the part of P above C with dimensions $\left(\frac{p_{s_1}a}{k} \times b \times c\right)$. The next cut C_2 is parallel to the YZ plane and splits P_1 into two parts such that the right prism will be cut in p_{s_2} pieces. That prism has dimensions $\left(\frac{p_{s_1}a}{k} \times \frac{p_{s_2}b}{p_{s_1}} \times c\right)$. We continue splitting the prism in this way until we finally get a prism with dimensions $\left(\frac{ap_{s_1}p_{s_4}\cdots p_{s_{(3i-2)}}}{kp_{s_3}p_{s_6}\cdots p_{s_{3i}}} \times \frac{bp_{s_2}p_{s_5}\cdots p_{s_{(3i-1)}}}{p_{s_1}p_{s_4}\cdots p_{s_{(3i-2)}}} \times \frac{cp_{s_3}p_{s_6}\cdots p_{s_{3i}}}{p_{s_2}p_{s_5}\cdots p_{s_{(3i-1)}}}\right)$. As you can see, the volume of the prism is $1/k$.

Each instance is unique since the dimensions of the final prism is determined by the prime numbers. The number of subsets is $\theta(2^{m-1})$ which implies the result since

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

∎

The exponential time algorithm has been implemented in order to study the results.

Conjecture 3 *The optimal solution for problem 2 contains at most 3 types of prisms when the input is a cube.*

An example of an optimal decomposition is given in Figure 3, where a cube has been cut into 7 pieces of equal volume by algorithm *Optimize3D*. But this conjecture is false when the input is not a cube.

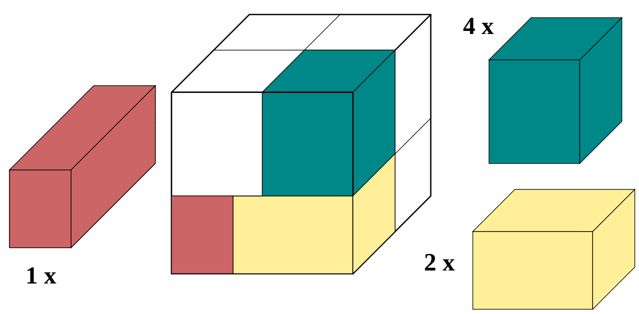

Figure 3: A cube cut into 7 pieces of equal volume. The solution contains 3 types of prisms.

We have found an instance of the problem 2 where the optimal solution contains 4 different smaller prisms. There is no optimal solution with at most three types of prisms for it. Figure 4 shows that instance: a $1 \times 1 \times \frac{1}{2}$ prism cut into 12 pieces optimally. The total surface area of the cuts is $\frac{419}{168}$.

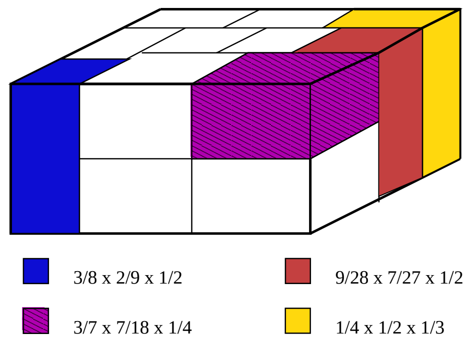

Figure 4: An example where the optimal solution contains 4 different prisms: a $1 \times 1 \times \frac{1}{2}$ prism cut into 12 equal volume pieces. The total surface area is $\frac{419}{168} \approx 2.494$.

By observing the optimal solutions produced by the exponential algorithm, we were able to find a simple decomposition of a cube into k equal volume prisms such that the total surface area of the cuts is near optimal.

Theorem 7 *There exists an approximation algorithm solving problem 2 for a cube such that its solution is no more than* $1 + \Theta\left(\frac{1}{\sqrt[3]{k}}\right)$ *times the optimal solution, for* $k \geq 8$.

Proof: Without loss of generality, assume that the cube has volume 1. The number of pieces k may be written as $a^3 + z$, where $0 \leq z \leq a^3 + 3a^2 + 2a$. Let the ranges $[a^3,\ a^3 + a^2]$, $[a^3 + a^2,\ a^3 + 2a^2 + a]$, and $[a^3 + 2a^2 + a,\ (a+1)^3]$ be R_1, R_2 and R_3 respectively. First, we show how to partition a cube where $k \in R_1$ is a multiple of a. In that case,

$$k = a^2(a - y) + ay(a + 1) \text{ where } 0 \leq y \leq a.$$

The solution contains 2 types of pieces. There are $a^2(a - y)$ pieces below the first cut and $ay(a + 1)$ pieces above it. An example is shown in figure 5 where a cube has been cut into 33 pieces. Below the first cut there are 9 pieces and above it 24 pieces, according to our partitioning scheme.

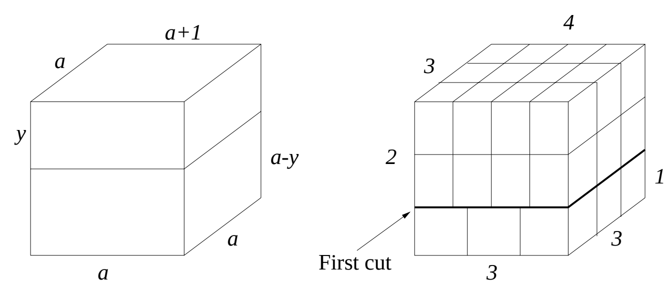

Figure 5: Cutting a prism in k pieces, k is a multiple of a and lies in the first range. On the right, a cube has been cut into 33 equal volume pieces. The solution contains 2 types of pieces.

The following table shows how to cut a cube into k pieces, where k is a multiple of a if $k \in R_1$ and a multiple of $a+1$ otherwise. The partitioning produces only 2 types of pieces. From left to right, the number of pieces increases from a^3 to $(a+1)^3$, covering R_1, R_2 and R_3.

$a \times a$	a	$a-1$	$a-2$	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0
$a \times (a+1)$	0	1	2	$\cdots$	a	$a-1$	$a-2$	$\cdots$	0	a	$a-1$	$\cdots$	0
$(a+1) \times (a+1)$	0	0	0	$\cdots$	0	1	2	$\cdots$	a	1	2	$\cdots$	$a+1$

Let us bound the quality of the solution, versus the theoretical lower bound from theorem 3. Recall that the cube has volume 1. We will show the upper bound when $k \in R_1$ but the proof is similar when this is not the case. Let $C_{opt} = 3\sqrt[3]{k} - 3$ from theorem 3. The total surface area of the cuts given by our partitioning scheme is $C = 3a - 2y - 3 + \frac{ay(a+1)}{k}$. Its upper is the following

$$C \;\leq\; 3\lfloor \sqrt[3]{k} \rfloor - 3 + \frac{\left(\lfloor \sqrt[3]{k} \rfloor\right)^2 \left(\lfloor \sqrt[3]{k} \rfloor + 1\right)}{k} \;\leq\; 3\sqrt[3]{k} - 2 + \frac{1}{\sqrt[3]{k}}$$

Therefore

$$\frac{C}{C_{opt}} \;\leq\; \frac{3\sqrt[3]{k} - 1}{3\sqrt[3]{k} - 3} \;=\; \frac{3\sqrt[3]{k} - 3 + 2}{3\sqrt[3]{k} - 3} = 1 + \frac{2}{3\sqrt[3]{k} - 3}.$$

It still remains to show how to cut a cube in k pieces where k is not a multiple of a or $a+1$ (depending in which range k lies). Let $k = a^2(a-y) + ay(a+1) + b = a^3 + ay + b$. We will describe our partitioning scheme only for the first range. The way to partition a cube is similar when k lies inside R_2 or R_3. We split the method into 3 cases:

Case 1: $k \in (a^3, a^3 + a)$

We first cut the prism such that $a^3 - a^2$ pieces lie below the first cut. (The first cut being parallel to the XY-plane.) Below the first cut there will be only one type of prism while two types will be partitioned in the remainder of the cube. Let P_1 and P_2 be the prisms below and above the first cut respectively. We put $a \cdot a \cdot (a-1)$ pieces in P_1 ($a-1$ layers of $a \cdot a$ pieces). The last layer of the partitioning contains $x = k - a^3 + a^2$ pieces. Since the top face P_2 is a square, we use the $O(1)$ time algorithm from [BCK+98] to cut that square into x pieces, where $a^2 < x < a^2 + a$. Therefore, the solution for P_2 will have one layer of the best partitioning of the square. (We say *best partitioning* because it is still not known if the algorithm from [BCK+98] gives an optimal solution.) The total surface area of the cuts is

$$C = 1 + (a-2) + \frac{2(a^3 - a^2)(a-1)}{k} + L\left(1 - \frac{a^3 - a^2}{k}\right)$$

where L is the total length of the cuts of the partitioning of a square into x equal area pieces of the algorithm[2] from [BCK+98]. Their algorithm won't have more than $a^2 + a - 1$ pieces to cut. Hence

$$L \leq 2a - 2 + \frac{(a-1)(a+1)}{a^2 + a - 1}$$

Then the upper bound of C is

$$\begin{aligned} C \;&\leq\; a - 1 + 2\frac{a^4}{k} - 4\frac{a^3}{k} + 2\frac{a^2}{k} + \left(2a - 2 + \frac{(a-1)(a+1)}{a^2 + a - 1}\right)\left(1 - \frac{a^3}{k} + \frac{a^2}{k}\right) \\ &\leq\; 3a - 3 + \frac{a^2 - 1}{a^2 + a - 1} \;\leq\; 3\sqrt[3]{k} - 2 \end{aligned}$$

[2] See case 1 of theorem 3.2 on page 3 of [BCK+98].

Therefore

$$\frac{C}{C_{opt}} \leq 1 + \frac{1}{3\sqrt[3]{k} - 3}.$$

Case 2: $k \in (a^3 + a^2 - a, a^3 + a^2)$

The first cut partition the prism in such a way that a^3 pieces lie in P_1 (the prism below that cut). And P_2 (prism above the first cut) will be cut into $k - a^3$ pieces. As shown in figure 6, P_1 has only one type of pieces, i.e. it is splitted into a layers of $a \cdot a$ pieces. It remains only one layer of $k - a^3$ pieces to cut in P_2 using the algorithm from [BCK+98] as explained in case 1. The total surface area of the cuts is given by

$$C = 1 + (a-1) - 1 + 2\left(\frac{a^3}{k}\right)(a-1) + L\left(1 - \frac{a^3}{k}\right).$$

The upper bound for L is the following[3]

$$2(a-1) - 1 + \frac{a(r-a+1)}{a^2-1}$$

Thus,

$$\begin{aligned}
C &\leq a + 2\frac{a^3}{k}(a-1) + \left(2a - 3 + \frac{a(r-a+1)}{a^2-1}\right)\left(1 - \frac{a^3}{k}\right) \\
&\leq 3a - 3 + \frac{ar}{a^2-1} - \frac{a^4 r}{k(a^2-1)} + \frac{a^5 r}{k(a^2-1)} - \frac{a^4}{k(a^2-1)} \\
&\leq 3\sqrt[3]{k} - 3 + \frac{2k^{2/3}}{a^2-1}
\end{aligned}$$

Then for $k \geq 8$ we have

$$\frac{C}{C_{opt}} \leq 1 + \frac{2}{3\sqrt[3]{k} - 3}.$$

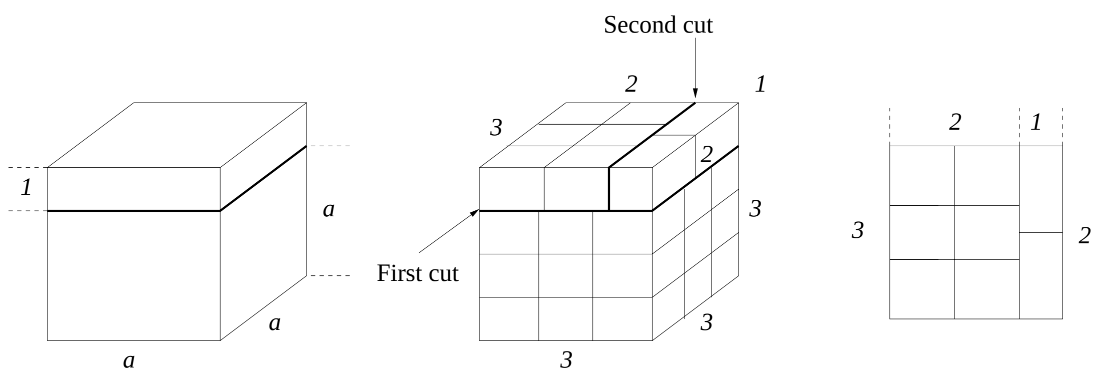

Figure 6: Cutting a prism in k pieces, $k \in R_1$ (case 2) and not a multiple of a. The square on the right has been cut into 8 equal area pieces.

[3] See case 2 of theorem 2.3 on page 4 from [BCK+98].

Case 3: $k \in (a^3 + a, a^3 + a^2 - a)$

Let P_1 and P_2 be the prisms below and above the first cut. P_1 is cut into $a - 1$ layers of $a \cdot a$ pieces. Let $x = k - a^2(a-1)$. P_2 is cut into one layer of x pieces if x is odd and in 2 layers of $x/2$ pieces otherwise. Again, we use the $O(1)$ time algorithm from Bose et al. for the partitioning of the top face of P_2 into x equal area pieces. It remains to analyze the quality of the solution given by this strategy. We will only consider the case where k is odd. The proof is similar when k is even. The total area of the cuts is given by

$$C = 1 + (a-2) + 2\left(\frac{a^2(a-1)}{k}\right)(a-1) + L\left(1 - \frac{a^2(a-1)}{k}\right).$$

The upper bound of L is the following (observe that the number of pieces being cut is smaller than or equal to $2a^2$)

$$L \leq 2(a+1) - 1 + \frac{(r-a-1)(a+2)}{2a^2}$$

We assumed the second case of the theorem 3.2 from [BCK+98]. But the proof is similar for the first case. Therefore

$$\begin{aligned}
C &\leq a - 1 + 2\left(\frac{a^3}{k} - \frac{a^2}{k}\right)(a-1) + \left(2a + 1 + \frac{(r-a-1)(a+2)}{2a^2}\right)\left(1 - \frac{a^3}{k} + \frac{a^2}{k}\right) \\
&\leq 3a - \frac{7a^3}{2k} + \frac{7a^2}{2k} + \frac{2}{a} + \frac{1}{a^2} - \frac{3}{k} + \frac{a}{k} \\
&\leq 3\sqrt[3]{k} + \frac{7}{2} + \frac{2}{\lfloor\sqrt[3]{k}\rfloor} + \frac{1}{\lfloor\sqrt[3]{k}\rfloor^2} + \frac{1}{k^{2/3}} - \frac{3}{k}
\end{aligned}$$

Since $r \leq 2(a+1)$, $a = \lfloor\sqrt[3]{k}\rfloor$ and $k \geq 8$.
Therefore

$$\frac{C}{C_{opt}} \leq 1 + \frac{5}{3\sqrt[3]{k} - 3}.$$

Which completes the proof. ■

References

[BCL98] Bose, P.K., Czyzowicz, J., and Lessard, D. (1998), *Cutting Rectangles into Equal Area Pieces*, Proceedings of the Tenth Canadian Conference on Computational Geometry, pp. 94–95.

[BCK+98] Bose, P.K., Czyzowicz, J., Kranakis, E., Krizanc, D., and Maheshwari, A. (1998), *Cutting Circles and Squares in Equal Area Pieces*, FUN'98.

[CFG91] Croft, H.T., Falconer, K.J., and Guy, R.K. (1991), *Unsolved Problems in Geometry*, Springer-Verlag.

[KMR88] Kong, T.Y., Mount, D.M., and Roscoe, A.W. (1988), *The Decomposition of a Rectangle Into Rectangles of Minimal Perimeter*, SIAM Journal on Computing, Vol. 17, n. 6, pp. 1215–1231.

[KMW87] Kong, T.Y., Mount, D.M., and Werman, M. (1987), *The Decomposition of a Square Into Rectangles of Minimal Perimeter*, Discrete Applied Mathematics, Vol. 16, pp. 239–243.

[OW85] Overmars, M.H. and Welzl, E. (1985), *The Complexity of Cutting Paper*, Proceedings of the 1st ACM Symposium on Computational Geometry, Baltimore, Maryland, USA, pp. 316–321.