Optimisation in Diamond Cutting

Ambili Viswambharan
Department of Engineering Science
University of Auckland
New Zealand
ambiliv@hotmail.com

Abstract

Diamonds are cut from rough polyhedral kimberlite stones. One of the most difficult tasks of the diamond cutter is to know where to place the diamond within the stone in order to minimise waste. The problem, in operations research terms, is to find the location, orientation, and magnification of the diamond shape within a given rough stone so that the volume of the diamond shape is maximised. Both linear and non-linear formulations of this problem are presented and solved and some numerical examples are also provided.

1 Introduction

The current method of deciding how to transform a cut stone into a jewel-quality diamond is unable to ensure maximum use of the rough stone, and doing so may lead to significant economic benefits for the diamond industry. Thus the aim of this project was to work towards optimising and computerising the diamond cutting decision. The question being addressed was: How should a diamond be cut from a given rough stone so that the volume of the diamond is maximised?

Currently, an expert diamond cutter makes this decision after a careful visual examination of the rough stone. Although this method is likely to produce *very good* solutions, this project was aimed at ensuring optimal diamonds are cut from every rough stone.

Maximising the volume of cut diamonds and thus minimising waste, is an important objective in the diamond industry. This is mainly due to the particular feature of the diamond industry that allows the price of cut diamonds to increase exponentially with weight. A one-carat stone can fetch around NZ$10,000, while a two-carat stone of the same shape, colour, clarity and cut can be expected to sell for NZ$40,000. Also, given that over three million tonnes of kimberlite is mined each year for the jewellery industry, small savings in each stone can add up to significant economic benefits for any diamond cutting business.

The structure of diamond also lends itself particularly well to optimisation; not only are almost all of the standard diamond cuts convex, but rough stones of kimberlite are also composed entirely of planar surfaces that are easy to model mathematically.

1.1 The Ideal Diamond Shape

In the diamond industry the ideal shape for a diamond is one that allows the stone to refract the light passing into it, and reflect it back into the eye of the viewer. The resulting multi-coloured sparkle is the mark of a truly exceptional diamond and is what a diamond cutter will aim to achieve from a cut stone. The exact proportions required to achieve this property in a round cut diamond have been researched and presented by Tolkowsky[1], and the resulting shape is called the "Tolkowsky Cut" and what has been used in this project and will hereafter be referred to as the "diamond shape".

2 Problem Formulation

The diamond cutting problem could be described as that of fitting a diamond of given shape, inside a rough stone, so that the volume of the diamond was maximised. So we can start formulating our model by defining some variables:

The diamond shape can be denoted 'D',
the rough stone can be denoted 'S'.

Examining the problem further, it can be shown that there are only three possible transformations that need to be considered, namely: rotation, magnification, and translation.

So more variables can be defined:

Denote the rotation matrix 'R',
the magnification factor 'M',
and the optimal location 'Z'.

A tentative model can then be formulated as follows:

Maximise M
s/t R*M*D(Z) lies entirely within S
(as maximising magnification/dilation factor is equivalent to maximising volume)

If rotation was ignored, and the only transformations of interest were translation and magnification, the diamond-cutting problem can be solved using Linear Programming.

2.1 The Linear Model [LP]

The following discussion and formulation is summarised from *Horst & Tuy* [2]:

The diamond-cutting problem can be described as a special application of the design-centring problem, which is formulated as follows:

$$\max_{\substack{M \in \Re_+ \\ Z \in S}} \{M : Z + MD \subset S\}$$

$$\text{for } D \in \Re^n \ \ D \text{ compact, convex, } 0 \in \text{int}(S)$$
$$\text{for } S \in \Re^n \ \ S \text{ compact, non-empty}$$

Assuming that S is convex, we can re-define S and view it as an intersection of the halfspaces whose associated hyperplanes describe the faces of S. It is then possible to define another variable $r_S(Z)$ such that:

$$r_{sj}(Z) = \begin{Bmatrix} \max\{M : Z + MD \subset S\} & \text{if } Z \in S \\ 0 & \text{if } Z \notin S \end{Bmatrix}$$

Which can be interpreted as the maximum achievable magnification of D (located at a point Z) before it is constrained by halfspace S_j.

Clearly, if $S = \bigcap_{j \in J} S_j$ J ∈ N then $r_s(Z) = \inf\{r_{s_j}(Z) : j \in J\}$.. ①

I.e: The maximum magnification of K can only be as big as the smallest magnification of all the given S_j otherwise D would not be in the intersection of the S_j's. What follows is an outline of a Lemma that is essential to the understanding of the final Linear Programme:

Lemma: if H is a closed half-space, then $r_h(z)$ is given by $r_H(z) = \max\left\{0, \dfrac{\alpha - cz}{\bar{v}}\right\}$ where $\bar{v} = \max\{cz : z \in D\} > 0$

Proof:
We can describe H as follows: $H = \{y : cy \leq \alpha\}$ $c \in \Re^n$, $c \neq 0$, $\alpha \in \Re$, where C is the normal of H, and α is the right hand side. It is then possible to observe that the problem given by: $\max\{cx : x \in D\}$, has a non-zero solution $\{\bar{x} : \bar{v} = c\bar{x} > 0\}$; since $0 \in \text{int}\, D$ and $c \neq 0$.

Then, for each $z \in H$, a new variable $\rho(z)$ can be defined where:

$$\rho(z) = \frac{\alpha - cx}{\bar{v}}$$

$$(\rho(z) > 0 \text{ and } z + \rho(z)D \subset H \text{ if } \rho(z) = 0).$$

If p(z) > 0 and x ∈ D then:

$$c(z + \rho(z)x) = cz + \rho(z)cx \leq cz + \rho(z)c\bar{x} = \alpha,$$

so $$z + \rho(z)D \subset H.$$

But $$c(z + r\bar{x}) > c(z + \rho(z)\bar{x}) = \alpha \quad \text{whenever } r > \rho(z),$$

so $$r_H(z) = \rho(z).$$

i.e. $$r_H(z) = \max\left\{0, \frac{\alpha - cz}{\bar{v}}\right\} \quad \forall z \in \Re^n$$ .... ②

End of Proof.

So from ① and ② we obtain that if $\bar{v}_i = \max\{c^i z : z \in D\} > 0$ for $i = 1,2,....s$, then the maximum magnification with fixed orientation is given by:

$$\boxed{\begin{array}{ll} \max\ M & \\ s/t & \alpha^i - c^i z \geq M\bar{v}_i \quad \text{for } i = 1,2,....s \end{array}}$$

Simply put, this meant that the maximum magnification could be found by considering each hyperplane s_i from the convex set of the rough stone S and finding the point in D that will first collide with s_i and [we call this $\bar{v}_i$]. Then maximising the magnification, with the constraint that each point found above should satisfy its corresponding hyperplane. So for a rough stone S with *s* number of sides, one needs to solve *s* + 1 linear programs, in order to find the maximum magnification of any shape D within S.

2.2 The Non-Linear Model [NLP]

The basic formulation given previously was this:

Maximise M
s/t R*M*D(Z) lies entirely within S

where:
D is the diamond shape
S is the rough stone shape
R represents the rotation
M is the magnification of D within S
Z is the point that D is located at within S

Let us look at the representation of rotation. The best way to deal with rotations in this problem is to use a rotation matrix, which in this case would be a 3x3 matrix (given that the problem is three dimensional). A rotation not only preserves angles but also preserves magnification. So to make sure that all angles were preserved in the transformation, it is sufficient to add a constraint to ensure the rotation matrix would be orthogonal[1]. So our model is further refined as follows:

Maximise M
s/t R*M*D(Z) lies entirely within S
$R^T R = I$.

Obviously the constraint "R*M*D(Z) lies entirely within S" needs to be expressed in a form that is easier to code. If we use the principles gained from the LP formulation process, we can re-write the constraint in the following way. Let the Rough Stone S be expressed as a convex combination of the set of s hyperplanes that constitute its faces:

$$
\begin{gathered}
S = \\
a_1x + b_1y + c_1z \le d_1 \\
a_2x + b_2y + c_2z \le d_2 \\
\ldots \\
a_sx + b_sy + c_sz \le d_s
\end{gathered}
$$

[1] Note that the definition of an orthogonal matrix Q, is that $Q^TQ = I$, and that to "preserve angles" is to preserve inner products.

Or if expressed in matrix form: $Ax \le b$.

As long as each extreme point of the diamond shape D satisfies all of these equations, the solution will be feasible [as D is convex]. So given that D is expressed by the set of all its d extreme points:

$$D = \begin{bmatrix} x_1, y_1, z_1 \\ x_2, y_2, z_2 \\ \ldots \\ x_d, y_d, z_d \end{bmatrix},$$

then the solution Z, M, R will be feasible if $A(Z+MRD) \le b$. The final form of the NLP model is this:

$$\text{Maximise } M$$
$$\text{s/t } \; e_i^T A(Z+MRD_j) \le b_i$$
$$R^T R = I$$
$$\text{For } i = 1\ldots s,\; j = 1\ldots d$$

2.3 Quaternion Algebra

Looking ahead, it is necessary to describe and define rotations if different staring solutions need to be found for the NLP. The best way to visually describe a rotation is using quaternions. To do this, an axis of rotation, and an angle of rotation about that axis are needed. A quaternion is a 4-tuple that can be constructed given a unit axis of rotation, and angle:

$$\tilde{n} = [n_1, n_2, n_3]$$
$$\varnothing$$

From these we can construct the corresponding quaternion:

$$q = \left[\mathrm{Cos}\left(\frac{\phi}{2}\right),\; n_1 \mathrm{Sin}\left(\frac{\phi}{2}\right),\; n_2 \mathrm{Sin}\left(\frac{\phi}{2}\right),\; n_3 \mathrm{Sin}\left(\frac{\phi}{2}\right) \right]$$

The quaternion description of a rotation and the matrix representation are clearly equivalent. Since the matrix description is simpler to deal with computationally, we construct the rotation matrix from the quaternion q, once the visual description is completed. From this quaternion, we can construct a 3x 3 rotation matrix such that:

$$\text{For } \; q = [w, x, y, z]$$

$$R = \begin{bmatrix} 1-2(y^2+z^2) & 2xy-2zw & 2xz+2yw \\ 2xy+2wz & 1-2(x^2+z^2) & 2yz-2xw \\ 2xz+2yw & 2yz+2xw & 1-2(x^2+y^2) \end{bmatrix}$$

R is constructed by considering where the quaternion multiplication maps the co-ordinate vectors to; i.e. if:

$$qxq^T = \begin{cases} [0,\ a,\ b,\ c] & \text{if } x = [0,1,0,0] \\ [0,\ d,\ e,\ f] & \text{if } x = [0,0,1,0] \\ [0,\ g,\ h,\ i] & \text{if } x = [0,0,0,1] \end{cases}, \text{ then } R = \begin{bmatrix} a, & b, & c \\ d, & e, & f \\ g, & h, & i \end{bmatrix}.$$

3 Code Implementation

The first use of programming came into this project when it became necessary to find the equations of the hyperplanes making up D and S. MATLAB was the language chosen for this task as it is inherently mathematical, so was the natural choice. Subsequent modelling tasks were also coded in MATLAB, following on from the work previously done.

The choice of using AMPL for the actual optimisation process was made because it is an optimisation orientated package. It gives the user access to a range of solvers [linear and non-linear], and the syntax is aimed at making model formulation and entry as easy, and intuitive as possible.

3.1 Generating Equations of Hyperplanes

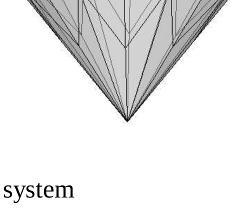
As mentioned previously, the formulation of the linear programming model required that we know the equations of the planes that constitute the faces of D and S. Both D and S are described by their extreme points, and so code needed to be written that was able to find the planes of interest, given the set of extreme points.

An algorithm was developed for this task that finds the relevant planes by creating a plane for each possible combination of the supplied extreme points[2], and then testing these for validity. To be valid, a plane has to be facet defining; so all the extreme points must lie either on, or to one side of the plane.

3.2 Displaying the Diamond Shapes

Obviously, an important part of the solution to the diamond-cutting problem is to be able to represent it visually. Diamond cutters, like most, are not mathematically orientated people, and so a solution in the form of one coordinate point, a magnification factor and a rotation matrix is not sufficient to ensure complete understanding of the solution. It is for this reason that a visual representation of the diamond shape and subsequent solution was required.

The 'patch' function in MATLAB was found to be particularly effective as it creates a three-dimensional 'patch' given a set of defining co-ordinates, as illustrated by Figure 1. To display the diamond shape, it was sufficient to create a patch for each equation by finding the extreme points that satisfied it.



[2] This was done by solving a simple linear system

Figure 1. Diamond Shape generated using MATLAB

3.3 Solving the Linear Model

Once the formulation of the problem was obtained, implementing it in AMPL was a simple process. Two model files were required for the LP model. The first calculated the colliding extreme point given a hyperplane of S and the set of extreme points that described D. The second took the results from the solution of the first LP and found an optimal magnification and location of D of fixed orientation within S.

A script file was required since the first LP needed to be solved for each hyperplane in S, and the results of each LP needed to be stored and passed into the second LP to find the optimal magnification and location.

3.4 Solving the Non-Linear Model

Again, once the formulation was obtained, the code implementation in AMPL was a simple matter. The model file for this problem was developed by Lectureres M.J.O'Sullivan and C.Walker[3], and the corresponding script file was derived from the one used for the linear model.

3.5 Integrating MATLAB and AMPL

At this stage, the project was quite fragmented. There existed MATLAB code to completely describe a polygon given its extreme points, AMPL code to solve the diamond-cutting problem using an LP without rotation, or an NLP with rotation, and MATLAB code to display polygons, and thus solutions to the AMPL optimisation.

In order to make the system easier to use, it was necessary to write MATLAB and AMPL files to make information accessible to the other program.

This was done as follows: Once MATLAB obtained the equations of planes forming the convex hull of the given extreme points, code was written to allow an AMPL data file to be created and stored where the AMPL program had access to it. Then the AMPL LP script was re-written to make use of this file and another AMPL script was written to allow the NLP model to use information from this MATLAB-generated data file. Both AMPL scripts mentioned above were then further modified to be able to write the results of the optimisation to a MATLAB file, which could then be used to display AMPL results in MATLAB.

4 Comparing the Linear and Non-Linear Models

Having coded the problem, the next natural step was to obtain some results and compare the performance of LP and NLP formulations. Obviously the NLP was not guaranteed to give globally optimal solutions, and so it was necessary to apply both formulations to different orientations of D within S. This then raised the question of which orientations were the best starting points; i.e. how should the staring solutions be chosen to maximise likelihood of gaining a globally optimal solution for both the LP and NLP?

[3] Department of Engineering Science, University of Auckland, New Zealand

In view of the likely structure of the rough stone S and D, it was concluded that the most likely placement of D within S was at a corner point - lying along the longest diagonal from that corner point; or lying on a diagonal joining the midpoints of two opposite edges.

To explore all the solutions using this heuristic, it was necessary to work out the longest diagonal for each corner point of S and to work out the rotation that will transform D so that it lies in the orientation given by this diagonal. Then this would be repeated for all the midpoints of all the edges of S. Note that although two extreme points (or mid-points of edges) will share a diagonal, it is still necessary to consider both, since the rough stone may not be symmetrical about those two points.

To work out the longest diagonal radiating from a given extreme point in S, it is sufficient to find the extreme point that is furthest away from the one in question, and find the line joining those two. Having obtained this information, it only remained to work out the relevant axes of rotation and angles for each corner point.

The axis of rotation can be obtained by taking the cross product of two vectors: The first can be derived from the diagonal, and the second is the vertical z-axis. This is because the original orientation of D is vertical. The angle of rotation could then be obtained from the following formula:

$$x \bullet y = |x||y|\mathrm{Cos}(\phi),$$

where ø is the angle between the vectors x and y.

The axis of rotation and angle were then used to construct a quaternion and the corresponding rotation matrix that could be easily applied to the Diamond Shape before solution.

5 Finding the Global Optimum

As in every optimisation problem, the best possible outcome is finding the global optimum. Although the LP is perfectly able to find the optimal solution for the diamond-cutting problem for a given orientation, it is unable to find the global optimum. The NLP can find local optima quickly by looking at rotation, but again, is never guaranteed to find the global optimum, as it is non-convex. However, the LP combined with parametric analysis seems promising, so a brief outline for a possible method of finding the global optimum using this approach is given below.

The linear model finds the optimal location and magnification of the diamond within the rough stone for a fixed orientation.

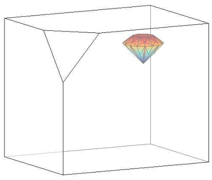
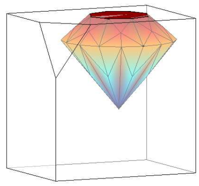
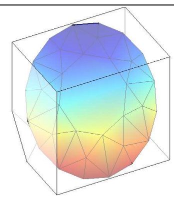
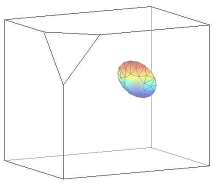
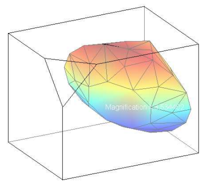
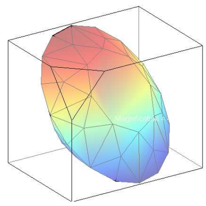
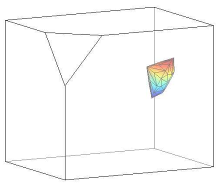
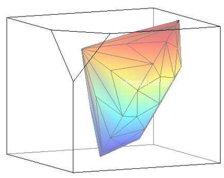
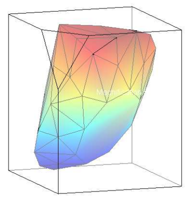
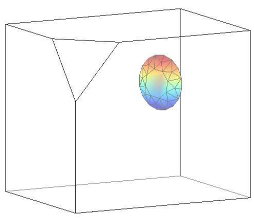
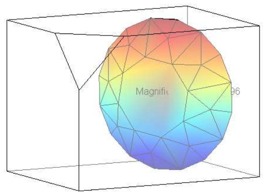
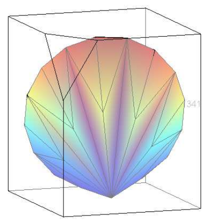
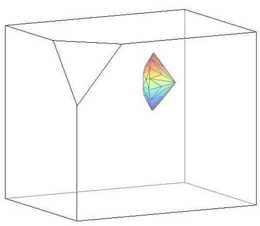
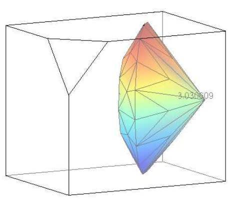
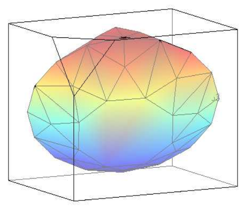
Now observe that a change in the orientation of the diamond within the rough stone is equivalent to fixing the orientation of the diamond but changing (in a different manner) the orientation of the rough stone enclosing it, i.e. it is only the relative position of the diamond and the rough stone that is of any consequence.

The linear model allows us to consider the effects of the planes of the rough stone on the magnification of the diamond separately. Therefore using the above observation, we can consider the rotations of the planes of the rough stone by parametric analysis (on the objective costs in the LP's) and obtain the optimal vertices $(\bar{v}_i)^k$ where the new subscript k indexes a range of rotations over which the optimal vertex remains fixed.

Once we tabulate ranges of rotation pertaining to an n-tuple of fixed vertices $(\bar{v}_1, ..., \bar{v}_n)$, then we can rank the magnifications in decreasing order. The next task is to investigate if a feasible rotation of the stone will match a row of the magnification.

The outline given above does not yet incorporate the geometry of the diamond and the stone. We are currently working on combining that information to make the process substantially easier.

6 Analysis of Results

BFS	LP Solution	NLP Solution
Vertical Orientation 	 Magnification = 3.78	 Magnification = 4.15
Orientated along diagonal between corner points 	 Magnification = 3.53	 Magnification = 4.15
	 Magnification = 3.53	 Magnification = 4.15
Orientated along diagonal between edge mid-points 	 Magnification = 4.00	 Magnification = 4.15
	 Magnification = 4.05	 Magnification = 4.15

These results show that the NLP gives solutions with a better objective function when compared to the LP. It must be noted, however, that since we are dealing with the entirely physical problem of diamond cutting, it may be possible that the LP is more useful in practice. This could be because the diamond cutter using this software needs to fix the orientation of D for some reason, e.g. to avoid an inclusion or to make use of a feature of the rough stone, etc.

The LP is also useful for two other reasons:

- It will demonstrate what the cutter will end up with if the stone is cut optimally with the cutters choice of orientation; as opposed to the orientation that the NLP will suggests [which is inevitably a better solution].
- Combined with parametric analysis, the linear method is promising for finding the global optimal solution.

7 Conclusions

The aim of this project was to attempt to optimise the diamond cutting decision, and this has been achieved to a great extent by implementing linear and no-linear optimisation models using MATLAB and AMPL.

The project has developed over several stages, starting with researching the cutting process, identifying areas for improvement, formulating optimisation models, and finally implementing software solutions.

Both models used in this project are guaranteed to find local optima, and an approach has also been developed to find the global optimum of a diamond-cutting problem. Of the two models outlined, results show that the non-linear formulation is more efficient than the linear –formulation, having a faster running time as well as yielding higher magnifications. However, the linear formulation would also be useful in instances where the cutter needs the orientation of the diamond to be fixed within the rough stone.

Comprehensive software tools have been developed in MATLAB and AMPL to support the solution process. These programs have the capability to compute all information necessary to solve and display the results of the diamond-cutting problem, given the extreme points of the rough stone and diamond shape. The MATLAB program will also allow a user to specify a rotation for the diamond before solving, as well present the option of running a heuristic to explore various starting solutions.

Acknowledgements

I would like to thank my project supervisor Proffessor Golbon Zakeri for her tireless patience and guidance during my work on this paper. I would also like to thank Professors Mike O'Sullivan Jnr and Cameron Walker for their input to the rotation formulation; and would finally like to mention my colleague Mr. Cameron Mellor, for his help with MATLAB.

References

[1] Burton,E (1978). *Diamonds.* Fakenham Press Limited. Fakenham, Norfolk, Great Britain

[2] Horst, R. & Tuy, H. (1990). *Global Optimization:Deterministic Approaches.* p[558-561]. Springer-Verlag: Berlin, Heidelberg, Germany.