Evolutionary Intelligence (2021) 14:1065–1073
https://doi.org/10.1007/s12065-020-00542-1

SPECIAL ISSUE

A novel algorithm for efficient utilization of gemstone using genetic algorithm

Hiten M. Sadani[1] · **Nikhil Kumar Singh**[2] · **Manish Khare**[3]

Received: 9 September 2020 / Revised: 22 November 2020 / Accepted: 28 November 2020 / Published online: 14 January 2021

Abstract
In this paper, a novel method is used for fitting a polished and faceted object which is also called as a gem or diamond in a given rough gemstone using genetic algorithm. The goal of proposed Genetic Algorithm based Multiple Object Fitting algorithm is to maximize the utilization of the volume of rough gemstone by fitting the largest number of polished gemstones inside rough gemstone. One of the most difficult tasks is to fit the number of polished gemstones and positioning of each and every polished gemstone within the rough gemstone in order to minimize the waste. This is an optimization problem that is used to find the position, orientation, and scaling parameters of all the polished gemstones within a given rough gemstone such that the sum of volumes of all polished gemstones is maximized. The major novelty of proposed work is to fit more than one object in a given rough stone. The simulation results demonstrate the efficiency of our proposed algorithm.

Keywords Gemstone cutting · Genetic algorithm · Calibration body · Optimization

1 Introduction

Gemstones are being mined and traded since long and are a complex large Industry, which is very poorly researched [1]. There are various issues in the gemstone sector like mining, processing and trading. In this paper we have focused on efficient utilization of gemstone, which is related to minimization of wastage while gemstone cutting. Gemstone cutting starts with a planning process in industry. For small stones, usually the planning consists of checking the stone for obvious flaws and selecting a particular shape of polished gemstone which results into the best fitting of the shape in a rough gemstone which fulfill the objective of maximizing the utilization or minimizing the wastage of precious stone. In brief, such approach tries to maximize the yield of polished gemstone from a rough gemstone in order to minimize loss of weight which results into better valuation. The Brilliant round shape is the most famous shape of polished gemstone in diamond industry.

It is always tried to find out the best positions of cuts in polished gemstone by carefully analyzing the rough gemstone, aiming to maximize the total summation volume of all shapes embedded in a rough gemstone.

As far as our knowledge, this is the first attempt to propose an algorithm for fitting more than one objects in a given rough gemstone using genetic algorithm. There is one patent by Ceulemans [2] which claims to fit more than one gemstone in rough gemstone using the number of defined regions for each object fitting. The result published in the patent [2] is based on different object of same type only. Whereas the approach proposed in this paper may have more than one object of different shapes. Apart from this, few approaches have been reported in the literature to fit only one polished gemstone into rough which are discussed in section II. It is very important to note that using our proposed approach, it is possible to fit more than one objects in rough gemstone which is significant achievement in comparison of other methods available in the literature. We demonstrate the efficiency of our proposed approach by fitting more than one

✉ Manish Khare
mkharejk@gmail.com

Hiten M. Sadani
sadani1989@gmail.com

Nikhil Kumar Singh
nikhil.singh31@gmail.com

[1] U. V. Patel College of Engineering, Mehsana, Gujarat, India

[2] Government Engineering College, Gandhinagar, Gujarat, India

[3] Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT), Gandhinagar, Gujarat, India

object in rough stone. The simulation result is described for fitting two gemstones which are not even carried out by any of the previous approaches [2, 3]. By fitting more than one object we get more volume utilization of rough stone and also computational time of our approach is less than time taken by Heuristic approach [3]. In Sect. 2, we discuss related work. Our work is inspired from heuristic approach [3]. Our proposed approach is described in Sect. 3. Section 4 contains the Result and Analysis and Sect. 5 concludes the paper.

2 Related work

In the literature, the number of approaches has been proposed which treat the object fitting as an optimization problem. Winterfeld's lapidary cutting problem [4] is formulated as design centering problem (DC) [5] and the solution for DC deals with the optimization of volume, of a polished gemstone (the design), which must be contained in a rough gemstone (the container). DC problem [4] is formulated as a general semi-infinite problem (GSIP). Using Stein's GSIP solver [6], Winterfeld [4] optimizes the GSIP problem in an adaptive refinement procedure (ARP) such that obtain solution of ARP is more feasible for the lapidary cutting problem. At the end of ARP, an optimal solution is found and time taken by this method is in minutes.

Viswambharan [7], proposed both linear and non-linear formulations to solve diamond cutting problem. The problem with linear programming (LP) method is that it ignores the rotation of diamond shape. Only translation and magnification are calculated by LP method. Whereas the non-linear programming method (NLP) considers rotation, translation and magnification of diamond shape in a given rough stone together.

Fiorest et al. [8], introduced Gems 3D optimizer. The Gems 3D optimizer looks virtually into the scanned gemstone for better use of its volume by fitting gem into stone. The optimizer takes two files as input, one containing the 3D virtual representation of the rough gemstone with its points and faces, and another containing the critical points. In second version of algorithm, critical points are selected by genetic algorithm rather than manually. The algorithm generates all possible plans by the combination of critical points, three at a time. For each of these plans, the contour of gem is extracted by finding points which provides a polished gemstone in the plan. This contour is then used for registration waist model stoning, for each of the models available.

da Silva et al. [3], solved the colored gemstone cutting problem using genetic algorithm [9–11] such that the largest faceted cut design fits inside a given rough gemstone. The calibration body concept introduced by da Silva et al. [3] preserves the properties of gemstone. Heuristic approach [3] supports Round and Oval shapes of polished gemstone. Using genetic algorithm, they found the position and orientation of one polished gemstone within a rough gemstone. The objective function verifies every point of rough gemstone whether it is inside calibration body or not. If it is inside then scale factor is calculated to reduce the size of polished gemstone and returns the volume of polished gemstone as fitness value.

3 Proposed approach

Based on literature review it can be concluded that most of the approaches deal with the fitting of only one gem (object) in a given rough stone with an objective of maximum yields or minimum wastage of precious stone using optimization method. The proposed novel algorithm is used for efficient utilization of rough stone by cutting or fitting more than one object in a given rough gemstone. Our approach is inspired from heuristic approach proposed by da Silva et al. [3] which also uses genetic algorithm for object fitting. The major difference between our proposed algorithm and heuristic approach [3] is that our algorithm is able to fit or cut more than one object (gem) from a given rough stone and results into more yields comparatively. Problem formulation of our approach uses genetic algorithm, and is based on heuristic approach [3]. So, here first heuristic approach proposed is discussed before discussing our proposed method.

3.1 Heuristic approach: for polished gemstone fitting

da Silva et al. [3], the author uses virtual calibration body to preserve quality of gemstone. To utilize volume of rough gemstone, it is required to fit polished gemstone with maximum volume but at the same time, it is necessary to preserve properties of polished gemstone. The characteristic of a polished gemstone depends on the amount of the light reflected when entered from its top. By the refraction index of gemstone [12], the critical angle is obtained which is primary restriction in the design process of polished gemstone. This critical angle must be followed by the planar facets of polished gemstone.

Figure 1 shows different polished gemstone which is obtained by using different critical angles. When this critical angle used for facets of polished gemstone and if it is lower or bigger than the recommended angle, there is dispersion of the incident lights which does not return. It reduces the brightness and results in poor quality of polished gemstone [12].

For preserving the property of polished gemstone, the calibration body might be used instead for the optimization of polished gemstone. Calibration body for different shapes

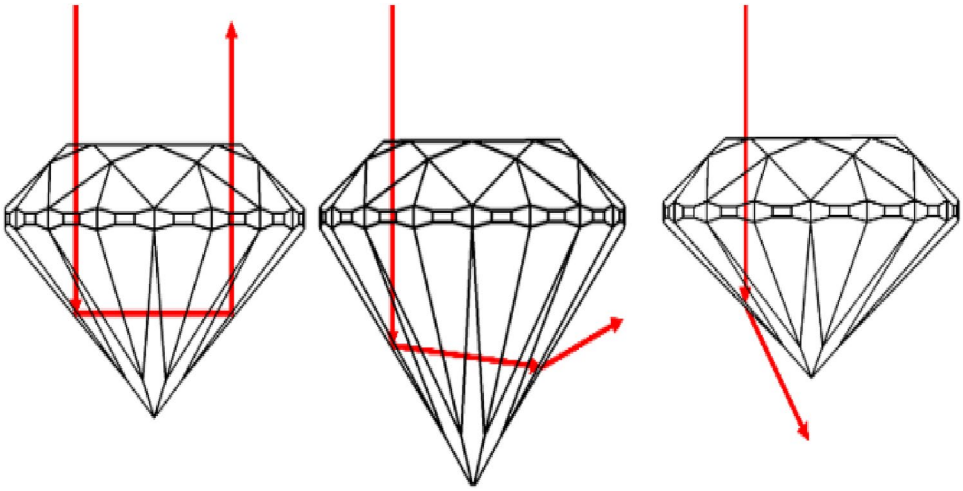

Fig. 1 Optical effects in polished gemstone with different critical angles [12]

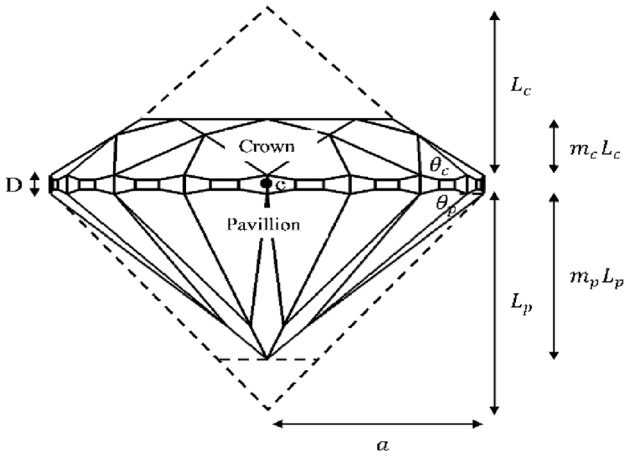

Fig. 2 A representation of calibration body

Table 1 The calibration body parameters [3]

Variable	Particulars
a	Major semi-axis length
b	Minor semi-axis length
e	Eccentricity
D	Height of girdle
L_p	Lower cone height
m_p	$(0 \le m_p \le 1)$: $m_p L_p$ is the lower cone height
L_c	Upper cone height
m_c	$(0 \le m_c \le 1)$: $m_c L_c$ is the upper cone height
θ_{bc}	Angle between the basis semi-major axis and the upper cone
θ_{bp}	Angle between the basis semi-major axis and the lower cone

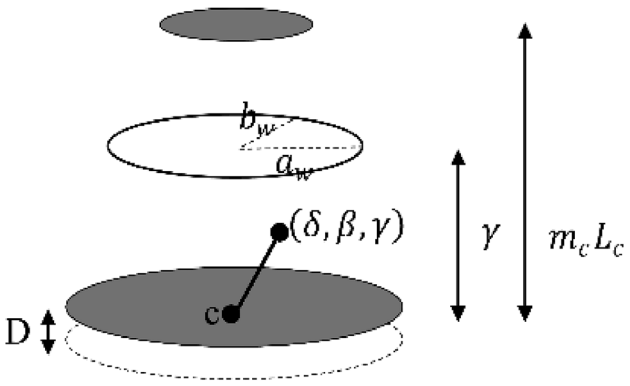

Fig. 3 Coordinates of the calibration body

like round and oval is created such that these shaped polished gemstone can be represented into two, juxtaposed and elliptical cones, with same major axis separated by distance D, as shown in Figs. 2 and 3.

In this strategy, first step is to verify if the calibration body is inside the rough gemstone or not. For this, check all points of the rough gemstone and verify if point is outside the virtual calibration body or not. If all points of rough gemstone are outside the virtual calibration body, then the polished gemstone of that virtual calibration body is inside the rough gemstone. The density of triangular faces of the rough gemstone affects the accuracy of this strategy because, if the virtual calibration body enters the rough gemstone in between the face, this strategy cannot detect it. Therefore,

it is necessary to keep density of point cloud of rough gemstone high. Different parameters of virtual calibration body [3] are shown in Table 1.

3.2 Proposed algorithm

In our approach, it is possible to fit more than one gemstone in a given rough gemstone. The goal of our proposed GAOF algorithm is to maximize the total sum of volumes of various shaped polished gemstone, finding the scaling factor, orientation and position within rough gemstone with minimum execution time.

Due to the optical properties of gemstone, the height and width of polished gemstone are strongly related to each other. So, the height and width of polished gemstone cannot be changed separately. For shapes of polished gemstone, two shaped cuts: Round shape and oval shape are considered and calibration body is defined for them in our algorithm. Our GAOF algorithm is demonstrated by fitting two polished gemstone within rough gemstone to get maximum utilization of volume of rough gemstone. Here, first mathematical equations used in our approach are described and then proposed algorithm is described using these equations.

3.2.1 Mathematical equations

Consider rough gemstone as G and a calibration body of polished gemstone as C_i. Consider centre $\left(x_{0_i}, y_{0_i}, z_{0_i}\right)$ as position of polished gemstone and angle $\left(\theta_i, \psi_i, \sigma_i\right)$ as orientation of polished gemstone with respect to axis (x, y, z) respectively. Consider α_i as largest scaling factor of calibration body of polished gemstone for a given center $\left(x_{0_i}, y_{0_i}, z_{0_i}\right)$ and orientation $\left(\theta_i, \psi_i, \sigma_i\right)$, such that the calibration body of polished gemstone can be contained in rough gemstone as given in Eq. (2).

$$max\alpha_i\left(x_{0_i}, y_{0_i}, z_{0_i}, \theta_i, \psi_i, \sigma_i\right) \tag{1}$$

$$\left\{\left(x_i, y_i, z_i\right) = \left(x_{0_i}, y_{0_i}, z_{0_i}\right) + \alpha_i * \left(C_{\theta_i,\psi_i,\sigma_i}\right)\right\} \subseteq G \tag{2}$$

$$0 \leq \theta_i, \psi_i, \sigma_i \leq 2\pi \tag{3}$$

$$0 < i \leq m \tag{4}$$

Representation of rough gemstone or irregular 3D objects is triangular mesh representation which is framed by points in R^3 and by triangular faces. Rough gemstones for our algorithm are taken from repository [13].

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{5}$$

$$R_y(\psi) = \begin{bmatrix} \cos\psi & 0 & \sin\psi \\ 0 & 1 & 0 \\ -\sin\psi & 0 & \cos\psi \end{bmatrix} \tag{6}$$

$$R_z(\sigma) = \begin{bmatrix} \cos\sigma & -\sin\sigma & 0 \\ \sin\sigma & \cos\sigma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

Rotation of point $p(x, y, z)$ with sequence of axis x, y, z and resultant coordinate (δ, β, γ) will be-

$$(\delta, \beta, \gamma) = R_z(\sigma)(R_y(\psi)(R_x(\theta)(x - c, y - c, z - c))) \tag{8}$$

where c is the centre of Calibration body.

Parameters of calibration body as shown in Fig. 2 are used in following Eqs. (9)–(16) [1]. Equation (9) is used to verify z-axis of 3D point (δ, β, γ) is in γ-plane or not and γ as shown in Fig. 3 can be calculated using Eq. (8). Equations (10), (11), (12), (13) are used to check that 3D point is in $\delta\beta$-plane or not. Equations (14), (16) are used to calculate scale factor in both γ-plane and $\delta\beta$-plane. In our algorithm, the explanation of how these equations are used is detailed in the later part.

$$-m_pL_p - d < \gamma < m_cL_c + d \tag{9}$$

$$\frac{\delta^2}{a_w^2} + \frac{\beta^2}{b_w^2} < 1 \tag{10}$$

where

$$d = D/2, \tag{11}$$

$$a_w = \begin{cases} a\left(1 - \frac{\gamma - d}{L_c}\right), & if\, \gamma - d \in [0, m_cL_c], \\ a, & if\ |\gamma| \leq d, \\ a\left(1 + \frac{\gamma + d}{L_p}\right), & if\ -\gamma - d \in [0, m_pL_p], \end{cases} \tag{12}$$

$$b_w = a_w\sqrt{1 - e^2}, \tag{13}$$

$$s_\gamma = \begin{cases} \frac{\gamma}{m_cL_c + d}, & if\ \gamma > \varepsilon \\ \frac{\gamma}{m_pL_p + d}, & otherwise \end{cases} \tag{14}$$

$$t = \tan^{-1}\left(\frac{\beta}{1 - \delta\sqrt{1 - e^2}}\right) \tag{15}$$

$$s_\delta = \frac{\left(\frac{\delta}{\cos t}\right)}{a_\gamma} \text{ using equation (15)} \tag{16}$$

If unit vector is (x_u, y_u, z_u) and point is (x, y, z) then after shifting of this point with distance *dist*,new point (x', y', z') is calculated as shown below,

$$(x', y', z') = dist * (x_u, y_u, z_u) + (x, y, z) \tag{17}$$

3.2.2 Genetic algorithm based multiple object fitting (GAOF) algorithm

Initially, the rough gemstone is processed using library mixkit [14] such that sufficient density of point cloud is maintained and also reduces the number of points and faces of a given rough gemstone. After completion of rough gemstone processing, genetic algorithm is executed which generates position and orientation of calibration body and these parameters are given to objective function. By using these position and orientation, objective function calculates volume of calibration bodies embedded in rough gemstone and then returns total volume of calibration bodies to genetic algorithm as fitness value.

First, parameters of Genetic Algorithm are initialized as given by heuristic approach [3] like $p_{size} \in \{50, 75, 100\}$, $p_{mut} \in \{0.2, 0.3, 0.4\}, p_{cross} \in \{0.5, 0.75, 0.9\}$ and the number of generation is 1000 or 2000. The reason for considering the parameters of heuristic approach [3] is that they are well tested to give good results in case of single gemstone fitting. Objective Function used is available in GALib Library [15] which is free open source C++ library of genetic algorithm. Steady State Genetic Algorithm (SSGA) [3] is also used in our approach.

In the initial population of p_{size} individuals, genetic algorithm begins and it executes until the criteria for stopping the algorithm is satisfied. Applying mutation and crossover operations to individuals of current generation, it generates a new generation of temporary population having l ($l = p_{size} * r$) new individuals, for replacement percentage r. For crossover and mutation operations, the individuals are selected based on their fittest value. In the next generation, the new individuals are used. To keep population size constant, from the previous population $p_{size} - l$ best individuals are selected for the next generation. At the end of algorithm, from all the population the best individuals are returned as the problem solution.

In our approach, the real-valued genome consists of genes $x_0, y_0, z_0, \theta, \psi, \sigma$ is implemented using class GARealGenome available in library GALib [15]. Objective function evaluates each genome also called individual. The individual fitness is the total volume of the two calibration bodies embedded in the rough gemstone calculated by $v = v_0\alpha_1^3 + v_0\alpha_2^3$. Volume v_0 is the starting volume of both calibration bodies. After calculating $x_0, y_0, z_0, \theta, \psi, \sigma$ for one calibration body, with respect to first calibration body, position and orientation of second calibration body is calculated. In first step, the population for first calibration body is initialized for parameters θ, ψ, σ by selecting random values and set x_0, y_0, z_0 as the center point of the rough gemstone's bounding box and after that $x_0, y_0, z_0, \theta, \psi, \sigma$ of second calibration body is calculated using parameters of first calibration body. If the center of first calibration body is not in the interior part of the rough gemstone, the other center point is selected randomly. If it lies in rough gemstone mesh then it is selected as center point of first calibration body. The Gaussian Mutator [3] is used for mutation in genetic algorithm.

There are two methods of crossover operation: real blending crossover and uniform crossover. In the real blending crossover method, each gene of the child genome obtains the value using formula $\alpha f + (1 - \alpha)m$, where the value of α lying in the interval $[-0.5, 0.5]$ is chosen randomly. Variables f and m represents gene's values of father and mother genome, respectively. One of genes from the parents are passed to the child gene with equal probability in the uniform crossover. For stopping criterion, our algorithm sets a fix value for number of generations.

Objective function returns fitness value to genetic algorithm and there are also another output of objective function is center point of second calibration body where as orientation of second calibration body is reverse orientation of first calibration body.

a. Objective Function:

c_1, c_2—Center of Calibration body C_1 and C_2, respectively
α_1—Scale Factor of Calibration body C_1
α_2—Scale Factor of Calibration body C_2
v_0—Initial volume of Calibration body C_1 and C_2

Input: $G, C_1, C_2, x_0, y_0, z_0, \theta, \psi, \sigma$

Output: v

1. **if** x_0, y_0, z_0 is not inside in G **then**
2. return 0
3. **end if**
4. $c_1 = (x_0, y_0, z_0)$
5. $\alpha_1 \longleftarrow diagonal(G)/C_1.a$
6. $\alpha_2 \longleftarrow diagonal(G)/C_2.a$
7. $\alpha_1 \longleftarrow GetScale(G, C_1, c_1, \theta, \psi, \sigma)$
8. **if** $\alpha_1 > prev(\alpha_1)$**then**
9. dist = 3D distance between upper flat surface of C_1 and surface of G
10. Current_Height_of_C_2 = $m_p L_p + d + m_c L_c$
11. ScaleRatio = dist/Current_Height
12. $C_2.a = C_2.a *$ ScaleRatio
13. $c_2 =$ Shift point c_1 using dist and unit vector of center point of upper flat surface of C_1 and C_2 by equation (17)
14. $\alpha_2 \longleftarrow GetScale(G, C_2, c_2, \theta, \psi, \sigma)$
15. **else**
16. return 0
17. **end if**
18. $prev(\alpha_1) = \alpha_1$
19. $v = v_0{\alpha_1}^3 + v_0{\alpha_2}^3$
20. return v

In objective function, the method of finding largest scale of two calibration body inside rough gemstone to utilize more volume of it is defined. The Objective Function takes one Rough gemstone R, two Calibration body C_1, C_2, point (x_0, y_0, z_0) and angle (θ, ψ, σ) as input. Initially, the objective function verifies, if the center x_0, y_0, z_0 lies inside mesh of rough gemstone G; otherwise it returns value 0 for given input. For point-interior testing, the algorithm proposed in [16] is used. Then, parameters α_1, α_2, c_1 are initialized in line 4–6. First calibration body center c_1 is initialized to (x_0, y_0, z_0). The maximum diagonal distance of the bounding box of rough gemstone G over semi-axis a is used to initialize as an upper bound on scale factors α_1, α_2. Call Function *GetScale()* for getting scale factor α_1 for calibration body C_1 to embedded rough gemstone G in line 7. If this scale factor α_1 is not larger than previous scale factor α_1 for same calibration body C_1, then return 0 because previous volume of calibration body C_1 is more than current volume. If the condition test is satisfied in line 8 then calculate position c_2 of other Calibration body C_2 and after that scale factor α_2 using method *GetScale ()* which is applied for calibration body C_2. Lines 9–13 are used for calculating position of calibration body C_2. Using point-interior testing algorithm [16] and center point of upper flat surface of calibration body C_1, surface point of rough gemstone is calculated then, calculates distance between center point of upper flat surface of C_1 and

intersected surface point of rough gemstone in line 9. Then, calculate height of calibration body C_2 using equation given in line 10. In line 11, distance between upper flat surface of C_1 and surface of G over height of C_2 is used as scale ratio for semi-major axis a of calibration body C_2. Then, in line 9, distance is calculated between upper surface of calibration body C_1 and C_2 and this distance is used to calculate center point of calibration body C_2. In line 13, center point c_2 of C_2 is calculated to shift original point c_2 at calculated distance in line 9 and unit vector between center point of upper flat surface of C_1 and C_2 using Eq. (17). For new position c_2 of Calibration body C_2, largest scale factor α_2 is calculated in line 14 using same method *GetScale()* to calculate scale factor for calibration body C_2. This objective function can evaluate for only two shapes round and oval. The objective function can be modified for other shaped polished gemstone, by changing Eq. (13) to an equation that checks if a point of rough gemstone is inside the other shaped calibration body or not. *GetScale* Function is discussed which returns scale factor of calibration body corresponding to position and orientation.

b. GetScale Function

Table 2 Features of the 10 scanned rough gemstones [13]

Gem	Vol.	No. of vertices	No. of faces
1	1787.38	18,334	36,664
2	1853.73	18,536	37,068
3	1504.81	16,688	33,372
4	2394.03	23,101	46,198
5	2283.48	22,377	44,750
6	1827.47	18,943	37,882
7	1553.91	16,679	33,354
8	1680.09	18,392	36,780
9	1498.00	17,241	34,478
10	1946.99	20,348	40,692

Table 3 The roundshaped calibration body parameters [3]

Variable	Value
θ_{bc}	33°
θ_{bp}	42°
L_c	$a \tan \theta_{bc}$
$m_c L_c$	0.45 mm
L_p	$a \tan \theta_{bp}$
$m_p L_p$	1.32 mm
a	1.5 mm
b	1.5 mm
D	0.15 mm

Input: $G, C, c(x_0, y_0, z_0), \theta, \psi, \sigma$

Output: α

1. **for** each point $p(x, y, z)$ in G **do**
2. $(\delta, \beta, \gamma) =$ Rotate point p using equation (8).
3. **If** γ satisfies equation (9) **then**
4. calculate a_w according to equation (12)
5. calculate b_w according to equation (13)
6. **If** a_w, b_w, δ, β satisfies eq. (10) **then**
7. calculate s_γ according to equation (14)
8. calculate s_δ according to equation (16)
9. $\alpha \leftarrow max(min(s_\gamma, s_\delta), \alpha)$
10. **end if**
11. **end if**
12. **end for**
13. **return** α

Table 4 The test results of heuristic approach [3] for the round shaped polished gemstone

Gem	Max (%)	t (s)
1	37.28	92
2	35.72	90
3	39.95	95
4	38.69	116
5	42.09	125
6	38.70	103
7	39.59	91
8	37.95	102
9	36.29	96
10	36.33	105

Table 5 The test results of proposed approach for the round shaped polished gemstone

Gem	Max (%)	N_C	N_F	t(s)
1	39.42	3668	7332	23
2	40.39	3708	7412	20
3	42.22	3339	6674	22
4	37.49	4621	9238	29
5	46.16	4477	8950	31
6	42.20	3790	7576	26
7	43.20	3337	6670	22
8	35.17	3680	7356	25
9	39.13	3449	6894	23
10	41.24	4071	8138	25

This *GetScale()* Method is used to calculate scale factor of calibration body C to fit calibration body within rough gemstone G for corresponding position and orientation. Line 1takes each point p of G. In line 3, it is checked that point p satisfies the γ-plane test using Eq. (9), if this test is satisfied, then in line 6 it is checked that the point p is inside an ellipse with semi-axis (a_w, b_w) using Eq. (10) or not. If satisfies, then point p is inside the virtual calibration body, and therefore the polished gemstone is not fullyinside the rough gemstone. If point p satisfies conditions in lines 3 and 6, thenin lines 7–9, reduced scale factor α is a calculated using Eqs. (14) and (16). After examining all points of G, the largest possible scaling factor α is obtain such that C is within G. Finally, *GetScale* function gives output as scaling factor α.

4 Experimental results

To evaluate our proposed algorithm, a set of 10 scanned rough gemstones are taken from 3D Gems Repository [13]. Table 2 contains the volume in mm^3, no of vertices and no of faces for each of the 10 rough gemstones [13]

Table 3 represents the values of calibration body parameters used in our proposed algorithm. RealBlendCrossover method [3] and Gaussian mutator [3] is used for Crossover operation and mutation operation respectively in our approach. To measure the performance, volume yield and execution time is taken as used by da Silva et al. [3]. The various parameters of genetic algorithm like population Size (p_{size}), mutation percentage (p_{mut}), the crossover percentage (p_{cross}), and the replacement percentage r are taken as 100, 0.3, 0.75, 0.9 respectively [3].

4.1 Testing environment

All the experiments have been conducted on a PC with processor Intel Core i5, running at 3.20 GHz, 4 GB of main memory and 64 bit Windows 7 operating system. The algorithm is developed in Microsoft visual studio 2008 platform using the library GALib [15], Mixkit Library [14], OpenGL Library [17] and compiled with Microsoft Visual C++ 9.0.

5 Results

Table 4 represents the results of our proposed algorithm. For each of the 10 gemstones Table 4 consist of maximum volume utilization (Max (%)), new number of vertices (N_C), new number of faces (N_F) and execution time (t (s)) in seconds, after pre-processing of rough gemstone.

According Winterfeld [4], volume of the round shaped polished gemstone yields 35%, with average execution time of 608 s, by executing the algorithm min a Linux machine with an Intel Xeon (3 GHZ) with 4 GB RAM. It is important to note that in approach by Winterfeld [4] instances of rough gemstone are not available; therefore its comparison with our proposed algorithm is difficult and is not possible directly.

Rough gemstone instances of experimental results for heuristic approach [3] are available in repository and comparison of volume yield is directly possible. da Silva et al. [3] have used the processor Intel Core i7, running at 2.80 GHz and 12 GB of memory to conduct the experiment. The approach of da Silva et al. [3] is conducted with processor Intel Core i5 running at 3.20 GHz, 4 GB of main memory and 64 bit Windows 7 operating system and results of experiments are represented in Table 4 as below.

Our approach optimizes volume of rough gemstone. The results of our proposed approach as shown in Table 5 represent the maximum volume utilization from all 10 rough gemstone as 46% whereas in heuristic approach [3], it is 42%. In our approach, the average volume utilization of different rough gemstones is 41% whereas it is 36% for heuristic approach [3]. Maximum execution time in heuristic

approach [3] is 125 s for 10 instances of rough gemstone while for our approach maximum execution time is 31 s for the same number of instances. Average execution time for 10 instances of rough gemstone, in our proposed approach and heuristic approach [3] are 25 s and 102 s respectively.

6 Conclusions

The proposed Genetic Algorithm based Multiple Object Fitting (GAOF) algorithm solves the gemstone cutting rapidly for round and oval shapes. This approach fits two polished gemstones in a given rough gemstone to utilize more volume. Other shape of polished gemstone can also be solved by the proposed approach. Experimental result shows that proposed approach is achieving maximum volume of 46% whereas in heuristic approach it is 42%. Hence, it can be concluded that our results are better than the results obtained by heuristic approach. Average execution time of our proposed approach and heuristic approach are consequently 25 s and 102 s which indicates that execution time of proposed approach is much less than heuristic approach. In future work, our aim is to optimize the volume of rough gemstone with multiple polished gemstones fitting in rough gemstone using more shapes.

References

1. Cartier LE (2019) Gemstones and sustainable development: perspectives and trends in mining, processing and trade of precious stones. Extr Ind Soc 6(4):1013–1016
2. Ceulemans T (2007) Device and kit for visualizing a cutting regime of a diamond, and a method for determining a cutting regime. US Patent US20070186918 A1, 16 Aug 2007
3. da Silva V, Ritt M, da Paz Carvalho J, Brusso M, da Silva J, Zanatta A (2012) A real-valued genetic algorithm for gemstone cutting. In: XXXVIII Conferencia Latinoamericana En, Informatica (CLEI), Medellin, 2012
4. Winterfeld A (2008) Application of general semi-infinite programming to lapidary cutting problems. Eur J Oper Res 191(3):838–854
5. Nguyen VH, Strodiot J-J (1992) Computing a global optimal solution to a design centering problem. Math Program 53(1–3):111–123
6. Stein O (2006) A semi-infinite approach to design centering. In: Dempe S, Kalashnikov V (eds) Optimization with multivalued mappings, Springer optimization and its applications. Springer, Berlin, pp 209–228
7. Viswambharan A (1988) Optimisation in diamond cutting. Technical report, University of Auckland, New Zealand
8. Fiorest MB, Brusso MJ (2012) Algoritmos Genéticos Na Otimização Da Lapidação De Gemas Coradas. Revista CIATEC-UPF 3(2):44–55
9. Mitchell M (1999) An introduction to genetic algorithms. MIT Press, Cambridge
10. Lambora A, Gupta K, Chopra K (2019) Genetic algorithm: a literature review. In: 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon). IEEE, pp 380–384
11. Haldurai L, Madhubala T, Rajalakshmi R (2016) A study on genetic algorithm and its applications. Int J Comput Sci Eng 4(10):2347–2693
12. Mol AA, Martins-Filho LS, da Silva JDS, Rocha R (2007) Efficiency parameters estimation in gemstones cut design using artificial neural networks. Comput Mater Sci 38(4):727–736
13. 3D Gemas Repository [Online]. http://usuarios.upf.br/~3dgemas/repositorio/index.php. Accessed 06 Dec 2013
14. Garland M (2004) Mixkit: QSlim simplification software [Online]. http://mgarland.org/software/qslim.html. Accessed 01 Dec 2019
15. Wall M (2019) GAlib: a C++ library of genetic algorithm components [Online]. http://lancet.mit.edu/ga/. Accessed 01 Dec 2019
16. Sunday D (2019) Intersections of rays, segments, planes [Online]. http://geomalgorithms.com/a06-_intersect-2.html. Accessed 06 Dec 2019
17. OpenGL: the industry's foundation for high performance graphics [Online]. http://www.opengl.org/. Accessed 06 Dec 2019