

École polytechnique de Louvain

Geometric Analysis of Historical Violin Outlines: A Study of Approximation by Circular Arcs

Author: **Marie DUTRON**

Supervisors: **François GLINEUR, Anne-Emmanuelle CEULEMANS**

Readers: **Paul FISETTE, Pierre-Antoine ABSIL, Philémon BEGHIN**

Academic year 2025–2026

Master [120] in Mathematical Engineering

Abstract

Violins that are constructed nowadays are copies of existing models. It is believed that luthiers from the 17th and 18th centuries designed the outline of violins using circles. We will here attempt to approximate the outline of historical violins using a set of circles. We are using a corpus of 14 violins from the Low Countries, as well as a synthetic violin made of circles, as reference.

We first analyze the performance of a greedy algorithm that builds circles sequentially around the contour, bounded by a user-defined tolerance. Second, we develop a global optimization method, where a predefined number of circles is optimally selected out of a set of candidate circles, built by least-squares approximation of local neighbourhoods of points of the contour. Three different variations of this second method are explored. We evaluate those methods with error-based metrics, as well as a symmetry score.

The greedy algorithm produced poor results, as it is heavily conditioned on the starting point. The global optimization methods resulted in recurring patterns across the violin corpus. While these results do not constitute proof of historical evidence, they can form a basis for future organological research.

Acknowledgments

I would like to start by thanking my supervisors, Prof. Anne-Emmanuelle Ceulemans and Prof. François Glineur, as well as Philémon Beghin, for their availability, their guidance, and their support throughout the development of this thesis. I always wished to combine my love for history with my love for mathematics. Thank you for making that possible.

I would also like to thank Prof. Harry Mairson for kindly sharing his code and methods, which allowed the extraction of the contour of the Digital Amati.

Finally, I would like to thank my family and friends for their support during the realisation of this thesis. Special thanks to my parents for their encouragement and their suggestions, and especially to my mother, for her critical reviews of my work. Thank you, Odile, for being the voice of reason when the stress got to my head. Finally, thank you to my partner in crime, Yasmine, for your presence and your answers to my never-ending doubts. I could not have gotten through this master's degree without you.

AI Usage

In this master's thesis, Claude was used to improve sentence formulation and figure design, as well as assistance for code debugging. Grammarly was used for grammar correction. All content was reviewed critically.

Contents

1	Context and Motivation	2
1.1	Studied Instruments	2
1.2	Problem Formulation	9
1.3	Evaluation Metrics	9
2	State of the Art	12
2.1	Historical and Organological Perspectives	12
2.2	Geometric and Computational Methods	14
2.2.1	Computational Approaches to Violin Research	14
2.2.2	Circle-Based Outline Approximation	14
3	Analysis of a Greedy Algorithm	16
3.1	Description of algorithm	16
3.1.1	Best Single Arc Approximation	16
3.1.2	G^0 Piecewise Arc Approximation	18
3.2	Results and Limitations	19
3.2.1	Tolerance Analysis	21
3.2.2	Dependence on starting point	22
3.2.3	Separating parts	25
3.3	YangDu Algorithm: Summary	27
4	Development of a Global Approach	28
4.1	First optimization model: SetCover- ϵ	28
4.1.1	Formulation	28
4.1.2	Results and Limitations	29
4.1.3	SetCover- ϵ : Summary	32
4.2	Second optimization model: SetCover-K	33
4.2.1	Formulation	33
4.2.2	Results and Limitations	34
4.2.3	SetCover-K: Satisfying results	37
4.3	Further Adaptations	37
4.3.1	Minimization of the Maximum Error	37
4.3.2	Relaxation: Allowing for skipped points	40
4.4	Set Covering Approach: Summary	42
5	A Study of Symmetry	45
5.1	Development of a symmetry score	45
5.2	Symmetry score study	46
5.2.1	SetCover-K	47

5.2.2	SetCover-KM	48
5.3	Enforcing symmetry	48
5.3.1	Formulation	49
5.3.2	Results	50
5.4	Symmetry Study: Summary	52
6	Results and Discussion	53
6.1	Comparison of methods	53
6.1.1	Digital Amati	53
6.1.2	Historical Violin Corpus	55
6.2	Reduced violins	61
6.3	Selecting the best method	63
6.4	Future directions	63
	Conclusion	64
	Glossary of Mathematical Notations	66
	References	67
	Figures references	69
A	Violin Corpus	70
B	YangDu: Enforcing Symmetry	71
B.1	Enforcing symmetry	71
B.2	Realigning points	72
C	SetCover-ε – All Results	74
C.1	Results for a tolerance of 1mm	74
C.2	Results for a tolerance of 2mm	77
D	SetCover-K – All Results	80
D.1	Results for 16 circles	80
D.2	Results for 18 circles	83
D.3	Number of circles study	86
E	SetCover-KM – All Results	91
E.1	Results for 16 circles	91
E.2	Results for 18 circles	94
E.3	Number of circles study	97
F	Reduced Instruments – All Results	102
F.1	SetCover-K – $K = 18$	102
F.2	SetCover-KM – $K = 18$	104

Chapter 1

Context and Motivation

Violins are musical instruments that benefit from worldwide recognition and popularity. However, their origins are unclear: it is believed that the violin as we know it appeared in northern Italy, in the late 16th century. It was probably first used by itinerant musicians from a lower-class background, which would explain why the origin of the violin lacks documentation. However, very quickly, violins gained popularity and were played and produced throughout Europe [4].

Violins that are produced nowadays are designed as copies of existing models. However, the shape and curved outline of such instruments raise questions. Many books formulated the theory that such a shape was obtained by an arrangement of circles and possibly straight lines [8][17][7]. Such a construction only requires a compass and ruler, tools that were heavily used. However, if such a method existed, this knowledge has been lost. In this thesis, we will attempt to rebuild such a method and to approximate the outline of violins using circular arcs.

1.1 Studied Instruments

Morphology of the violin

The violin body has a distinctive hourglass shape divided into three sections: the upper bout, the lower bout, and the C-bouts, the narrowed waist between them (see Figure 1.1). This waist allows the bow to reach all strings without obstruction. The small pointed projections where these sections meet are called corners. At the top of the instrument, the neck extends upward from the body.

Seen from the side, the body consists of three distinct components: the sound board, the back, and the ribs: thin strips of wood forming the sides that connect the two plates and define the body's depth.

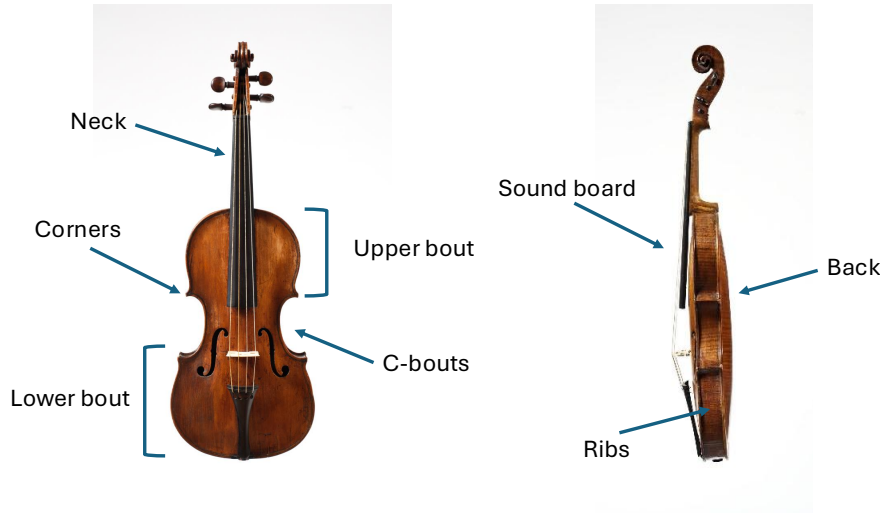


Figure 1.1: Anatomy of the violin

Origin of the studied instruments

We will conduct our study on a collection of 23 instruments from the Musical Instruments Museum (MIM, Brussels). In this thesis, all instruments will be referred to by their inventory number¹. This collection of 23 instruments contains instruments of varying sizes: violins, violas, and tenors. For simplicity, the word "violin" will be used to refer generically to instruments of the violin family. All studied violins were made in the 17th or 18th century in the Low Countries, covering approximately the area of present-day Belgium, Luxembourg, and the Netherlands.

It is important to note the origin of those instruments, as they were built quite differently from the Italian violins, which have been more thoroughly studied. Italian violins were built from the ribs, which were molded to fit the desired shape. Then the back and the sound board were glued on either side of it. On the contrary, violins from the Low Countries were built from the back: it was cut to the desired shape, and the outline was carved to fit the ribs (see Figure 1.2). The ribs were thus slid in those grooves, and the sound board was glued on top of it [5]. According to Karel Moens, this technique originates from the first violins, made by the musicians themselves, as it is easier to achieve, and requires fewer tools [15]. In this thesis, we will thus solely study the back of those instruments, as they were built first, and therefore defined the shape of the violin.

¹A complete list of the instruments is available in Appendix A

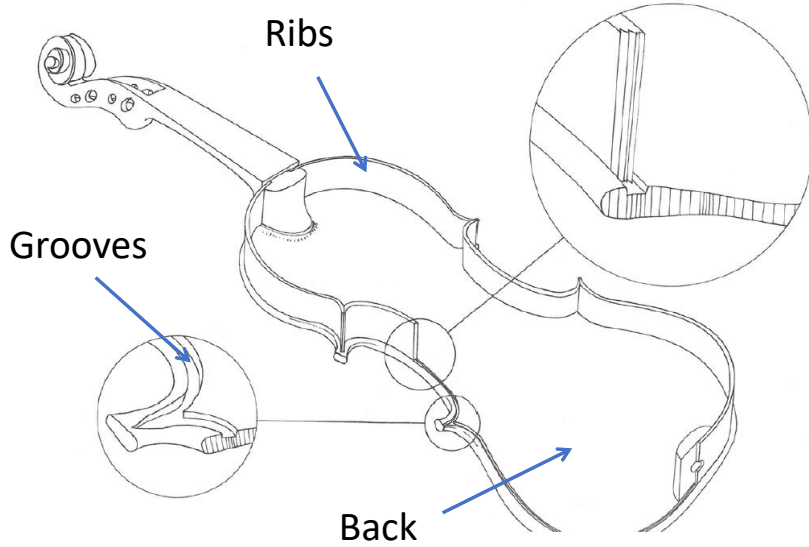
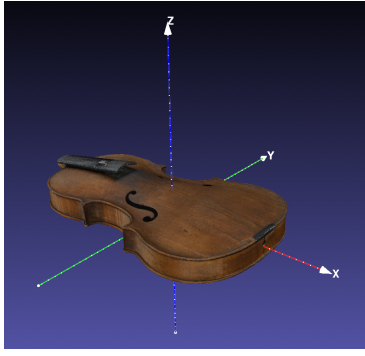


Figure 1.2: Violin made on grooves. The back is built first, and grooves are carved. The ribs are slid into the grooves, and the sound board (not shown) is glued on top of the grooves.

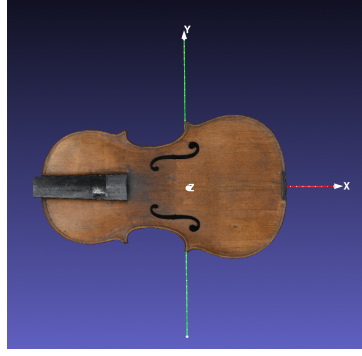
Among the 23 instruments we consider in this study, 9 instruments are believed to have been constructed with different dimensions originally. In the 18th century, the various instruments within the violin family were classified, and sizes were standardized in three categories: the violin, the viola, and the cello. From 1750 onwards, ancient instruments were reshaped to fit the standards. Therefore, the original shape of those instruments has been changed, and it is difficult to know the original morphology of these instruments. These instruments will therefore not be used as reference instruments for this study. Since that data is available, we will try the best-performing methods on those violins to see if the reduction is detectable from the circles. For the sake of clarity, all the reduced violins will be referred to with an * next to their inventory number.

Outline Acquisition

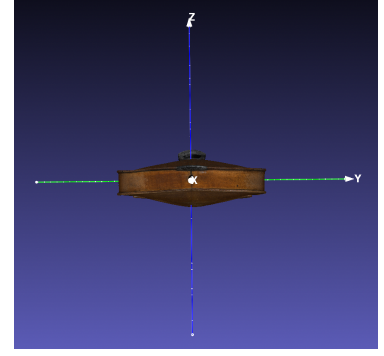
Obtaining a clear outline for these violins was a challenge tackled by Philémon Beghin in his master's thesis and later in his doctoral thesis. The method is explained thoroughly in [2]. The violins were first digitized by photogrammetry: they were photographed from every angle (every 10°) and at different heights. About 160 pictures per violin were given to the photogrammetry software Agisoft Metashape, which extracted the mesh from the pictures. Once the mesh is obtained, the neck is manually removed, and Principal Component Analysis is used to align the approximate plane of symmetry between the sound board and the back with the Oxy plane and the left-right plane with the Oxz plane (see Figure 1.3).



(a) General view of the recentered violin



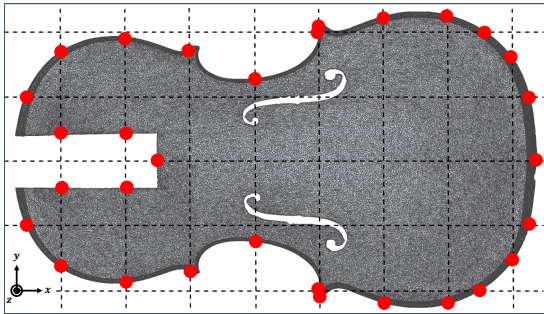
(b) Recentering the left-right plane with the Oxz plane



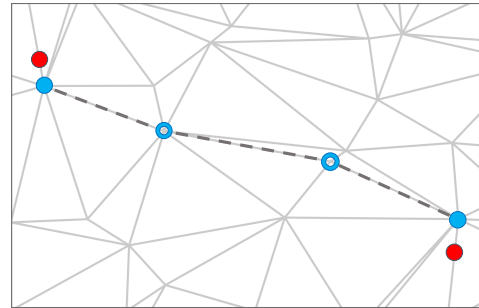
(c) Recentering the sound board-back plane with the Oxy plane

Figure 1.3: Recentering with Principal Component Analysis

Once this mesh is obtained and pre-processed, a rough outline of the sound board and the table is drawn manually. From this, "extreme points" are extracted from vertical cross-sections of the plate at every millimetre, mapped onto the nearest actual mesh vertices, and then connected into a closed contour loop using a Travelling Salesman Problem (TSP) solver combined with a shortest path algorithm (see Figure 1.4). Any gaps between non-adjacent vertices are filled with intermediate points to ensure a fully connected loop. Once the closed loop is established, all vertices and faces lying outside it are removed, retaining only the largest connected component (the "inner mesh"), which corresponds to the plate itself.



(a) Extreme points computed on a wide grid



(b) Mapping of the extreme points (red) onto the nearest vertices (filled blue), shortest path algorithm to add intermediate vertices (empty blue)

Figure 1.4: Contour delineation process

The final outline of the back is then mirrored (according to the y -axis) using a rotation matrix so that the final contour displays the back looking from above (as a "valley" rather than a "mountain"), which is coherent with how the violins were built (see Figure 1.2).

Deformations and distortions

The violins we are working with are old artefacts: all originate from the 17th and 18th century. It is therefore fair to consider that they may have been distorted through

time and conditions. Most violins have been readapted throughout history to the playing trends of that time, and as such, some parts may have been reshaped or replaced. As mentioned above, some violins have also been fully reduced to fit the established size standards.

Besides voluntary choices by various luth makers, such old violins are also damaged by humidity, heat, or accident. For example, violin 2780 shows a dent on the back, which is also visible in the outline used.

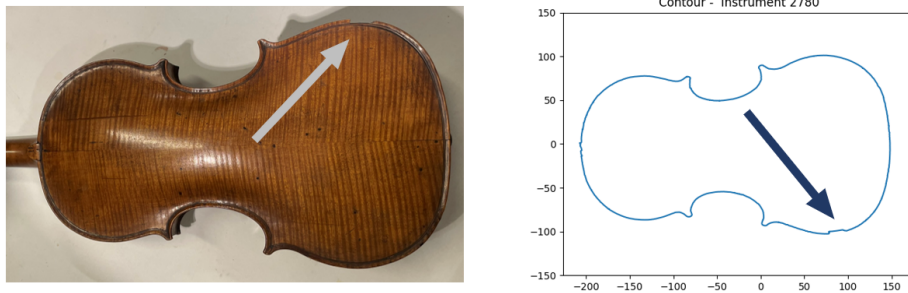


Figure 1.5: Violin 2780 displaying a dent, and its extracted contour. The contour is rotated because we are studying the backs looking from above (as a "valley", rather than a "mountain"), as they were made.

Two parts of the outline present important distortions or noise. The first part is the corners, which are understandably more likely to be damaged due to their outside position and thinness. The second part is the junction between the body and the neck. When defining the extreme points from the mesh, the delimitation is less visible than for the body, resulting in an approximate outline. To prevent these imprecise outlines from affecting the results of the performed algorithms and research methods, we chose to cut them out, as displayed in Figure 1.6.

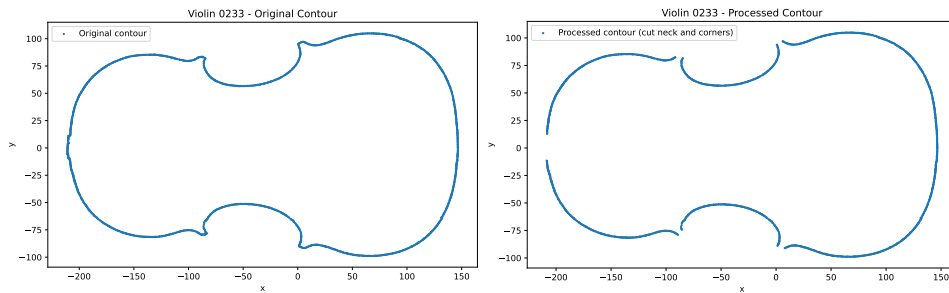


Figure 1.6: Comparison of raw and cut contours - Violin 0233

Recentering

As we consider symmetry a good indication of how good a circle fitting method is, it is important to center the violins on the x-axis, thus between the left and right parts of the violin. To achieve that, we identify at every millimeter one point on the top half ($y > 0$), and one point on the bottom half ($y < 0$), and define the midpoint function:

$$\mu(x) = \frac{y_{\text{top}} + y_{\text{bottom}}}{2}$$

The vertical offset can then be estimated as:

$$k = \text{median}_x(\mu(x))$$

and the translation $y \leftarrow y - k$ can be applied to every contour point (see Figure 1.7 (b)). The median is chosen rather than the mean as it is more robust to outliers (such as the dent seen in Figure 1.5).

As we can see in Figure 1.7 (b), some violins are off rotation after the first step. This manifests as a midpoint line $\mu(x)$ that has a nonzero slope. The geometric intuition is the following: a line at angle θ to the horizontal satisfies $y = x \tan \theta$, so the midpoint of a perfectly symmetric but tilted object obeys

$$\mu(x) \approx x \tan \theta \quad \implies \quad \text{slope}(\mu) \approx \tan \theta$$

We fit a linear model

$$\mu(x) \approx \beta_0 + \beta_1 x$$

by least squares approximation and extract $\hat{\beta}$, the estimated slope. To optimize further and find the best rotation angle, we define the following procedure: first, we rotate the contour by θ around its centroid $(\bar{x}, \bar{y})$ with

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x - \bar{x} \\ y - \bar{y} \end{pmatrix} + \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}$$

Then, we recompute the midpoint function $m_\theta(x)$ from the rotated contour, fit a new line to it, and extract the slope $\hat{\beta}(\theta)$. This gives a function $\theta \mapsto \hat{\beta}(\theta)^2$ that measures how far the rotated contour is from being centered. We then minimize it:

$$\theta^* = \arg \min_{\theta} \hat{\beta}(\theta)^2$$

The rotation by θ^* is the one that makes the midpoint line as flat as possible.

After rotation, k is recomputed and reapplied to correct the small residual vertical offset that rotation introduces.

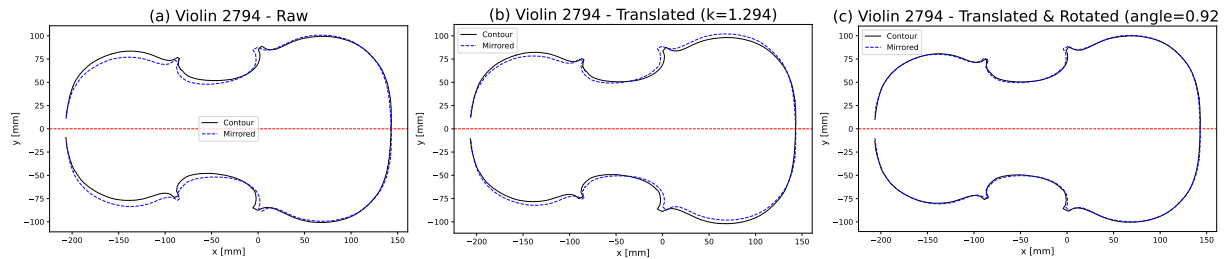


Figure 1.7: Recentering process of violin 2794. (a) displays the original contour of the violin (with cut neck and corners). (b) displays instrument 2794 after translation. (c) displays instrument 2794 after translation and rotation.

A special instrument

François Denis, a French luthier, developed a geometric method in his *Traité de lutherie* to trace violin outlines using 18 circles, a technique derived from classical Italian violins (as explored in greater depth in Section 2.1). He especially recreated an Amati² mould by averaging various instruments built from the same mould. Building on this work, Professor Harry Mairson of Brandeis University designed a specialized programming language to reproduce Denis's theoretical constructions (see Figure 1.8) [13]. We have extracted the outline generated by this model to serve as a reference for validating the various algorithms and methods examined in this thesis. On the display below, the blue line represents the ribs, which were the starting point of the design for Italian violins. The red line next to it represents the soundboard. This violin will be referred to as "Digital Amati" in this thesis.

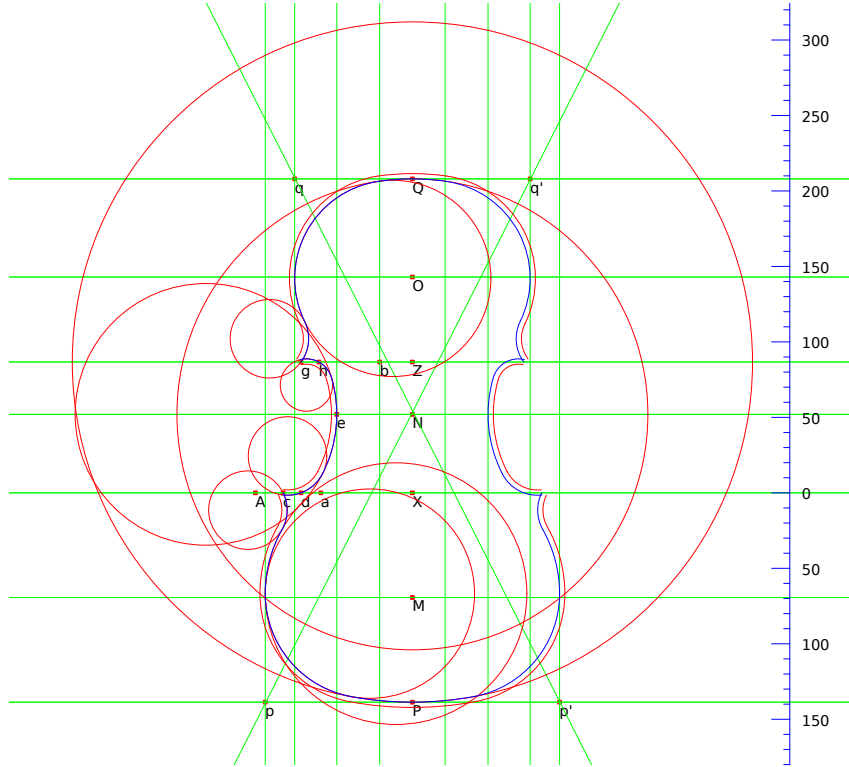


Figure 1.8: Output of Prof. Mairson displaying a theoretical Amati violin as designed by Denis

As the violins we are investigating are not perfectly shaped anymore, the methods should be robust to noise. We thus added a bit of noise to the Digital Amati as follows: for each point (x_i, y_i) in the contour, the noisy point $(\tilde{x}_i, \tilde{y}_i)$ is:

$$\begin{aligned}\tilde{x}_i &= x_i + \epsilon_x, & \epsilon_x &\sim \mathcal{U}(-0.01, 0.01) \\ \tilde{y}_i &= y_i + \epsilon_y, & \epsilon_y &\sim \mathcal{U}(-0.01, 0.01)\end{aligned}$$

Where $\mathcal{U}(a, b)$ denotes a continuous uniform distribution on the interval $[a, b]$. The two noise components ϵ_x and ϵ_y are independent of each other and independent across points. This "new" violin will be referred to as Noisy Digital Amati.

²Andrea Amati was a Cremonese luthier from the 16th century.

1.2 Problem Formulation

The problem tackled in this thesis will thus be to approximate the outline of violin backs using circles. In some methods, we will allow for some loose points, considering that free-hand drawing is a possible hypothesis, but not the main focus. We also consider the possibility that ellipses can be found in violins, either by design or by distortion of circles. However, we chose to exclude this possible theory from our research and to focus solely on circles. Another area that was left out is spirals, as they are an ill-defined mathematical concept, and seem difficult to draw perfectly with historical tools.

In order to approximate those outlines, we will develop different methods and compare them using five different evaluation metrics (more details on this in Section 1.3):

- Number of Circles
- Average error
- Average maximum error
- Maximum Error
- Symmetry score

1.3 Evaluation Metrics

One challenge we encountered is that we do not know what could be considered a good approximation. Therefore, we are considering five distinct evaluation metrics, knowing that none of them can describe how accurate the approximation is on their own. Depending on the method used, some of these metrics may be fixed or bounded by the method. A glossary containing recurring mathematical notations can be found on page 66.

Number of Circles

The first metric we consider is the number of circles found by the algorithm. This is a delicate metric to work with, as we do not know how many circles were used to design violins. However, we can still see if the number of circles found is within a reasonable range (roughly between 10 and 40), as it does not seem likely that a luthier would use much less or much more than that to draw a violin. Secondly, elbows in parameter study plots may reveal an optimal number of circles.

Average Error

The second metric we consider is the average error, meaning the distance between each point on the outline and its assigned circle, averaged over all points of the outline. In the case that a point is assigned to two circle arcs, the smallest value is retained. The average error is thus computed as follows:

$$\bar{\delta} = \frac{1}{n} \sum_{p_i \in \mathcal{P}} \min_{C_k^* \in \mathcal{S}_i} ||p_i - \mathbf{c}_k|| - r_k| \quad (1.1)$$

$\mathcal{P} = \{p_1, \dots, p_n\}$ is the set of contour points and $\mathcal{C}^* = \{C_1^*, \dots, C_{K'}^*\}$ is the set of circle arcs approximating the outline. $\mathcal{S}_i \subseteq \mathcal{C}^*$ is the subset of $\mathcal{C}^*$ of circles encompassing point p_i . Each circle C_k^* has center $\mathbf{c}_k = (x_k^c, y_k^c)$, and radius r_k .

As illustrated in Figure 1.9, each **green segment** represents the individual distance for one contour point. The average error averages all such segments over each point of the full outline.

Average Maximum Error

The next metric is the average maximum error, meaning the maximum distance between each arc and its assigned points, averaged over all circle arcs. It is computed as follows:

$$\bar{\delta}_{max} = \frac{1}{K'} \sum_{C_k^* \in \mathcal{C}^*} \max_{p_i \in \mathcal{P}_k} ||p_i - \mathbf{c}_k|| - r_k \quad (1.2)$$

$\mathcal{P} = \{p_1, \dots, p_n\}$ is the set of contour points and $\mathcal{C}^* = \{C_1^*, \dots, C_{K'}^*\}$ is the set of circle arcs approximating the outline. $\mathcal{P}_k \subseteq \mathcal{P}$ is the subset of contour points assigned to circle C_k^* , $\mathbf{c}_k = (x_k^c, y_k^c)$ is the center of circle C_k^* , and r_k is its radius. This corresponds to the largest **magenta segment** in Figure 1.9 averaged over all circle arcs.

Maximum Error

The fourth metric we use is the maximum error, meaning the worst-case distance on the entire violin outline. It is expressed as follows:

$$\delta_{max} = \max_{C_k^* \in \mathcal{C}^*} \max_{p_i \in \mathcal{P}_k} ||p_i - \mathbf{c}_k|| - r_k \quad (1.3)$$

$\mathcal{P} = \{p_1, \dots, p_n\}$ is the set of contour points and $\mathcal{C}^* = \{C_1^*, \dots, C_{K'}^*\}$ is the set of circle arcs approximating the outline. $\mathcal{P}_k \subseteq \mathcal{P}$ is the subset of contour points assigned to circle C_k^* , $\mathbf{c}_k = (x_k^c, y_k^c)$ is the center of circle C_k^* , and r_k is its radius. This corresponds to the largest **magenta segment** in Figure 1.9 across all arcs and all points.



Figure 1.9: Illustration of circle fitting and error metrics on a single arc of a violin outline. Each green segment is the individual error $|||p_i - \mathbf{c}_k|| - r_k|$ for one contour point p_i . The magenta segment highlights the maximum error on this arc. Note that the evaluation metrics are computed over the full outline; this view shows a single arc for clarity.

Symmetry Score

The last evaluation metric is a symmetry score reflecting how symmetrical the approximating circles are between the left and right halves of the violin. It is indeed unlikely that violins were designed asymmetrically, but historical distortions may have affected that symmetry. This score also differs in the question it answers: rather than measuring how faithfully the circles approximate the outline, it captures how historically plausible the configuration is. For this reason, symmetry is studied separately in Chapter 5, where the symmetry score is developed and formulated in Section 5.1. In Chapters 3 and 4, symmetry is assessed qualitatively.

Chapter 2

State of the Art

2.1 Historical and Organological Perspectives

Construction of musical instruments using a straightedge and a compass has been attested since the Middle Ages: Henri Arnaut de Zwolle, a Franco-Flemish physician and astronomer, wrote a treatise on the design of musical instruments, among others, a lute drawn with a compass [12]. Another lute from the Brussels Musical Instruments Museum, built in Venice by luthier Johann Hieber around 1600 (number of inventory 1561), shows marks of a compass. Those marks vary quite significantly from the ones from Arnaut de Zwolle. It is thus attested that lutes were drawn with a compass, but the design methods vary depending on the maker.

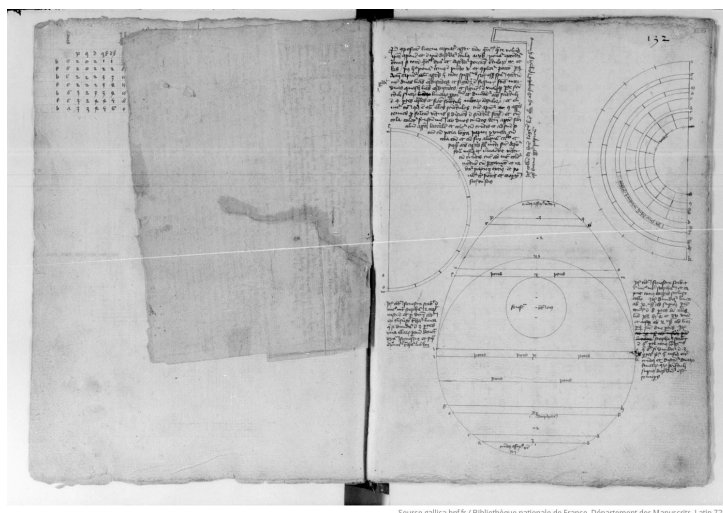


Figure 2.1: Design of a lute by Henri Arnaut de Zwolle

For the violin itself, no source from the Low Countries has been found. However, there exist indirect sources from Italy, such as models or moulds from Stradivari's workshop¹ [3]. In 1972, Simone Sacconi presents an attempted reconstruction of one of Stradivari's moulds [17]. As explained in Section 1.1, Italian violins were built quite differently from violins from the Low Countries. Given this, it is reasonable to assume that design meth-

¹Antonio Stradivari (1644–1737) was a Cremonese instrument maker who achieved broad recognition, with his instruments finding their way into the collections of courts across Europe.

ods varied accordingly. The approaches detailed here will therefore not serve as a direct starting point for our analysis, but rather as a point of comparison and contextual reference.

In 1987, Kevin Coates published a book named "Geometry, Proportion and the Art of Lutherie" [7] which first presents various mathematical concepts that were used for craftsmanship: units and numbers, proportionality, means, ratios, the Golden Section and more. Various instruments are then presented, and analysed as geometrical constructions (see Figure 2.2). However, the authenticity of those instruments has later been questioned [16].

As already introduced in Section 1.1, François Denis, a French luthier, wrote in 2006 a *Traité de lutherie* presenting theoretical constructs of famous Italian violins [8]. Denis first analyses the historical drawing of a lute made by Henri Arnaut de Zwolle mentioned earlier. As mentioned in Section 1.1, the last part is the core of what interests us: he fully recreates an Amati mould from the averaged contours of different instruments stemming from the same mould. He then attempts to describe its shape through a construction of circles. This is thus the opposite of our process: he arranges circles to obtain the desired shape. It is this process that American professor from Brandeis University, Harry Mairson, followed to create the "Digital Amati": a computer software used by some luthiers to design their own instruments. As Mairson put it, "Digital Amati is, in effect, a programmable straightedge and compass". As mentioned in Section 1.1, we extracted the outline of this theoretical violin to use as a reference for our methods. While Denis's *Traité* focuses on moulds, and thus cannot be applied to our violins, he also worked directly on outline in [9]. He uses the same method as in his *Traité*: starting from circles, he constructs an outline as close as possible to his target instruments, rather than fitting circles to an existing outline.

Overall, various theories have arisen for the construction of violins, but while they differ in some ways, there are still observable patterns.

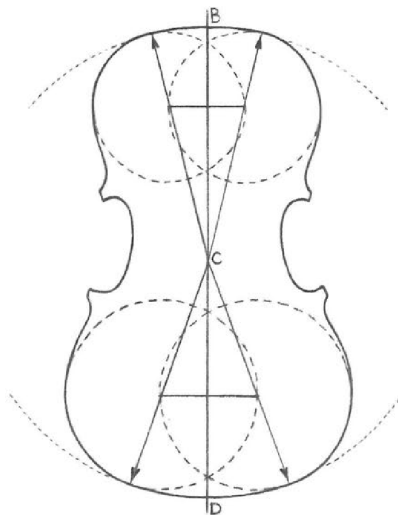


Figure 2.2: Drawing from Kevin Coates

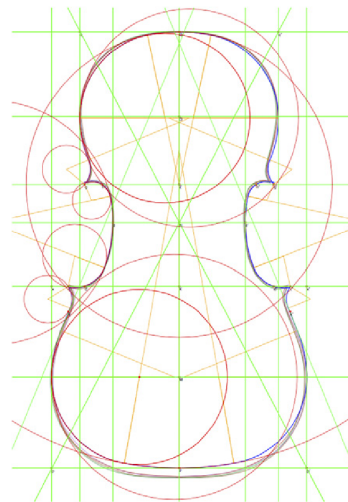


Figure 2.3: Digital Amati by François Denis, digitalised by Harry Mairson. The figure only displays the construction circles on the left-hand side of the instrument.

As shown in Figures 2.2 and 2.3, the upper and lower bouts share a similar construction: two symmetrical circles joined by a larger circle to form the flatter section. These features have remained consistent throughout the construction attempts and are worth keeping in mind.

2.2 Geometric and Computational Methods

2.2.1 Computational Approaches to Violin Research

Above the lutherie theories detailed above, some quantifiable, mathematical studies have been conducted on violins. In 2014, Daniel H. Chitwood published an article studying the morphology of violins to trace genetic lineage within a 9000-instrument corpus ([6]). Chitwood works exclusively with violin outlines, obtained from iconography with ImageJ macros, using built-in functions "Find Edge", "Make Binary" and "Fill Holes". The first part of this article aims to distinguish instruments from the same family (violin, viola, cello or double bass) using Principal Components Analysis and Linear Discriminant Analysis. LDA and hierarchical clustering were then used to study the relatedness of violin shapes and luthiers. While this study is not directly related to the topic of this thesis, this general approach highlights the wide variety of violin shapes across the corpus and suggests that different luthiers may have developed distinct tracing methods over time.

In [10], the authors predict the modal frequencies of violins using geometric parameters. While this article uses the shape as a starting point for the exploration of other characteristics, it is interesting to note that the outline they use to conduct their research is a parametric construction generated from circles (based on a drawing from the workshop of Enrico Ceruti²). Overall, acoustic characterization of the violin using various mathematical techniques is a topic that has been thoroughly researched (see [11]).

2.2.2 Circle-Based Outline Approximation

The idea of describing shapes using circles is very generic. Theoretically, any object or shape can be accurately described by a set of circles, as long as the tolerance or size of the set is loosely constrained. The specificity of our research question is that we want those circles to have some historical significance, whether it comes from a low number of circles, approximate symmetry, or repeated patterns between violins. As such, the key resides more in twisting straightforward techniques to fit our needs rather than exploring complex shape representation methods.

Overall, many methods exist to approximate outlines using curves. Bézier curves, for instance, represent shapes through parametric polynomial curves defined by control points, and are widely used in computer graphics and font design for their smoothness and flexibility [14]. However, those methods do not use circles specifically, but parametric curves. Other methods, such as the Medial Axis Transform, represent a shape by its skeleton, that is, the set of all maximal inscribed circles [18]. This approach is fundamentally oriented toward the interior of a shape rather than its outline, and completely disregards outside circles. It also aims at filling the shape rather than accurately describing the outline. More

²Enrico Ceruti (1806 – 1883) was a Cremonese luthier.

broadly, the main purpose of these approaches is data reduction, representing a shape with as few parameters as possible, which is a different objective from ours, where the circles themselves must carry geometric and historical meaning.

One article retained our attention. Yang and Du (1996) [19] propose an iterative algorithm to approximate digitized curves by piecewise circular arcs, seeking the smallest number of arcs such that the maximum deviation from the original curve remains within a specified tolerance. Their approach is particularly relevant to our problem, and the geometric and iterative nature of their method also leaves room for adaptation to our specific constraints, namely, the historical plausibility of the circles. This method will therefore be studied in detail in Chapter 3, where we explore how it can be applied and adapted to the approximation of violin shapes.

Chapter 3

Analysis of a Greedy Algorithm

To approach our problem, the first algorithm we considered and implemented is described in [19] by Yang and Du. It is a greedy algorithm that ensures that the distance between every constructed circle and the outline of the violin remains below a tolerance. For simplicity, the algorithm will be referred to as the YangDu algorithm in this thesis.

3.1 Description of algorithm

The algorithm can be split into two parts: arc detection and shape separation. Firstly, given a defined arc of the outline with a beginning and an endpoint, one has to compute the best-fitting circle. The second part takes the entire shape of the violin and performs a search, cutting the shape into arcs to make sure that all circles are below a defined tolerance. This explanation of the algorithm follows the article's notation and, therefore, is not fully compliant with the glossary on page 66.

3.1.1 Best Single Arc Approximation

The purpose of this step is to compute the best-fitted arc for a digitized curve $\Delta = \{V_1, V_2, \dots, V_n\}$ where $V_1, \dots, V_n$ are the points of the curve. In order to facilitate the computations, we start by normalizing the arc we are working with, by centering it on the Cartesian plane, and placing both endpoints on the x-axis. This step can be seen in Figure 3.1 (a-b). Let $V_1 = (-a, 0)$ and $V_n = (a, 0)$ be the two endpoints of the normalized digitized curve Δ_n . Let Υ_m be the unique circular arc with endpoints V_1 and V_n that crosses the y -axis at $M = (0, m)$ with $m \in \mathbb{R}_0$, which is called the midpoint of Υ_m . Υ_m is one possible approximation of Δ_n , and the error of the approximation can be measured by the distance between Δ_n and Υ_m , i.e.,

$$E_\infty(\Delta_n, \Upsilon_m) = \max_i \{e_i(m)\} \quad i = 1, \dots, n$$

where $e_i(m)$ is the distance between point i of Δ_n and the circular arc Υ_m , defined as $e_i(m) = \min\{d(V_i, P) : P \in \Upsilon_m\}$ and d is the Euclidean distance between two points (see Figure 3.1 (b)).

The single arc approximation problem is equivalent to finding an m^* such that the error

$$E_\infty(m) = E_\infty(\Delta_n, \Upsilon_m) = \max_{i=1}^n \{e_i(m)\}$$

is minimum at $m = m^*$.

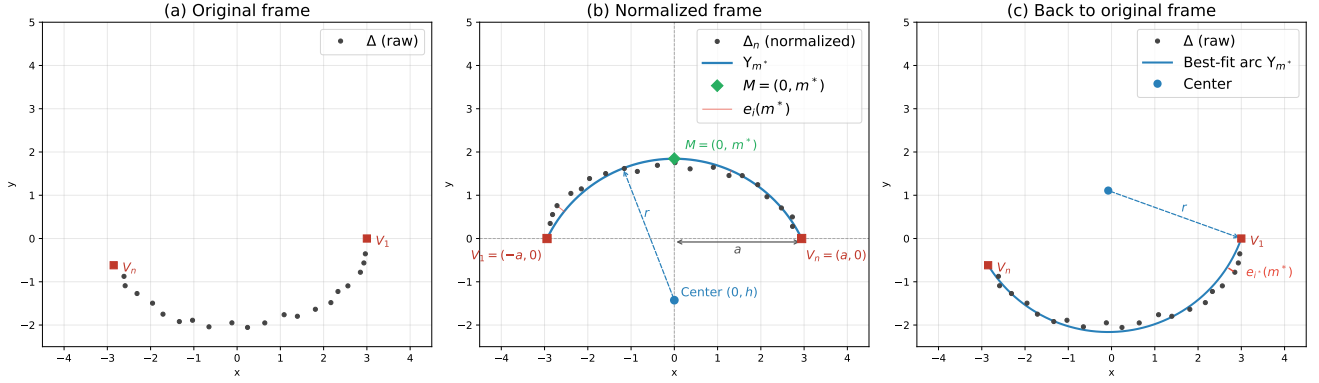


Figure 3.1: Depiction of best single arc approximation. (a) displays the digitized curve Δ before normalization. (b) displays the normalized digitized curve Δ_n , with best fitting arc Υ_{m^*} and corresponding parameter $M=(0, m)$ and error $e_i(m^*)$. (c) displays the digitized curve Δ and best fitting arc Υ_{m^*} in original frame.

To compute this m^* , the first step is to determine the search interval. For each $V_i = (x_i, y_i)$, $2 \leq i \leq n-1$, there corresponds a unique Υ_{m_i} , passing through V_i with midpoint $M = (0, m_i)$, while maintaining the fixed endpoints at $(-a, 0)$ and $(a, 0)$. The search interval for optimal point m^* is located in the search interval $I = [m_{\min}, m_{\max}]$, where $m_{\min} = \min\{m_2, m_3, \dots, m_{n-1}\}$ and $m_{\max} = \max\{m_2, m_3, \dots, m_{n-1}\}$ (see Figure 3.2 (a)). Once we have the search interval, it iteratively divides the interval into three parts, evaluates the error at two interior points (m_a and m_b), and eliminates the third of the interval with the worst error (see Figure 3.2 (b)). This process continues until the interval is smaller than a tolerance defined at $1 \times 10^{-8}mm$, or the number of iterations reaches 200 (see Figure 3.2 (c) and (d)). Once m^* is found, the center offset and radius of the optimal arc are recovered via $h = \frac{m^{*2}-a^2}{2m^*}$ and $r = \sqrt{h^2 + a^2}$, fully characterising the fitted circle. The normalized digitalized curve Δ_n and the center can be restored to the original coordinates (see Figure 3.1 (c)).

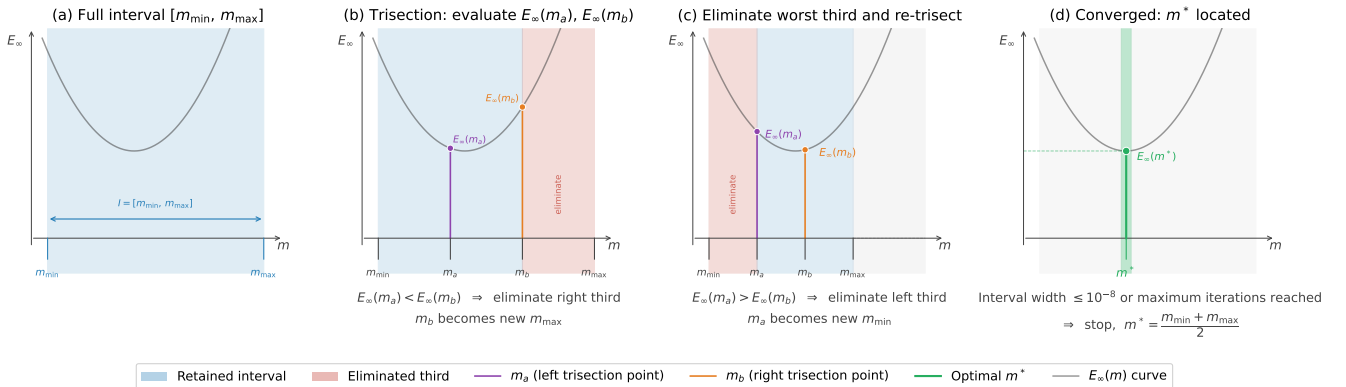


Figure 3.2: Depiction m^* search by ternary steps. (a) presents the full interval $I = [m_{\min}, m_{\max}]$. (b) and (c) present steps of the search process by trisection. (d) presents the converged search process, with m^* located.

The key theoretical insight enabling this search procedure is that the error function

$E_\infty(m)$ is unimodal over the search interval $I = [m_{\min}, m_{\max}]$. This follows from the fact that each individual distance function $e_i(m)$ is unimodal, and the maximum of finitely many unimodal functions is itself unimodal. Unimodality guarantees the existence of a single global minimum with no other local minima, which means that interval-reduction techniques such as golden section search (see Section 3.1.2) are guaranteed to converge to the optimal arc Υ_{m^*} without getting trapped.

3.1.2 G^0 Piecewise Arc Approximation

Once we know how to compute an arc for a defined set of points, we need to extend this to the entire shape of the violin. The purpose is to find as few arcs as possible while remaining under a chosen tolerance ε for $E_\infty(m)$. This tolerance is expressed in millimeters. The strategy works in two phases per segment. Figure 3.3 displays the entire process.

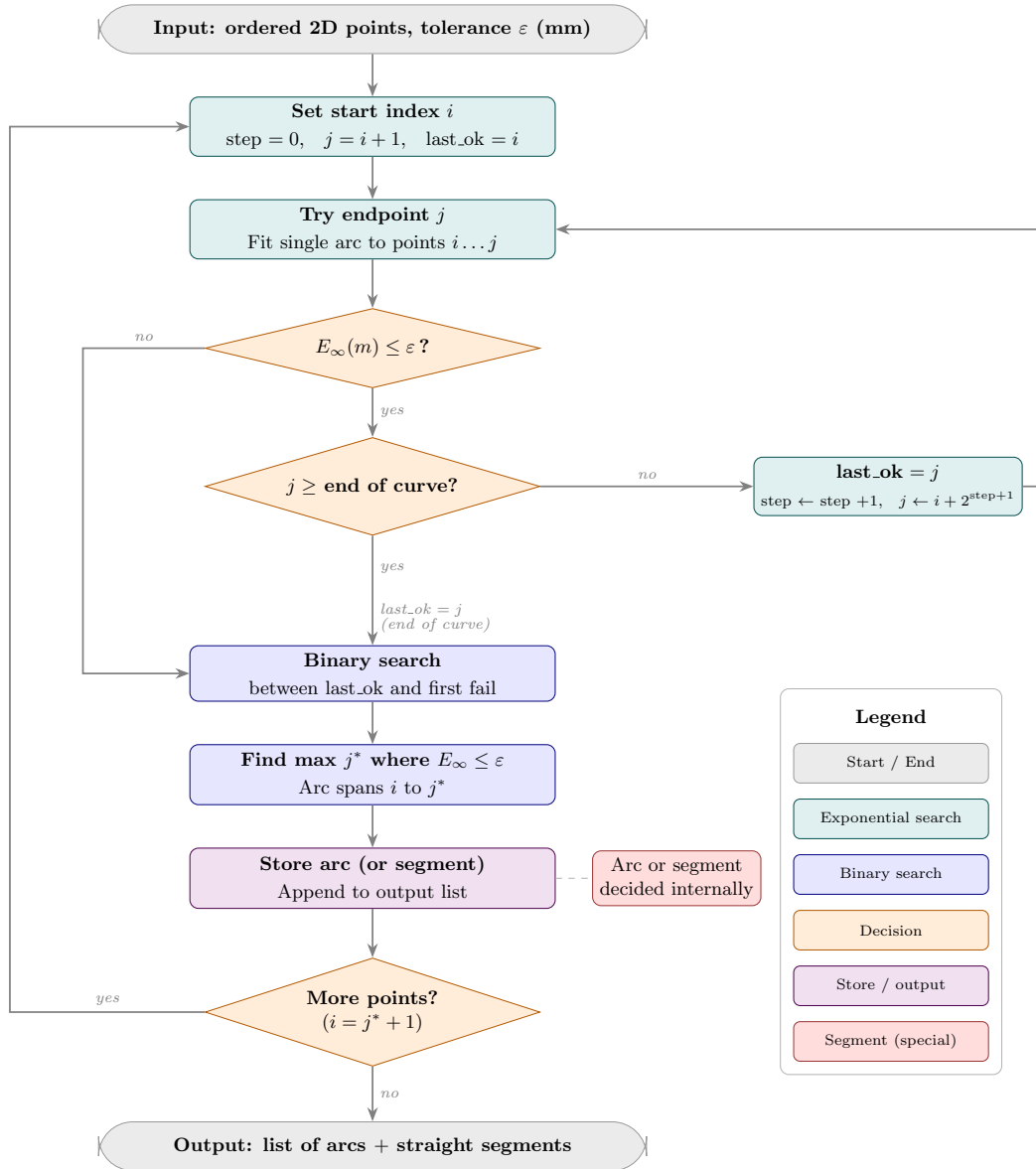


Figure 3.3: YangDu Algorithm Flowchart

In the exponential search phase, starting from point i , it tries increasingly larger segments by doubling the endpoint at positions $i + 1, i + 2, i + 4, i + 8, i + 16$, etc. For each attempt, it fits a single arc and checks if the error is less than or equal to tolerance ε . It keeps track of the last successful endpoint and continues doubling until either the error exceeds ε or it reaches the end of the curve. This phase quickly identifies a rough upper bound for how far the arc can extend.

Once the exponential search fails or reaches near the end, the binary search refinement phase begins. It performs a binary search between the last valid endpoint and the first failed endpoint to precisely find the furthest point where the error remains below the tolerance. Once a maximum-sized circle is found, it is stored, and the process repeats from the next point onward.

In the case of failure to fit an arc to a group of points, the arc is classified as a "segment", thus a straight line encompassing this group of points. It is hard to classify whether finding a segment is a good thing, as it is possible that luthiers drew segments as well as circles, or if it is a failure of the algorithm that could not find a better-fitting circle for this zone.

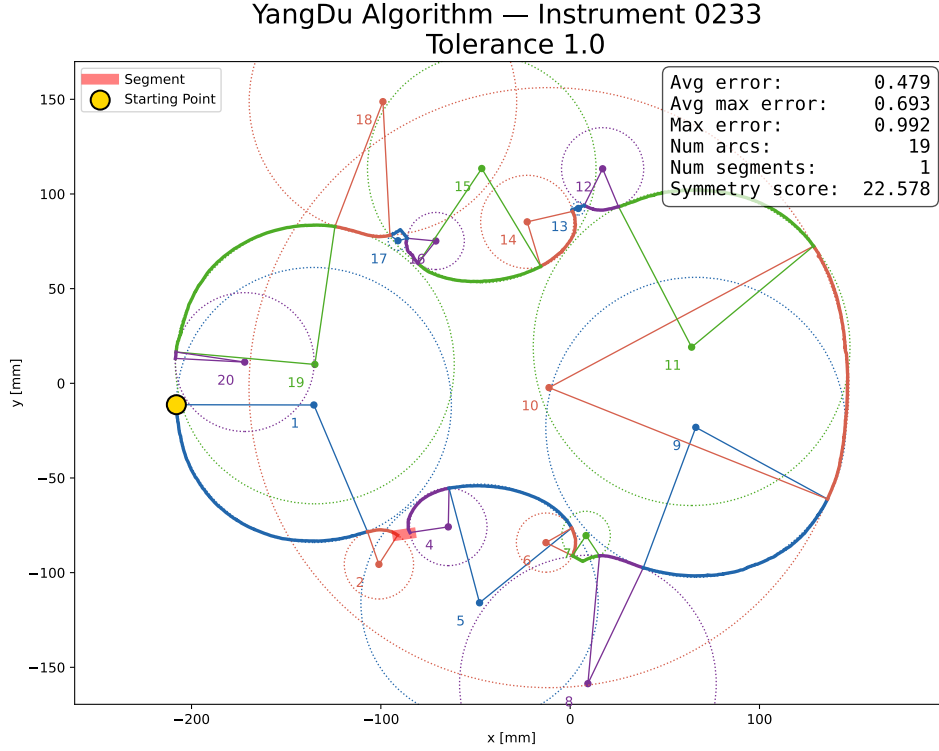
3.2 Results and Limitations

When using the algorithm, we vary one main parameter: the tolerance, which defines the maximum distance between the outline and the circles, and is expressed in millimeters. A study of this parameter can be found in Section 3.2.1.

As we can see in Figure 3.4, the YangDu algorithm gave a satisfying number of circles, ranging between 19 and 26 circles for a tolerance ε of 1 mm (see Section 3.2.1 for a discussion on tolerance). The YangDu Algorithm retrieves one segment for instrument 0233, between circles 2 and 4. This segment forms the corner, a place where we would not expect to find a circle. However, the algorithm does not retrieve segments for the other corners, leading to poorly fitted circles (see circle 7 on Figure 3.4). This is likely explained by corner curvature: circle 4 approximates a corner whose curvature is too sharp to admit even a rough circular fit.

We do notice several key weaknesses. Firstly, even though some circles are symmetrical (see 1 and 19 for instance), most lack their symmetric counterpart (2 and 18 have different radii, the same goes for 4 and 16, 8 and 12, and 20 lacks any symmetrical circle). Since we are working with ancient (and possibly damaged) violins, it is hard to attribute this to a fault in the algorithm with certainty, as it could also be caused by non-symmetrical distortions or modifications. Indeed, the wood twists or cracks with age.

Another weakness, which this time can more surely be attributed to the algorithm itself, is its dependence on the starting point. The YangDu Algorithm is a greedy algorithm, and is therefore by nature heavily conditioned by the starting point. The solution is not guaranteed to be the best overall, and the algorithm will never question the previous steps to adjust for better circles or lower evaluation metrics. This is particularly evident for circle 20. As the algorithm approached the end of the outline, it was forced to insert a final circle that covers only a small portion of the contour, a portion that is structurally comparable to the part already represented by circle 19. This will be further explored in

Figure 3.4: YangDu Algorithm with tolerance $\varepsilon=1.0\text{mm}$ – Instrument 0233

Let us now investigate how the algorithm performs on the Digital Amati. As we are looking for perfectly fitted circles, the tolerance is set much lower, and we can see in Figure 3.5 that the algorithm performs well and finds 18 circles, which was the number used to build the Digital Amati. However, the fit is not perfect, and a few circles are wrongly placed: circles 2 and 6 are very different, and circles 14, 15, and 16 are slightly shifted to the right. Another key observation is that the method finds 3 segments, and we know this violin is only made of circles. These segments are barely visible, but are placed at the corners of the violin, between circles 7 and 9, 11 and 13, and 17 and 19. This explains why the circle count reaches 21 circles even though only 18 circles are retrieved. This suggests that the algorithm falls back to a segment without expanding the search enough. In the theoretical construction, the corners are slightly overlapping, which may cause these results.

While the YangDu Algorithm performs fairly well on the perfect Digital Amati, it gives inferior results for the noisy Digital Amati, giving diverging inferior results (see Figure 3.6). It does find 18 circles, but these are wrongly placed and mostly asymmetrical. The evaluation metrics are comparable to violin 0233, implying that the YangDu algorithm does not detect any underlying construction in the Noisy Digital Amati.

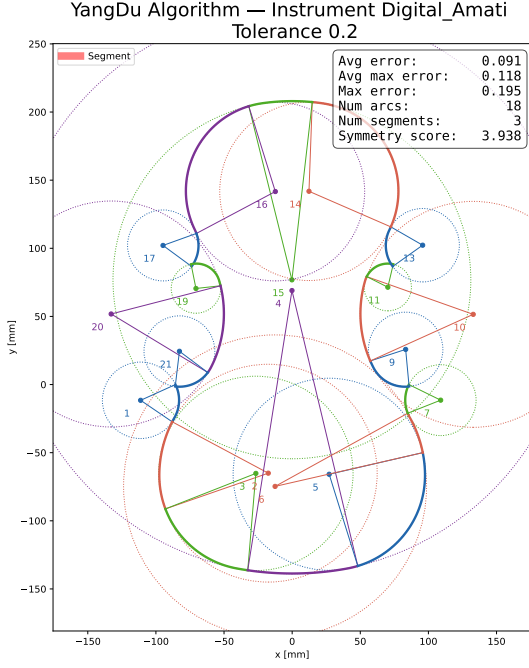


Figure 3.5: YangDu Algorithm with tolerance $\varepsilon=0.2\text{m}$ – Digital Amati

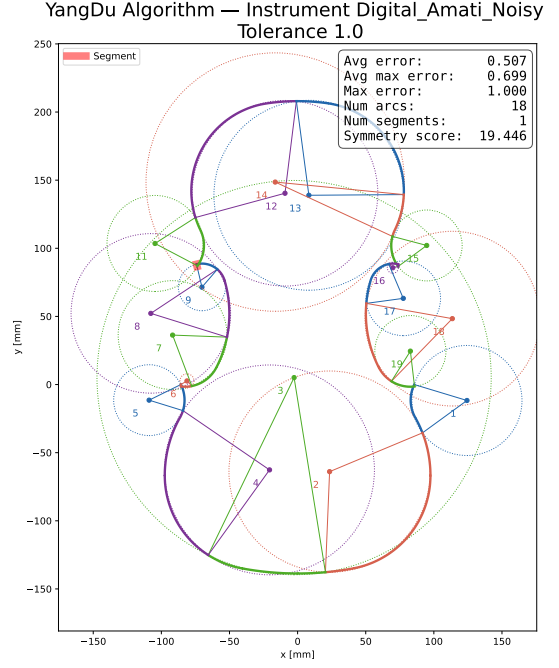


Figure 3.6: YangDu Algorithm with tolerance $\varepsilon=1.0\text{mm}$ – Noisy Digital Amati

3.2.1 Tolerance Analysis

As explained above, the tolerance, expressed in millimeters, is the tuning parameter of this algorithm. Working with such ancient violins can be complex, as they might have been distorted or modified through time. As such, choosing a low tolerance prevents the algorithm from adapting to those possible dents or imperfections. On the contrary, a high tolerance may extend the circles too far and give poorly fitted results. In this section, we will discuss how the tuning of the tolerance affects various evaluation metrics: the number of circles, the average error, and the average maximum error. The maximum error is not discussed here, as it is structurally correlated with the tolerance.

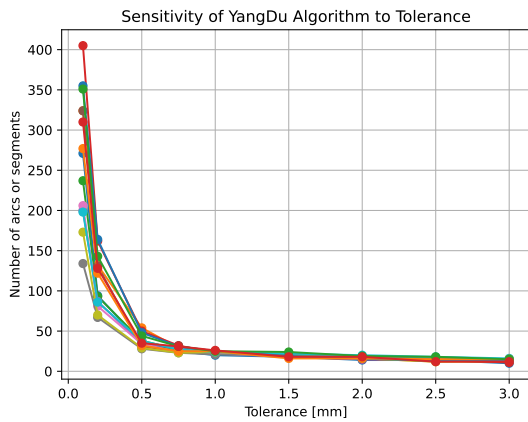


Figure 3.7: Number of circles and segments found for varying tolerances

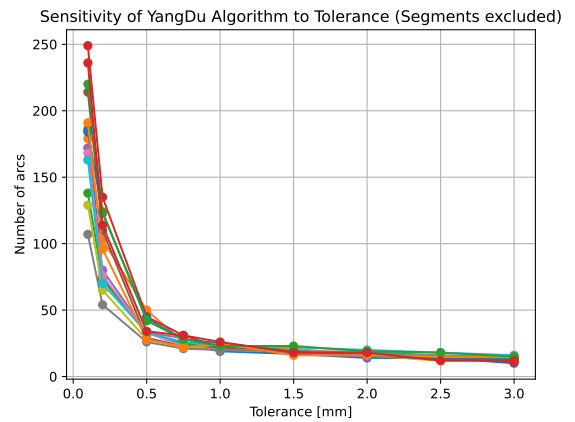


Figure 3.8: Number of circles found for varying tolerances, segments excluded

Each line on Figures 3.7, 3.8, 3.9 and 3.10 represents a different violin. The legend

is excluded to improve readability, as we are looking for trends rather than studying specific violins. As we can see on Figures 3.7 and 3.8, the number of circles (with or without segments) quickly drops to 20-25 for a tolerance of 1mm. This is a strength of the algorithm: even with a relatively low tolerance, the number of circles stays within an acceptable range. For lower values, the algorithm finds more segments (about 200 with a tolerance of 0.1mm). With a higher tolerance, the algorithm is allowed to search further and is more likely to detect circles.

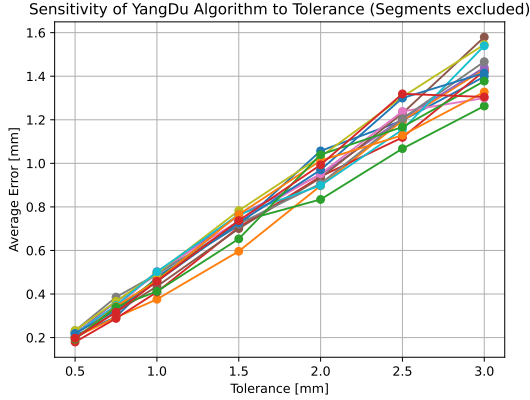


Figure 3.9: Effect of Tolerance on Average Error $\bar{\delta}$

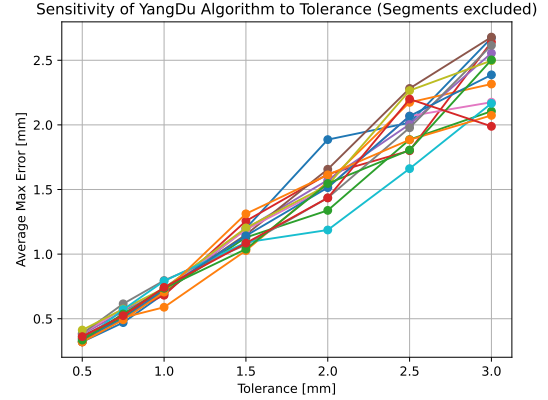


Figure 3.10: Effect of Tolerance on Average Maximum Error $\bar{\delta}_{\max}$

As we can see on Figures 3.9 and 3.10, both the average error $\bar{\delta}$ and average maximum error $\bar{\delta}_{\max}$ increase with the tolerance, without any clear breakpoints. It is thus difficult to draw a definite conclusion on the best parameter choice for this algorithm. For the next experiments, a tolerance of 1mm was chosen as a reference, as it seems reasonably precise, and the point where the number of arcs seems to settle on Figures 3.7 and 3.8.

3.2.2 Dependence on starting point

The biggest drawback of this algorithm is its dependence on the starting point. Below are presented two runs of the algorithm on the same violin with the same tolerance, but different starting points.

This change in starting point gives very different results on the bottom half of the violin (right side of the image). The difference settles after a while, as we can see from circle 12 on Figure 3.11 and circle 5 on Figure 3.12. This suggests that, although the algorithm appears to be stable in the long run, the differences are significant and cannot be ignored. It also raises the question of how to choose a “good” starting point. As it stands, the choice seems somewhat arbitrary.

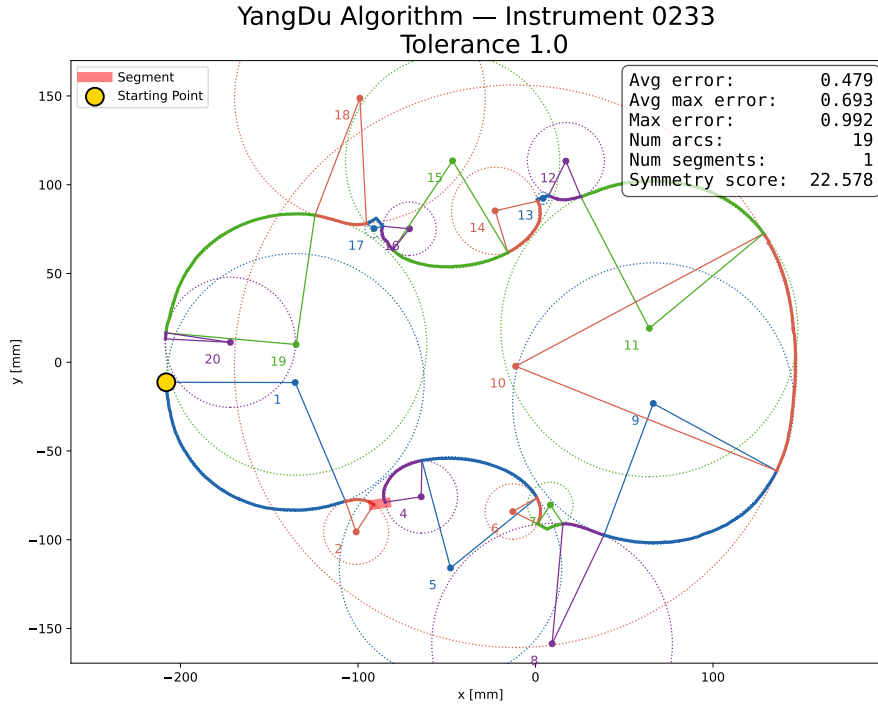


Figure 3.11: YangDu Algorithm with tolerance $\varepsilon=1.0\text{mm}$ – Instrument 0233

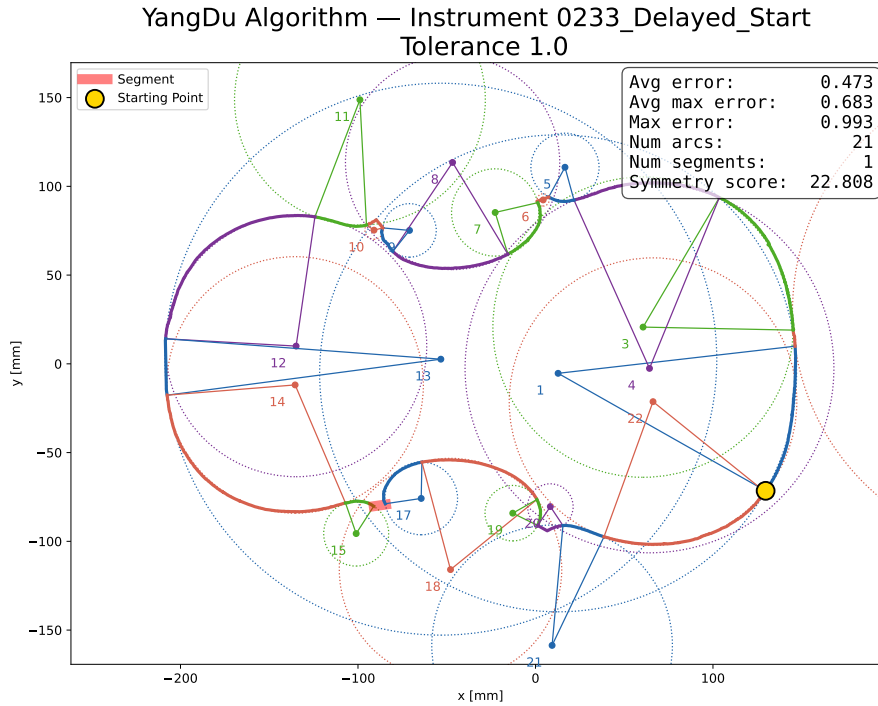


Figure 3.12: YangDu Algorithm with tolerance $\varepsilon=1.0\text{mm}$ – Instrument 0233 with delayed start¹

¹Circle 13 encompasses both sides of the neck cut. In fact, the algorithm does not differentiate between points that are close together and those that are farther apart. As such, the algorithm will "jump over" the gap and fit a circle to both sides at the same time.

Figure 3.13 presents three metrics: number of fitted circles, average error $\bar{\delta}$, and average maximum error $\bar{\delta}_{\max}$ evaluated for every tenth contour point used as a starting index. None of the three metrics exhibits a clear trend or a well-defined minimum, and they do not co-vary: starting indices that reduce the average error do not systematically reduce the number of circles or the average maximum error. The fluctuations appear largely irregular, suggesting that the piecewise arc fitting is sensitive to the starting point but without an exploitable structure.

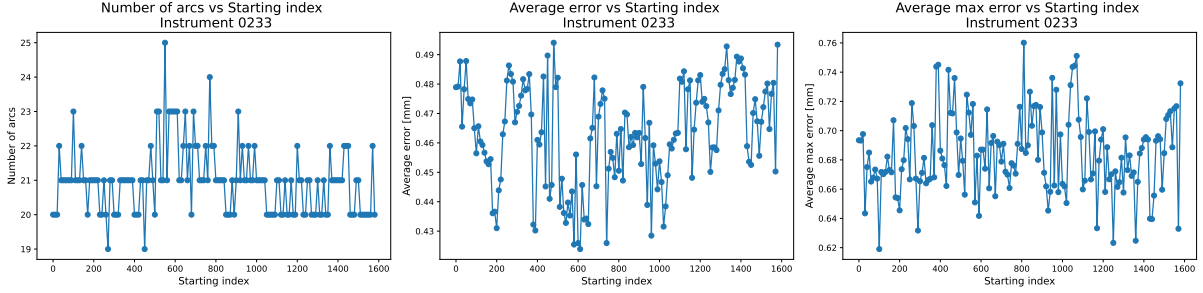


Figure 3.13: Number of fitted arcs (left), average error $\bar{\delta}$ (centre), and average maximum error $\bar{\delta}_{\max}$ (right) as a function of the starting contour index for violin 0233.

This is corroborated by Figure 3.14, which maps these metrics back onto the contour of violin 0233. Each colored point of the contour corresponds to a distinct starting point, and the color indicates lower (green) to higher (red) error for the YangDu Algorithm with that starting point. The average error $\bar{\delta}$ (left) shows some spatial structure: higher errors concentrate around the upper bout and the C-bouts, and there is a low-error zone on the lower right of the outline. This suggests that this region is more robustly fitted regardless of the starting index. This is likely due to its smoother, low-curvature geometry, but it is worth noting that this region does not appear symmetrically on the upper right part. The average maximum error $\bar{\delta}_{\max}$ (right) tells a different story: the low-error regions do not spatially overlap with those identified by the average error, indicating that the two metrics capture different aspects of the fit quality. The absence of a starting point where both metrics simultaneously reach a minimum suggests that the sensitivity to initialization is an intrinsic limitation of the greedy arc-fitting approach rather than a correctable weakness. It is also worth noting that geometrically intuitive starting points, such as the corners, which one might expect to serve as natural arc breakpoints, do not score particularly well on either metric in Figure 3.14.

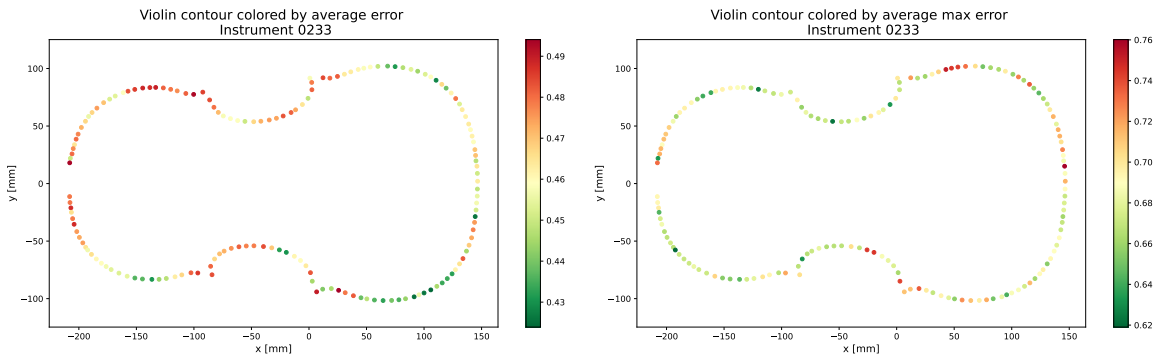


Figure 3.14: Contour of violin 0233 coloured by the average error $\bar{\delta}$ (left) and average maximum error $\bar{\delta}_{\max}$ (right) resulting from each contour point used as a starting index.

3.2.3 Separating parts

We attempted another approach to correct this dependence on the starting point: to split the whole contour into different parts rather than taking the outline as a whole. The idea is to follow the logic of the construction of a violin rather than simply follow mathematical results. We thus cut the violin at each of the corners, as the concavity around this zone makes it impossible for one circle to approximate both sides of the corners. We obtain the four parts depicted in Figure 3.15.

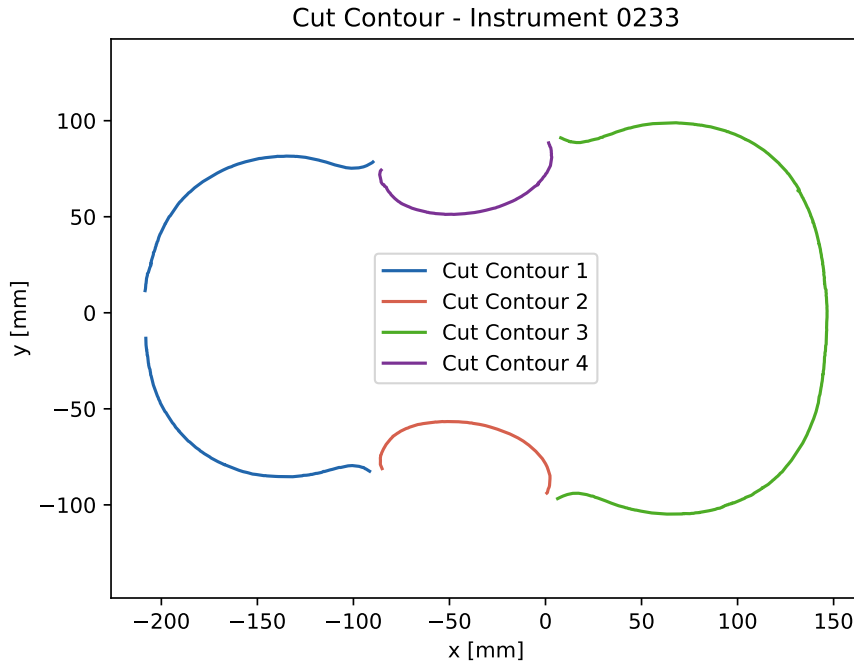


Figure 3.15: Violin 0233 outline divided into 4 parts, separated by the corners.

The idea is to enforce "logical" starting and ending points, hopefully getting better evaluation metrics or a more symmetrical result (see Table 3.1). As we can see in Figure 3.16, the obtained results are quite different from those obtained for Figure 3.4. The biggest advantage of splitting the outline is that the algorithm does not find any circle inside the corners, such as circles 13 or 17 on Figure 3.4, which have the wrong concavity. Besides this, the splitting does not present any significant advantage. The result seems to be less symmetric (see circles 10 and 11 against 13), and the evaluation metrics are very similar. One other thing that is noticeable, and unlikely in violin design, is the gigantic circle 14². This circle is centered at the point (1008.44, -3344.33) and has a radius of 3573.27mm. Such a big radius (≈ 3.5 m) seems very impractical to draw.

The results of the YangDu algorithm when separating the contour also especially highlight that the YangDu algorithm focuses on fitting the biggest arcs possible, rather than the best-fitted. We can see in Figure 3.16 that the circles found first, right after the starting point, are systematically bigger than their symmetric counterparts (see circles 1 and 5, for example).

²The number of the circle 14 does not show on the figure as the circle is too big, but we can see the green line crossing the violin, and describing the outline between circles 13 and 15.

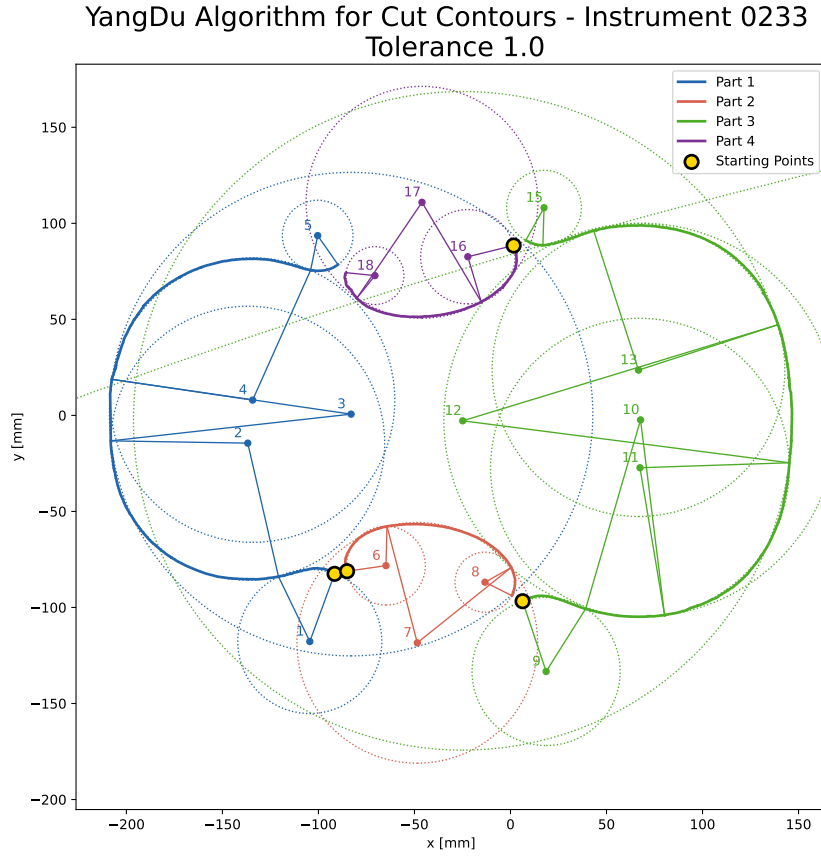


Figure 3.16: YangDu Algorithm with tolerance $\varepsilon=1\text{mm}$ – Instrument 0233 with splitted outline

Part	Arcs	Segs	Avg Error $\bar{\delta}$	Avg Max Error $\bar{\delta}_{\max}$	Max Error $\delta_{\max}$
Part 1	5	0	0.3624	0.5292	0.9794
Part 2	3	0	0.5134	0.6983	0.9802
Part 3	7	0	0.3973	0.6675	0.9984
Part 4	3	0	0.5545	0.8975	0.9923
Global Part 1-4	18	0	0.4569	0.6981	0.9984
Not splitted	19	1	0.479	0.693	0.992

Table 3.1: Piecewise arc fitting metrics for instrument 0233. For comparison, the last line presents the evaluation metrics for the algorithm when the outline is taken as a whole, as presented in Figure 3.4, thus with starting point at the side of the neck.

The evaluation metrics for the splitted YangDu Algorithm (Table 3.1) do not seem particularly better than the metrics for the full violin. As such, this last attempt to improve the YangDu Algorithm does not seem promising.

3.3 YangDu Algorithm: Summary

The YangDu algorithm is a solid first approach: it is fast (less than 10 seconds per instrument), keeps the number of arcs reasonable (19-26 for a tolerance of 1mm), and the tolerance parameter is intuitive to tune. That said, it comes with clear limitations. Being a greedy algorithm, it does not find the optimal decomposition: it finds the longest arcs it can, one at a time, without ever reconsidering earlier choices. This means it tends to produce large, rough arcs rather than the best-fitting ones overall.

The dependence on the starting point is also a key problem. Two runs on the same violin with a different starting index can give quite different results, and there are no exploitable results or metrics that would guide the selection of an optimal one. This makes any result, to some extent, arbitrary.

For these reasons, the YangDu algorithm serves as a useful baseline but is unlikely to recover the geometric intent behind the original violin design. The next chapters explore approaches that address some of these shortcomings.

Chapter 4

Development of a Global Approach

The main drawback of the YangDu algorithm is that it cannot guarantee the best solution overall. Due to its greedy nature, it is heavily conditioned on the starting point and will always select the biggest circle, regardless of whether it is the best. As such, the next natural step was to develop a global optimization method that tried to optimize the arcs on the entire outline rather than building the arcs one by one.

4.1 First optimization model: SetCover- ε

The global approach has two stages. First, candidate circles are generated to approximate portions of the violin outline. Second, an optimal subset is selected from these candidates: the subset must be small and yet cover all points as effectively as possible according to a defined criterion. This selection problem is formulated as a set covering problem. The method is formalized below, and Figure 4.1 illustrates the overall process.

4.1.1 Formulation

Let $\mathcal{P} = \{p_1, \dots, p_n\}$ be an ordered set of contour points. A maximum tolerance $\varepsilon > 0$ is fixed. For each point $p_i \in \mathcal{P}$, the neighborhood is expanded by incrementing the half-width $j = 0, 1, \dots$ and fitting a circle to the sub-sequence $\{p_{i-j}, \dots, p_{i+j}\}$ using least squares approximation. The half-width refers to a set of j points on each side of p_i . The expansion continues as long as the maximum distance from any point in the sub-sequence to the fitted circle remains below ε . The largest valid neighborhood is retained, and the corresponding circle is added to the candidate set $\mathcal{C} = \{C_1, \dots, C_n\}$, containing all the candidate circles. As only the candidate circle encompassing the most points from its neighbourhood is retained for each point, there are also n circles. The algorithm description of Chapter 3 followed the article notation, which explains why some notations may differ. Essentially, the digitalized curve Δ represents a section of $\mathcal{P}$.

Each candidate C_k covers a subset $\mathcal{P}_k \subseteq \mathcal{P}$ of contour points. The incidence matrix $A \in \{0, 1\}^{n \times n}$ is defined by each element $A_{i,k}$

$$A_{i,k} = \begin{cases} 1 & \text{if } p_i \in \mathcal{P}_k, \\ 0 & \text{otherwise.} \end{cases}$$

By construction, every point $p_i \in \mathcal{P}$ appears in at least one candidate's coverage set, so

the set covering problem is always feasible. The optimization model is:

$$\begin{aligned} \min_{x \in \{0,1\}^n} \quad & \sum_{k=1}^n x_k \\ \text{s.t.} \quad & \sum_{k=1}^n A_{i,k} x_k \geq 1, \quad \forall i = 1, \dots, n \end{aligned} \tag{4.1}$$

where $x_k \in \{0,1\}$ is a binary variable indicating whether candidate circle C_k is selected. The objective function minimizes the number of selected circles. The constraint ensures that every contour point is covered by at least one selected circle. This problem is solved using the Gurobi optimizer (version 13.0.1), and will be referred to as *SetCover- ε* throughout this thesis. Figure 4.1 displays the process of constructing the optimisation model.

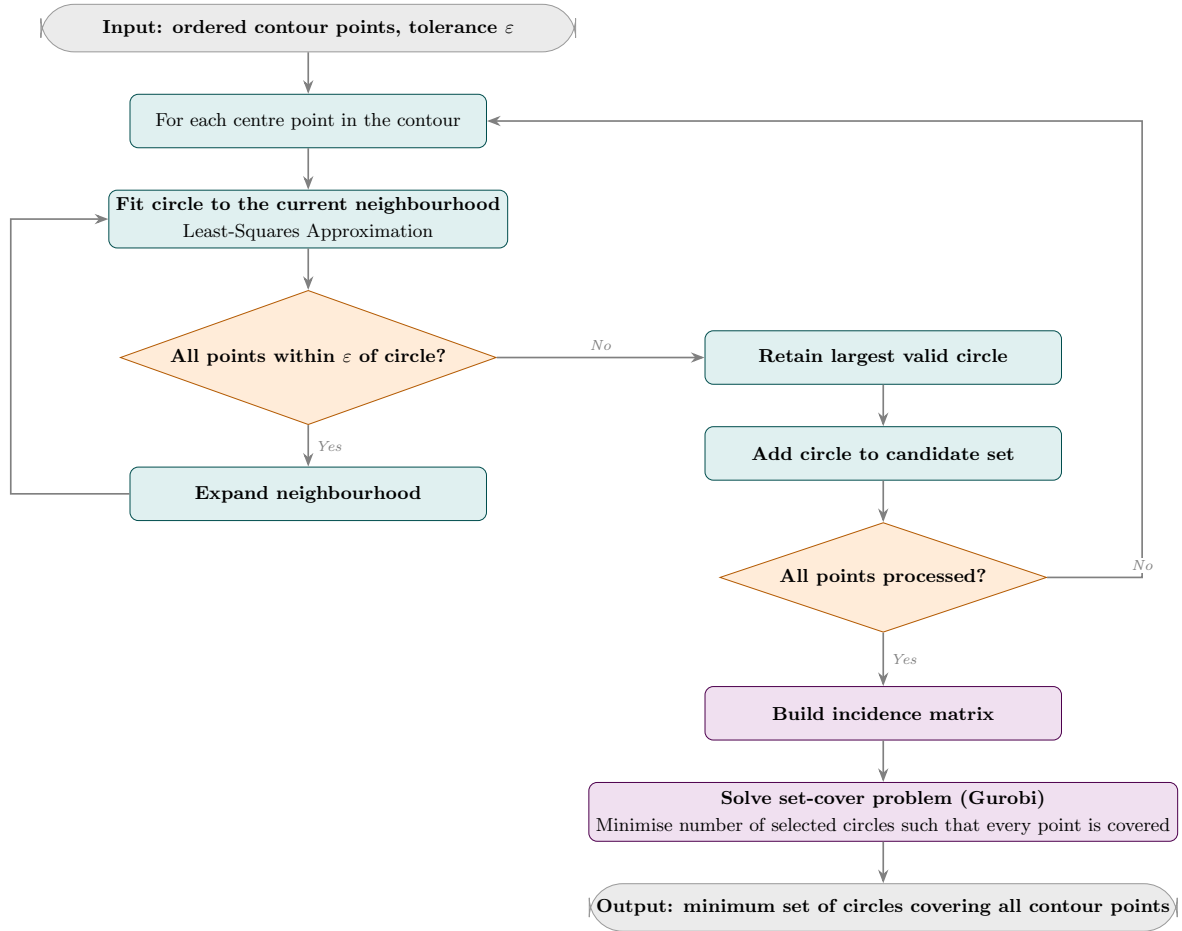


Figure 4.1: SetCover- ε flowchart

4.1.2 Results and Limitations

As shown in Figure 4.2, the SetCover- ε method shows promising results for instrument 0233: it produces 16 circles, three fewer than the YangDu algorithm, while halving the average error $\bar{\delta}$. The higher average maximum error $\bar{\delta}_{\max}$ is a direct consequence of the algorithm's design: circles are extended point by point until they reach the pre-specified

maximum tolerance, and only the largest acceptable circle is retained. The YangDu algorithm, by contrast, constrains each circle to pass through the start and end points of its corresponding arc, which constrains the fitting circles more, limits how far circles can be extended, and thus keeps the maximum error lower.

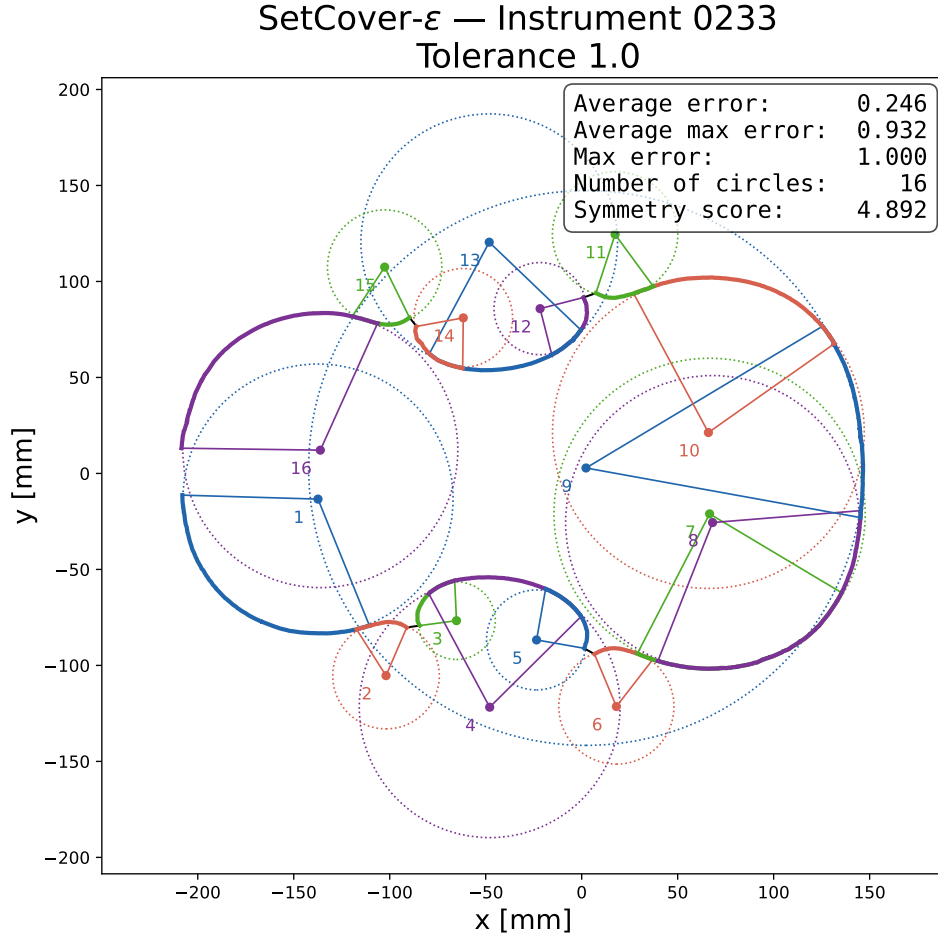


Figure 4.2: SetCover- ε for Instrument 0233 - Tolerance 1.0

One noticeable characteristic is that circles largely overlap, for instance, circles 7 and 8 in Figure 4.2. In fact, circle 7 describes only a very small part of the outline. These large overlaps suggest that there are many equivalent solutions for the same number of circles, and the Gurobi optimizer will stop at the first one that reaches the optimal value, disregarding the other evaluation metrics. It is thus quite possible that another solution with 16 circles improves the three evaluation metrics, leading to a better fit. This also questions whether the number of circles is the most meaningful parameter to minimize. Our purpose is to find circles that accurately describe the outline, not to describe the outline with as few circles as possible.

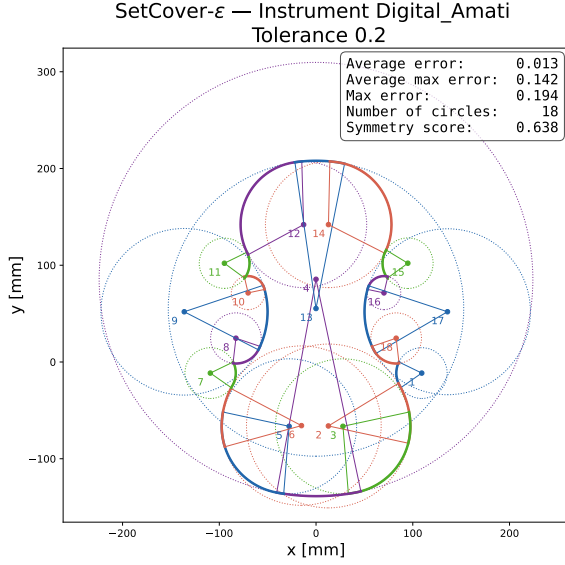


Figure 4.3: SetCover- ϵ for Digital Amati - Tol 0.2

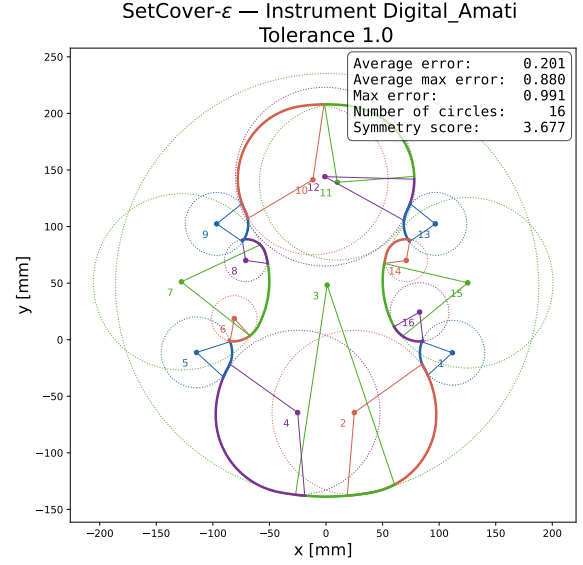


Figure 4.4: SetCover- ϵ for Digital Amati - Tol 1.0

As we can see, SetCover- ϵ gives very satisfying results for the Digital Amati for a small enough tolerance: the error metrics drop very low, and the results are qualitatively almost perfectly symmetrical¹. However, when we increase the tolerance to 1mm, the algorithm performs worse. It finds fewer circles than expected (16 instead of 18), and the circles do not correspond to the expected circles from the theoretical construction. This deterioration is a direct consequence of the candidate set construction: with a higher tolerance, each arc can expand further before being rejected, producing large candidate circles that individually cover more points but approximate the outline less faithfully.

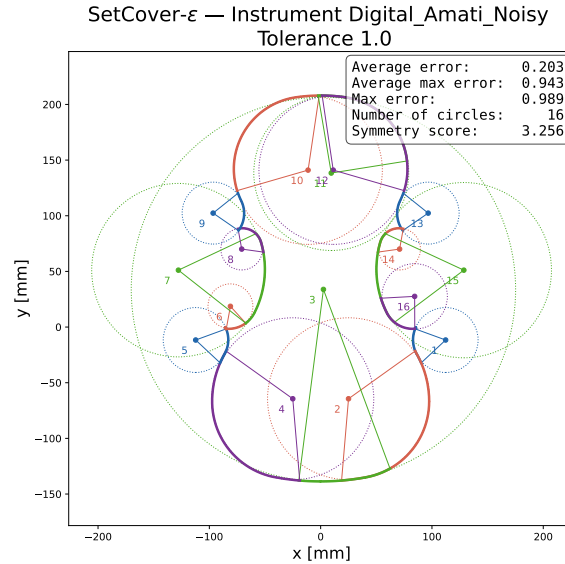


Figure 4.5: SetCover- ϵ on Noisy Digital Amati – Tolerance 1.0mm

For the Noisy Digital Amati at tolerance 1 mm, the solution closely resembles that obtained on the noiseless version at the same tolerance, as illustrated in Figure 4.4. Since

¹The symmetry score will be developed and studied in Chapter 5.

the model does not consider circle accuracy when selecting a subset, with a tolerance of 1mm, it completely disregards the noise.

Tolerance Analysis

To better understand how the choice of tolerance affects the results, the SetCover- ε method was run on all instruments for tolerance values ranging from 0.5 mm to 3.0 mm. Two metrics were measured: the number of circles selected and the average error. The average maximum error and the maximum error are not considered here, as they are structurally determined by the candidate generation procedure: arcs are expanded point by point until they reach the tolerance limit, so the maximum error of any selected circle will always closely approach ε by construction. Just as in Chapter 3, every instrument is represented by a line, but these are not labeled as we are examining trends rather than the behaviour of specific instruments.

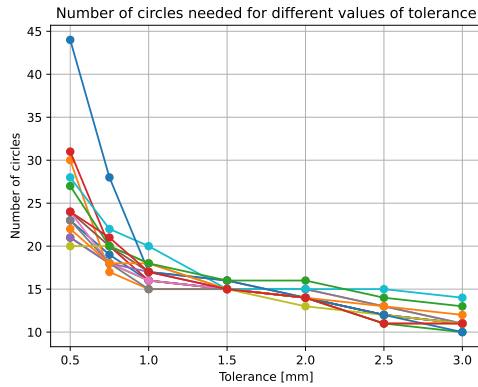


Figure 4.6: Effect of Tolerance on Number of Circles

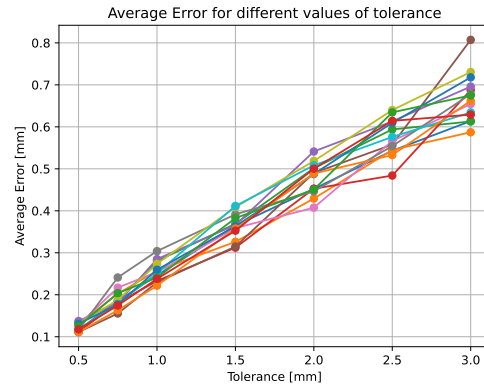


Figure 4.7: Effect of Tolerance on Average Error $\bar{\delta}$

As we can see in Figure 4.6, fewer circles are needed to cover the contour as the tolerance increases. This makes sense, since a looser tolerance allows each arc to stretch further before being rejected, so each circle ends up covering more points. The number of needed circles drops sharply between 0.5 and 1.0, after which the curves flatten out. Beyond a certain point (tolerance of 1.5mm or 2mm, depending on the instrument), relaxing the tolerance further has little additional effect on the number of circles. Most instruments stabilise in the range of 10 to 16 circles for large tolerances.

There are no noticeable trends with the average error study (Figure 4.7). Overall, average error grows steadily as tolerance increases, and does so in a remarkably consistent way across all instruments.

4.1.3 SetCover- ε : Summary

The SetCover- ε corrects the faults of the YangDu algorithm. By constructing a set of candidate circles and choosing among those, it avoids conditioning on the starting point. The solution is by construction guaranteed to be the best overall. However, this model has two main shortcomings. First, for each contour point, only the largest circle that stays within tolerance is kept as a candidate. This means that smaller, tighter-fitting

arcs centered on the same point are discarded entirely, even when they would produce a better geometric approximation. Second, minimizing the number of circles is not the most meaningful objective: two solutions using the same number of circles can differ greatly in how accurately they trace the contour, and the model has no way to prefer one over the other. To address both issues, we introduce a new model, referred to as *SetCover-K*.

4.2 Second optimization model: SetCover-K

In order to correct those issues, we extended the candidate circles set to include all intermediate circles. The objective function was also adapted to minimize the sum of the maximum error of each selected circle, rather than the number of circles. Figure 4.8 displays the flowchart of the new model, and Section 4.2.1 details the mathematical formulation of the model.

4.2.1 Formulation

The construction of the candidate set is extended so that, for each contour point p_i , every intermediate circle generated during the neighborhood expansion is retained. Concretely, for each center point $p_i \in \mathcal{P}$ and each half-width $j = 0, 1, \dots$, a circle is fitted by least squares to the sub-sequence of points centered on p_i . As long as the maximum fitting error stays below ε , the circle is added to the candidate set and the window is expanded further. The process stops when the error first exceeds ε .

The full candidate set is $\mathcal{C} = \{C_1, \dots, C_m\}$. Each candidate C_k is characterized by:

- a center $\mathbf{c}_k = (x_k^c, y_k^c)$ and radius r_k , obtained by least squares $k = 1, \dots, m$,
- the subset $\mathcal{P}_k \subseteq \mathcal{P}$ of contour points it covers,
- its maximum fitting error $e_k = \max_{p_i \in \mathcal{P}_k} \left| \|p_i - \mathbf{c}_k\|_2 - r_k \right|$.

The incidence matrix $A \in \{0, 1\}^{n \times m}$ encodes which circles cover which points:

$$A_{i,k} = \begin{cases} 1 & \text{if } p_i \in \mathcal{P}_k, \\ 0 & \text{otherwise.} \end{cases}$$

The errors e_k are stored separately and entered into the model as objective coefficients. The optimization model is:

$$\begin{aligned} \min_{x \in \{0,1\}^m} \quad & \sum_{k=1}^m e_k x_k \\ \text{s.t.} \quad & \sum_{k=1}^m A_{i,k} x_k \geq 1, \quad \forall i = 1, \dots, n \\ & \sum_{k=1}^m x_k \leq K \end{aligned} \tag{4.2}$$

where $x_k \in \{0, 1\}$ indicates whether candidate C_k is selected, and K is the maximum number of circles allowed. The objective minimizes the sum of maximum fitting errors

accumulated over all selected circles. The first constraint ensures that every contour point is covered by at least one selected circle. The second constraint fixes the maximum number of selected circles: without it, the model could trivially minimize the error by selecting a large number of very small circles, which would defeat the purpose of seeking a compact geometric representation. This problem is solved as an integer linear program using the Gurobi optimizer, and will be referred to as *SetCover-K* in this thesis. In this entire thesis, the maximum tolerance is set at $\varepsilon = 2.0\text{mm}$. This value was chosen as a reasonable balance between acceptable error, accounting for possible distortion or dents in the violin, and computational feasibility. With this tolerance, execution time ranges from 15 to 55 minutes, depending on the violin, running on the UCLouvain shared `lemaitre4` cluster. Increasing the tolerance would have resulted in significantly longer execution times, making this method impractical for analysis.

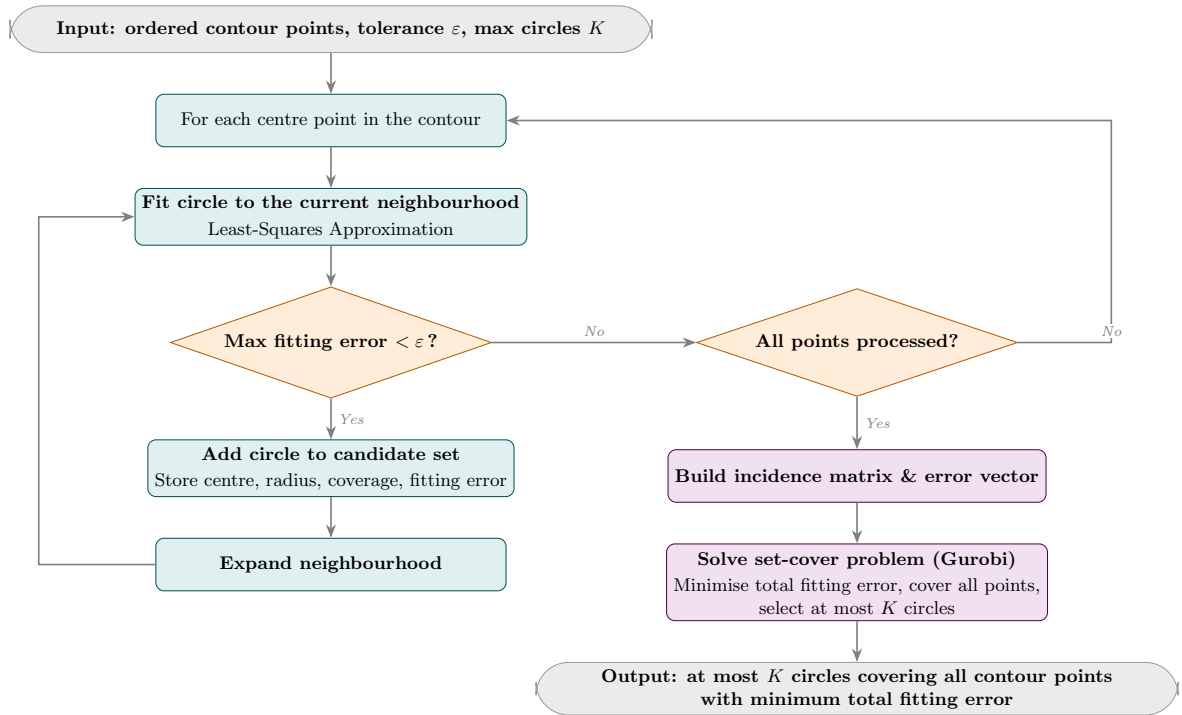


Figure 4.8: SetCover-K flowchart

4.2.2 Results and Limitations

Figure 4.9 shows the results of the SetCover-K method for violin 0233, with $K = 16$. The results are compelling: a very low average error $\bar{\delta}$ and average maximum error $\bar{\delta}_{\max}$. The circles barely overlap, and visually, the results are symmetric². It is worth noting that, while the results are very satisfying for violin 0233, the performance of SetCover-K depends heavily on the instrument. Some violins obtain asymmetrical results or large circles that only describe a small part of the outline. Some results select small circles inside the corners of the C-bouts, which display the wrong concavity but fit locally. All results are presented in Appendix D.

²A thorough study of symmetry can be found in Chapter 5.

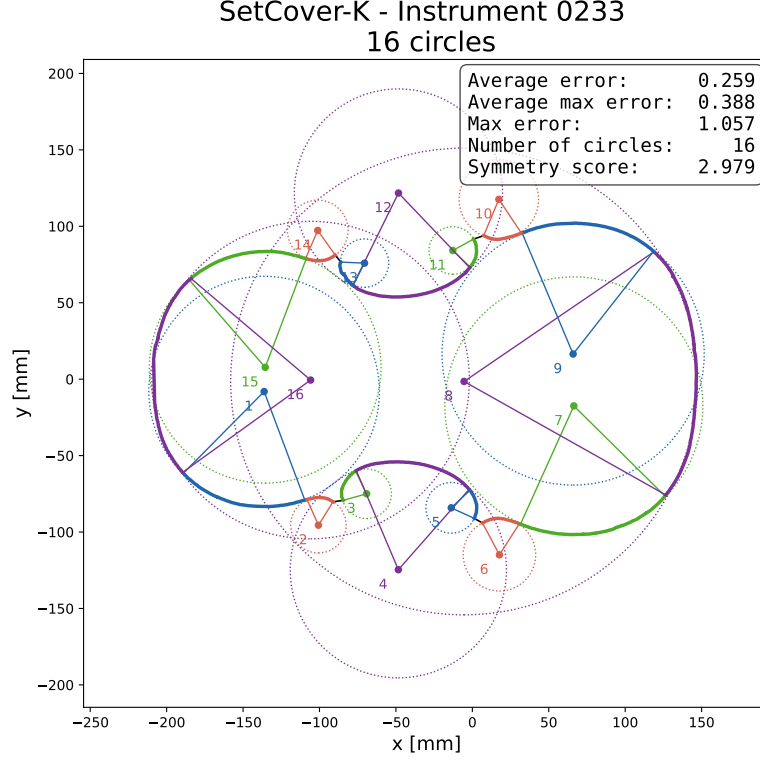


Figure 4.9: SetCover-K on Instrument 0233 – $K = 16$ and $\varepsilon = 2.0\text{mm}$

The SetCover-K method also performs almost perfectly for the Digital Amati (see Figure 4.10), even when raising the maximum tolerance to 2mm. Unlike the previous methods, which increase the circles as far as possible below the tolerance, SetCover-K focuses on the circles with the lowest error. This leads to a more accurate result despite the higher tolerance. This also explains that, for the first time, the method performs well on the Noisy Digital Amati (see Figure 4.11). For both, the errors remain negligible, and near-perfect symmetry is reached.

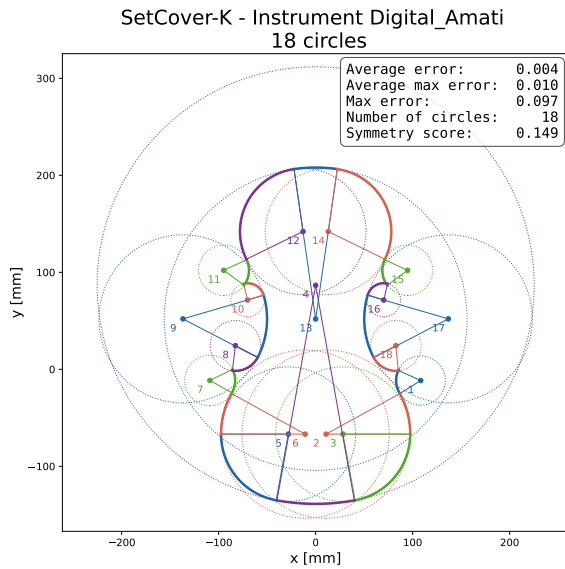


Figure 4.10: SetCover-K on Digital Amati – $K = 18$ and $\varepsilon = 2.0\text{mm}$

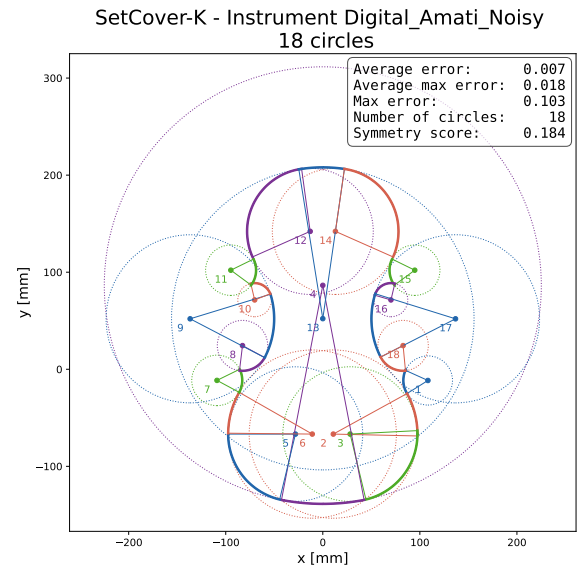


Figure 4.11: SetCover-K on Noisy Digital Amati – $K = 18$ and $\varepsilon = 2.0\text{mm}$

Number of circles study

Due to the execution time of each violin, it was impossible to compute an analysis for the number of circles for all violins at the same time. We will thus only show this analysis for violin 0233, all other parameter study graphs can be found in Appendix D. It is also possible to note that different luthiers may have different methods, and therefore use different numbers of circles. As we can see on Figure 4.12, there are no clear breakpoints either, and anywhere between 16 and 20 circles would be appropriate, with 16 and 17 circles still producing a large maximum error. Above 20 circles, the maximum error stops decreasing, and both the average error and average maximum error decrease at a much slower rate.

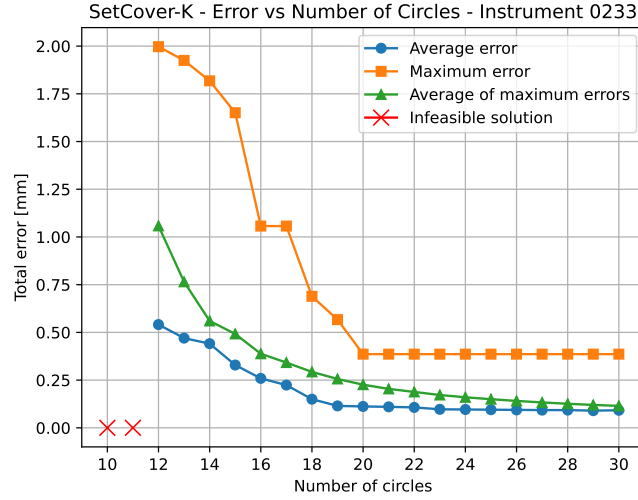


Figure 4.12: Evaluation metrics vs Number of circles – Instrument 0233

It is also worth noting that the model is feasible with 12 circles, which is four circles fewer than for SetCover- ε , which aimed at minimizing the number of circles. This is due to the increase of the maximum tolerance ε to 2.0mm (compared to 1.0mm for SetCover- ε). This might also be improved by the extension of the candidate set: with more candidate circles, the model can fit more closely to the outline, reducing the need for circles.

Qualitatively, above 20 circles, the additional circles are relatively small and come to correct locally, leaving the entire layout largely unchanged (see Figure 4.13 (b)). Here is highlighted circle 4, which is the circle that the model added when raising K to 22. As we can see, this circle comes to correct local distortion rather than improving the overall quality of the outline decomposition.

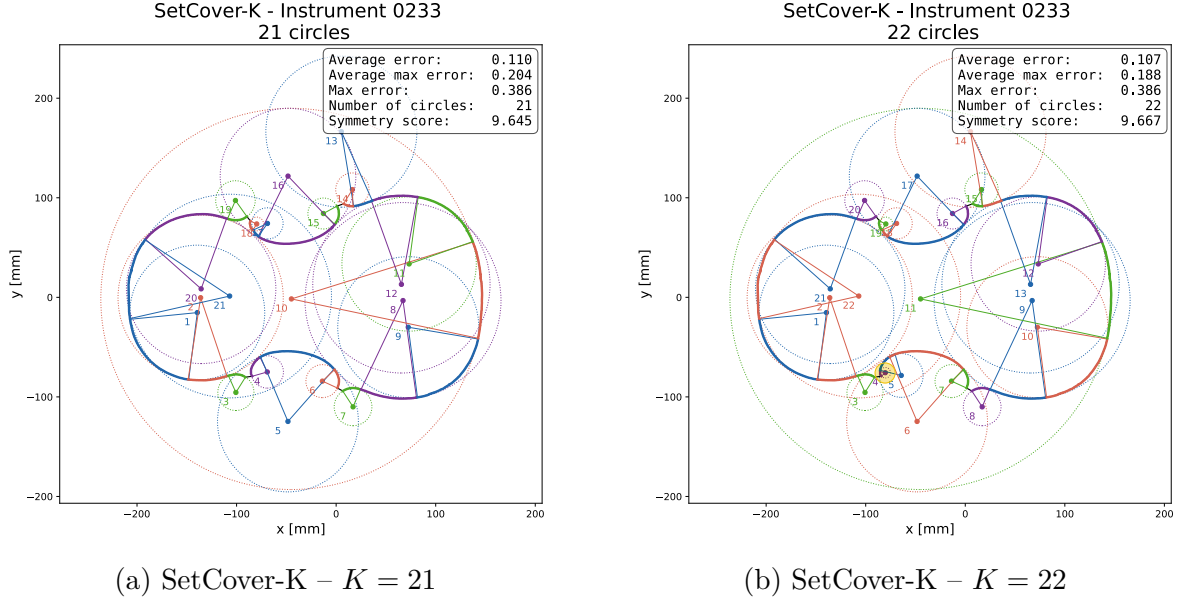


Figure 4.13: Effect of adding one circle when $K > 20$. Highlighted on (b) is the added circle when raising K from 21 to 22.

4.2.3 SetCover-K: Satisfying results

As we have seen, SetCover-K has promising results. For some violins, such as instrument 0233, the results are outstanding, and for others, the algorithm still performs reasonably well but leaves room for improvement³. Given these solid foundations, the next section will explore some adaptations of SetCover-K rather than starting from scratch with a completely different model.

4.3 Further Adaptations

4.3.1 Minimization of the Maximum Error

As explained in Section 4.2.1, SetCover-K minimizes the sum of all maximum errors of chosen circles for a pre-defined maximum number of circles and tolerance (which is equivalent to minimizing the average maximum error $\bar{\delta}_{\max}$). The following section explores the effect of replacing this objective with a different evaluation metric. Among the two possible evaluation metrics, the average error $\bar{\delta}$ and the overall maximum error $\delta_{\max}$, we prioritize the minimization of the overall maximum error. While the minimization of the average error would have produced interesting results, it is likely to produce a highly uneven fit: some circles approximating the arc very closely, others very poorly. The purpose of our problem is to ensure that all circles fit correctly to the outline of the violin, and the average error may mask some poorly fitted circles. This section will thus minimize the maximum error among the selected set.

³See Appendix D for all results.

Model Formulation

$$\begin{aligned}
& \min_{x \in \{0,1\}^m, \tau \in \mathbb{R}} \quad \tau \\
& \text{s.t.} \quad \sum_{k=1}^m A_{i,k} x_k \geq 1, \quad \forall i = 1, \dots, n \\
& \quad \quad \sum_{k=1}^m x_k \leq K \\
& \quad \quad e_k x_k \leq \tau, \quad \forall k = 1, \dots, m
\end{aligned} \tag{4.3}$$

The auxiliary variable $\tau \in \mathbb{R}$ represents an upper bound on the fitting error of any selected circle. The third constraint enforces that whenever a candidate C_k is selected (i.e. $x_k = 1$), its error e_k cannot exceed τ . Minimizing τ therefore drives the worst-case error among all chosen circles as low as possible. This new model will be referred to as *SetCover-KM* (M standing for Maximum Error). Similarly to SetCover-K, the maximum tolerance is fixed at $\varepsilon = 2.0\text{mm}$, to achieve a good trade-off between model size and flexibility to fit the data.

Results

Overall, the results are comparable to those of SetCover-K. The only metric that can be meaningfully compared between the two models is the average error $\bar{\delta}$, since the number of circles is fixed in both cases, and each model directly minimizes one of the other metrics (average maximum error or overall maximum error), which therefore trivially performs better in its respective model. We can also qualitatively appreciate which result feels more likely. Two results are presented below alongside the SetCover-K results for comparison: Figures 4.14 and 4.15 display instrument 0233, which has been the example used throughout this thesis, and Figures 4.16 and 4.17 show instrument 2792, for which SetCover-KM distinctively performs better. For both results presented in this section, $K = 16$ and $\varepsilon = 2.0\text{mm}$. All other results can be found in Appendix E.

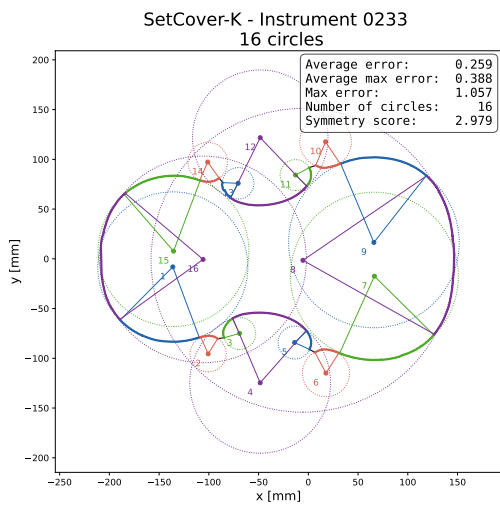


Figure 4.14: SetCover-K on Instrument 0233 – $K = 16$ and $\varepsilon = 2.0\text{mm}$

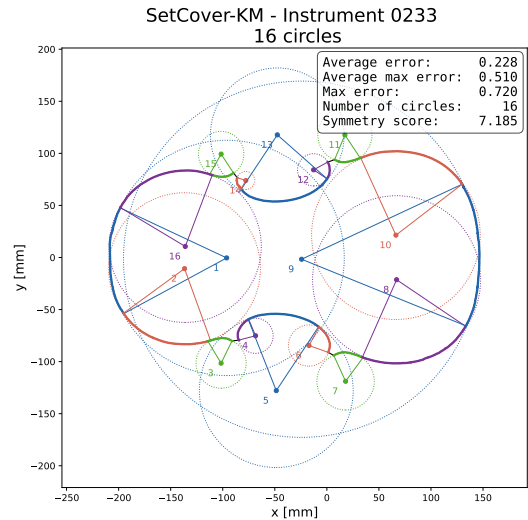


Figure 4.15: SetCover-KM on Instrument 0233 – $K = 16$ and $\varepsilon = 2.0\text{mm}$

As we can see in Figures 4.14 and 4.15, for Instrument 0233, the results are similar for both methods. Looking more closely, SetCover-KM performs better in terms of average error, but the difference is small (0.228 and 0.259). However, one might argue that SetCover-K is more symmetrical and therefore gives better results. A thorough study of symmetry can be found in Chapter 5.

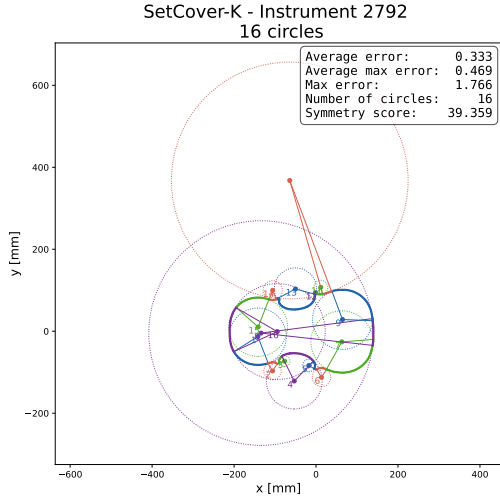


Figure 4.16: SetCover-K on Instrument 2792 – $K = 16$ and $\varepsilon = 2.0\text{mm}$

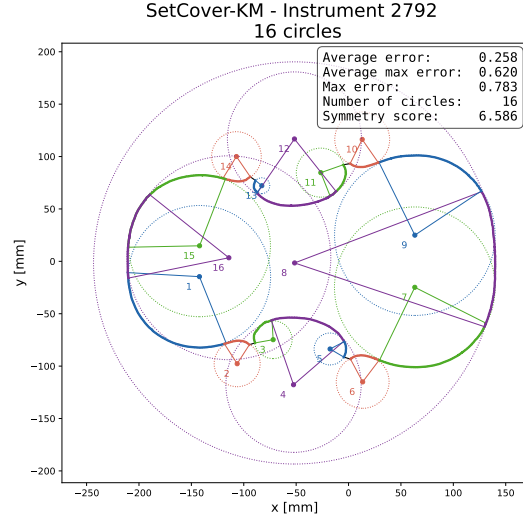


Figure 4.17: SetCover-KM on Instrument 2792 – $K = 16$ and $\varepsilon = 2.0\text{mm}$

For Instrument 2792, the advantage of SetCover-KM is clearer, as it opts not to choose circle 10 (orange), which is excessively bigger than the rest (see Figures 4.16 and 4.17). This is a trend observed in the entire corpus: overall, SetCover-KM tends to avoid bigger circles, leading to more consistent results. Differences persist, but results across all instruments are more similar.

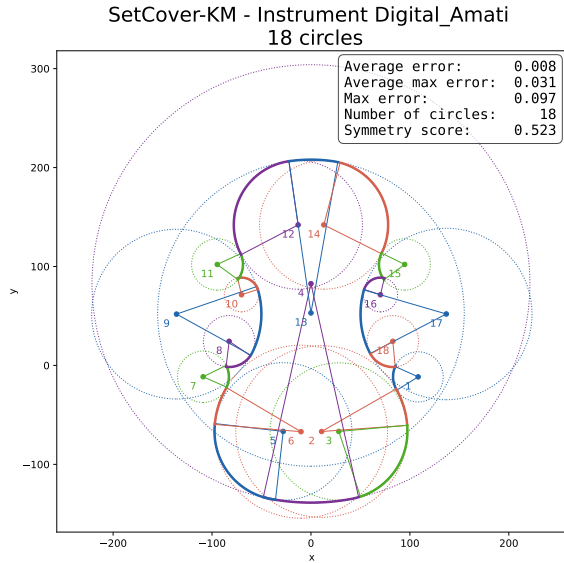


Figure 4.18: SetCover-KM on Digital Amati – $K = 18$ and $\varepsilon = 2.0\text{mm}$

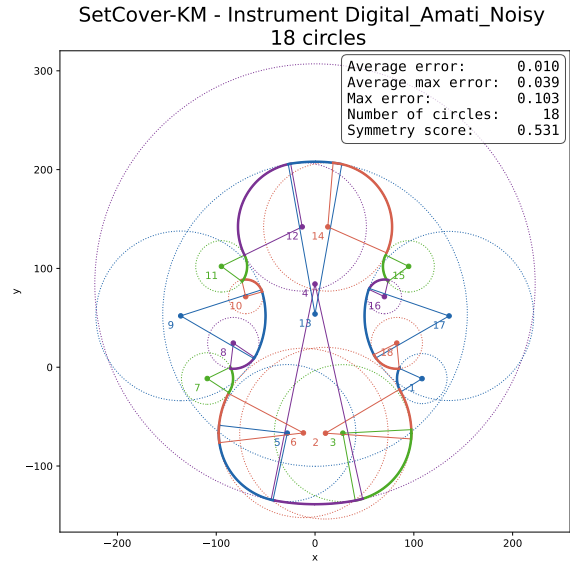


Figure 4.19: SetCover-KM on Noisy Digital Amati – $K = 18$ and $\varepsilon = 2.0\text{mm}$

A slight difference can be observed when it comes to the Digital Amati. While the

SetCover-K method performed perfectly, the result is slightly inferior for the SetCover-KM method: the circles overlap, the average error is doubled, and the average maximum error is tripled. Looking more closely, we can observe that both methods have exactly the same overall maximum error, meaning SetCover-K managed to minimize the maximum error as well. Once this value was reached, SetCover-KM did not attempt to improve the other evaluation metrics, leading to a poorer fit. The exact same observation happens for the Noisy Digital Amati: both methods reach the smallest overall maximum error, leading to a poorer fit by SetCover-KM, which does not attempt to correct other evaluation metrics.

Parameter Study

Similarly to SetCover-K, the most important parameter to tune is the number of circles the optimization model is allowed to select. However, unlike SetCover-K, the optimal number of circles for violin 0233 is clearer.

As we can see in Figure 4.20, all three metrics seem to present an elbow at 16 circles. It seems fair to assume that this would be the best trade-off between precision and the number of circles. Similar to SetCover-K, circles added above 20 present little interest, and provide local correction rather than large approximation.

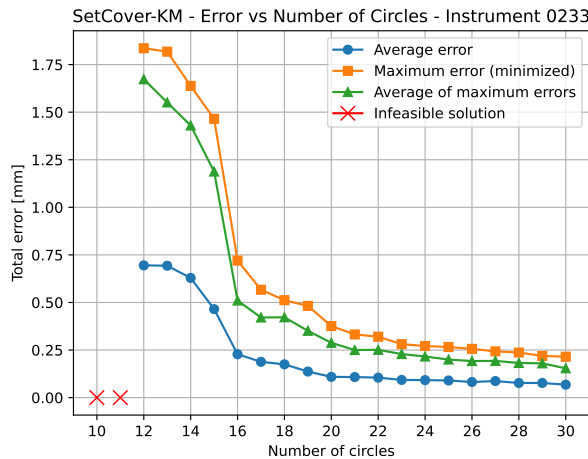


Figure 4.20: Evaluation metrics vs Number of circles - Instrument 0233

4.3.2 Relaxation: Allowing for skipped points

Another modification of the SetCover-K model we wanted to explore was to relax the model to allow for skipped points. This serves two purposes: first, it strengthens the model against outliers, such as the dent in instrument 2780, as can be seen in Figure 1.5. Secondly, it might help detect sections that were drawn freehand.

Formulation

Let $\mathcal{P} = \{p_1, \dots, p_n\}$ be the set of contour points, $\mathcal{C} = \{C_1, \dots, C_m\}$ the candidate circles, and $A \in \{0, 1\}^{n \times m}$ the incidence matrix defined as before. Let K be the maximum number of circles and $K_{\text{skip}} = \lfloor \alpha \cdot n \rfloor$ the maximum number of points that may be left uncovered, where $\alpha \in [0, 1]$ is a user-defined skip percentage. A percentage was chosen

rather than a fixed number to account for different sizes of outline⁴.

A second set of binary variables is introduced: $z_i \in \{0, 1\}$, where $z_i = 1$ indicates that point p_i is left uncovered. The model becomes:

$$\begin{aligned}
& \min_{x \in \{0,1\}^m, z \in \{0,1\}^n} \sum_{k=1}^m e_k x_k \\
& \text{s.t.} \quad \left(\sum_{k=1}^m A_{i,k} x_k \right) + z_i \geq 1, \quad \forall i = 1, \dots, n \\
& \quad \sum_{k=1}^m x_k \leq K \\
& \quad \sum_{i=1}^n z_i \leq K_{\text{skip}}
\end{aligned} \tag{4.4}$$

where $x_k \in \{0, 1\}$ indicates whether circle c_k is selected, and $z_i \in \{0, 1\}$ indicates whether point p_i is skipped. The objective still minimizes the sum of the maximum errors accumulated over all selected circles (like SetCover-K). The first constraint is the relaxed coverage condition: each point must either be covered by at least one selected circle or be explicitly flagged as skipped. Theoretically, a point could be both covered by a circle and flagged as skipped. However, if this solution is reached, it implies that skipping the point does not improve the objective function and thus does not affect the result. The second and third constraints limit the number of circles and the number of skipped points, respectively. This model will be referred to as *SetCover-KS* (S standing for Skipped Points).

Results

Figure 4.21 shows the results of SetCover-KS for violin 0233, with $K = 16$ and $\alpha = 0.05$. The skipped points concentrate almost exclusively at the corners of the outline, and in particular around the damaged upper-left corner between circles 13 and 14. This makes sense: the contour is locally unreliable in these regions due to physical damage, and the model correctly identifies these points as outliers rather than forcing a poorly-fitting circle through them. The fact that the upper right corner, which is undamaged, is not skipped further supports this interpretation. This suggests that the model is responding to actual noise in the contour rather than to corners being geometrically hard to fit in general.

⁴Depending on the instrument, outlines range from 1500 to 2800 coordinate points.

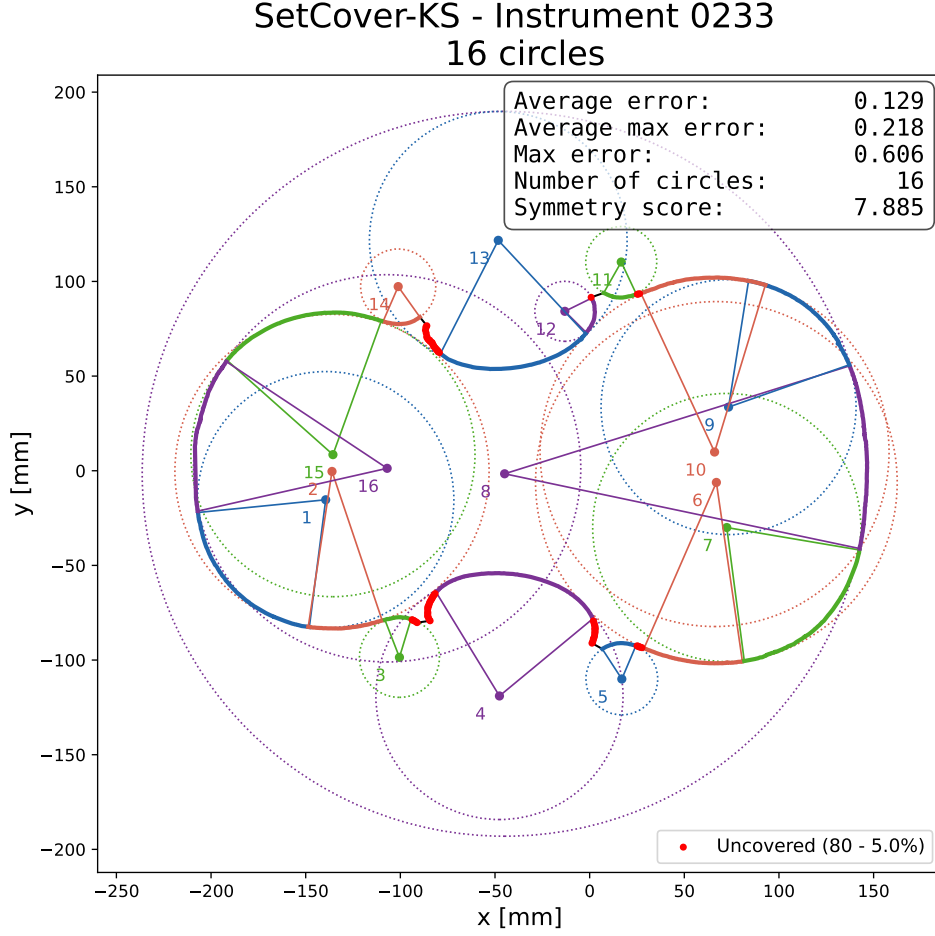


Figure 4.21: SetCover-KS for violin 0233 – $K = 16$, $\alpha = 0.05$ and $\varepsilon = 2.0\text{mm}$

The choice of α is, however, not straightforward. If used for outlier detection, it depends on the specificities of each violin outline. If chosen too large, the model starts skipping perfectly valid points simply because doing so reduces the objective function. Ideally, a parameter study similar to those conducted for the other methods would help determine a good value, but this is not feasible here: each run takes approximately one hour per violin and set of parameters, making any systematic exploration impractical. The choice of α is therefore left to the user, based on a visual inspection of the results.

SetCover-KS shows promising behaviour on damaged instruments, and the skip mechanism works as intended. The main limitation is not the method itself but the computational cost, which makes it difficult to use in practice without a good initial guess for α .

4.4 Set Covering Approach: Summary

This chapter traced the evolution of our optimization methods to approximate violin outlines using circles.

The first global approach we considered was SetCover- ε . It generates one candidate circle per contour point: the largest circle centered on that point that still stays within a fixed tolerance ε . Then, it solves a set covering integer program to find the minimum

number of circles that together cover every point. This was an improvement over YangDu, producing fewer circles and a lower average error. However, it had two flaws: only the maximal circle per point is kept as a candidate (while better-fitting smaller arcs are disregarded), and minimizing the number of circles is a poor objective: two solutions with the same count can differ greatly in accuracy.

SetCover-K addressed both problems. The candidate set is expanded to include every intermediate circle generated during the neighborhood expansion (instead of only the largest), and the objective is changed to minimize the sum of maximum fitting errors across selected circles. A constraint K is added to the total number of circles. This produced better results, with circles that barely overlap and near-perfect symmetry on some instruments. The tradeoff is computational cost: runs take 15-45 minutes per violin on a computing cluster.

SetCover-KM is a variant that minimizes the overall maximum error rather than the sum of errors. It tends to avoid large circles, producing more uniform results across the corpus, but performs slightly worse on some well-conditioned instruments such as the Digital Amati.

SetCover-KS introduces slack variables allowing a small fraction α of contour points to be left uncovered. This makes the model robust to damage and outliers. The downside is high computational cost, making parameter tuning for α impractical.

SetCover-K and SetCover-KM give very promising results, which will be studied thoroughly in Chapter 6. SetCover-KS is also promising, but its high computational cost makes a thorough study of the results impossible. Table 4.1 summarizes the characteristics, strengths, and weaknesses of each method.

Model	Candidate circles	Objective	Constraints	Key strengths	Key limitations
SetCover- ε	One per point (largest within ε)	Min number of circles	All points covered	Fast	Discards smaller arcs, many equivalent solutions
SetCover-K	All intermediate circles within ε	Min sum of max errors	All points covered; $\leq K$ circles	Minimal overlap, near-perfect symmetry on some instruments	Execution time, K must be chosen, inconsistent results across the corpus
SetCover-KM	All intermediate circles within ε	Min worst-case error	All points covered; $\leq K$ circles	More uniform results across the corpus	Execution time, K must be chosen, does not improve other metrics once min max error is reached
SetCover-KS	All intermediate circles within ε	Min sum of max errors	All points covered or skipped; $\leq K$ circles; $\leq \lfloor \alpha \cdot n \rfloor$ skips	Robust to damage and outliers, skipped points localize problematic regions	K must be chosen, approx. one hour per run; α must be chosen; over-skipping if α too large

Table 4.1: Summary of set covering methods

Chapter 5

A Study of Symmetry

The consideration of symmetry as an evaluation metric is complex. From a historical perspective, violins were almost certainly designed with symmetry in mind. However, damage and distortion accumulated over time may have affected the instrument asymmetrically. It is also said that some luthiers, with age and experience, begin to take greater liberties in their craft, introducing subtle imperfections that lend the instrument character, making it less perfect geometrically. Nevertheless, reaching approximate symmetry seems like a reasonable criterion to assess the historical likelihood of results.

Up to this point, nothing in our methods encouraged the selection of circles set symmetrically on the contour, and we did not have any quantitative tools to measure symmetry beyond qualitative observation. That said, we observed that for the Digital Amati, a violin theoretically constructed from symmetric circles, the results of SetCover-K are nearly perfectly symmetric (see Figure 4.10). SetCover-KM produced satisfying results as well, approaching but not quite reaching perfect symmetry. SetCover- ε , with a low tolerance, produced good results, though increasing the tolerance led to a general deterioration in quality, symmetry included. It therefore appears that our methods are capable of recovering approximate symmetry when it is structurally inherent to the instrument. This chapter will delve into symmetry more thoroughly, first with the development of a symmetry score, and later with an attempt to enforce symmetry rather than just observe it.

5.1 Development of a symmetry score

The purpose of the symmetry score is to provide an objective measure of observable symmetry, one that reflects what can be visually appreciated without overstating the impact of minor imperfections. In particular, an isolated misplaced circle should contribute only marginally to the score when the overall configuration remains largely symmetric. It is computed in the following way (see Figure 5.1 for illustration):

Let $\mathcal{C}^* = \{C_1^*, C_2^*, \dots, C_{K'}^*\}$ be the set of K' selected circles to approximate a violin outline, where $K' \leq K$ and each circle C_k^* is characterised by a centre $\mathbf{c}_k = (x_k^c, y_k^c) \in \mathbb{R}^2$ and a radius $r_k \in \mathbb{R}^+$.

Symmetry is assessed with respect to the x-axis via a reflection operator $\mathcal{R}$, defined as

$$\mathcal{R}(x, y) = (x, -y)$$

as can be seen in Figure 5.1 (b), where $\mathcal{R}$ maps each circle to its mirror across the x-axis. The dissimilarity between two circles C_k^* and C_j^* , measuring how closely C_j^* mirrors C_k^* across the chosen axis, is defined as

$$\text{dis}(C_k^*, C_j^*) = \|\mathbf{c}_k - \mathcal{R}(\mathbf{c}_j)\|_2 + |r_k - r_j|$$

The first term penalises positional error: the distance between C_k^* 's centre and the reflection of C_j^* 's centre, and the second penalises any difference in radii. Each circle C_k^* is assigned the dissimilarity of its closest match in $\mathcal{C}^*$ (see Figure 5.1 (c)).

$$s_k = \min_{j \in \{1, \dots, K'\}} \text{dis}(C_k^*, C_j^*)$$

The symmetry score is then the mean of these best-match residuals:

$$S(\mathcal{C}^*) = \frac{1}{K'} \sum_{k=1}^{K'} s_k = \frac{1}{K'} \sum_{k=1}^{K'} \min_{j \in \{1, \dots, K'\}} \left(\|\mathbf{c}_k - \mathcal{R}(\mathbf{c}_j)\|_2 + |r_k - r_j| \right) \quad (5.1)$$

A score of $S = 0$ indicates perfect symmetry; larger values reflect the average residual error after each circle has been matched as favourably as possible. It is worth noting that each circle is its own candidate when reflected. This is intentional: a circle whose center lies close to the axis of symmetry (the x-axis) is essentially its own mirror image, and should therefore be allowed to match with itself when reflected (see Figure 5.1).

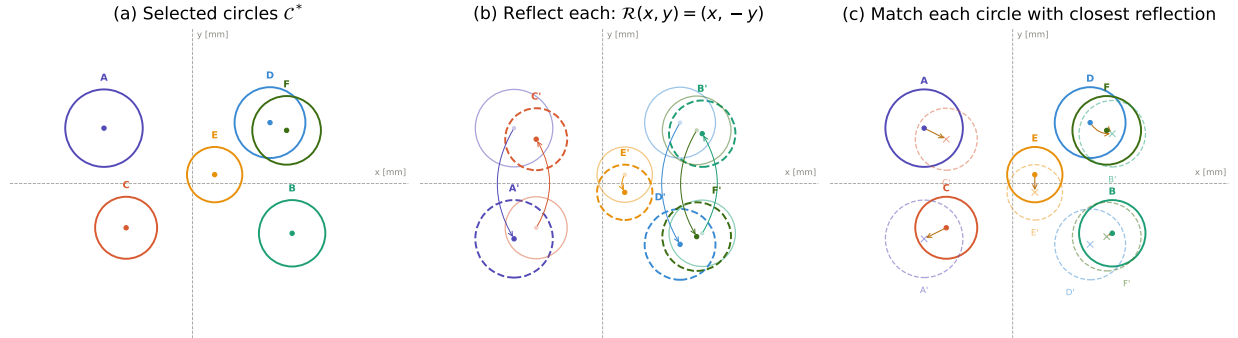


Figure 5.1: Illustration of the symmetry score computation. Distance is computed as $\|\mathbf{c}_k - \mathcal{R}(\mathbf{c}_j)\|_2 + |r_k - r_j|$, and the smallest distance for each circle is recorded as its score. Symmetry score is the mean of these minimum distances across all circles. Circles can be matched with themselves (see circle E), and several circles can be matched with the same reflected circle (see circles D and F with B').

The symmetry score is thus expressed in millimeters. As such, it is relative to the size of the instruments, and symmetry scores are not comparable across the corpus of instruments, but should be used to compare two methods for a given instrument.

5.2 Symmetry score study

In the corpus, and for $K = 16$, the symmetry scores vary between 2.98 mm and 41.32 mm for method SetCover-K and between 4.23 mm and 19.78 mm for SetCover-KM. For illustration, Figure 5.2 displays an example of a low symmetry score with instrument 0233, and Figure 5.3 displays a high symmetry score with instrument 2847.

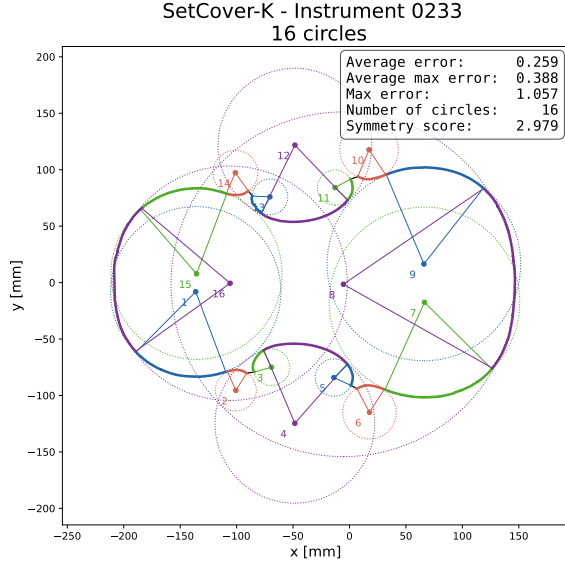


Figure 5.2: SetCover-K Instrument 0233 – $K=16$ – Symmetry score = 2.98mm

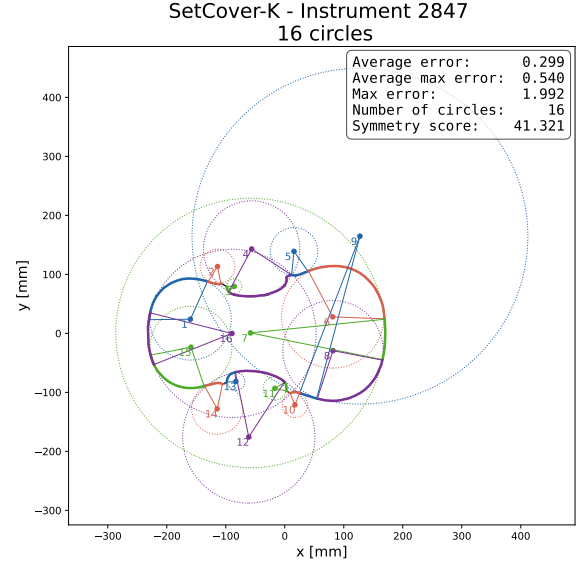


Figure 5.3: SetCover-K Instrument 2847 – $K=16$ – Symmetry score = 41.32mm

This section explores the effect of tuning the number of circles K on the symmetry score for methods SetCover-K and SetCover-KM, which were the most promising ones, as developed in Chapter 4. As mentioned in Section 4.3.2, method SetCover-KS is not thoroughly explored due to its long execution time. The results below are shown on violin 0233, as it was the recurring example used throughout this thesis. All other symmetry score analyses can be found in Appendices D for SetCover-K and E for SetCover-KM.

5.2.1 SetCover-K

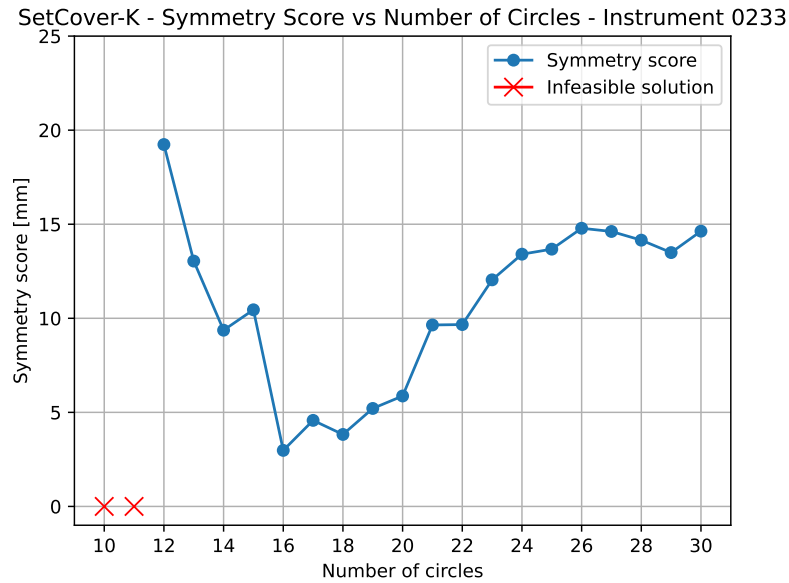


Figure 5.4: Analysis of the effect of the number of circles on the symmetry score – Instrument 0233 – SetCover-K

The symmetry analysis of SetCover-K (Figure 5.4) already gives a clear trend: one clear trough at $K=16$. This is revealing: below that, the model mostly tries to cover all points, as K is very limited. For bigger values $K > 21$, the model mostly adds small circles where the violin outline is more or less smooth, such as the corners, and where the circles thus have bigger errors. This mostly depends on the damage to the outline itself, which is not necessarily symmetrical. However, such a minimum for $K = 16$ does suggest a clear underlying structure. As mentioned before, the symmetry score should not be the only metric considered for parameter tuning, but in this case, it seems to confirm that the optimal value lies around 16-18 circles.

5.2.2 SetCover-KM

A similar trend is also visible for SetCover-KM in Figure 5.5, although not as clear. The symmetry score drops until $K = 17$, then oscillates between $K = 17$ and $K = 21$, and rises again when the model tries to correct local outliers rather than fit to the whole shape. The pattern is less clear, likely due to the nature of the objective function. At each step, the model will get rid of the circle with the biggest error, and replace it with two circles with lower maximum error and encompassing the same points (if possible). This might blur the underlying structure.

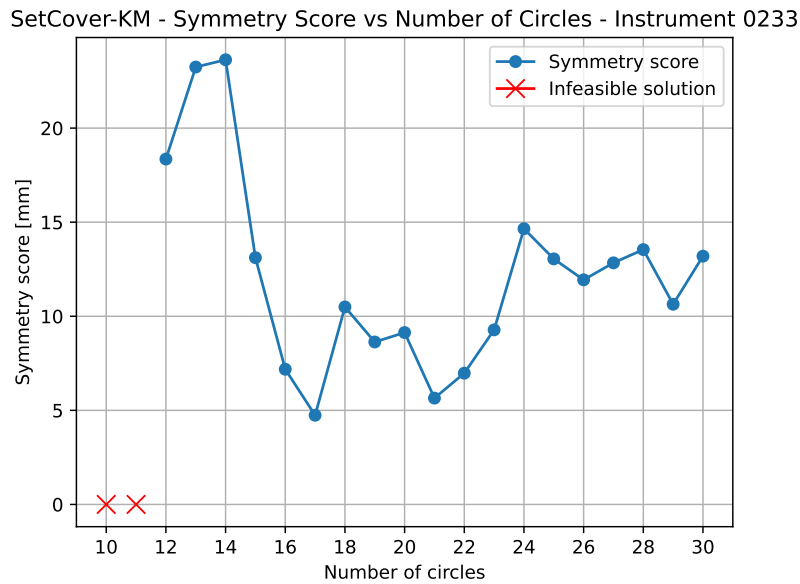


Figure 5.5: Analysis of the effect of the number of circles on the symmetry score – Instrument 0233 – SetCover-KM

5.3 Enforcing symmetry

Using symmetry as an evaluation metric already gives insight and comparison between the different methods and parameter choices. One step further is to enforce symmetry directly within the construction of the model, meaning forcing the algorithm to select symmetrical circles. In the first stages of this thesis, we attempted to enforce symmetry to the YangDu Algorithm, without much success. These attempts can be found in Appendix B.

Overall, enforcing symmetry is a tricky task. A possibility is to fit circles to one half, and to simply ensure that those circles fit within tolerance on the other half. That approach leads to two problems: first, this is not really finding a symmetrical solution, as it will always fit one side, and simply be acceptable for the other. The second problem is that it leads to an increase in the number of circles. Indeed, as the circles fit one side, they are more likely to be a wrong fit to the other side than if they were fitted to both sides directly.

That led to one final solution: generating candidate circles that fit both sides at the same time. To achieve that, the part of the outline lying below the x-axis was flipped, and both halves of the contour were intertwined to achieve one merged half-contour containing points from both halves. The SetCover-K method was applied to this merged contour. That way, candidate circles are generated, fitting points from both halves.

5.3.1 Formulation

Let $\mathcal{P}_{\text{fold}} = \{p_1, \dots, p_n\}$ is the set of folded contour points and $\mathcal{C} = \{C_1, \dots, C_m\}$ the set of candidate circles, with $A \in \{0, 1\}^{n \times m}$ the incidence matrix defined as before.

Each circle is classified according to the position of its center. Let

$$\mathcal{C}_{\text{axis}} = \{C_k \in \mathcal{C} : |y_k^c| < \tau_y\}, \quad \mathcal{C}_{\text{off}} = \mathcal{C} \setminus \mathcal{C}_{\text{axis}},$$

where y_k^c denotes the y -coordinate of the center of C_k and $\tau_y > 0$ is a user-defined tolerance. Circles in $\mathcal{C}_{\text{axis}}$ lie on the axis of symmetry and correspond to a single circle on the plate, whereas circles in $\mathcal{C}_{\text{off}}$ correspond to two circles (themselves and their reflection). A weight w_k is therefore attached to each circle:

$$w_k = \begin{cases} 1 & \text{if } C_k \in \mathcal{C}_{\text{axis}}, \\ 2 & \text{if } C_k \in \mathcal{C}_{\text{off}}. \end{cases}$$

The model becomes:

$$\begin{aligned} \min_{x \in \{0,1\}^m} \quad & \sum_{k=1}^m w_k e_k x_k \\ \text{s.t.} \quad & \sum_{k=1}^m A_{i,k} x_k \geq 1, \quad \forall i = 1, \dots, n, \\ & \sum_{k=1}^m w_k x_k \leq K, \end{aligned} \tag{5.2}$$

where $x_k \in \{0, 1\}$ indicates whether circle C_k is selected, and K represents the maximum number of circles in total (themselves and reflected). An off-axis circle, therefore, consumes two units of the budget K and contributes twice its fitting error e_k to the objective function. This model is solved with the Gurobi optimizer and will be referred to as *SetCover-KSym* in this thesis. Like in models SetCover-K and Set-Cover-KM, the maximum tolerance ε is set at 2.0mm.

Figure 5.6 illustrates the process of creating the merged contour and the creation of the candidate set of circles. Circles centered on the x-axis and encompassing parts of the

outline were manually added to allow the method to select circles that did not need a symmetrical counterpart.

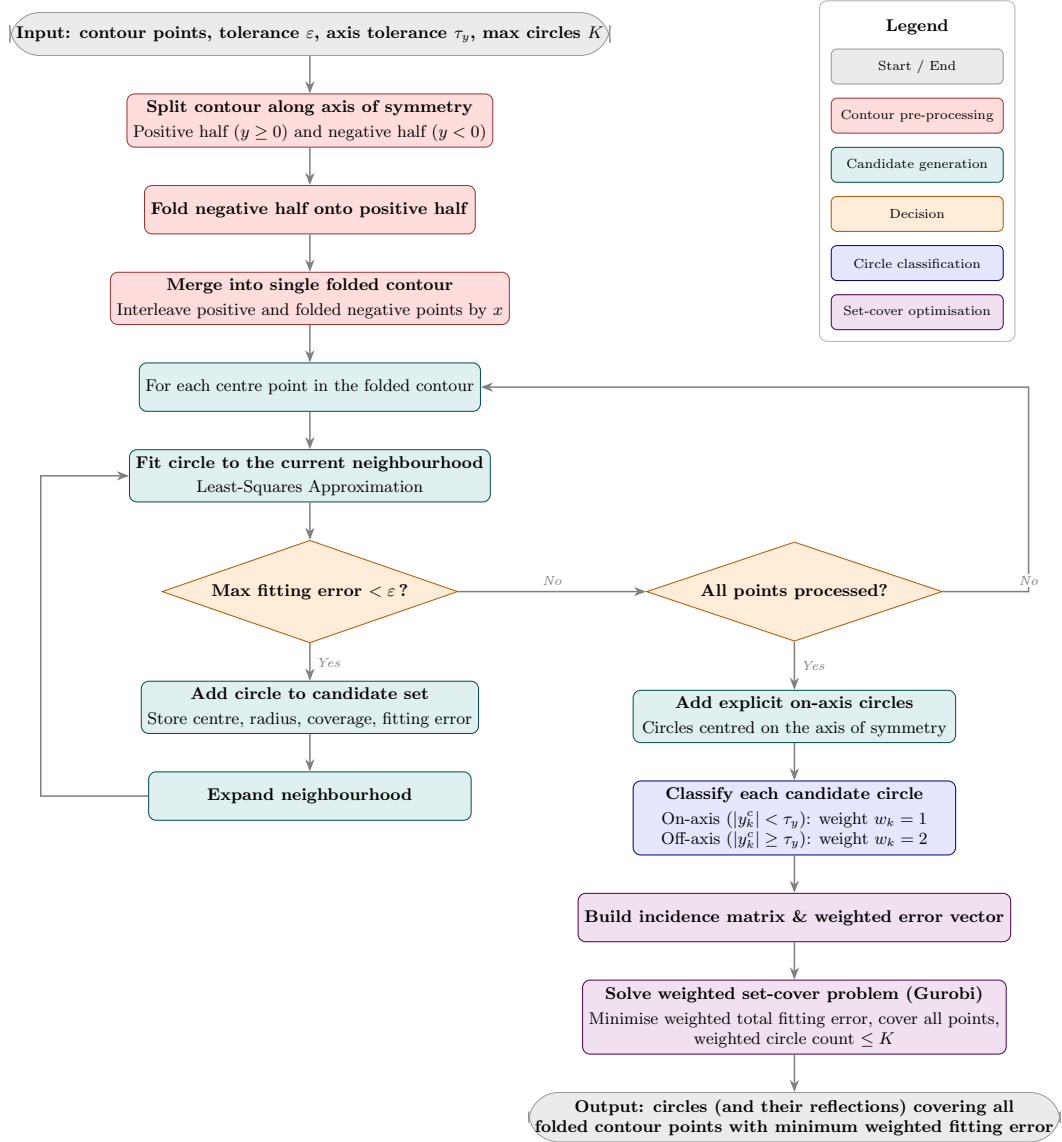


Figure 5.6: SetCover-KSym – Flowchart

5.3.2 Results

The first observation is that this model is infeasible for most of the violin corpus with $K=18$, and only becomes feasible for K values reaching 24 to 26. For those, the result is cluttered, with many circles overlapping. Circles fit respectively one half of the violin or the other, probably due to the inherent asymmetry of the violin (see Figure 5.7).

As an illustration, Figure 5.7 displays the result for violin 2771, with $K=26$ circles (which was the first feasible solution), and those results show large overlaps between the circles. Circles cannot fit both sides at the same time, so they are doubled to make up for that difference. It is interesting to note that violin 2771 performed reasonably well in terms of symmetry score (6.853 for SetCover-K with $K=18$, as can be seen in Appendix D).

This thus shows the weakness of enforcing symmetry: it is highly sensitive to (even slight) asymmetry. The attempt of folding the contour points does present one more possibility, which was not highlighted until then: the idea that one circle encompasses both the top and the bottom of the violin (as we can see for violin 2771 on Figure 5.7). The maximum error $\delta_{\max}$ exceeds the tolerance threshold due to the tolerance applied to axis-aligned circles. Specifically, any circle whose center lies within a distance $\tau_y = 1.5$ mm of the x -axis is considered on-axis. As a result, an on-axis circle whose center sits up to 1.5 mm away from the axis will be farther from the opposite side than the maximum tolerance $\varepsilon=2.0$ mm.

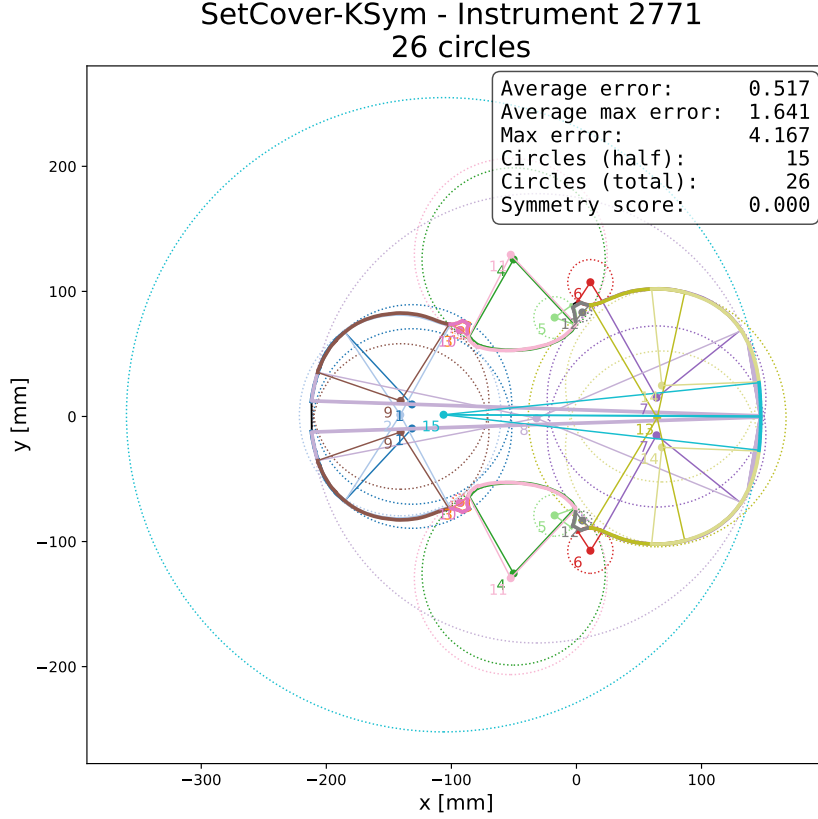


Figure 5.7: SetCover-KSym Instrument 2771 – $K = 26$, $\varepsilon = 2.0$ mm and $\tau_y=1.5$ mm

However, for violins 0233 and 2792, a solution was reached with 18 circles. Figure 5.8 displays the results for violin 0233. As we can see, all evaluation metrics besides the symmetry score are doubled when enforcing symmetry. It is also worth noting that the result differs from the SetCover-K (symmetry not enforced) result, which had a good symmetry score (3.826mm for $K = 18$) and appeared sufficiently symmetric. It seems like the SetCover-KSym model thus does not correct the symmetry found by the SetCover-K model with $K = 18$ (see Appendix D), but finds a different solution. It also seems heavily conditioned by local distortions, such as the one visible on the corners covered by circles 5 and 6. Overall, while it seems interesting to notice that the symmetry can be enforced, it is noticed that it produces interesting results only when the instrument already shows a remarkable symmetry. Even in those cases, the method fails to build upon the near-symmetry identified by the standard model, producing an unrelated solution rather than a corrected one. As such, this method is not recommended: it does not produce satisfying results for most instruments and focuses on local distortions rather than general

structure.

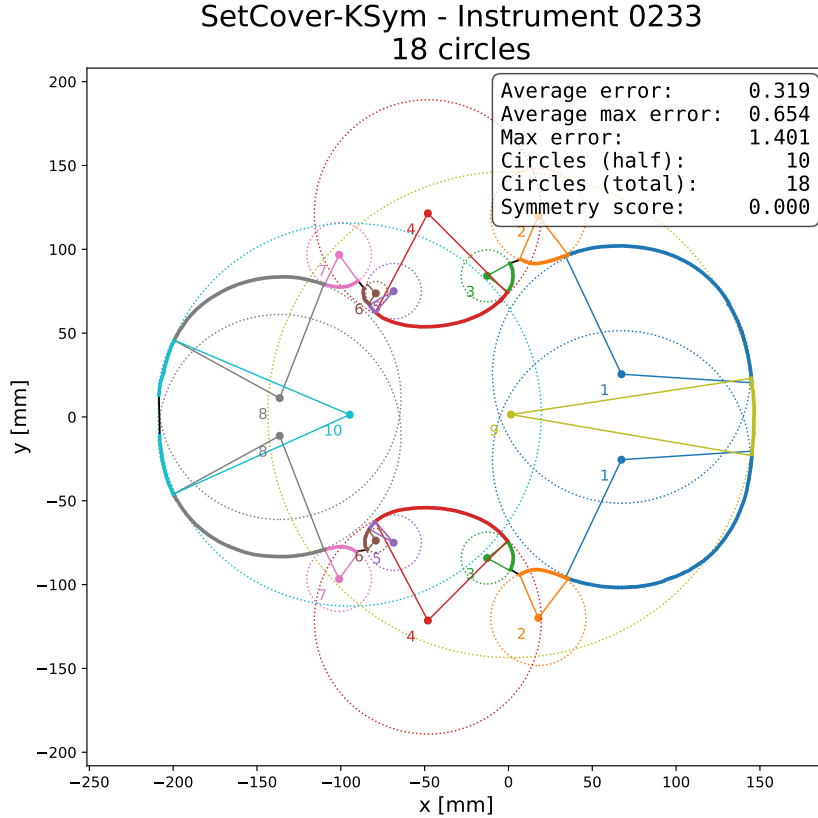


Figure 5.8: SetCover-KSym Instrument 0233 – $K = 18$, $\varepsilon = 2.0\text{mm}$ and $\tau_y = 1.5\text{mm}$

5.4 Symmetry Study: Summary

As explored in Section 5.1, we developed a symmetry score attempting to quantify the observable symmetry of our solutions. While this should not be the only evaluation metric considered, as it is heavily influenced by the symmetry of the instrument itself, it can help guide the choice of an optimal parameter.

An attempt to enforce symmetry within the method itself was unfruitful. The model struggled to find feasible solutions for most violins, and when it did, the results were cluttered and not particularly convincing: the overlapping circles were more a symptom of the violin’s own asymmetry or damage. Perhaps more surprisingly, it didn’t even improve on the solutions already found by SetCover-K, it just found different ones. The takeaway is that symmetry is most useful as a way to read and compare results, not as something to enforce directly into the optimisation. Given the infeasibility of the model without vastly overlapping circles, SetCover-KM was not attempted.

Chapter 6

Results and Discussion

This chapter presents and discusses the results obtained for methods SetCover-K and SetCover-KM, which were identified in Chapter 4 as the most promising. The YangDu, SetCover- ε , and SetCover-KSym methods are excluded from this comparison, for the intrinsic weaknesses identified in Chapters 3, 4, and 5, respectively. Method SetCover-KS is also discarded as its execution time prevents parameter study and corpus-wide evaluation necessary for a fair comparison. Two values of K are considered throughout: $K = 16$ and $K = 18$. While neither method displays a clear optimal value K , both 16 and 18 circles seem reasonable choices in terms of approximation error and symmetry score.

The comparison is structured in three parts. We first evaluate the methods on the Digital Amati, a synthetic violin outline constructed from a known set of circles, which provides a reference with a ground truth. We then confront these observations on the historical violin corpus. This part will compare evaluation metrics between both methods, but also examine recurring circle patterns, symmetry, and the proportionality of circle radii. Finally, we briefly consider the case of reduced violins, as introduced in Chapter 1, and discuss whether the reduction is perceptible from the approximated circles.

It should be highlighted that the methods are heavily guided: the number of circles K is fixed in advance, the candidate circle set is constructed from least square approximation, within a pre-defined tolerance, and the objective function focuses on one specific interpretation of approximation quality. While the global optimization model guarantees the best solution within this defined framework, the results are not unique in a broader sense: modifying the candidate set construction, the constraints, or the objective function would produce different (and potentially equally valid) results. The solutions presented here should therefore be interpreted as one possible optimal decomposition, rather than the reconstruction of a unique historical truth.

6.1 Comparison of methods

6.1.1 Digital Amati

A fair way to compare the methods is to do so on the Digital Amati, a synthetic instrument constructed from 18 circles. As such, we will only attempt our methods with K set at 18. This has been touched on in Chapter 4. The comparison of the results on the Digital Amati highlights one limitation of the SetCover-KM model: once it reaches

the minimum error, it does not attempt to reduce the other evaluation metrics. This is what occurs for the Digital Amati's: both methods reach the minimal maximum error, and the SetCover-K method continues to reduce the sum of all maximum errors, while SetCover-KM stops (see Figures 6.1 and 6.2). This leads to a more precise result with SetCover-K, where circles overlap less, and the evaluation metrics are lower. The same occurrence is observed on the Noisy Digital Amati, where both methods show robustness to the noise and give similar results (see Figures 6.3 and 6.4).

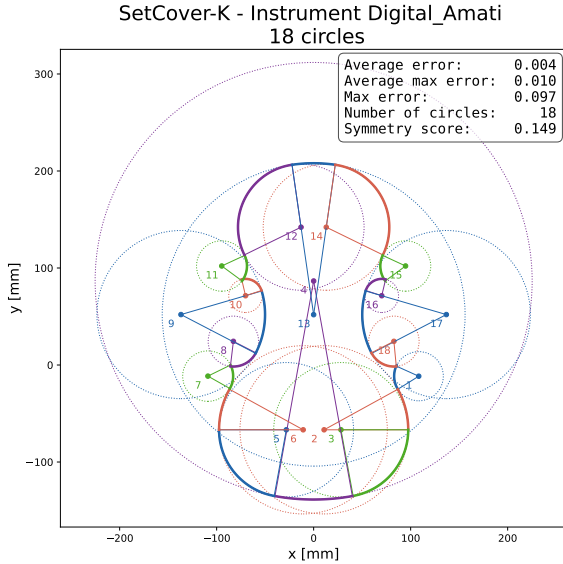


Figure 6.1: SetCover-K with $K=18$ – Digital Amati

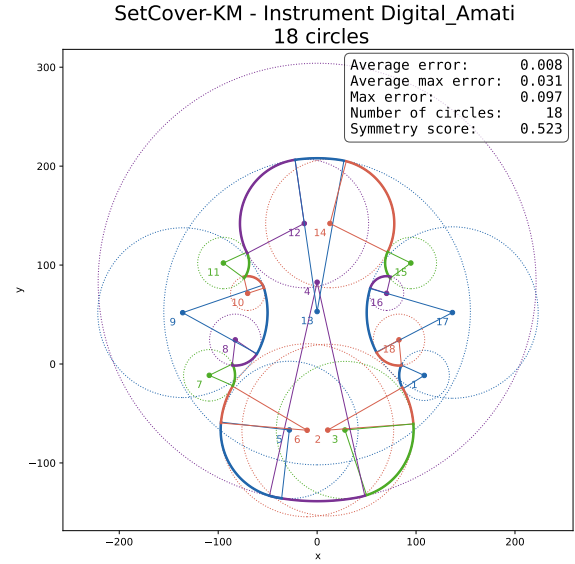


Figure 6.2: SetCover-KM with $K=18$ – Digital Amati

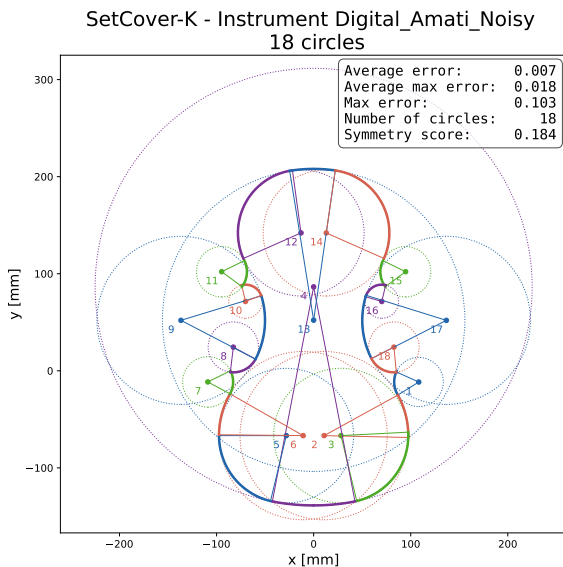


Figure 6.3: SetCover-KM with $K=18$ – Noisy Digital Amati

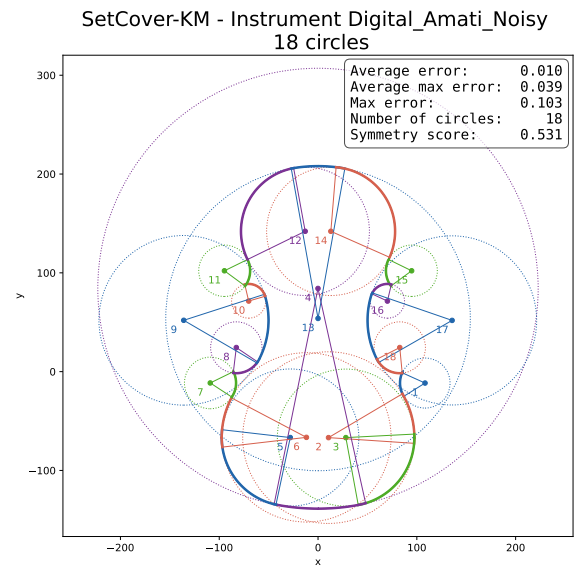


Figure 6.4: SetCover-KM with $K=18$ – Noisy Digital Amati

The Digital Amati is very useful to objectify the comparison of the methods, as we know the expected output. However, such a perfectly constructed violin is not representative of

the corpus we are working with. Furthermore, in the case of the Digital Amati, we know how many circles were used. For historical violins, we can only guess from qualitative observations and parameter study. Therefore, while the Digital Amati is a useful reference to discard poorly performing methods, it does not reflect the complexity and irregularity of real working conditions sufficiently to differentiate between methods that both perform well. Since both SetCover-K and SetCover-KM produce satisfactory results on the Digital Amati, the historical violin corpus is a necessary ground for a more meaningful comparison.

6.1.2 Historical Violin Corpus

The following section compares the results obtained by SetCover-K and SetCover-KM on the historical violin corpus, for $K = 16$ and $K = 18$. The first thing to note is that not all violins are equal in terms of approximation quality: some instruments produce consistently better results across all methods, while others perform more poorly. This is illustrated in Figure 6.5, where the medians of the four evaluation metrics over all methods and parameters are presented for each instrument. The median was chosen as it is robust against outliers and will not be skewed if one specific method performs poorly. Violins 0233, 1355, 2771, and 2772 are consistently well-approximated, while violins 2795, 2829, 2833, and 2836 prove more difficult. It is worth noting that instruments 2829, 2833, 2835, 2836, and 2847 are violas, thus larger-sized than the violins. This may cause more important distortions, leading to worse evaluation metrics. The ill-conditioned violins will not be excluded from the analysis, as we are trying to assess the best method over the entire corpus, despite the possible distortions. Nevertheless, these differences in instrument condition should be kept in mind when interpreting the results.

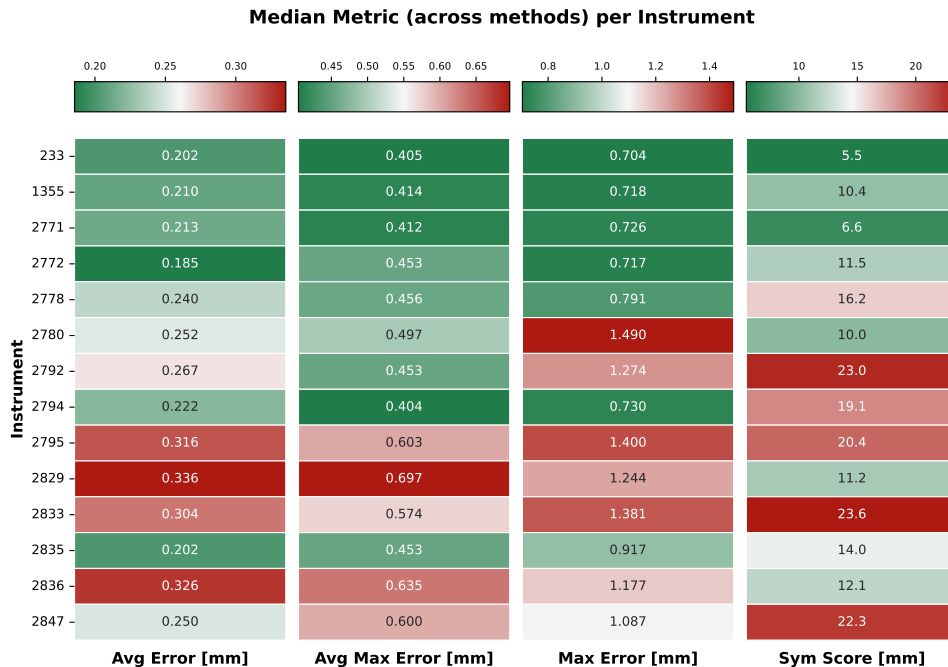


Figure 6.5: Median of the four evaluation metrics per instrument, computed across all four method-parameter combinations (SetCover-K and SetCover-KM, with $K = 16$ and $K = 18$ respectively). Lower values indicate better approximation for all metrics.

Overall, SetCover-K and SetCover-KM display similar results¹. SetCover-KM trivially achieves a lower maximum error, and SetCover-K reaches a lower average maximum error, while the average error shows no clear preference for either method. One noticeable difference between the two methods lies in the symmetry score. As can be seen in Table 6.1, this metric is highly variable for SetCover-K: it produces both the lowest and highest values across the corpus, and exhibits a large spread (see standard deviation (Std) and interquartile range (IQR)). On the contrary, SetCover-KM shows greater consistency, with maximum values approximately half of those produced by SetCover-K, and a markedly lower standard deviation and interquartile range. This suggests that SetCover-K outperforms SetCover-KM on a portion of the corpus, but SetCover-KM produces more uniform solutions overall.

Metric	SCK 16	SCK 18	SCKM 16	SCKM 18
Min	2.979	3.826	4.227	5.168
Max	41.321	41.930	19.782	23.214
Range	38.342	38.104	15.555	18.046
Mean	19.068	19.130	8.235	12.519
Median	14.631	14.337	7.005	12.161
Std	12.076	12.197	3.810	5.445
IQR	16.556	14.297	3.076	8.235

Table 6.1: Symmetry score statistics per method. Bold values indicate the smallest value per row. All values in [mm].

Across the entire violin corpus, and both methods, 18 circles proved to be a closer approximation than 16. Table 6.2 displays the mean performance of each method across the entire corpus, and methods with 18 circles outperform methods with 16 circles in all evaluation metrics besides symmetry score. This is trivial: with more circles, the model has more freedom to approximate the outline closely.

Method	Avg Error $\bar{\delta}$	Avg Max Error $\bar{\delta}_{\max}$	Max Error $\delta_{\max}$	Sym Score
SCK 16	0.312	0.498	1.551	19.068
SCKM 16	0.306	0.772	0.942	8.235
SCK 18	0.206	0.361	1.129	19.130
SCKM 18	0.207	0.510	0.641	12.519

Table 6.2: Mean Performance Across All Instruments. All values in [mm].

However, a more interesting question concerns the nature of the circles added or modified when moving from 16 to 18 circles: whether they approximate a large part of the outline and change the overall layout, or merely correct local distortions. This varies depending on the violin, but some observable patterns emerge. For some violins, two large circles are added in the lower bout (see Figure 6.7), which can also be found in the theoretical construction by François Denis that led to the Digital Amati. SetCover-K retrieves those exact two circles for violins 0233 and 2771 when raising K from 16 to 18. For some instruments, one of the two circles was already present with 16 circles, and the

¹All evaluation metrics are visible in Appendices D for SetCover-K and E for SetCover-KM.

rise to 18 retrieves the other (for instance, violins 2780 and 2829). Another recurring change concerns the C-bouts. On most violins, when K is large enough, five circles define the C-bouts: one large and four small ones wrapping the corners. For some violins, such as 2772 and 2778, increasing K from 16 to 18 completes this configuration on both sides.

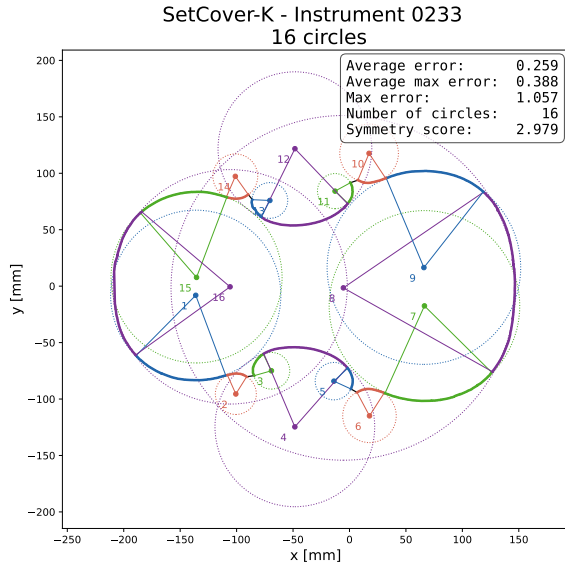


Figure 6.6: SetCover-K with $K=16$ – Instrument 0233

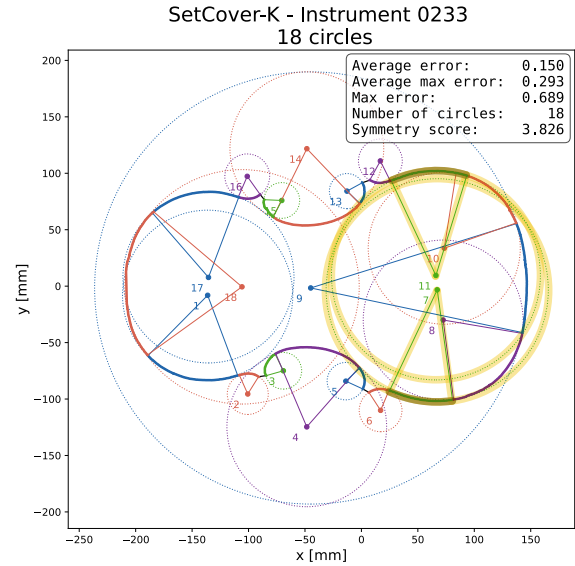


Figure 6.7: SetCover-K with $K=18$ – Instrument 0233. Highlighted are the added circles when raising K from 16 to 18.

The same trends can be observed for SetCover-KM, but not necessarily for the same instruments. For instance, raising K from 16 to 18 for violin 2792 adds the two lower bout circles, mirroring what was observed for violins 0233 and 2771 with SetCover-K. It thus seems that the emerging structure is not solely the product of one violin shape or one method, but of a combination of both. Overall, the same trends are noticeable for both methods, but SetCover-KM tends to prioritize smaller corrective circles addressing local specificities, more so than SetCover-K.

More generally, while the results vary depending on the violin and the method used, observable patterns emerge consistently across the corpus. Figure 6.8 introduces the labelling used to refer to these circle groups, illustrated on violin 0233. Table 6.3 then shows, for each group, the percentage of the corpus where the pattern is observed. The left value indicates the percentage of violins displaying the expected number of circles for that group, and the right value, the percentage of violins displaying at least some of them.

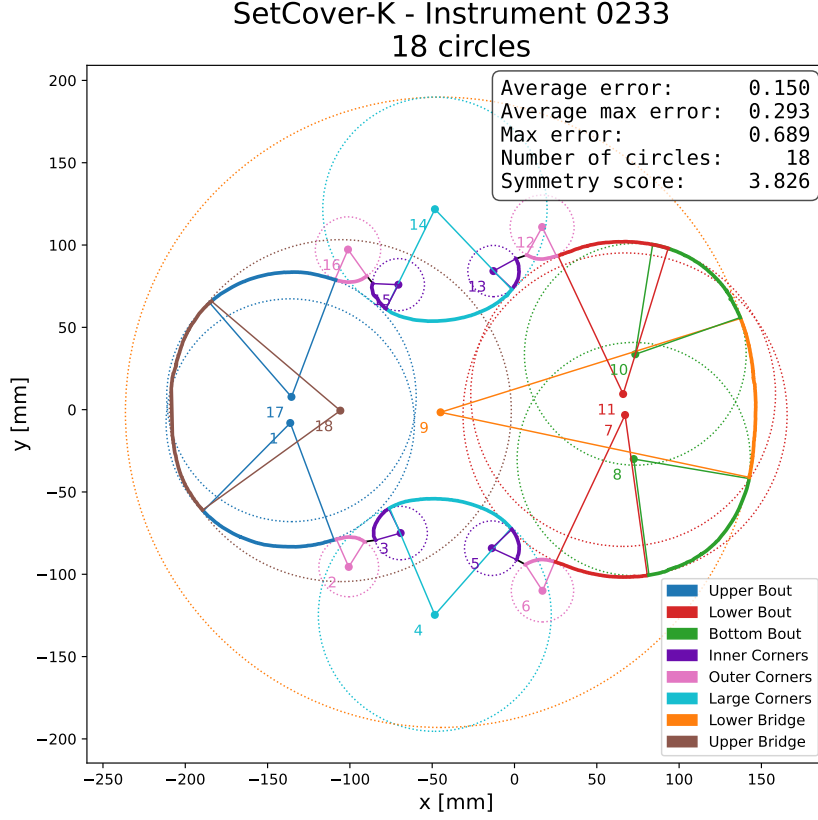


Figure 6.8: SetCover-K with $K=18$ – Instrument 0233. Colour-coded for circle labelling.

	SetCover-K 16		SetCover-K 18		SetCover-KM 16		SetCover-KM 18	
	Full	Partial	Full	Partial	Full	Partial	Full	Partial
Upper Bout	100%	100%	100%	100%	100%	100%	100%	100%
Lower Bout	0%	50%	35.7%	64.2%	0%	14.3%	35.7%	100%
Bottom Bout	100%	100%	100%	100%	100%	100%	100%	100%
Inner Corners	35.7%	100%	64.2%	100%	71.4%	100%	92.9%	100%
Outer Corners	85.6%	100%	85.6%	100%	71.4%	100%	64.2%	100%
Large Corners	100%	100%	100%	100%	92.9%	100%	100%	100%
Lower Bridge	100%	100%	100%	100%	100%	100%	71.4%	71.4%
Upper Bridge	85.6%	85.6%	85.6%	85.6%	85.6%	85.6%	100%	100%
Mean	75.86%	91.95%	83.89%	93.73%	77.66%	87.48%	89.03%	96.43%

Table 6.3: Percentage of recurring circles for each method. Full indicates when all of the expected circles can be observed, and partial when at least one of them is visible.

Table 6.3 shows that the upper bout and bottom bout circles can be observed across both methods and both values of K . The large corners also appear in all cases but for one violin in SetCover-KM 16. This suggests that these are structural circles of violins. The inner and outer corners always appear partially, but the damage concentrated in those zones sometimes prevents all corner circles from appearing.

While Table 6.3 allows us to observe recurring patterns and formulate possible historical hypotheses, it also highlights methods that manage to retrieve those structures more consistently. We can see that SetCover-KM with $K=18$ outperforms all other methods, for both showing the recurring circles fully or partially. This is both consistent with the observation that SetCover-KM tends to be more consistent across the corpus, while SetCover-K tends to perform better on well-conditioned violins. While SetCover-K manages to retrieve all identified circles for violin 0233 (see Figure 6.8), it struggles to perform optimally on damaged violins.

Another way to measure whether the obtained results have historical significance is to compare the radii of the obtained circles. A luthier would likely have worked with simple, whole-number proportions. For this, we studied violin 0233 with 18 circles and method SetCover-K. Table 6.4 displays the ratio between the radius of circle 2 (chosen because it is among the smallest, and therefore a plausible unit of measure) and the radii of all other circles. The third column displays a possible fraction that could have been used as a proportion, chosen to be as close to a simple rational number as possible, to reflect a plausible design choice. The last column shows the error (in percentage) between the ratio and the fraction, defined as:

$$\epsilon_i = \left| \frac{\rho_i - p/q}{\rho_i} \right| \times 100\%$$

where $\rho_i = r_i/r_2$ is the measured ratio and p/q is approximated fraction.

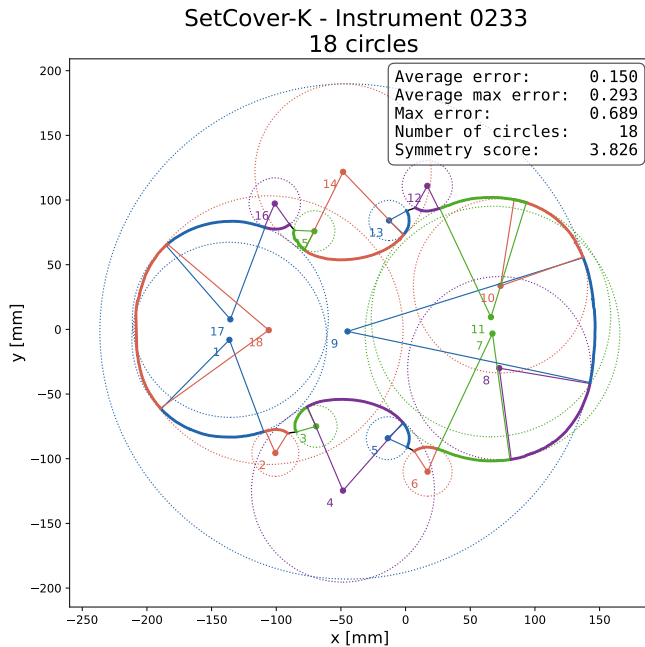


Figure 6.9: SetCover-K with $K = 18$ – Instrument 0233

#	Ratio r_i/r_2	Fraction	Error $\epsilon_i(\%)$
1	4.154	4/1	3.699
2	1.000	1/1	0.000
3	0.894	4/5	10.519
4	3.896	1/1	2.669
5	0.913	1/1	9.514
6	1.044	1/1	4.173
7	5.420	5/1	7.756
8	3.900	4/1	2.552
9	10.542	10/1	5.144
10	3.709	4/1	7.845
11	5.097	5/1	1.911
12	1.070	1/1	6.558
13	0.857	4/5	6.691
14	3.751	4/1	6.635
15	0.872	4/5	8.309
16	1.093	1/1	8.507
17	4.174	4/1	4.157
18	5.719	6/1	4.916

Table 6.4: Ratios r_i/r_2 with rational approximations

As we can see, using simple whole-number fractions, the resulting errors remain below 10% (with the exception of circle 3). Given the age of the instrument and the inherent

imprecision of hand craftsmanship, these errors seem reasonable. This further supports the historical plausibility of the circle layout retrieved by SetCover-K for instrument 0233.

One final observation worth noting is that, while achieved with different methods, the results obtained for violin 0233 with SetCover-K and violins 1355 and 2792 with SetCover-KM show a remarkable regularity with 18 circles. Despite noticeable irregularities, the circle layouts are qualitatively very similar. This is especially meaningful because those three violins are attributed to the same violin maker, Matthijs Hofmans IV (see Appendix A), and were thus most likely constructed using the same method. The fact that our algorithms independently retrieved similar circle layouts further supports the historical plausibility of the results. Moreover, the fact that this agreement was reached by two different methods reinforces the observation that SetCover-K performs better on well-conditioned violins, while SetCover-KM is more robust on damaged or irregular instruments.

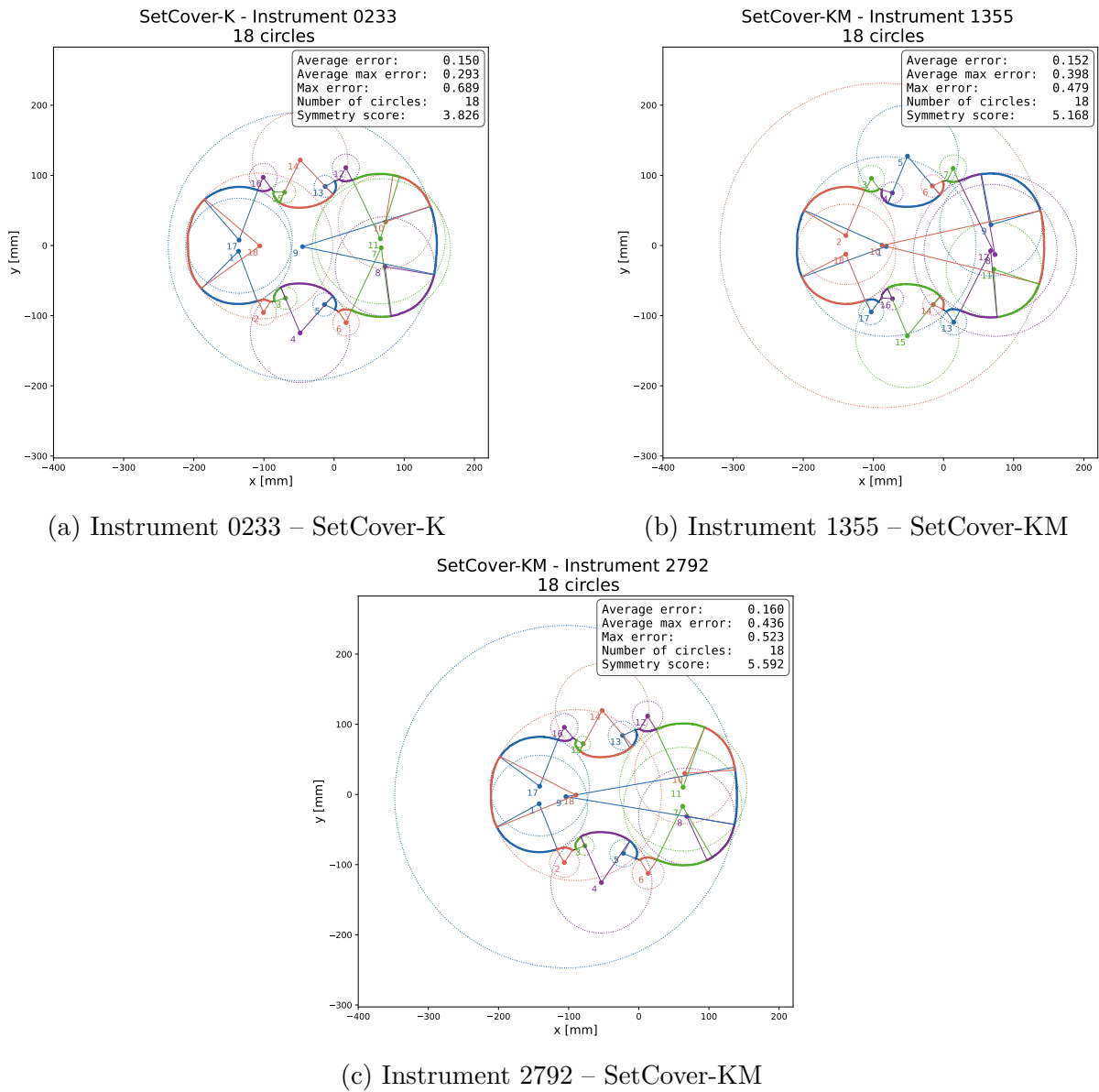


Figure 6.10: Three violins by Matthijs Hofmans IV, analyzed with different methods

6.2 Reduced violins

When the violin appeared, instrument sizes were not standardized, and the sizes and morphologies varied. During the 18th century, the instruments in the violin family were classified in three categories: the violin, the viola, and the cello. This led to the reshaping of many instruments to fit the new standards (see Figure 6.11).

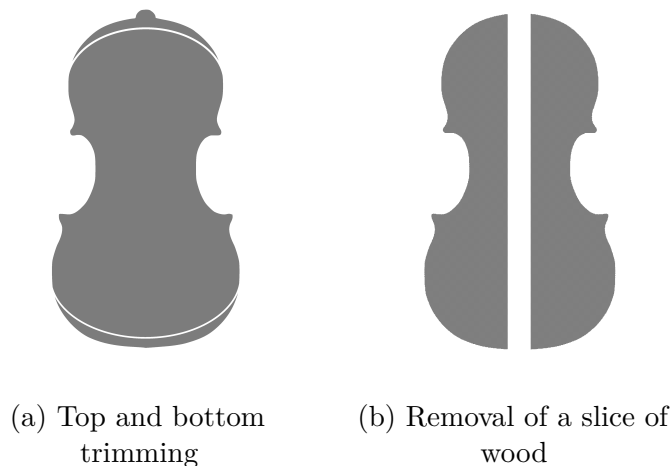


Figure 6.11: Different kinds of reduction

Throughout this thesis, we have only considered instruments that are not believed to have been reduced. However, the methods developed here naturally raise the question of whether such a reduction would be observable from the approximated circles. It should also be kept in mind that whether any given instrument was reduced remains a hypothesis rather than an established fact. As such, if patterns were to emerge and certain violins did not conform to them, this could raise questions about whether those instruments had indeed undergone reduction.

Overall, the circles obtained on the reduced violins do not differ much from the circles obtained on the non-reduced violins (see all results for reduced violins in Appendix F). Figure 6.12 shows one pattern observed on many of the reduced violins: the circle approximating the large bridge² is larger, maybe suggesting a reduction of the bottom of the body (see Figure 6.11 (a) for illustration). However, while this pattern is observed more often for reduced violins, this is not an absolute: not all reduced violins present this characteristic, and some non-reduced violins do present it (see instrument 2794 in Appendix D).

²See Figure 6.8 for circle labelling.

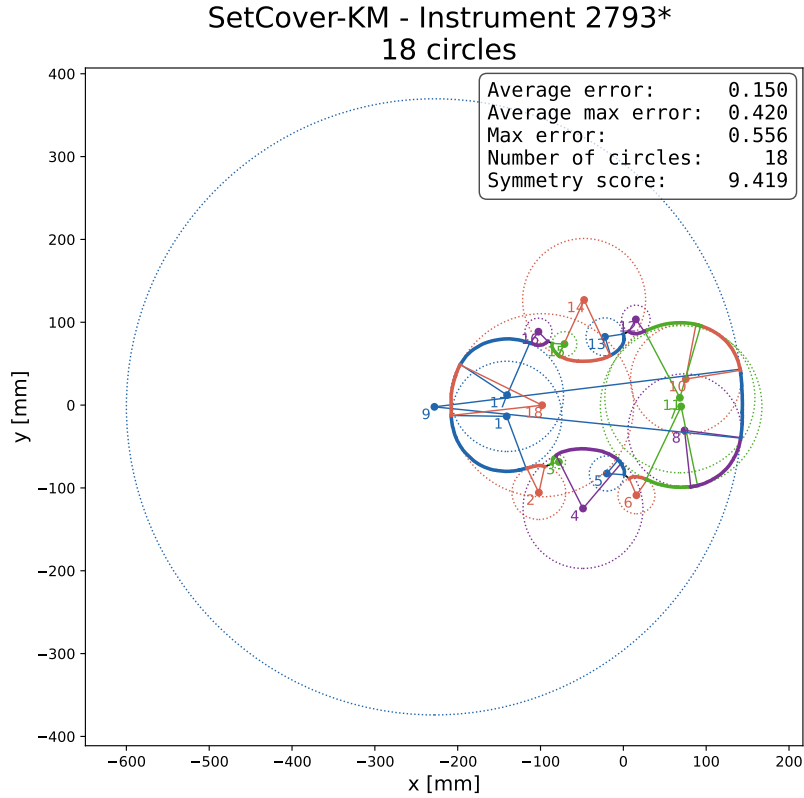


Figure 6.12: SetCover-KM with $K = 18$ – Instrument 2793* (reduced). Circle 9 (blue), defining the lower bridge, could suggest a reduction of the bottom of the violin.

Some reduced violins show very similar results to non-reduced violins (see Figures 6.13 (non-reduced) and 6.14 (reduced)). The reduction is thus imperceptible from a qualitative assessment of the circles. In the case of the removal of a slice of wood (see Figure 6.11(b)), the circles would not necessarily have been modified, but simply shifted. A thorough study of the position of the centers could investigate if that difference can be computed.

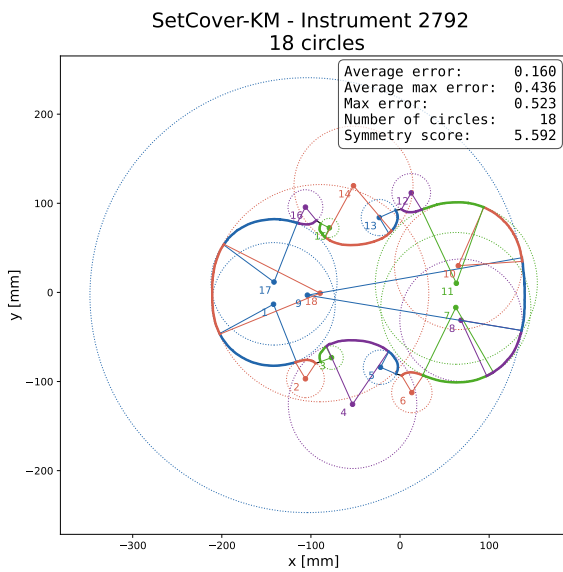


Figure 6.13: SetCover-KM with $K=18$ – Instrument 2792 (not reduced)

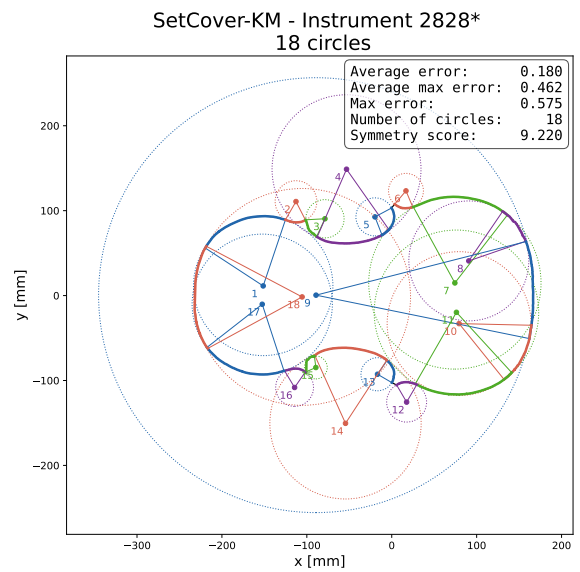


Figure 6.14: SetCover-KM with $K=18$ – Instrument 2828* (reduced)

6.3 Selecting the best method

All these observations lead to one question: what is the best method to approximate violins? The results show that neither SetCover-K nor SetCover-KM is unambiguously superior. SetCover-K tends to produce better results for well-conditioned instruments, such as the Digital Amati or violin 0233, meaning instruments that suffered little damage and were both produced and preserved with a high degree of symmetry. SetCover-KM gives more consistent results throughout the corpus and seems to be more robust for ill-conditioned instruments, where it avoids producing large circles and stabilizes the symmetry score.

It is worth noting that the results obtained for violin 0233 with SetCover-K are very similar to those obtained by SetCover-KM for violins 1355 and 2792. It thus seems that the best results are not tied to one method, but depend on the instrument.

When it comes to choosing the number of circles, it seems clear that K above 20 leads to overfitting the outline with small circles, while K below 15 tends to underfit it, leaving parts of the outline visibly poorly approximated. Setting $K = 18$ appears to be a good compromise, where both methods retrieve the underlying structure without overfitting. However, no parameter study yields a clear breakpoint, and the optimal K may also vary across the violin corpus.

6.4 Future directions

As observed throughout this thesis, the methods became increasingly consistent as they evolved. Our two final methods, SetCover-K and SetCover-KM, manage to retrieve recurring solutions across the violin corpus. However, the results are not fully uniform throughout the corpus, due to distortion or damage. The results of SetCover-K on the Digital Amati show that for a perfectly conditioned violin, the correct circles are retrieved. Several directions could guide the methods toward more homogeneous results for damaged violins, and as such, fill the gap created by those distortions. The first possibility would be to pre-process the violin outline to smooth out the contour, thereby avoiding circles that correct local distortions rather than capturing underlying structure.

Another direction would be to bound the radius of the candidate circles. SetCover-K occasionally approximates small parts of the outline with disproportionately large circles, which contribute little to the overall layout and whose radii seem unlikely to reflect a deliberate design choice (see Figure 4.14). Introducing an upper bound on the radius when generating the candidate circles would prevent such irregularities.

Another lead to explore would be to enforce approximate symmetry rather than exact symmetry. As we have seen in Chapter 5, enforcing perfect symmetry proves to be infeasible when the violin presents slight damage. However, a method allowing for some tolerance on the symmetry would allow a method to remain feasible.

Conclusion

Violins that are constructed nowadays are copies of existing models. It is believed that luthiers from the 17th and 18th centuries designed the outline of violins using circles. While some indirect sources from Northern Italy displaying circular constructions have been studied, no equivalent for violins from the Low Countries has been found. The purpose of this thesis is to approximate violin outlines using a set of circles, with the hope of finding recurring patterns that could have historical significance.

Several markers can indicate whether an approximation has historical plausibility. Patterns across the violin corpus, especially for instruments made by the same luthier, can suggest underlying structure. Another clue is symmetry between the left and right sides of the violin. These markers are not absolute: different luthiers may have had different methods for designing instruments, or may even have changed their approach throughout their lives, and symmetry may have been only approximate at the time of construction, and further altered by damage. Nevertheless, these remain useful indicators of whether a result reflects genuine historical practice. To quantify the symmetry of the obtained results, we develop in Chapter 5 a symmetry score. Finally, the study of radius length and center positions can inform whether the relative positions and proportions are simple enough to reflect a deliberate design choice.

The first method explored in this thesis is the implementation and study of the YangDu Algorithm, a greedy algorithm developed by Yang and Du in [19]. However, this algorithm gives uneven results and is heavily conditioned on the starting point. Furthermore, its objective is to compute the biggest circles, rather than the best-fitted. Being a greedy algorithm, it does not provide a guarantee to find the best solution overall, and does not question its previous choices.

To correct these shortcomings, we develop various global methods. The first one, SetCover- ε , corrects the weaknesses of the YangDu Algorithm, but presents its own: it minimizes the number of circles rather than an evaluation metric reflecting the approximation quality, and, given the discrete nature of the objective function, reaches several equivalent solutions, without differentiating between them. The SetCover-K method corrects those weaknesses by adapting the objective function to minimize the sum of the maximum error of each selected circle. The candidate circle set is also extended to contain all intermediary circles. This model produces satisfying results: outstanding for some violins, and satisfactory for others. The SetCover-K model is then adapted into a third model, SetCover-KM, which minimizes the worst-case maximum error rather than the sum of all maximum errors. This model led to more consistent results across the corpus. For well-conditioned instruments, it tends to yield slightly inferior results, but for the most damaged or irregular instruments, the results are more symmetrical and recurring

patterns are more observable

Both SetCover-K and SetCover-KM demonstrate that a set of circles can approximate violin outlines in a meaningful and recurring way. The two methods are complementary rather than competing: SetCover-K excels on well-preserved instruments, producing results of remarkable quality, while SetCover-KM provides more consistent and symmetric solutions across the entire corpus, including damaged or irregular instruments. The study of recurring circle patterns across the corpus, and particularly the regularity observed across instruments attributed to the same luthier, suggests that the results carry potential historical significance.

The main challenge when using these methods is to find the right balance between mathematical exactitude, approximating the outline as close as possible, and allowing a tolerance that reflects the historical possibility of imprecision on the part of the luthier, as well as the damages encountered over time. François Denis, who reconstructed mold outlines using circles, had to bend reality to fit the outlines into the desired shapes. This is not necessarily dishonest: some flexibility is needed when working with instruments built by craftsmanship and distorted through time. Nevertheless, this tradeoff between mathematical exactitude and adaptability should be carefully questioned and handled.

These methods remain limited: the results are sensitive to the condition of the violin, and noise, damage, or distortion prevent the methods from producing entirely consistent results. The study of the symmetry score shows that no method can produce exact symmetry across the corpus, and our attempts to enforce it led to the circles being multiplied to fit both sides. While showing recurring patterns, none of these methods manages to retrieve those patterns consistently. Finally, the choice of the optimal number of circles, a parameter chosen by the user, remains unclear. While 18 circles seem to produce satisfying results, our parameter studies do not reveal any clear breakpoints.

Several directions remain to be explored to perfect those results. A pre-processing of the outline could reduce noise or damage, which could prevent the methods from correcting those local distortions and focus on the general structure. Bounding the radii of the candidate circles could also prevent the methods from selecting ill-suited and unlikely circles. Finally, enforcing approximate symmetry rather than exact symmetry could guide the method toward a balanced design without increasing the number of needed circles.

Ultimately, this thesis proposes methods and a framework to analyse the geometrical construction of historical violins. While no definitive historical conclusion can be drawn on the luthiers' original intents and methods, the recurring structures observed across the corpus suggest one possible design method for instruments, further supported by the simple proportions of the radii. Together, these findings open possibilities for further musicological research.

Glossary of Mathematical Notations

$\mathcal{P} = \{p_1, \dots, p_n\}$	Set of contour points of a violin outline
$\mathcal{P}_k \subseteq \mathcal{P}$	Set of contour points approximated by circle C_k
p_i	Individual contour point
$\mathcal{C} = \{C_1, \dots, C_m\}$	Set of candidate circles
$\mathbf{c}_k = (x_k^c, y_k^c)$	Center of circle C_k
r_k	Radius of circle C_k
$\mathcal{C}^* = \{C_1^*, \dots, C_{K'}^*\} \subseteq \mathcal{C}$	Set of selected circles (output of optimization)
e_k	Maximum fitting error of circle C_k (mm)
ε	Maximum fitting tolerance (mm)
K	Maximum number of circles allowed
K'	Number of selected circles
$\bar{\delta}$	Average error (mm)
$\bar{\delta}_{\max}$	Average maximum error (mm)
$\delta_{\max}$	Maximum error (mm)
$A_{i,k}$	Incidence matrix entry: 1 if point p_i is covered by circle C_k , 0 otherwise
$x_k \in \{0, 1\}$	Binary variable: 1 if circle C_k is selected, 0 otherwise
$S(\mathcal{C}^*)$	Symmetry score of a circle configuration

References

- [1] Philémon Beghin. A Digital Tool at the Service of Organology: Validation of a Photogrammetric Approach. Master’s thesis, UCLouvain (EPL), 2021. Supervisors: P. Fisette, F. Glineur, and A.-E. Ceulemans.
- [2] Philémon Beghin, Anne-Emmanuelle Ceulemans, Paul Fisette, and François Glineur. Validation of a photogrammetric approach for the objective study of early bowed instruments. *Heritage Science*, 11:170, 2023.
- [3] Fausto Cacciatori, editor. *Antonio Stradivari – disegni, modelli, forme*. Museo del Violino, Cremona, 2016.
- [4] Anne-Emmanuelle Ceulemans. *De la vièle médiévale au violon du XVII^e siècle*. Épitome musical. Brepols, Turnhout, 2011.
- [5] Anne-Emmanuelle Ceulemans. The Violins of César Snoeck: Observations on the Origins of an Instrument Family. *Revue belge de Musicologie / Belgisch Tijdschrift voor Muziekwetenschap*, 76:165–192, 2022.
- [6] Daniel H. Chitwood. Imitation, Genetic Lineages, and Time Influenced the Morphological Evolution of the Violin. *PLOS ONE*, 9(10):e109229, 2014.
- [7] Kevin Coates. *Geometry, Proportion and the Art of Lutherie*. Clarendon Press / Oxford University Press, Oxford, 1985. First printed in paperback 1991. Printed by Butler and Tanner Ltd, Frome, Somerset.
- [8] François Denis. *Traité de Lutherie*. ALADFI, 2006.
- [9] François Denis. The Geometric Principles of String Instrument-Making in Brescia. In Luisa Baggi and Irene Dioli, editors, *Liutai in Brescia: 1520–1724*. Eric Blot, Brescia, 2008.
- [10] Sebastian Gonzalez, Davide Salvi, Daniel Baeza, Fabio Antonacci, and Augusto Sarti. A data-driven approach to violin making. *Scientific Reports*, 11:9455, 2021.
- [11] Evaggelos Kaselouris, Makis Bakarezos, Michael Tatarakis, Nektarios A. Papadogiannis, and Vasilis Dimitriou. A Review of Finite Element Studies in String Musical Instruments. *Acoustics*, 4:183–202, 2022.
- [12] John Koster. Arnaut de Zwolle, Henri. In *Grove Music Online*. Oxford University Press, 2001. Published in print: 20 January 2001.
- [13] Harry Mairson. Functional geometry and the “Traité de Lutherie”. *Journal of the Violin Society of America: VSA Proceedings*, XXVI(2):190–202, 2015.

- [14] Asif Masood and Muhammad Sarfraz. Capturing outlines of 2D objects with Bézier cubic approximation. *Image and Vision Computing*, 27(6):704–712, 2009.
- [15] Karel Moens. De viool in de 16de eeuw 2: Het ontstaan en de verspreiding van de ‘echte’ viool. *Musica Antiqua*, 11(1):5–12, February 1994.
- [16] Karel Moens. Renaissancegambas in het Brussels Instrumentenmuseum. Vragen rond toeschrijvingen, verbouwingen en authenticiteit. *Bulletin van de Koninklijke Musea voor Kunst en Geschiedenis*, 66:161–237, 1995.
- [17] Simone F. Sacconi. *I “segreti” di Stradivari*. Libreria del Convegno, Cremona, 1972. Con il catalogo dei cimeli stradivariani del Museo Civico “Ala Ponzzone” di Cremona.
- [18] Feng Sun, Yi-King Choi, Yizhou Yu, and Wenping Wang. Medial Meshes – A Compact and Accurate Representation of Medial Axis Transform. *IEEE Transactions on Visualization and Computer Graphics*, 22(3):1278–1290, 2016.
- [19] Shi-Nine Yang and Wei-Chang Du. Numerical methods for approximating digitized curves by piecewise circular arcs. *Journal of Computational and Applied Mathematics*, 66:557–569, 1996.

Figures references

All the figures in this thesis are my own production, except for:

- Figure 1.1: Picture of violin 2778, Brussels Musical Instruments Museum ([link](#)), annotated by me
- Figure 1.2: From K. Moens, *Europese strijkinstrumenten*, p. 22 (drawing: Els Jageneau), annotated by me
- Figure 1.3: From P. Beghin in [1]
- Figure 1.4: From P. Beghin in [2]
- Figure 1.8: Output of code by H. Mairson
- Figure 2.1: From H. Arnaut de Zwolle, Bibliothèque nationale de France, Department of Manuscripts, Latin 7295
- Figure 2.2: From K. Coates in [7]
- Figure 2.3: From H. Mairson in [13]
- Figure 6.11: From A.-E. Ceulemans

Appendix A

Violin Corpus

Unreduced Instruments

ID	Maker	Type	Date	URL
233	Matthijs Hofmans IV	violin	1645–1679	link
1355	Matthijs Hofmans IV	violin	before 1672 (uncertain)	link
2771	Hendrick Willems	violin	1701–1730	link
2772	Hendrick Willems	violin	1715	link
2778	Egidius Snoeck	violin	1701–1750 (uncertain)	link
2780	Marcus Snoeck	violin	c. 1740	link
2792	Matthijs Hofmans IV	violin	1665	link
2794	Jooris Willems	violin	1659	link
2795	Jooris Willems	violin	1659	link
2829	Hendrick Willems	viola	1650–1659	link
2833	Johannes Theodorus Cuypers	viola	1761	link
2835	Jean Hyacinthe Joseph Rottenburg ¹	viola	1753	link
2836	Gaspar Borbon	tenor viola	1692	link
2847	Matthijs Hofmans IV/Gaspar B	viola	before 1691 (uncertain)	link

Table A.1: List of unreduced instruments in the corpus

Reduced Instruments

ID	Maker	Type	Date	URL
2769	Hendrick Willems	violin	1723–1743	link
2774	Gaspar Borbon	violin	1650–1700	link
2776	Gaspar Borbon	violin	1701	link
2793	M. Hofmans IV? / G. Borbon?	violin	before 1679	link
2828	Hendrick Willems?	viola	1660	link
2831	Jooris Willems	viola	?	link
2840	Peeter I. Borbon? / J.-J.-A. De Lannoy?	viola	1771	link
2846	M. Hofmans IV? / G. Borbon?	viola	before 1679	link
2848	Matthys Hofmans IV?	viola	before 1672	link

Table A.2: List of reduced instruments in the corpus

¹The Rottenburgs were primarily wind instrument makers. The instrument may have been built by Marcus Snoeck at Rottenburg's request. It bore a label reading "Jean Hyacinth Rottenburgh / maior fecit à Bruxelles 1753".

Appendix B

YangDu: Enforcing Symmetry

As seen in Chapter 3, symmetry between the left and right parts of the instrument is imperfect with the YangDu Algorithm. If we consider that violins are surely built symmetrically, one option is to impose the symmetry. This is attempted in two ways: the first is to adapt the algorithm itself to only accept symmetrical arcs. The second way is to resample the points to align them across the x-axis and see if the obtained circles get more symmetrical.

B.1 Enforcing symmetry

The outline of an instrument is a single closed curve. First, we need to cut it into two halves that correspond to each other. For this, we take the full outline and flip it across the horizontal axis, then check, at each point, how far the original curve is from its flipped version. Where the curve is most symmetric, this distance is smallest. The point with the minimum distance becomes the starting point for the split. We then cut the outline in half from there, and reverse one of the halves so that both sequences travel in the same direction.

Then, the arc fitting can begin. We follow the YangDu algorithm on one half, but the acceptance condition changes: a segment is only accepted if the arc fits both curves within tolerance at the same time. If either one fails, the segment is rejected. This makes the algorithm more conservative than the standard Yang algorithm, since in asymmetric regions, both halves must agree, which forces shorter arcs and a finer segmentation.

As we can see, the result is disappointing: in Figure B.1, the number of circles almost triples, meaning the algorithm really struggles to fit arcs to the outline on both sides. Even when relaxing the tolerance parameter to 3mm (Figure B.2), the symmetric YangDu finds 38 circles, which is twice the number of circles for the standard YangDu with a tolerance of 1mm. As mentioned, the lack of symmetry may be due to an inherent fault of the algorithm, but also to a lack of symmetry in the violin itself.

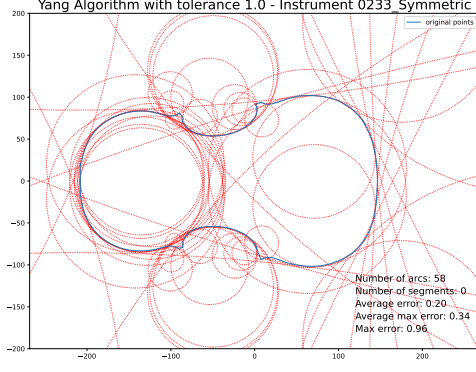


Figure B.1: Symmetric YangDu Algorithm with tolerance of 1.0 - Violin 0233

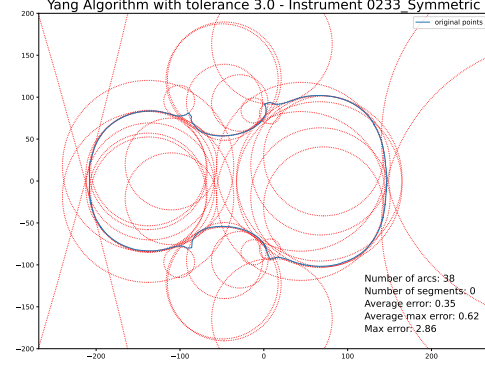


Figure B.2: Symmetric YangDu Algorithm with tolerance of 3.0 - Violin 0233

One limitation is worth acknowledging: the arc is always fitted to one side and simply tested against the other, rather than being fitted to both simultaneously. A truly symmetric approach would minimize the joint error across both curves, but this was not implemented here. As this adaptation is primarily an exploration, this is acceptable. Had the results been more promising, it would have been motivation to develop the method further.

B.2 Realigning points

Another attempt to improve the symmetry of the results was to realign the points vertically. As seen in Section 1.1, the outline of the instruments is extracted from a mesh. As such, the points are not necessarily aligned perfectly. The solution is to resample one half to match the other. Walking along the left half, for each point, the corresponding point on the right half at the same horizontal position is computed. If such a point exists exactly, it is used directly. If not, it is approximated by linear interpolation between the two nearest points. The result is a right half that shares the same horizontal positions as the left half, making the two sequences genuinely comparable.

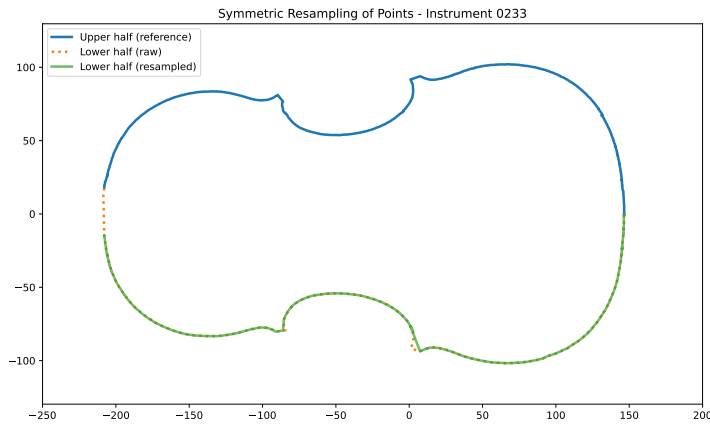


Figure B.3: Contour of violin 033 when realigning the outline. Blue line is the original points from the top. The orange dotted line is the original points from the bottom. The green line is the resampled bottom half according to the points of the top half.

The orange and green overlap almost perfectly everywhere, which confirms that the resampling introduces very little distortion along the smooth portions of the outline. The visible imprecision appears at the two corners, around the f-hole notches. These are the sharpest parts of the outline, where the curve changes direction very abruptly. Linear interpolation works well along smooth arcs but struggles at corners, since the x-coordinate is not monotone there and the local geometry cannot be captured by simply interpolating between two neighboring points. This is a known limitation of the resampling approach.

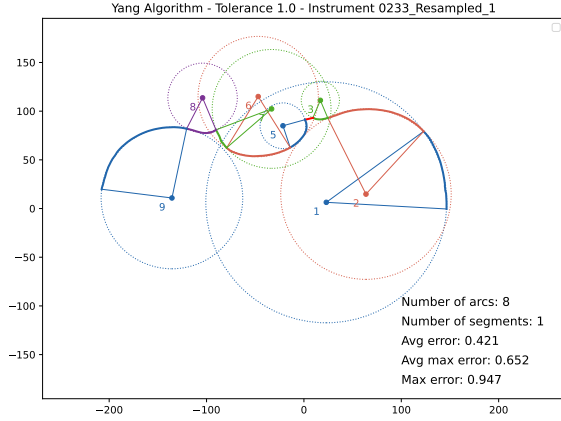


Figure B.4: Resampled Yang Algorithm - Violin 0233 (Reference Part)

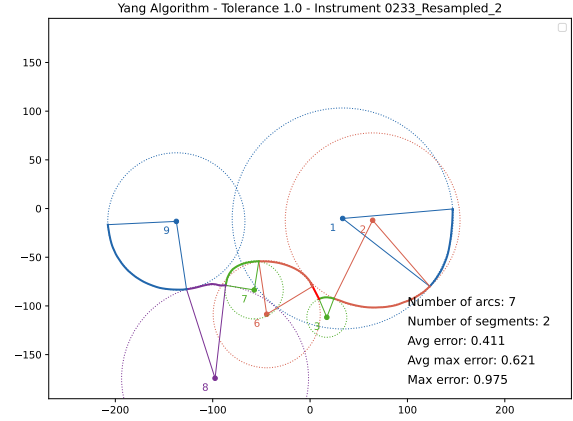


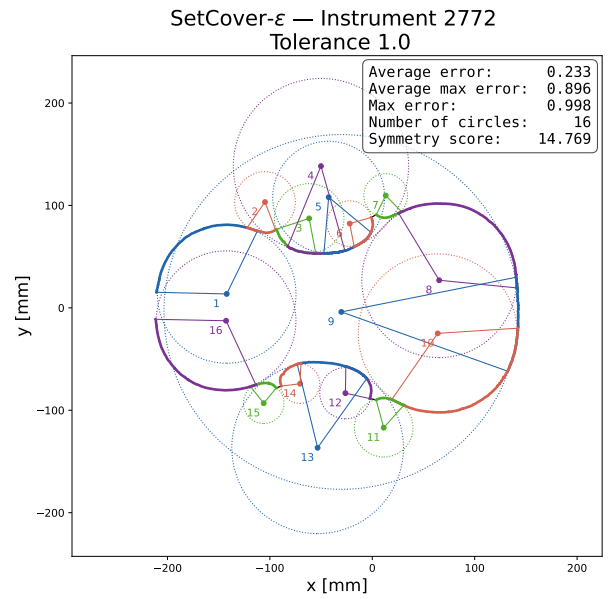
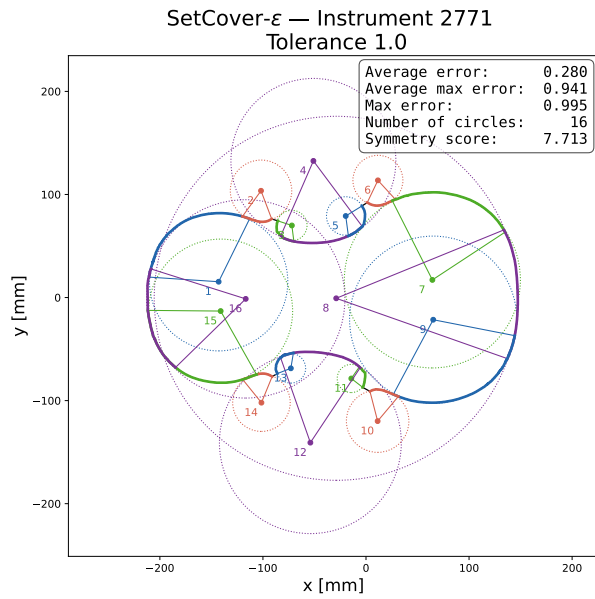
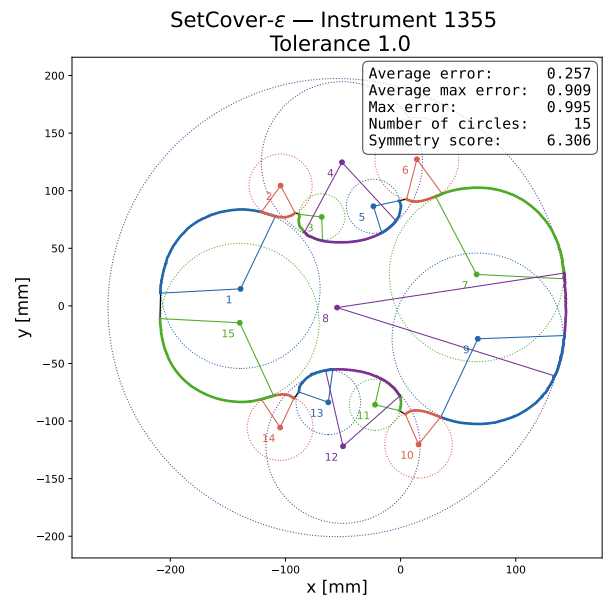
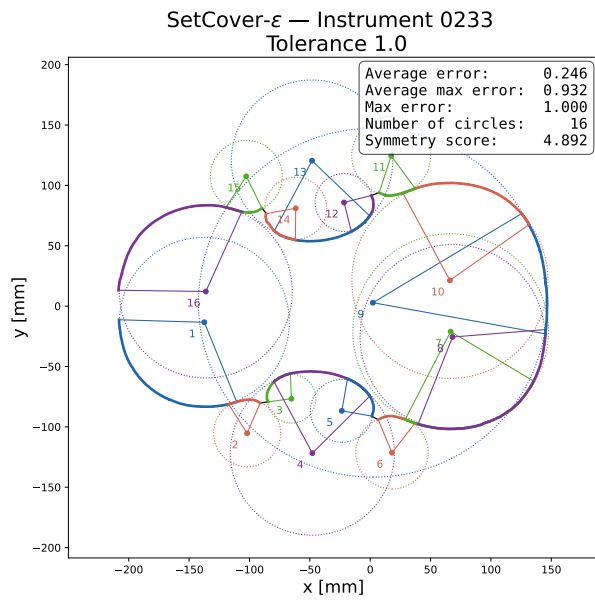
Figure B.5: Resampled Yang Algorithm - Violin 0233 (Resampled Part)

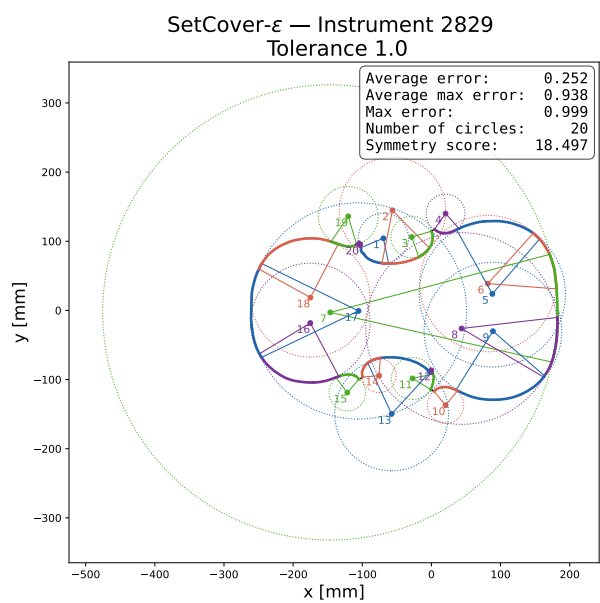
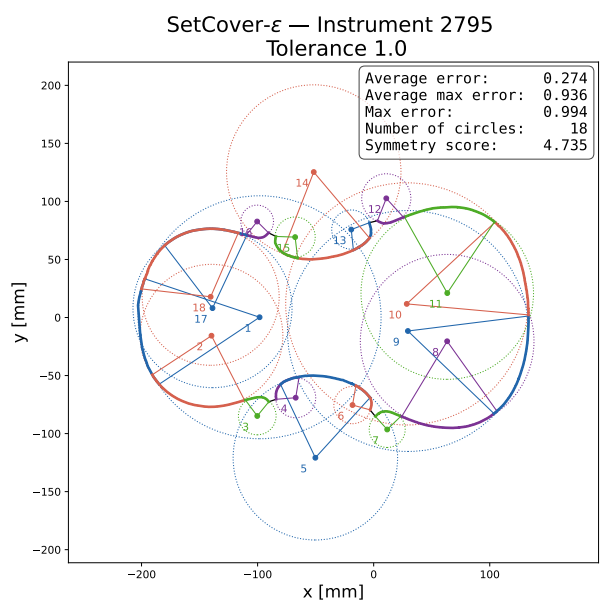
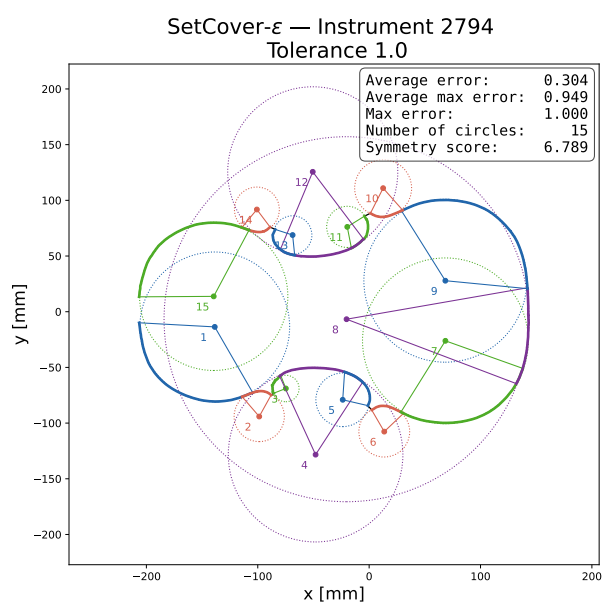
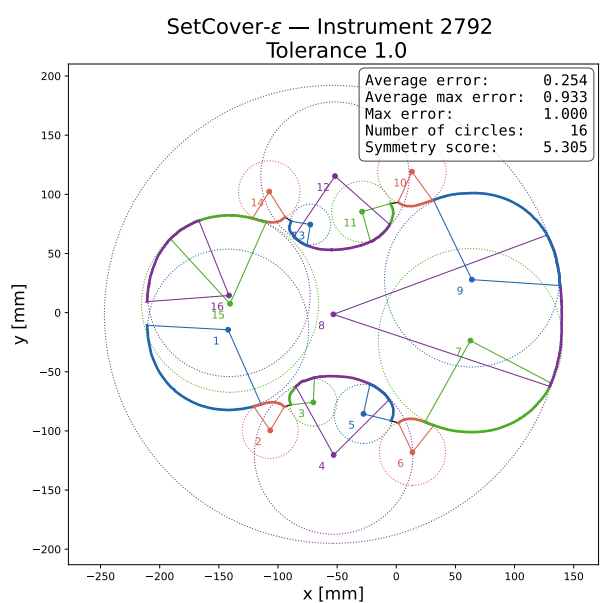
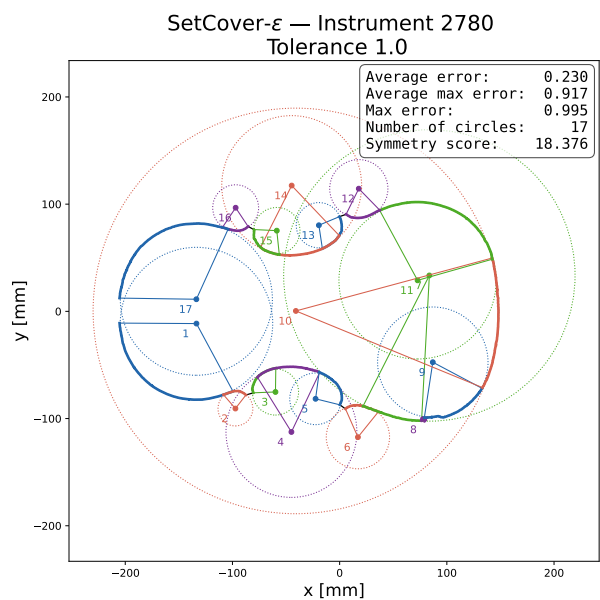
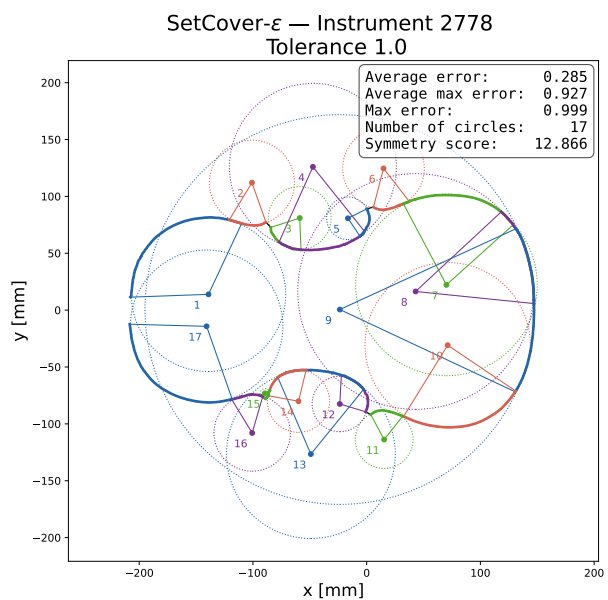
As we can see, the results are disappointing as well. The decompositions are reasonably consistent: both halves use roughly the same number of arcs, and the large arcs covering the upper and lower bouts are visible and geometrically sensible. The evaluation metrics are also quite similar. However, the two segmentations are not the same. The breakpoints do not fall at the same locations, and the arc numbering does not correspond between the two halves. This is exactly the problem the symmetric algorithm is meant to solve. Running the standard algorithm independently on each half gives no guarantee of alignment, making any direct arc-by-arc comparison between left and right impossible. Using this algorithm more confidently would also require the assurance that luthiers actually constructed their outlines from circles starting at symmetric breakpoints, which is not necessarily the case, as suggested by the historical drawings discussed in Section 2.1.

Appendix C

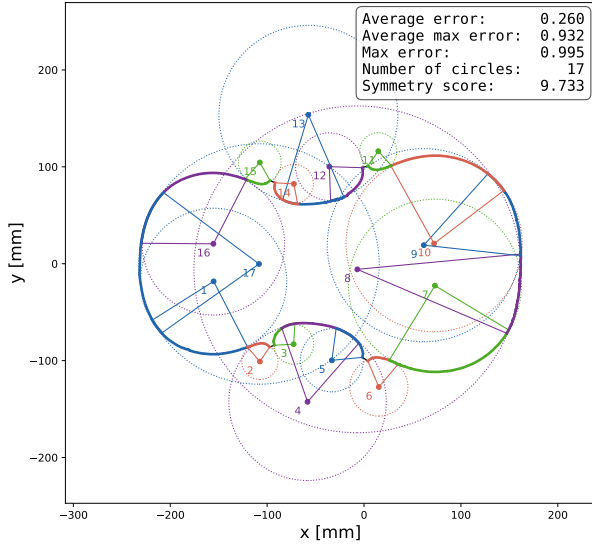
SetCover- ε – All Results

C.1 Results for a tolerance of 1mm

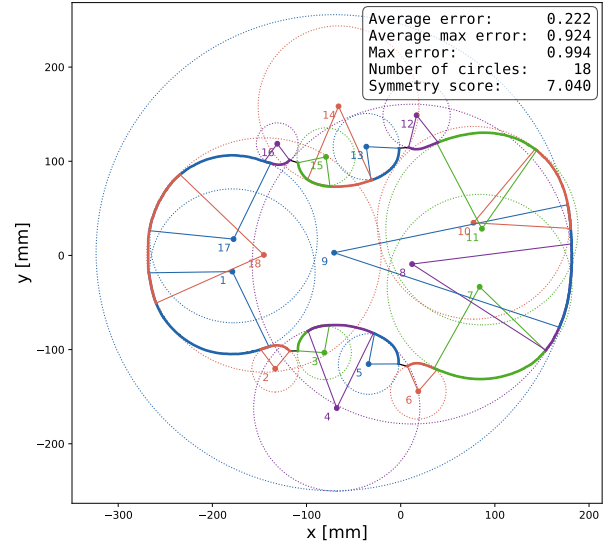




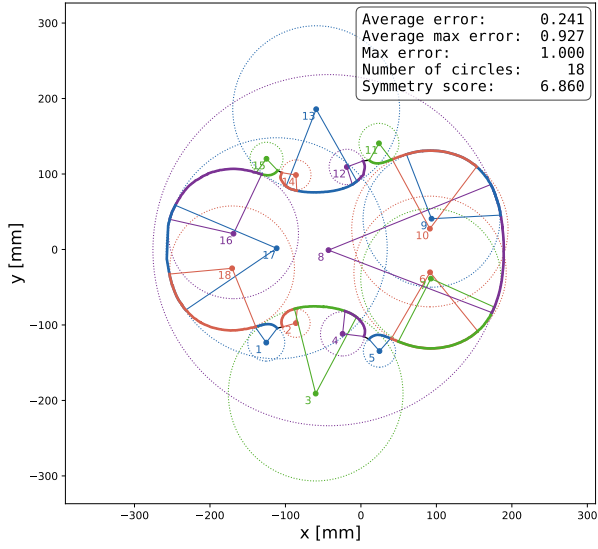
SetCover- ε — Instrument 2833
Tolerance 1.0



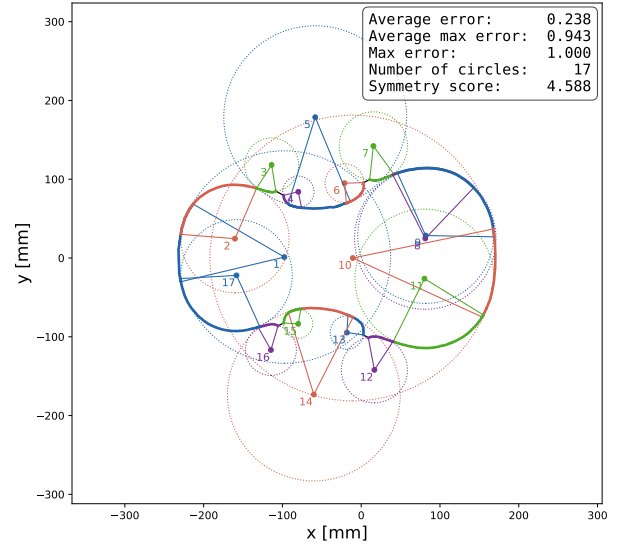
SetCover- ε — Instrument 2835
Tolerance 1.0



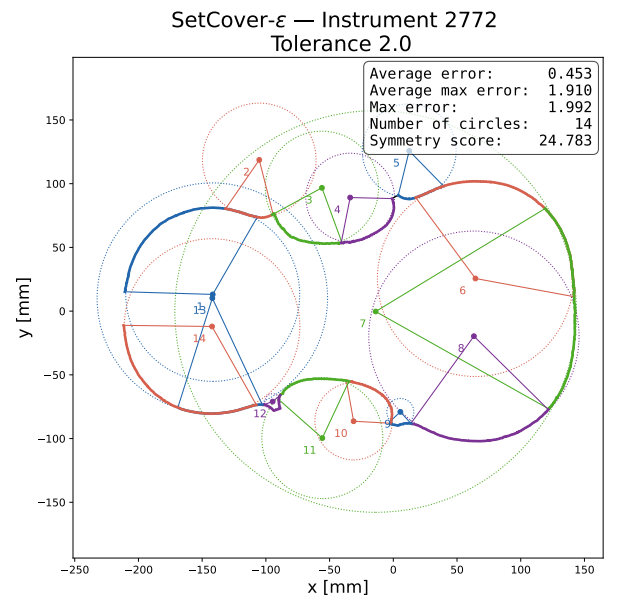
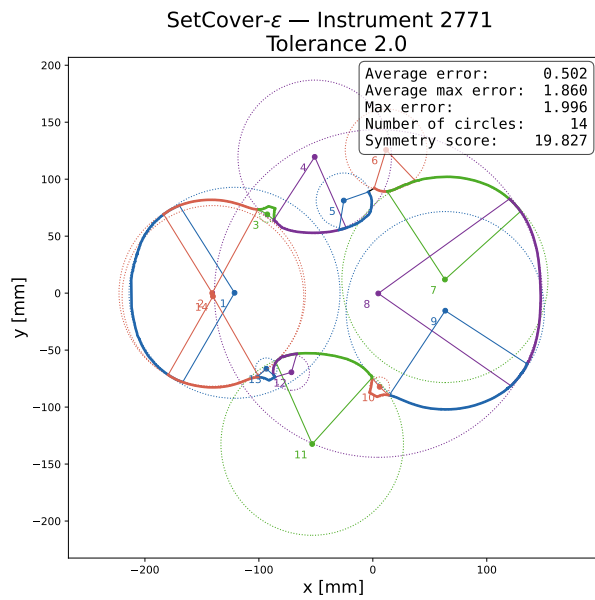
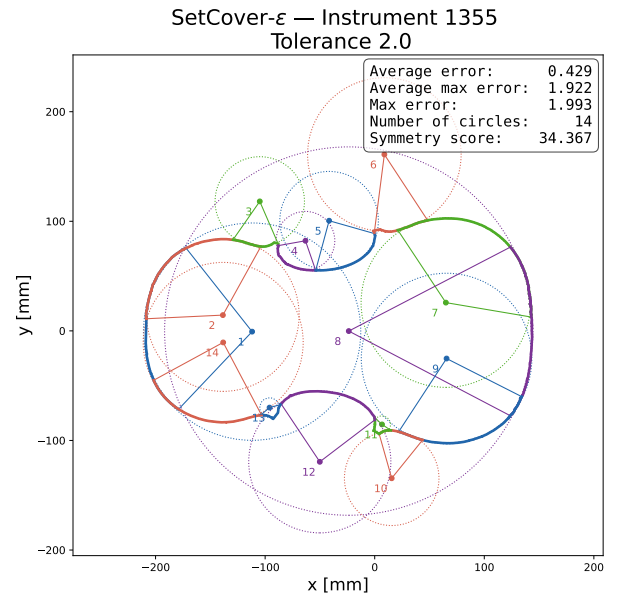
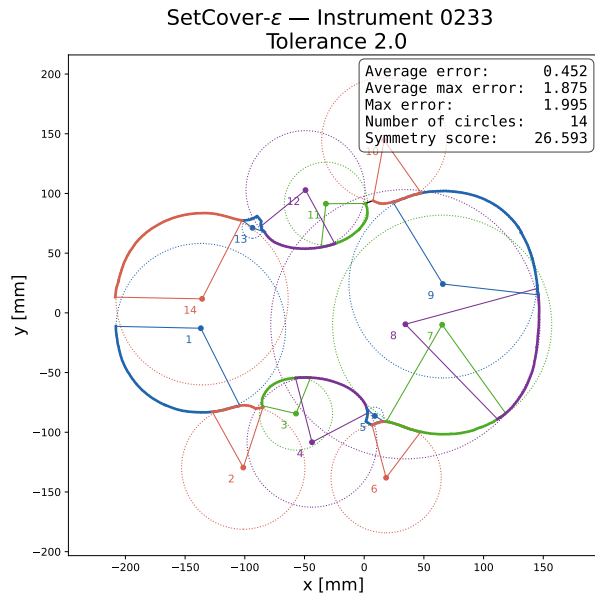
SetCover- ε — Instrument 2836
Tolerance 1.0

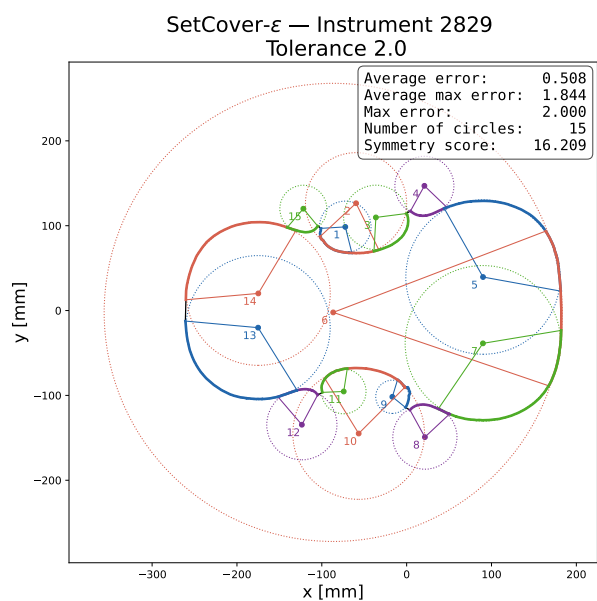
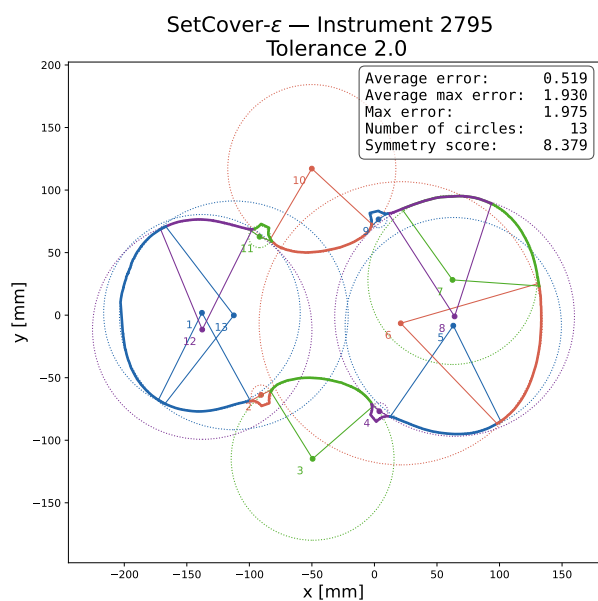
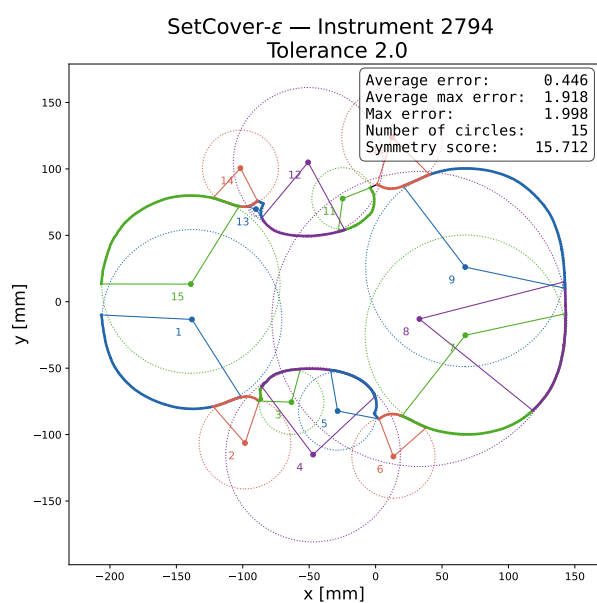
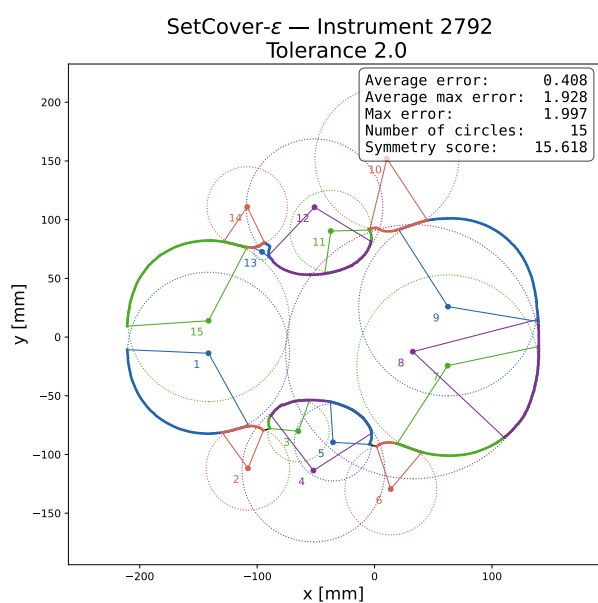
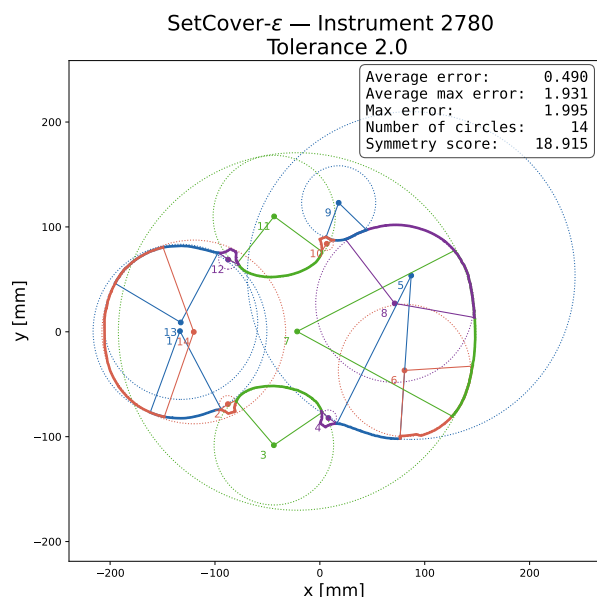
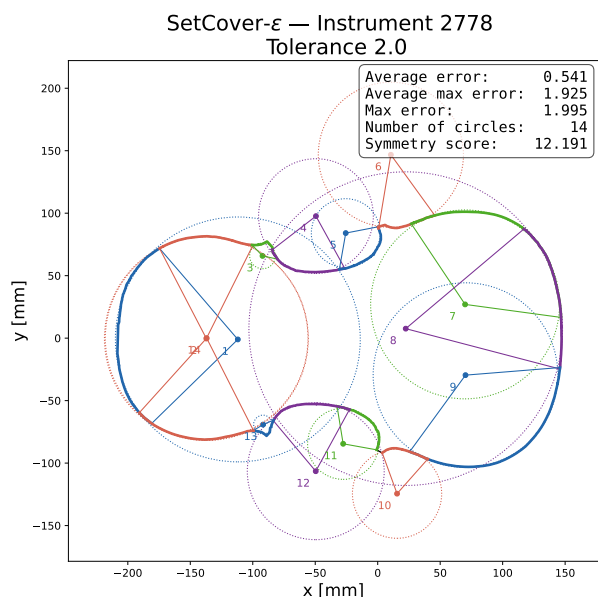


SetCover- ε — Instrument 2847
Tolerance 1.0

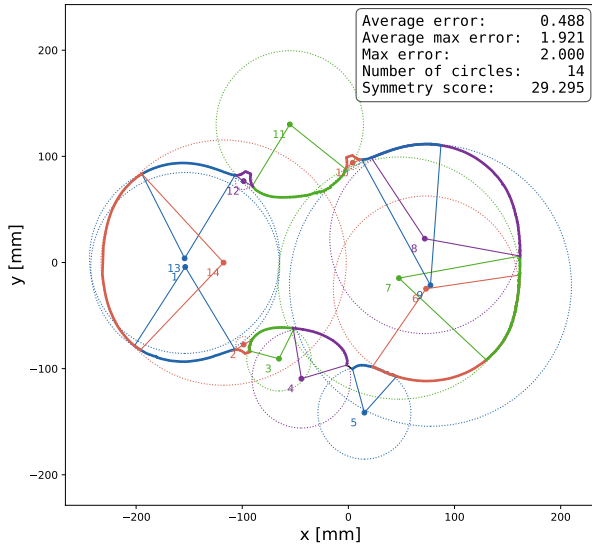


C.2 Results for a tolerance of 2mm

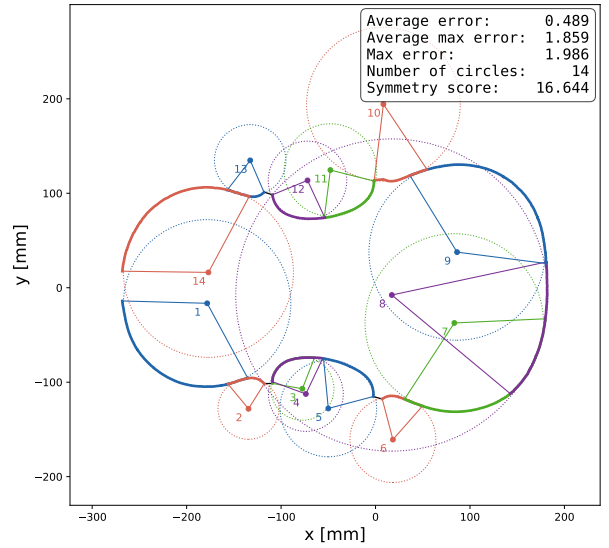




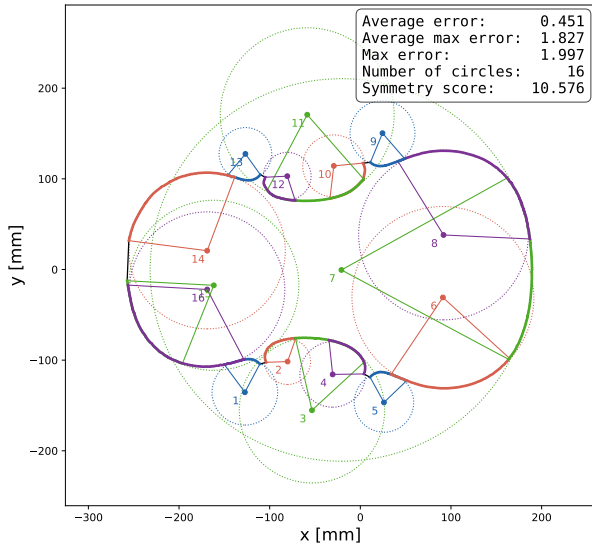
SetCover- ϵ — Instrument 2833
Tolerance 2.0



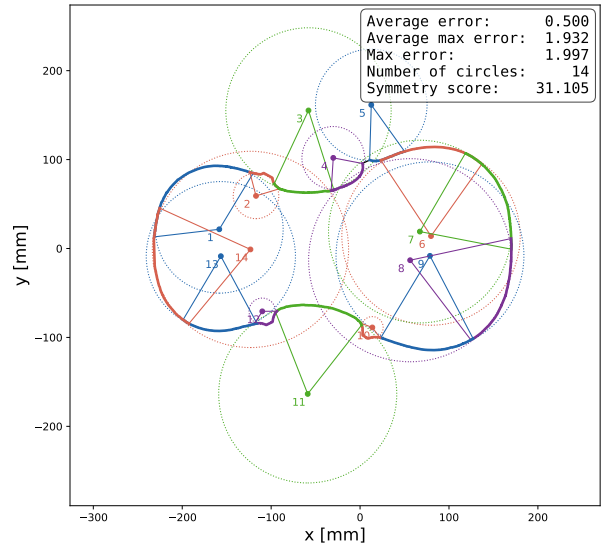
SetCover- ϵ — Instrument 2835
Tolerance 2.0



SetCover- ϵ — Instrument 2836
Tolerance 2.0



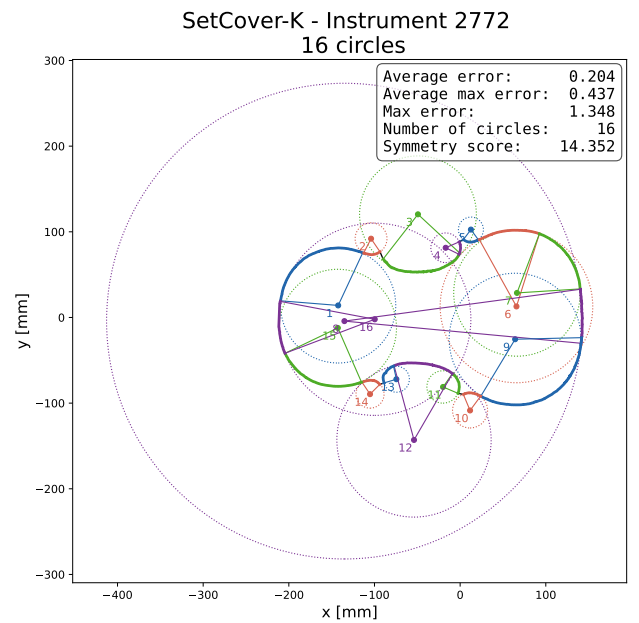
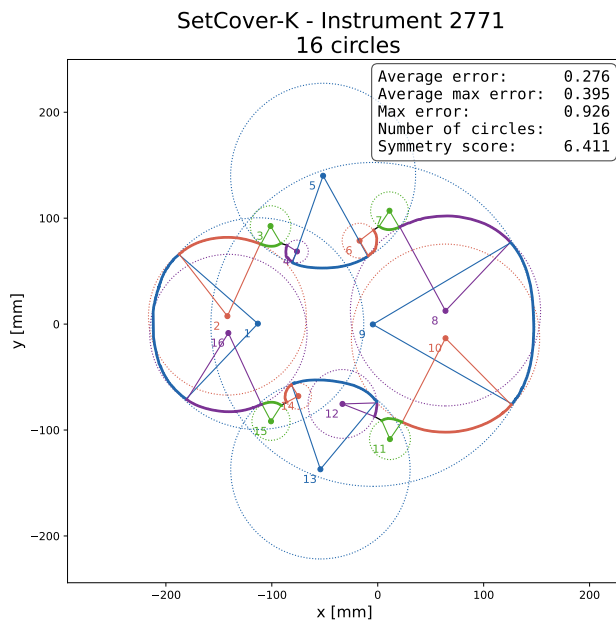
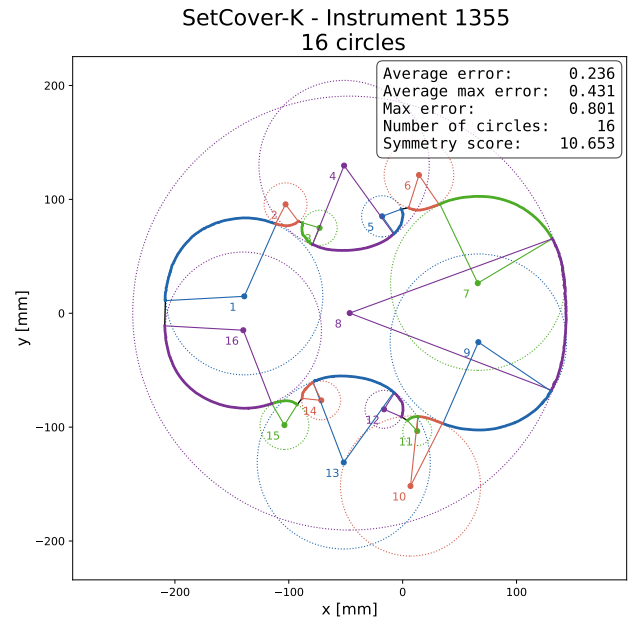
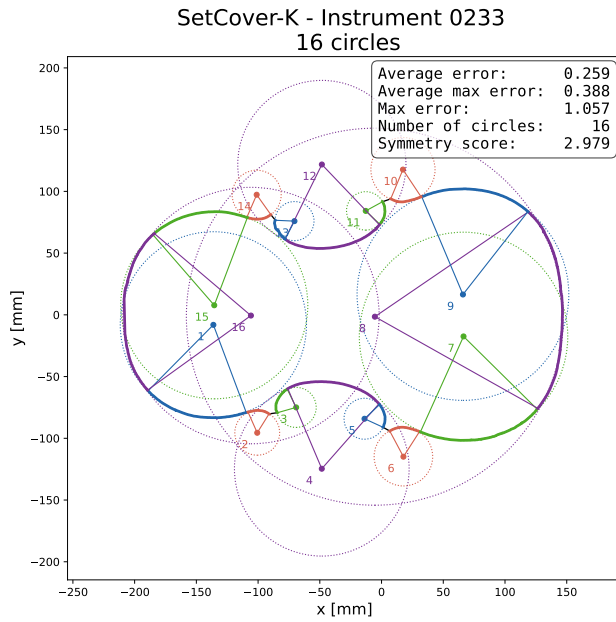
SetCover- ϵ — Instrument 2847
Tolerance 2.0



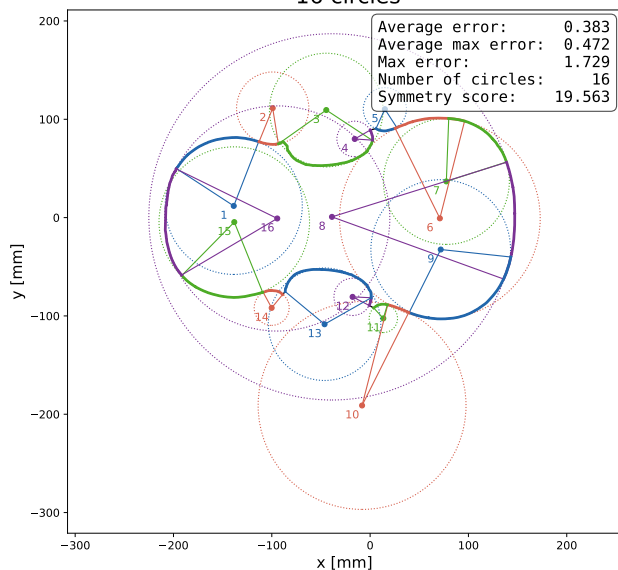
Appendix D

SetCover-K – All Results

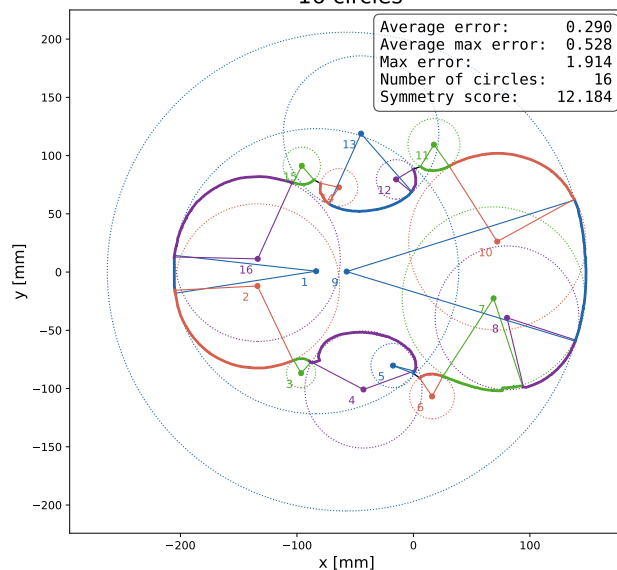
D.1 Results for 16 circles



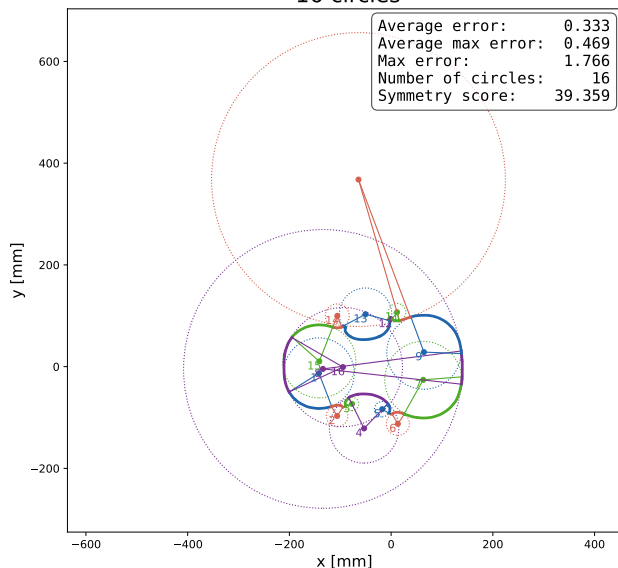
SetCover-K - Instrument 2778
16 circles



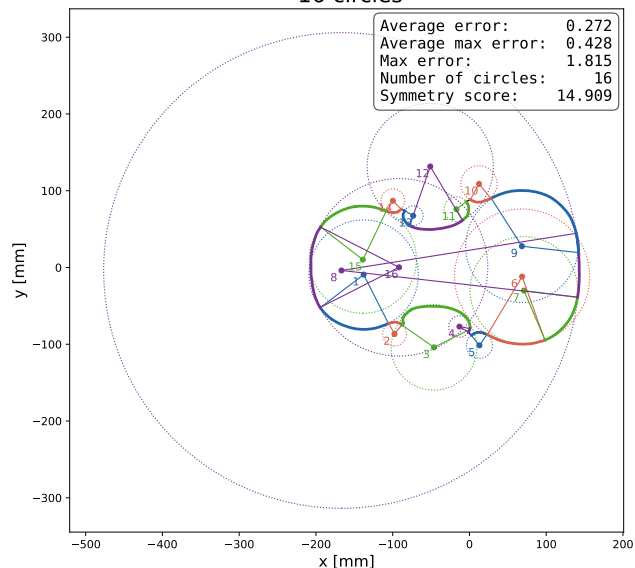
SetCover-K - Instrument 2780
16 circles



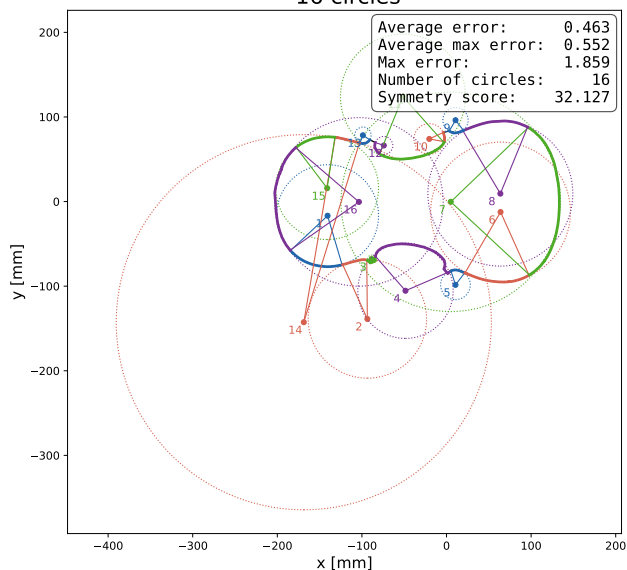
SetCover-K - Instrument 2792
16 circles



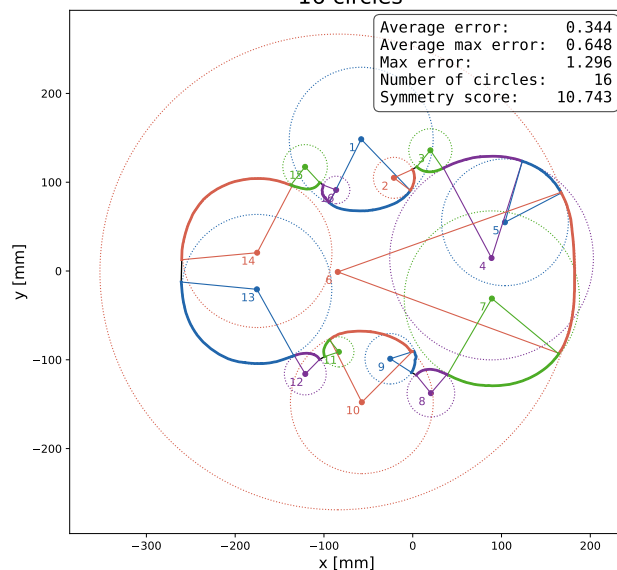
SetCover-K - Instrument 2794
16 circles



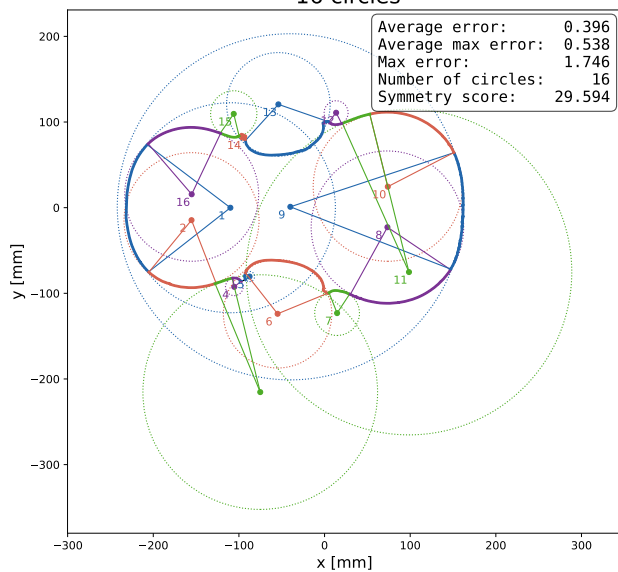
SetCover-K - Instrument 2795
16 circles



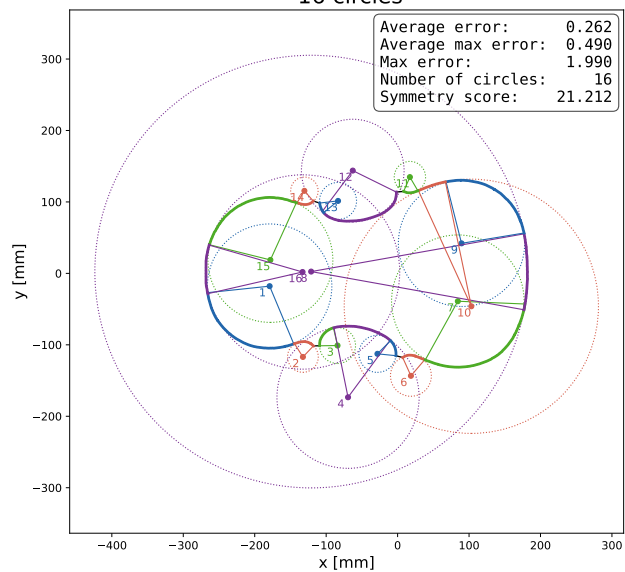
SetCover-K - Instrument 2829
16 circles



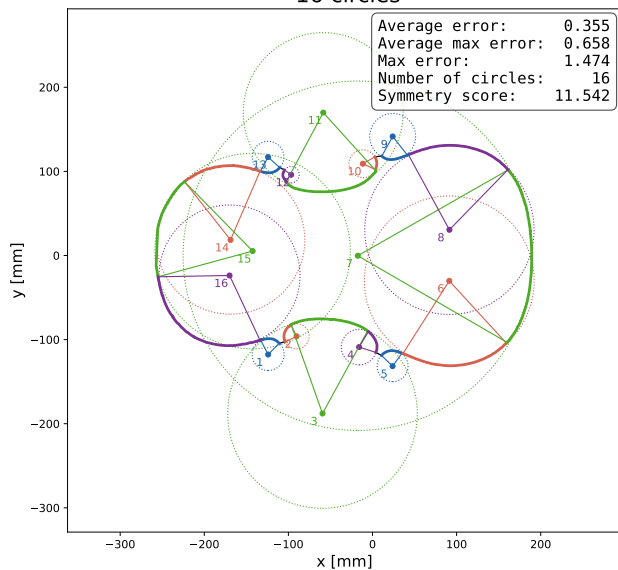
SetCover-K - Instrument 2833
16 circles



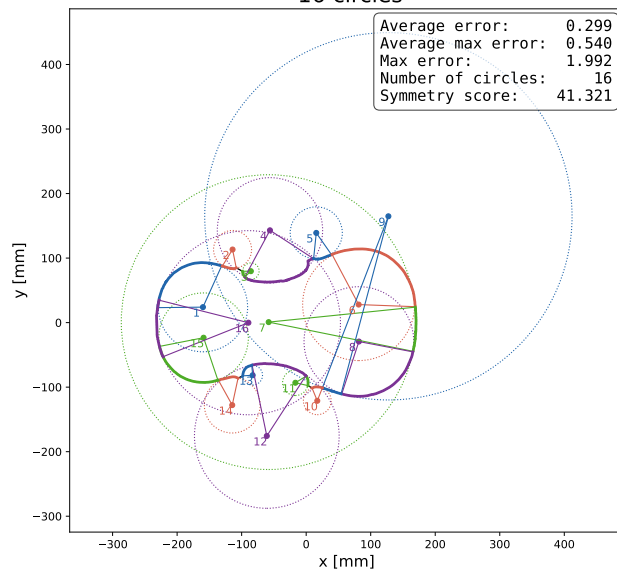
SetCover-K - Instrument 2835
16 circles



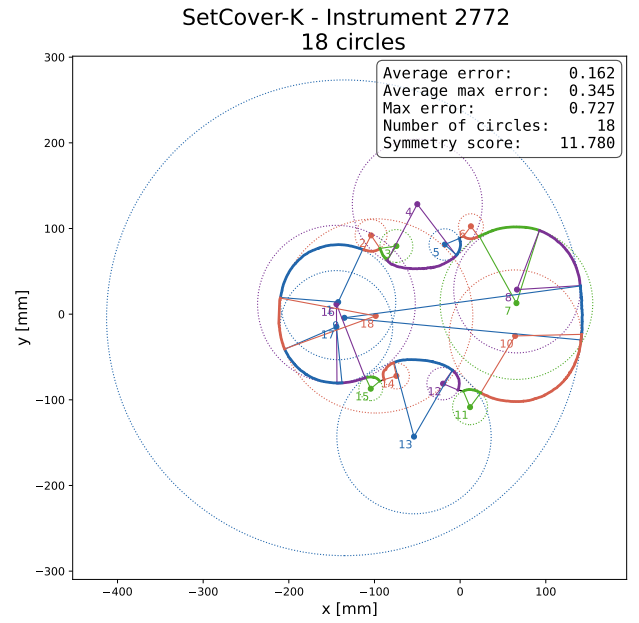
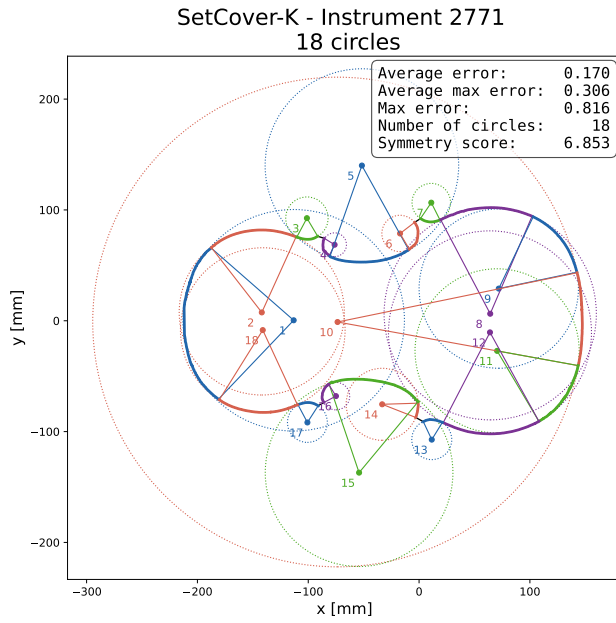
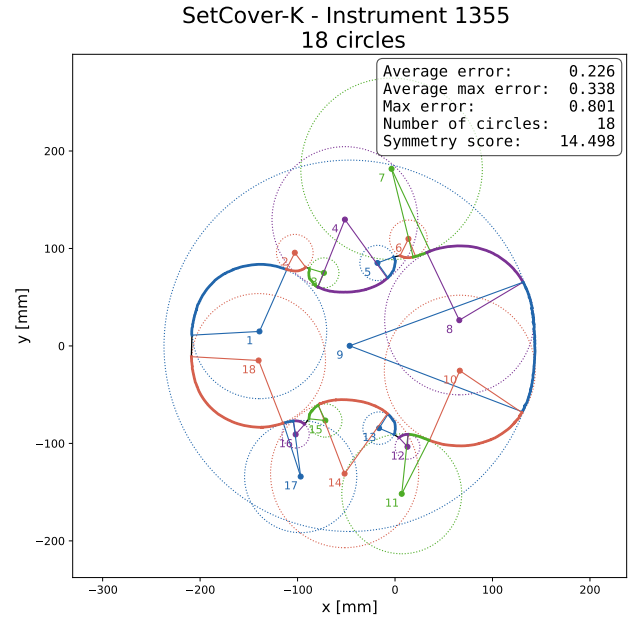
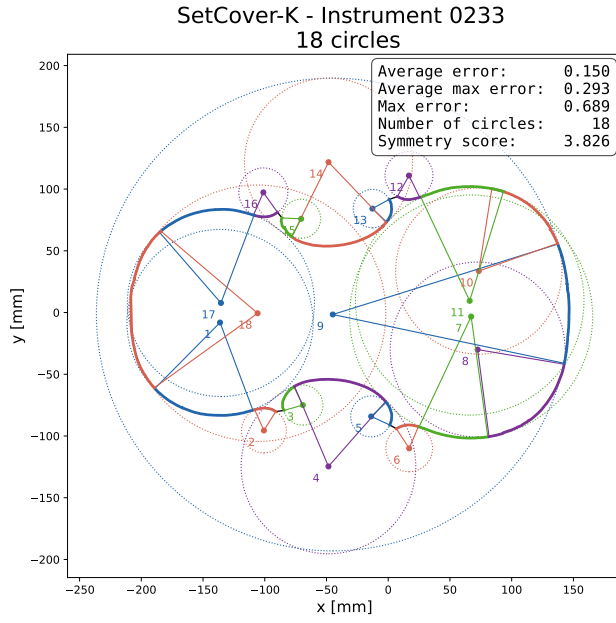
SetCover-K - Instrument 2836
16 circles



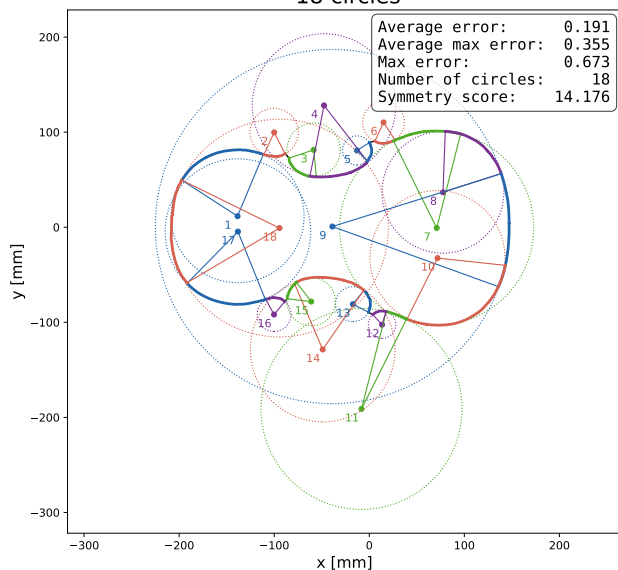
SetCover-K - Instrument 2847
16 circles



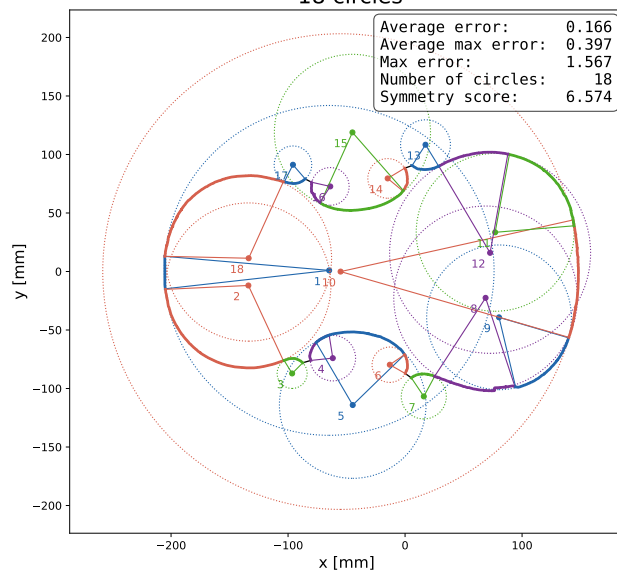
D.2 Results for 18 circles



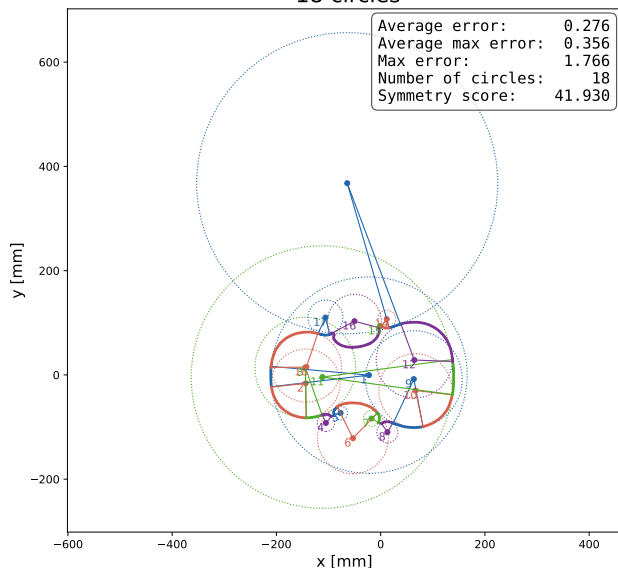
SetCover-K - Instrument 2778
18 circles



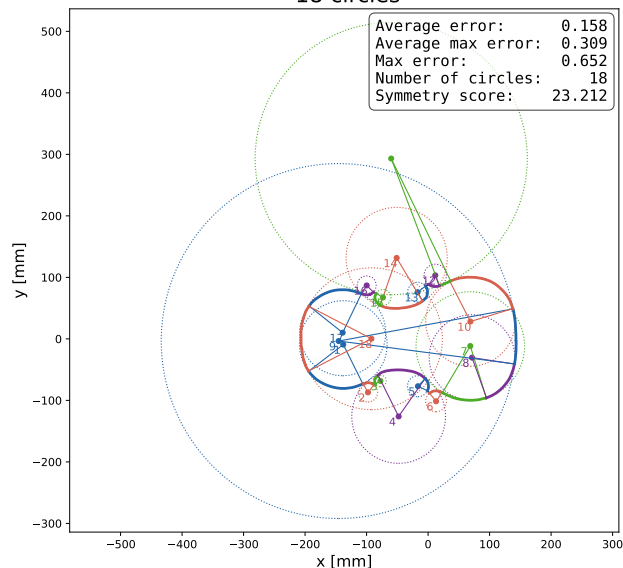
SetCover-K - Instrument 2780
18 circles



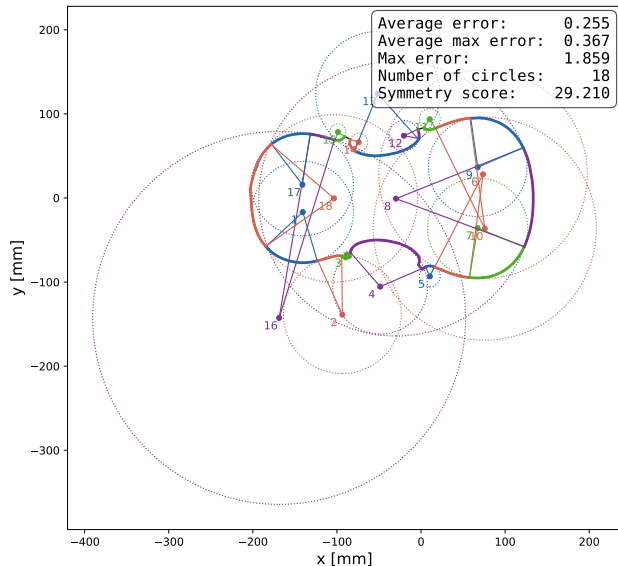
SetCover-K - Instrument 2792
18 circles



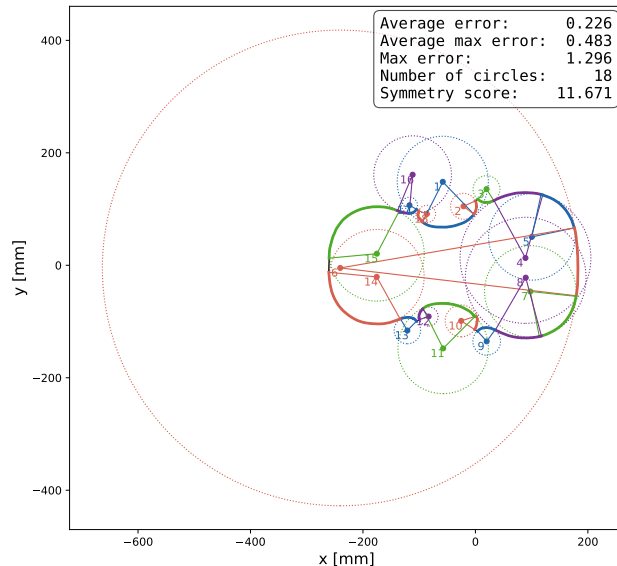
SetCover-K - Instrument 2794
18 circles



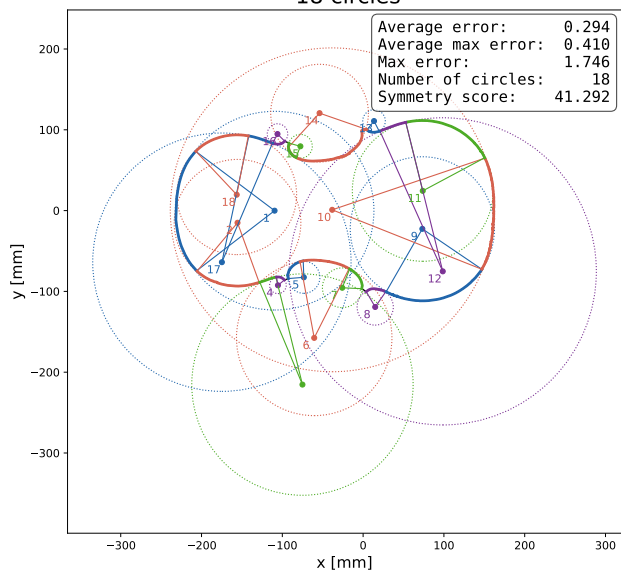
SetCover-K - Instrument 2795
18 circles



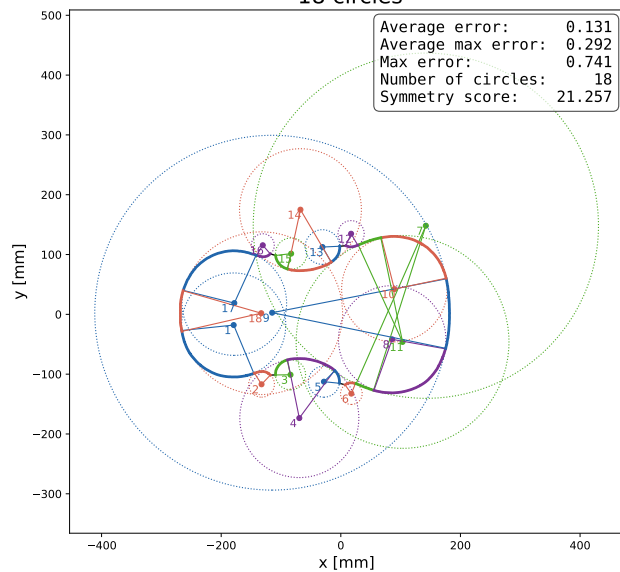
SetCover-K - Instrument 2829
18 circles



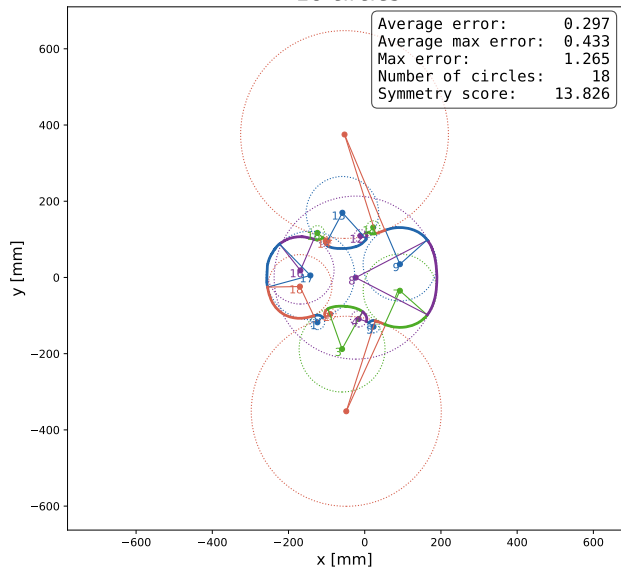
SetCover-K - Instrument 2833
18 circles



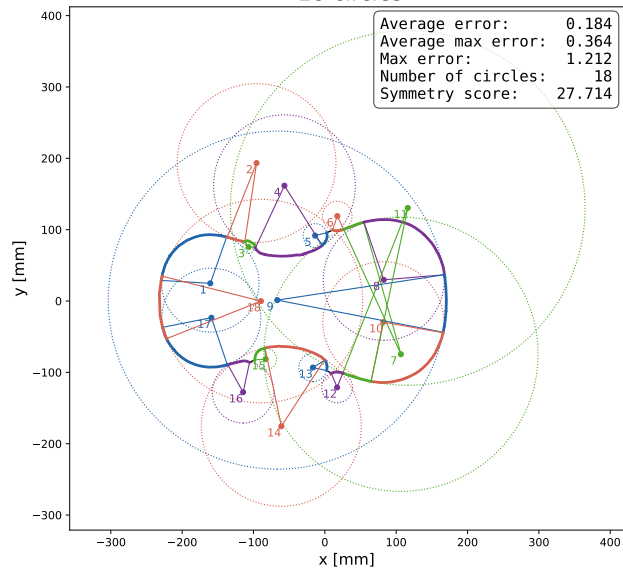
SetCover-K - Instrument 2835
18 circles



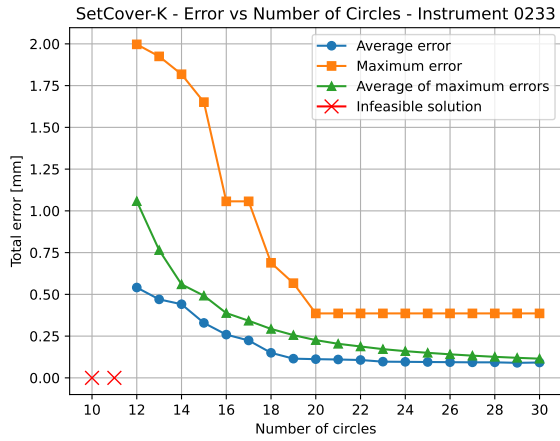
SetCover-K - Instrument 2836
18 circles



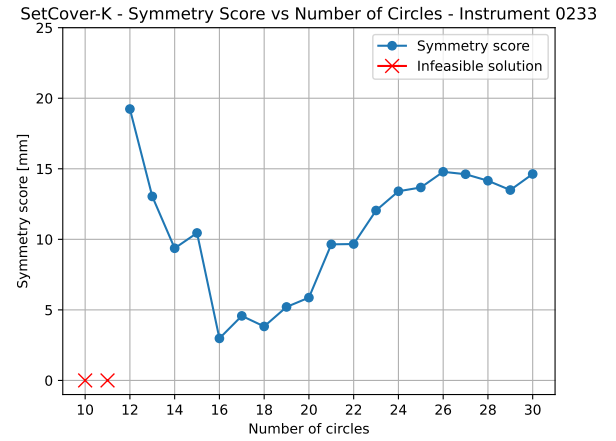
SetCover-K - Instrument 2847
18 circles



D.3 Number of circles study

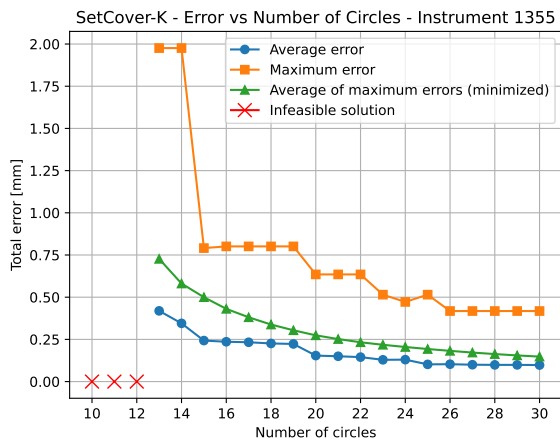


(a) Number of circles vs Error Metrics

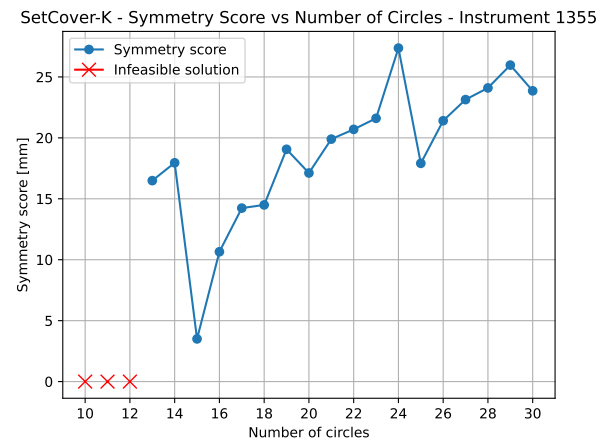


(b) Number of circles vs Symmetry score

Figure D.1: Number of circles study – SetCover-K – Instrument 0233

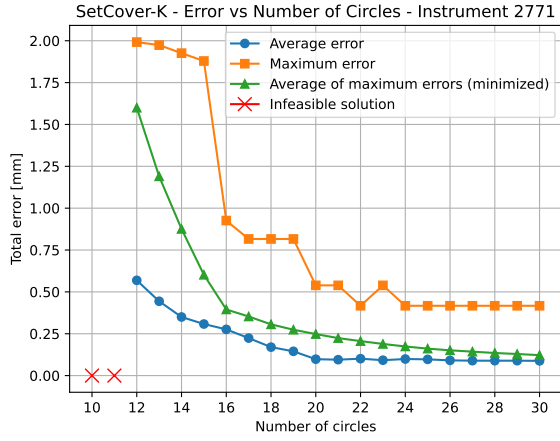


(a) Number of circles vs Error Metrics

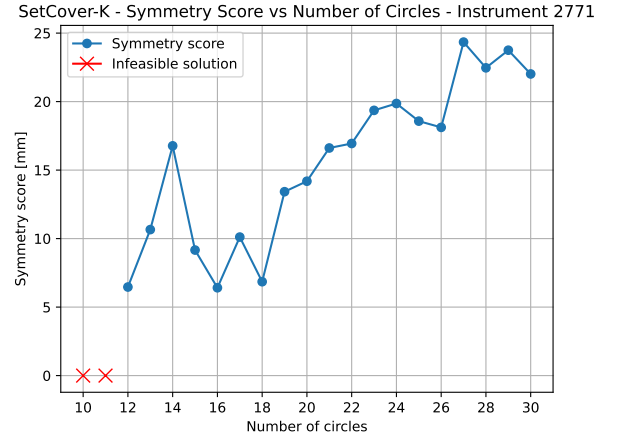


(b) Number of circles vs Symmetry score

Figure D.2: Number of circles study – SetCover-K – Instrument 1355

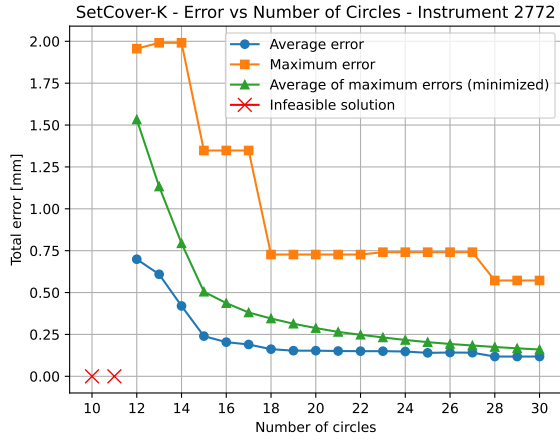


(a) Number of circles vs Error Metrics

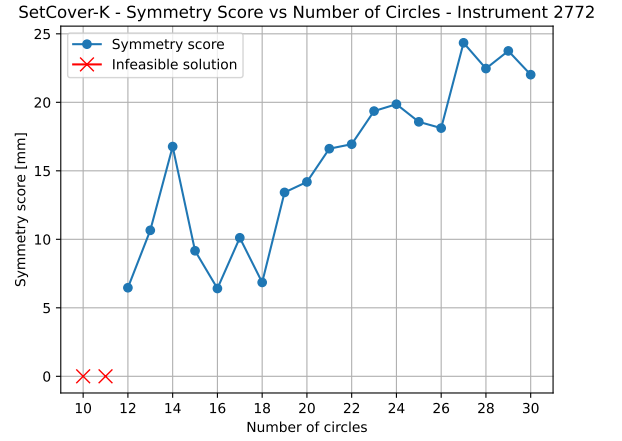


(b) Number of circles vs Symmetry score

Figure D.3: Number of circles study – SetCover-K – Instrument 2771

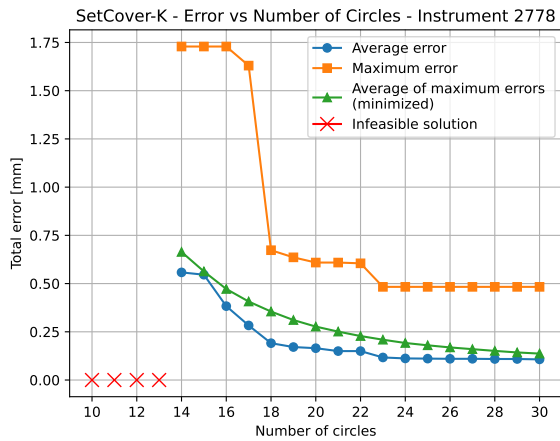


(a) Number of circles vs Error Metrics

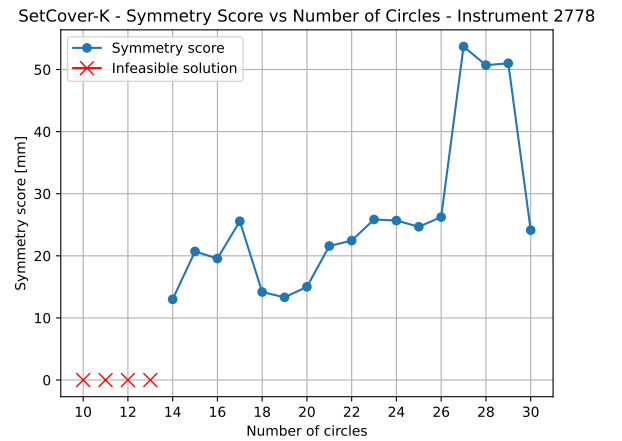


(b) Number of circles vs Symmetry score

Figure D.4: Number of circles study – SetCover-K – Instrument 2772

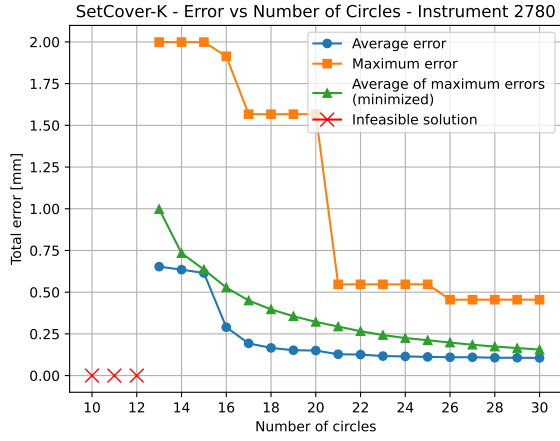


(a) Number of circles vs Error Metrics

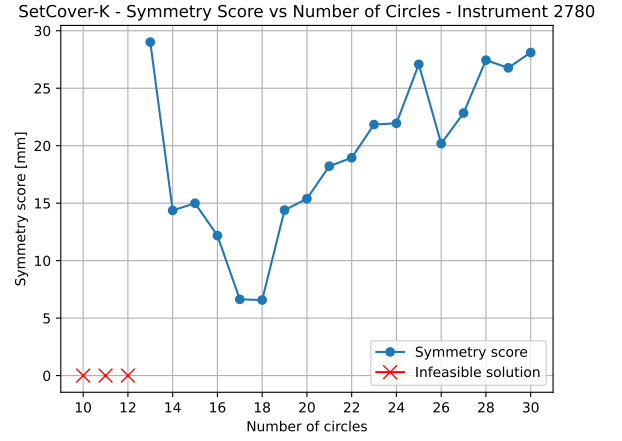


(b) Number of circles vs Symmetry score

Figure D.5: Number of circles study – SetCover-K – Instrument 2778

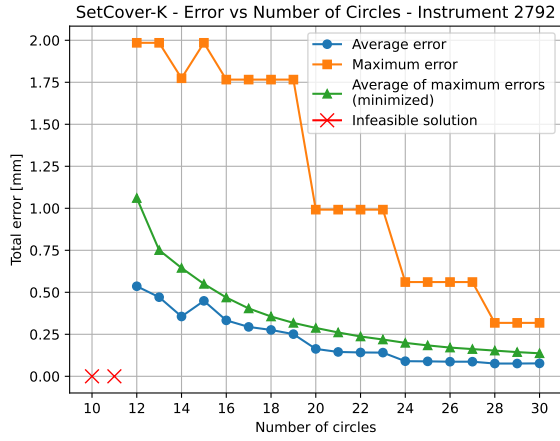


(a) Number of circles vs Error Metrics

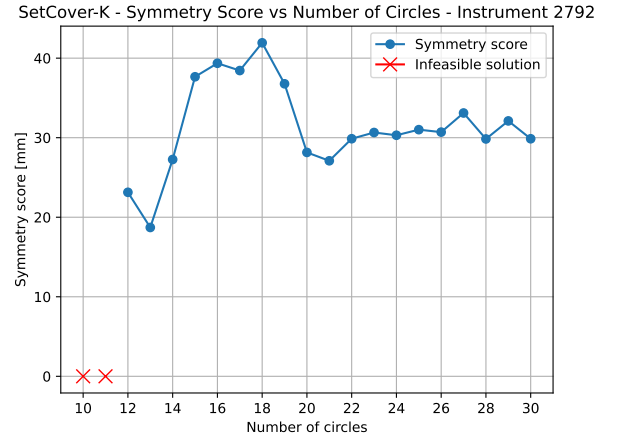


(b) Number of circles vs Symmetry score

Figure D.6: Number of circles study – SetCover-K – Instrument 2780

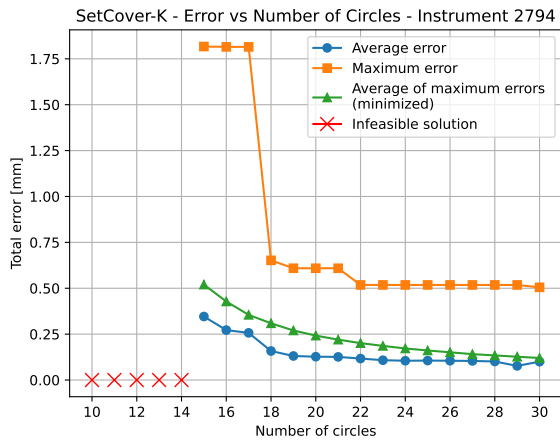


(a) Number of circles vs Error Metrics

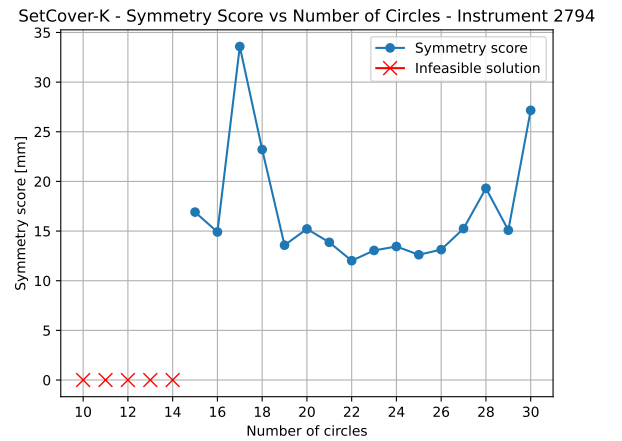


(b) Number of circles vs Symmetry score

Figure D.7: Number of circles study – SetCover-K – Instrument 2792

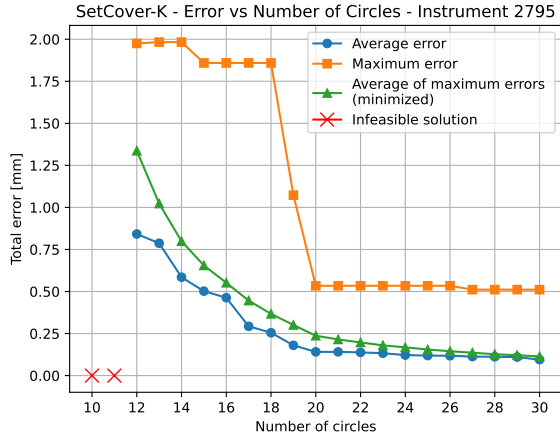


(a) Number of circles vs Error Metrics

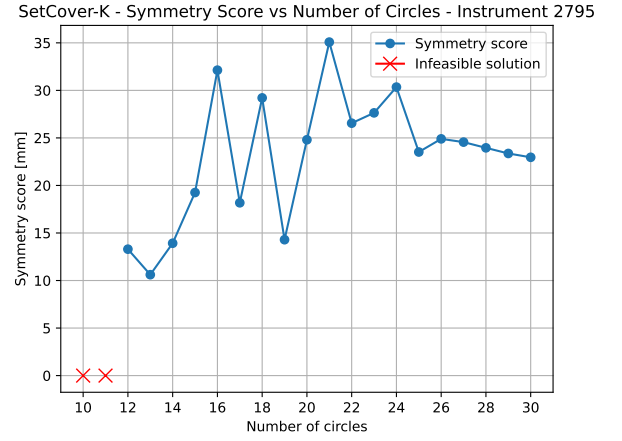


(b) Number of circles vs Symmetry score

Figure D.8: Number of circles study – SetCover-K – Instrument 2794

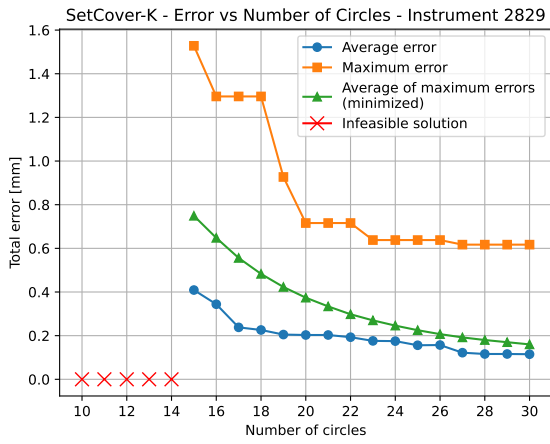


(a) Number of circles vs Error Metrics

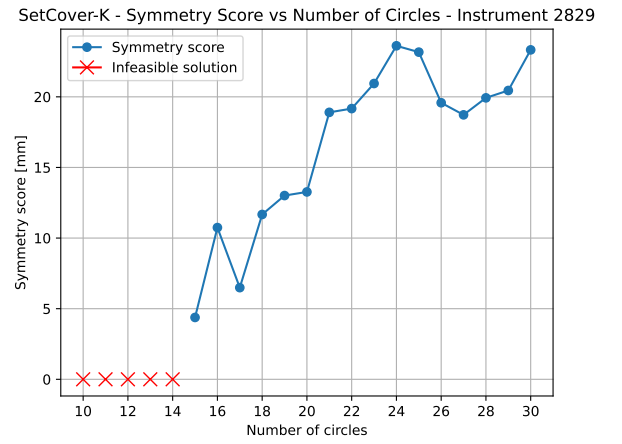


(b) Number of circles vs Symmetry score

Figure D.9: Number of circles study – SetCover-K – Instrument 2795

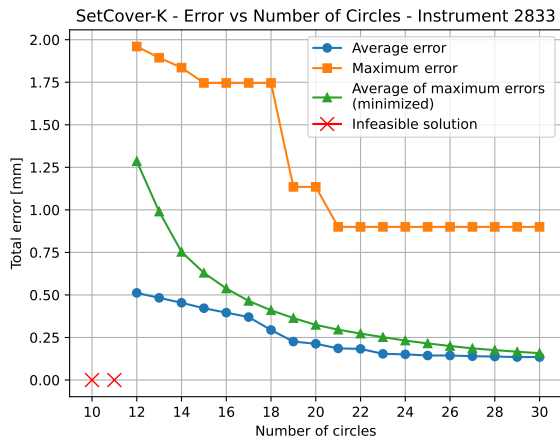


(a) Number of circles vs Error Metrics

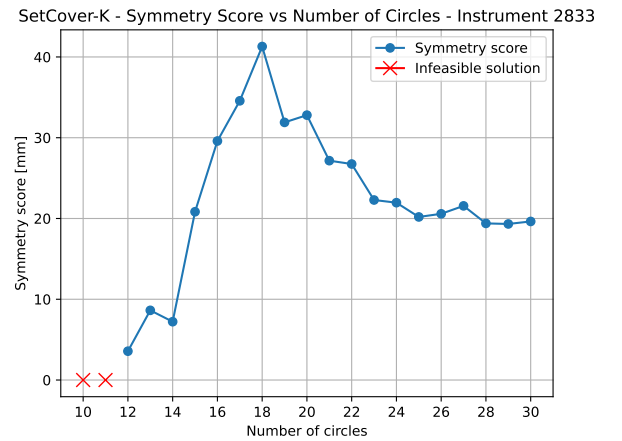


(b) Number of circles vs Symmetry score

Figure D.10: Number of circles study – SetCover-K – Instrument 2829

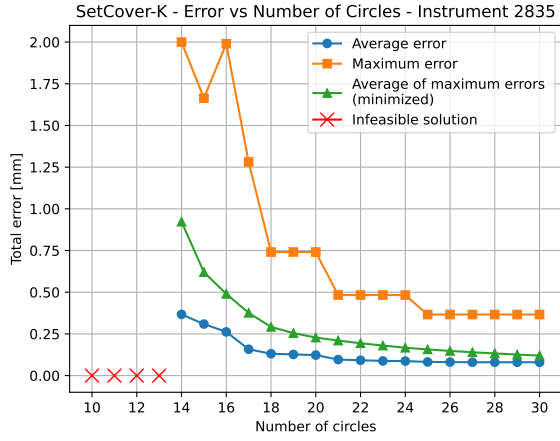


(a) Number of circles vs Error Metrics

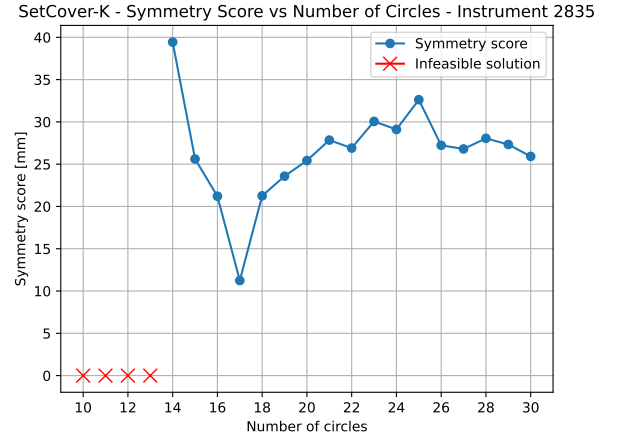


(b) Number of circles vs Symmetry score

Figure D.11: Number of circles study – SetCover-K – Instrument 2833

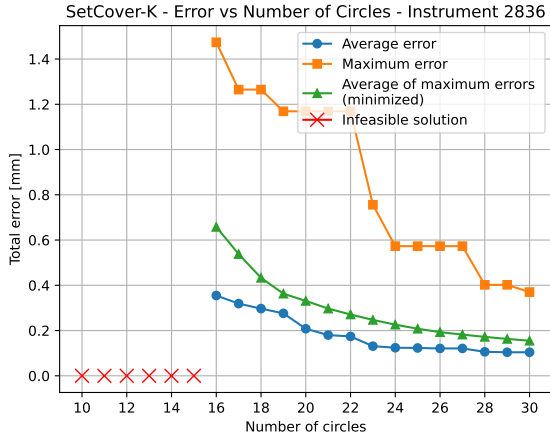


(a) Number of circles vs Error Metrics

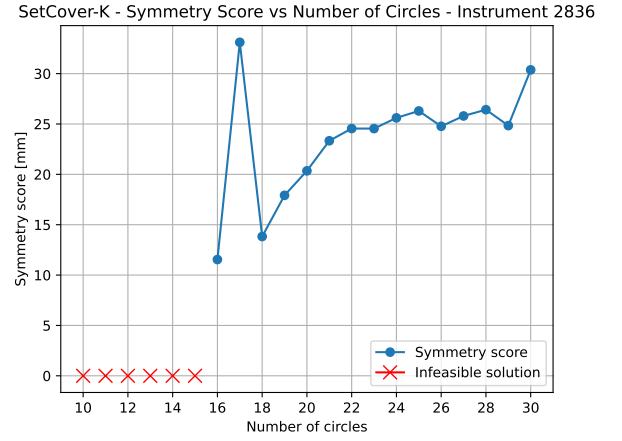


(b) Number of circles vs Symmetry score

Figure D.12: Number of circles study – SetCover-K – Instrument 2835

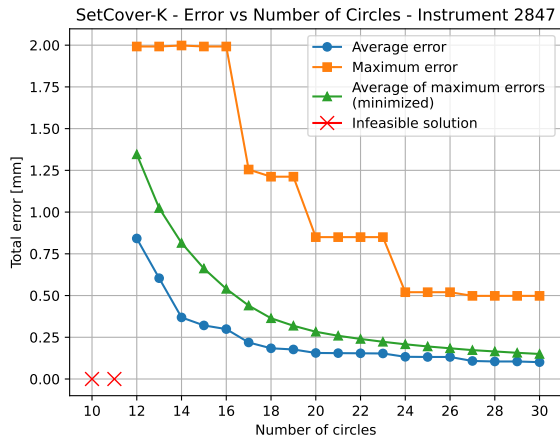


(a) Number of circles vs Error Metrics

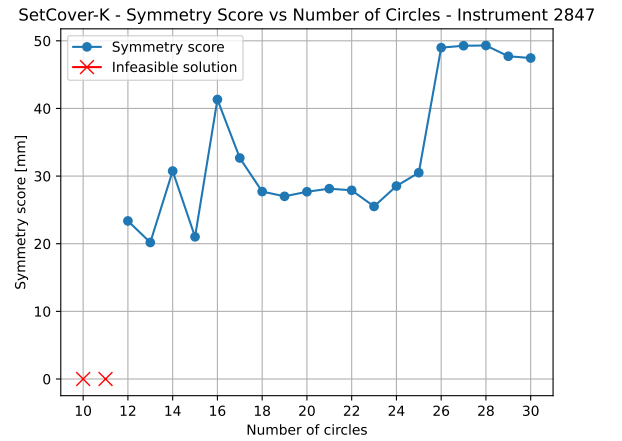


(b) Number of circles vs Symmetry score

Figure D.13: Number of circles study – SetCover-K – Instrument 2836



(a) Number of circles vs Error Metrics



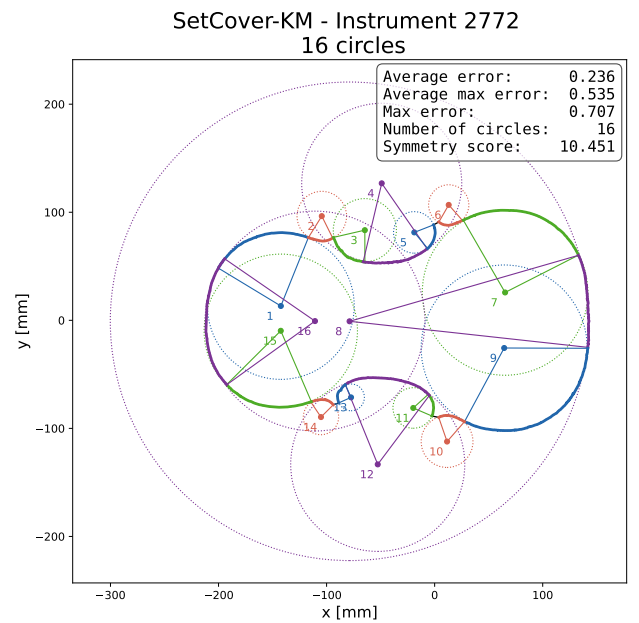
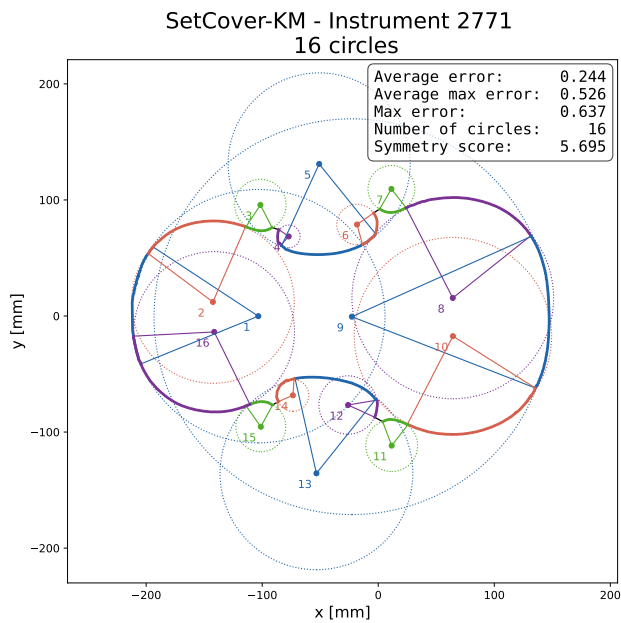
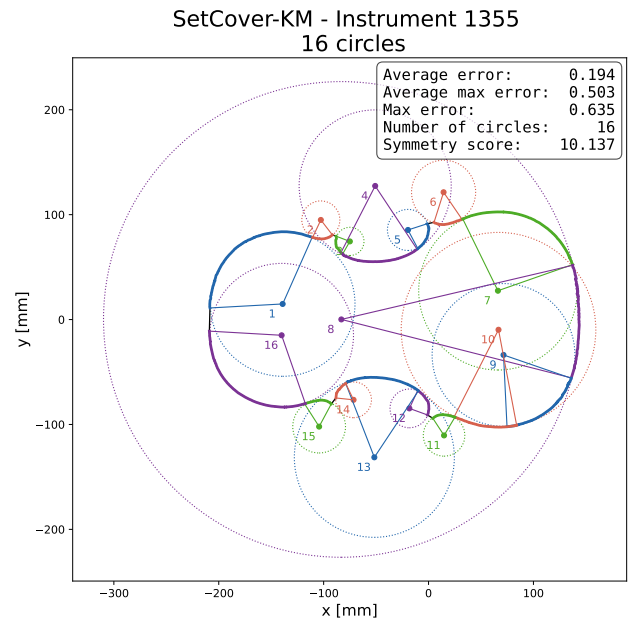
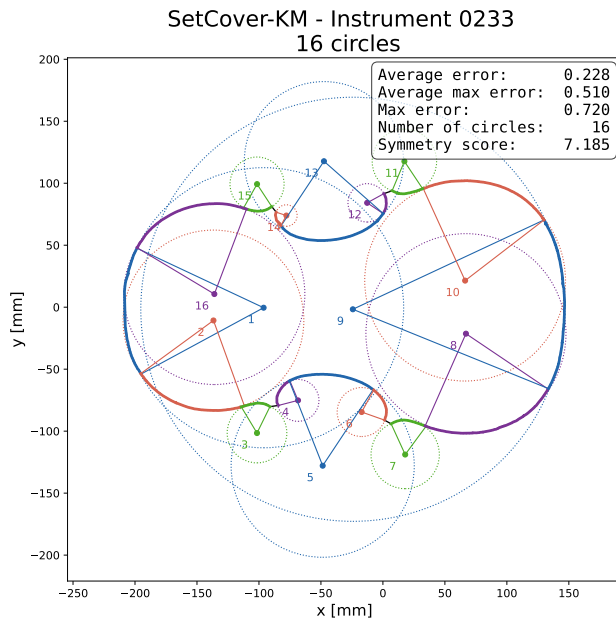
(b) Number of circles vs Symmetry score

Figure D.14: Number of circles study – SetCover-K – Instrument 2847

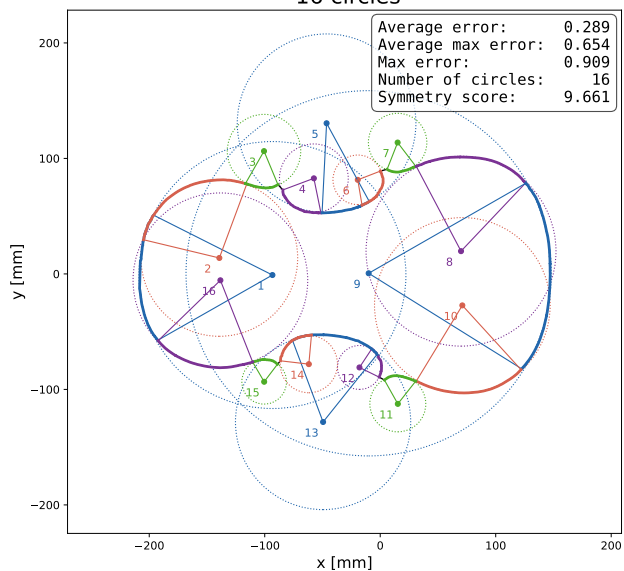
Appendix E

SetCover-KM – All Results

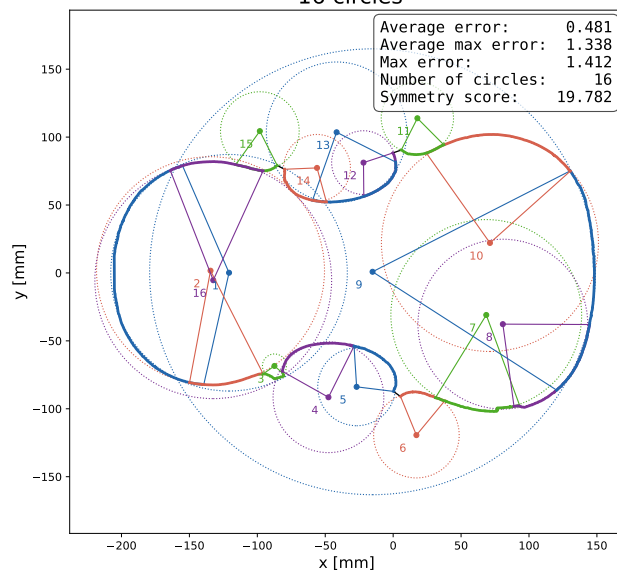
E.1 Results for 16 circles



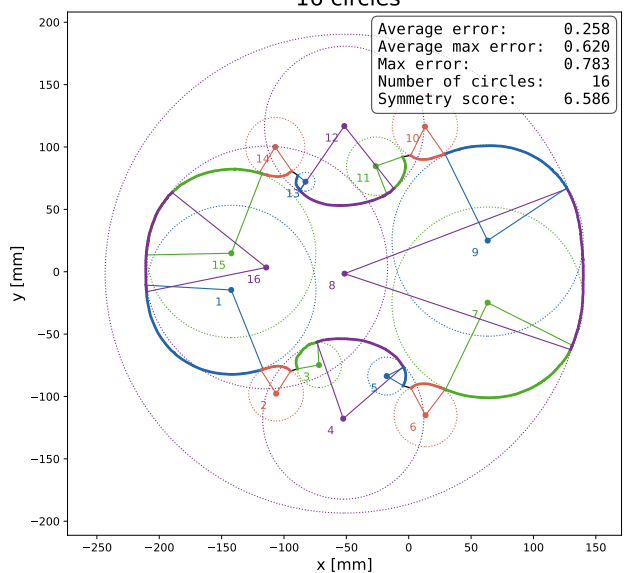
SetCover-KM - Instrument 2778
16 circles



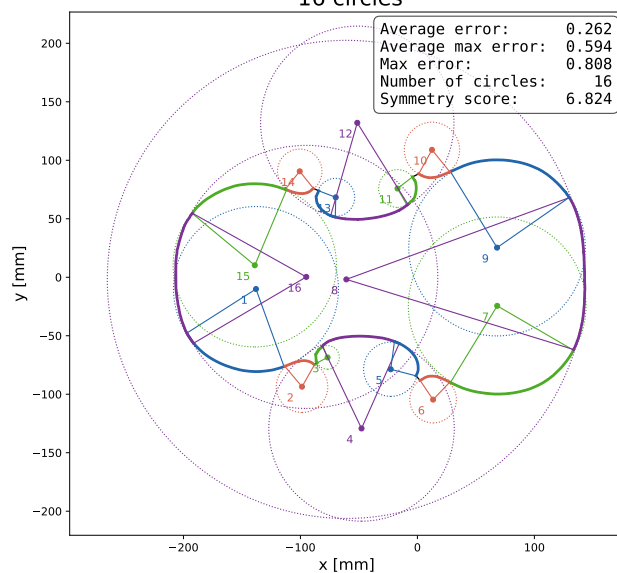
SetCover-KM - Instrument 2780
16 circles



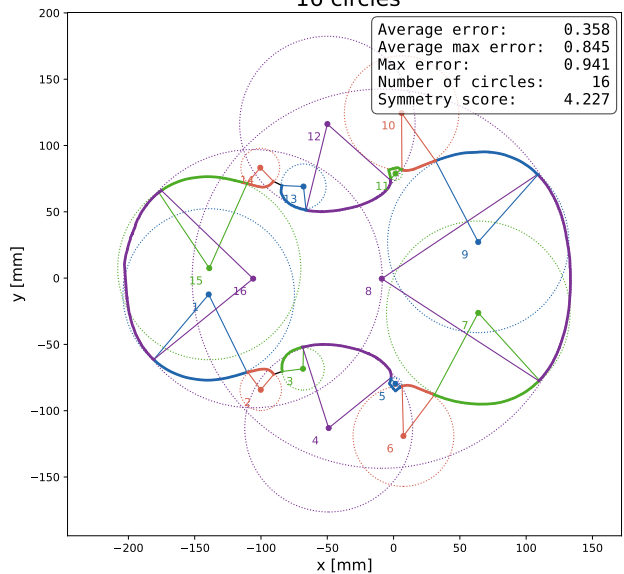
SetCover-KM - Instrument 2792
16 circles



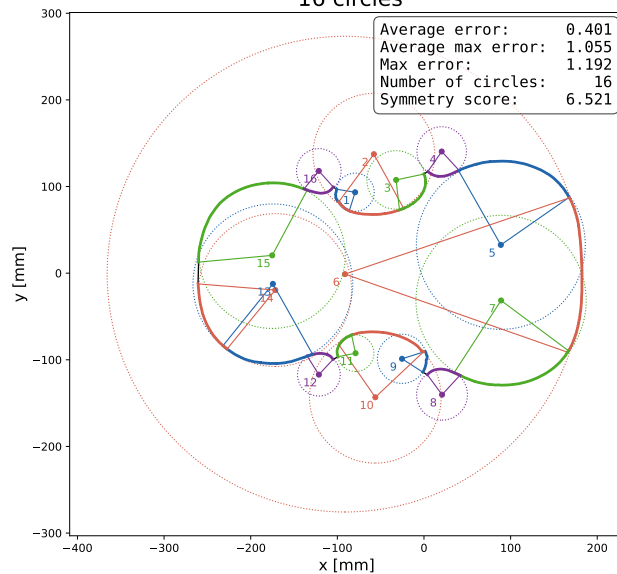
SetCover-KM - Instrument 2794
16 circles



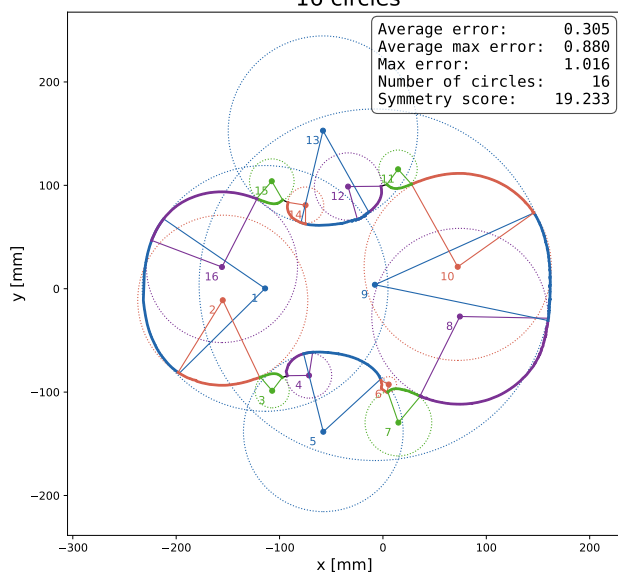
SetCover-KM - Instrument 2795
16 circles



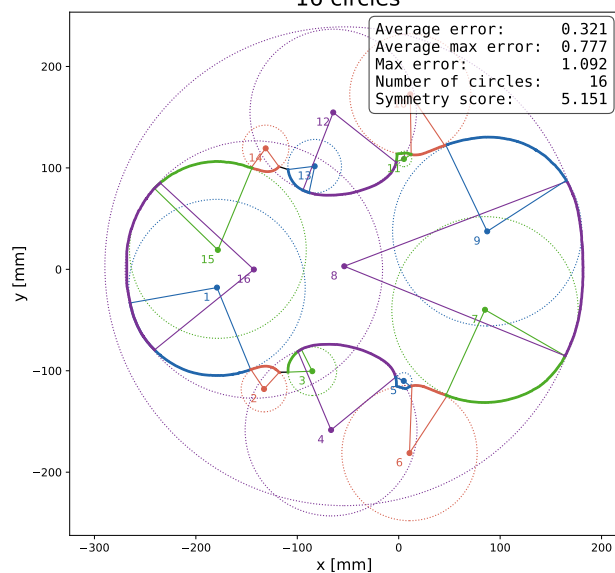
SetCover-KM - Instrument 2829
16 circles



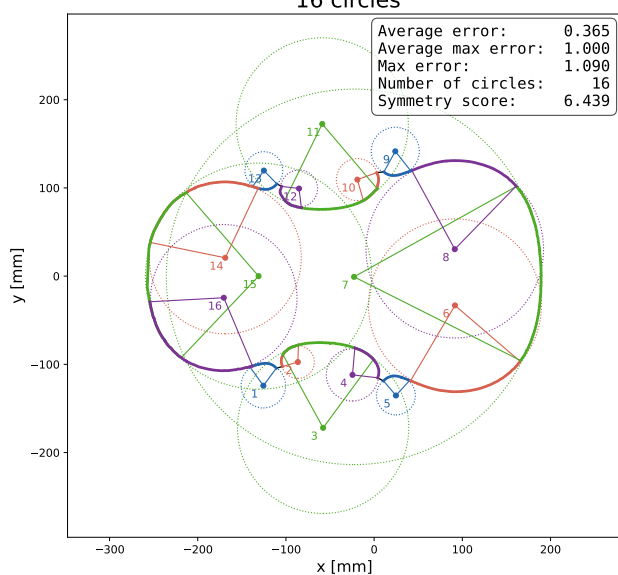
SetCover-KM - Instrument 2833
16 circles



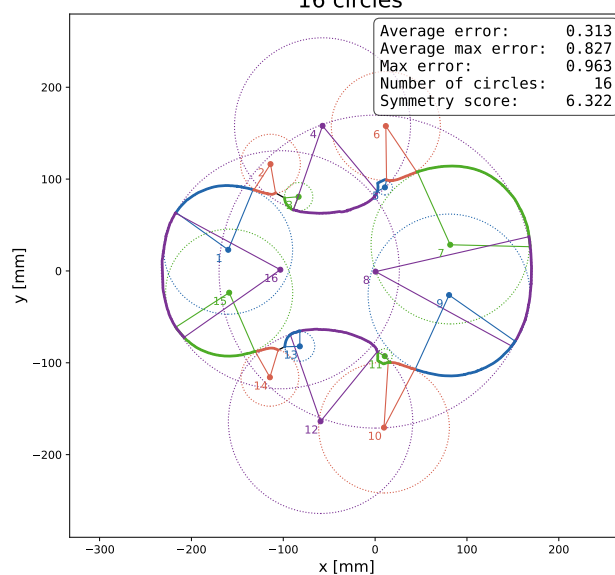
SetCover-KM - Instrument 2835
16 circles



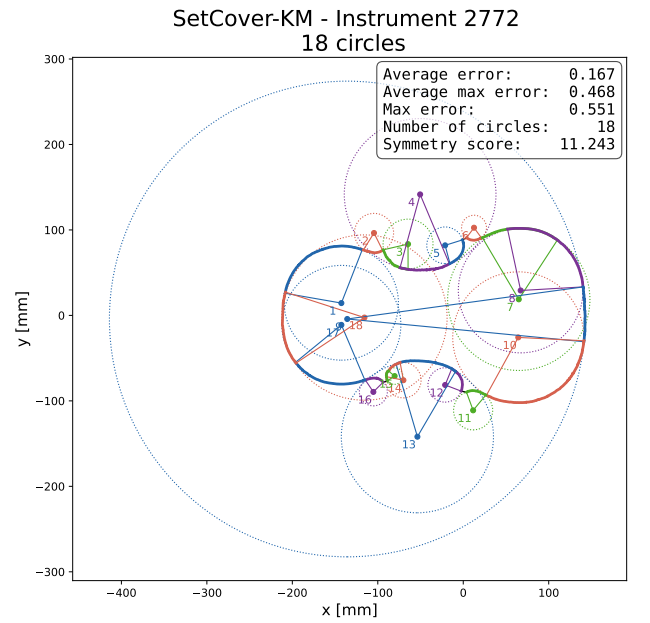
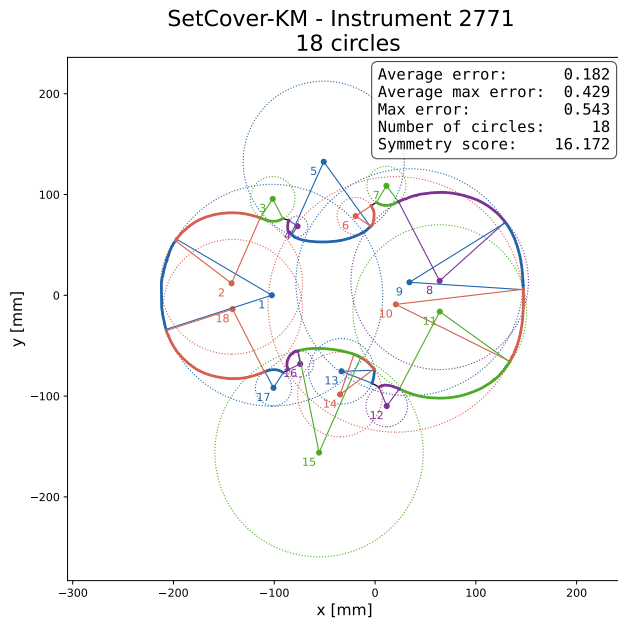
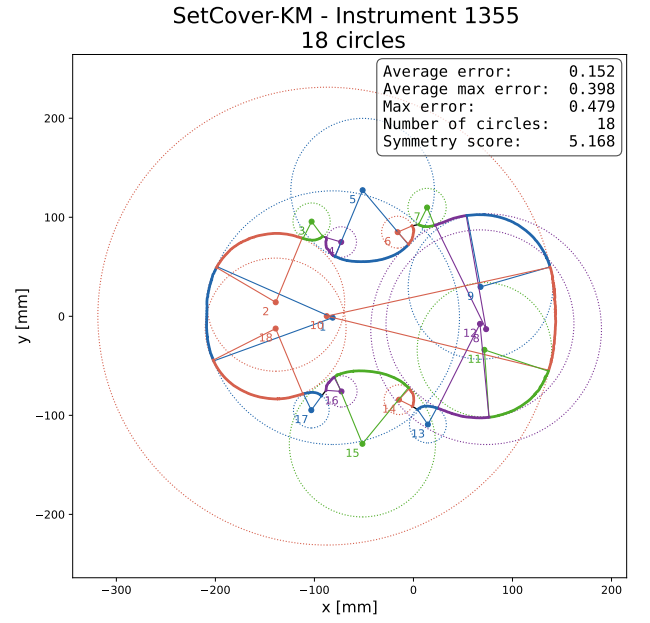
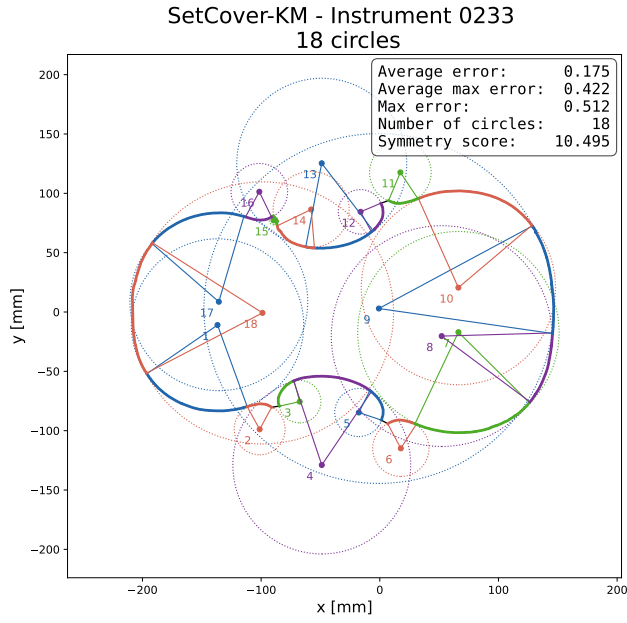
SetCover-KM - Instrument 2836
16 circles



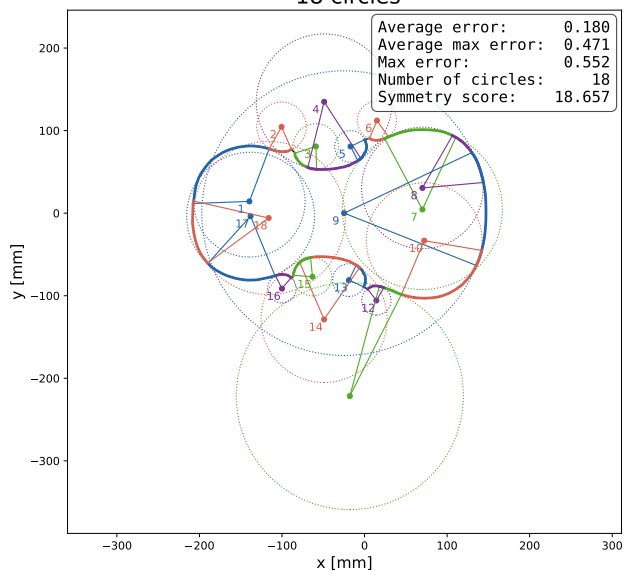
SetCover-KM - Instrument 2847
16 circles



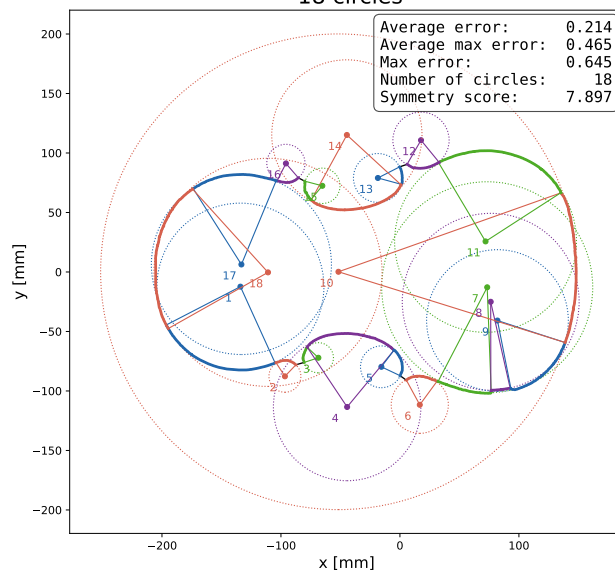
E.2 Results for 18 circles



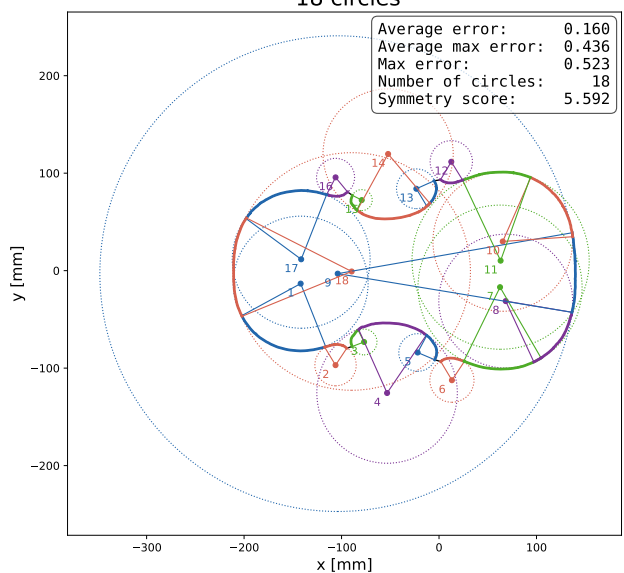
SetCover-KM - Instrument 2778
18 circles



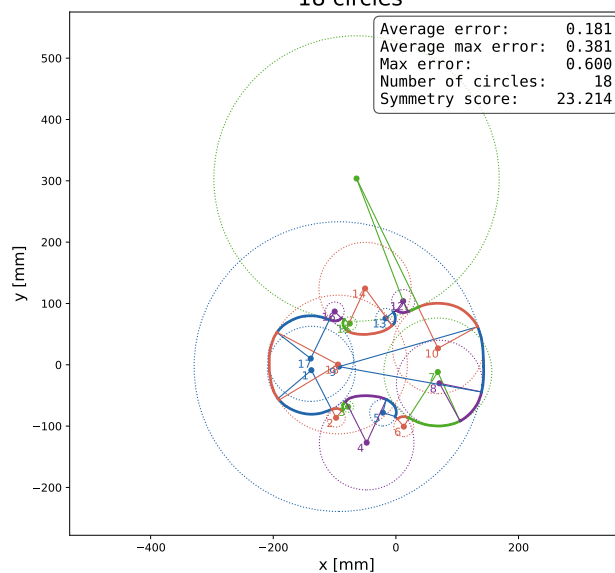
SetCover-KM - Instrument 2780
18 circles



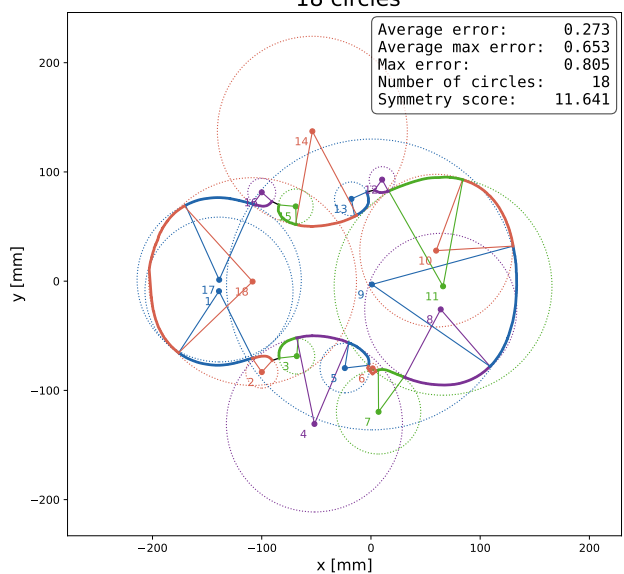
SetCover-KM - Instrument 2792
18 circles



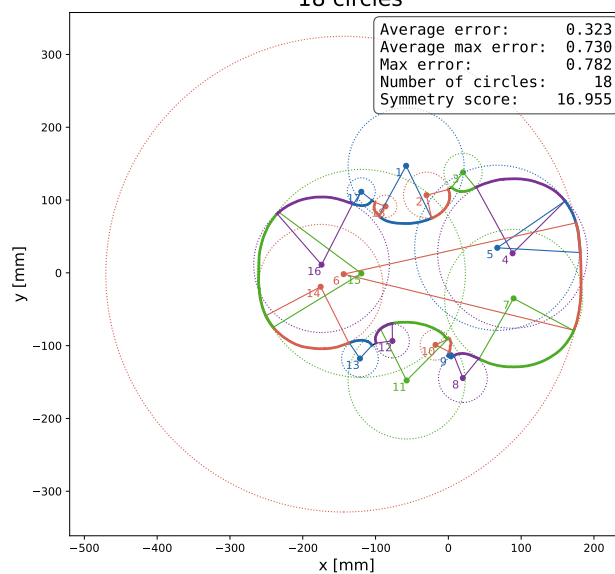
SetCover-KM - Instrument 2794
18 circles



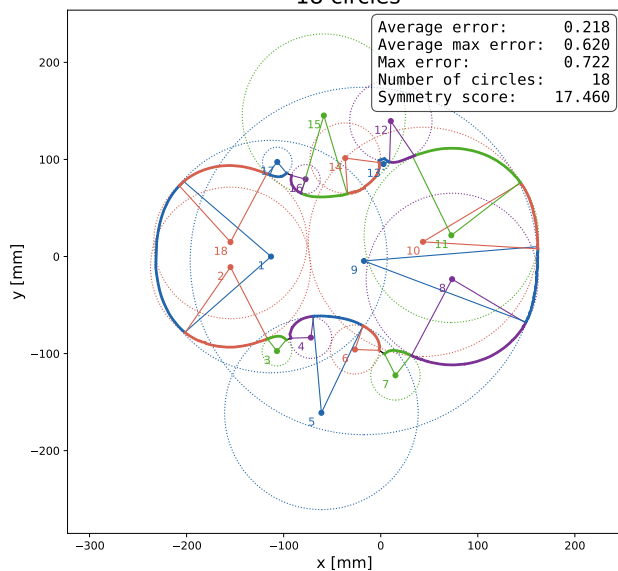
SetCover-KM - Instrument 2795
18 circles



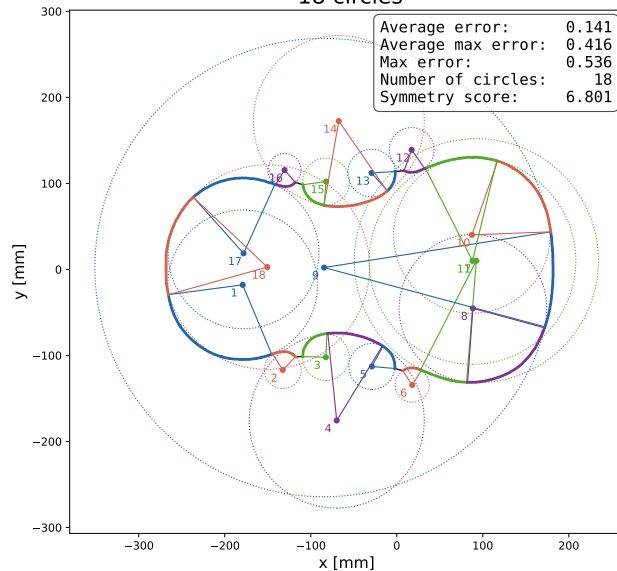
SetCover-KM - Instrument 2829
18 circles



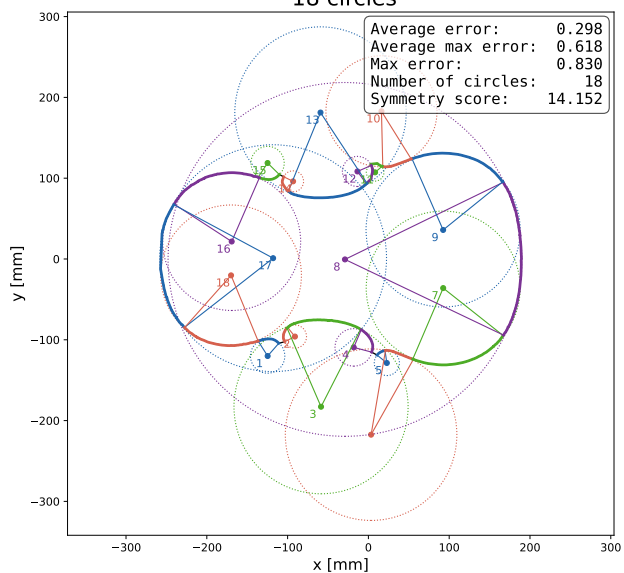
SetCover-KM - Instrument 2833
18 circles



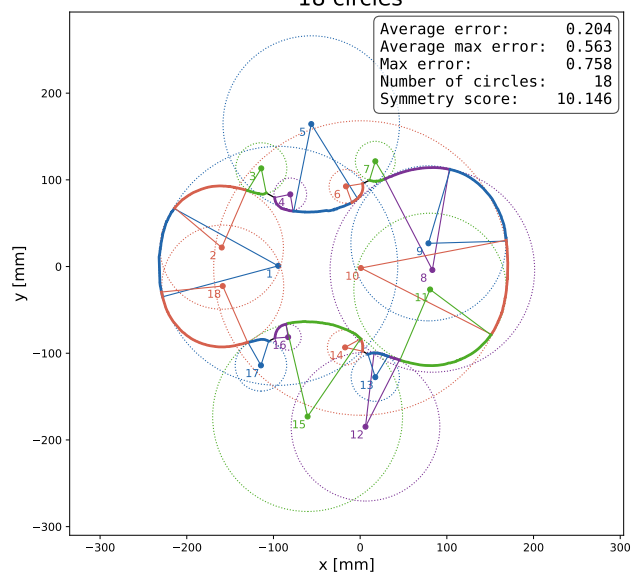
SetCover-KM - Instrument 2835
18 circles



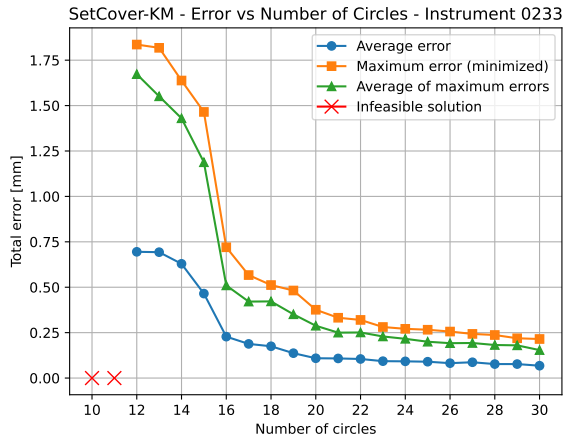
SetCover-KM - Instrument 2836
18 circles



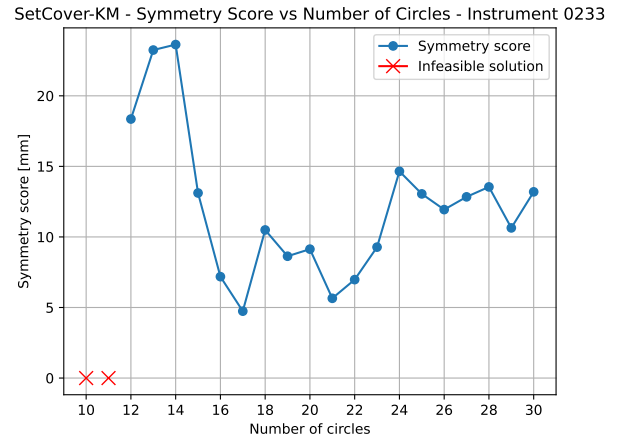
SetCover-KM - Instrument 2847
18 circles



E.3 Number of circles study

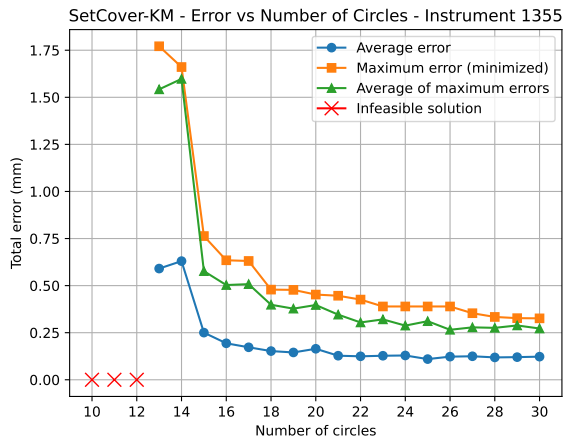


(a) Number of circles vs Error Metrics

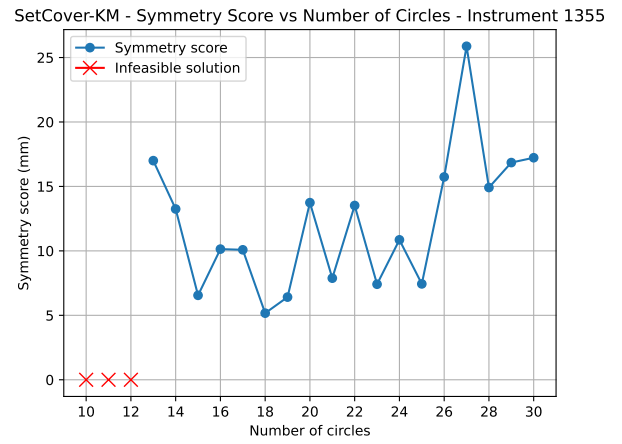


(b) Number of circles vs Symmetry score

Figure E.1: Number of circles study – SetCover-KM – Instrument 0233

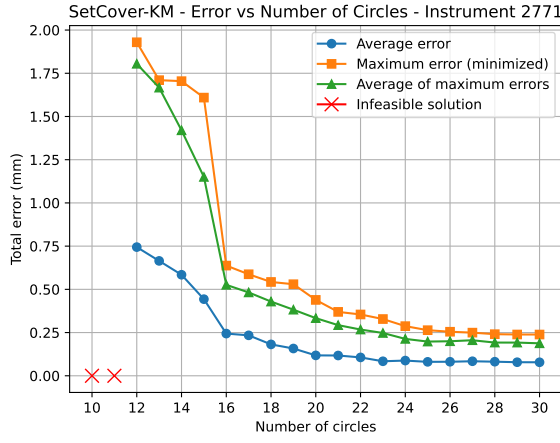


(a) Number of circles vs Error Metrics

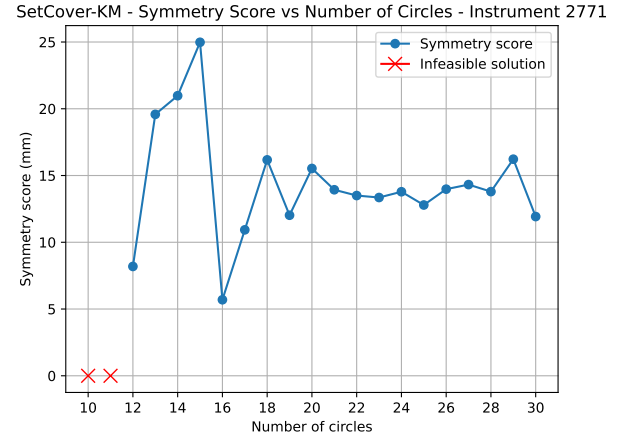


(b) Number of circles vs Symmetry score

Figure E.2: Number of circles study – SetCover-KM – Instrument 1355

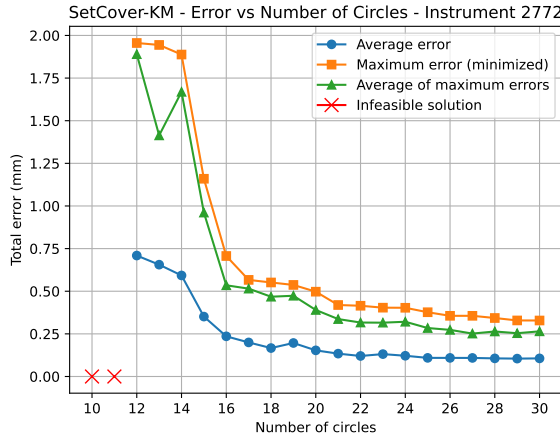


(a) Number of circles vs Error Metrics

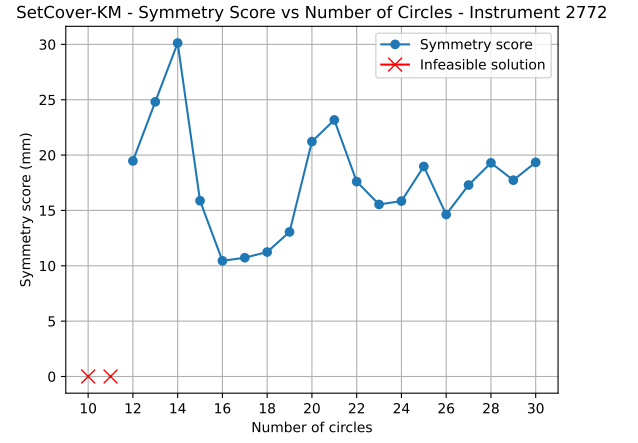


(b) Number of circles vs Symmetry score

Figure E.3: Number of circles study – SetCover-KM – Instrument 2771

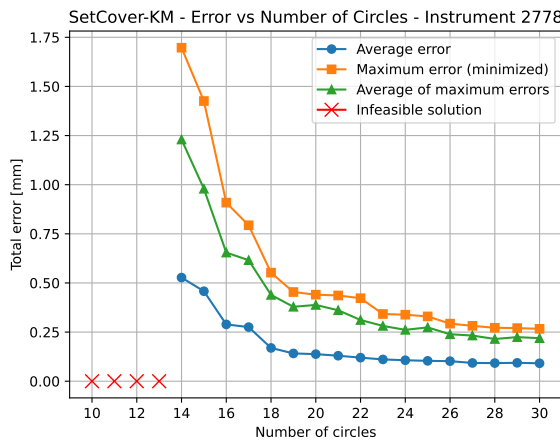


(a) Number of circles vs Error Metrics

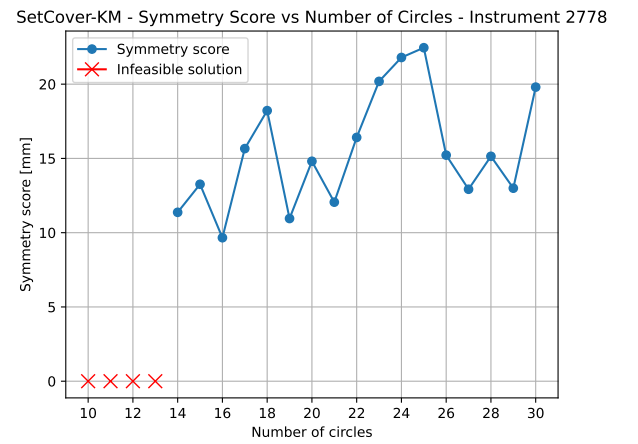


(b) Number of circles vs Symmetry score

Figure E.4: Number of circles study – SetCover-KM – Instrument 2772

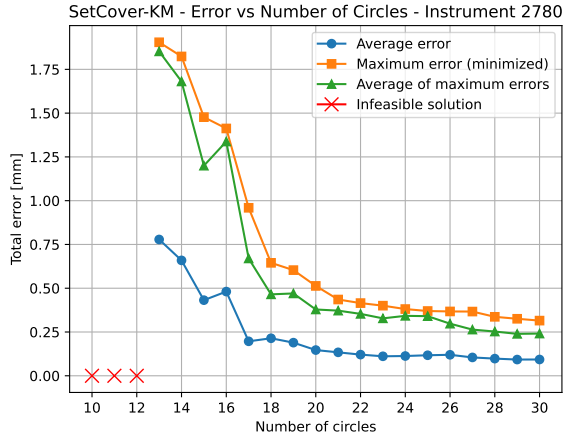


(a) Number of circles vs Error Metrics

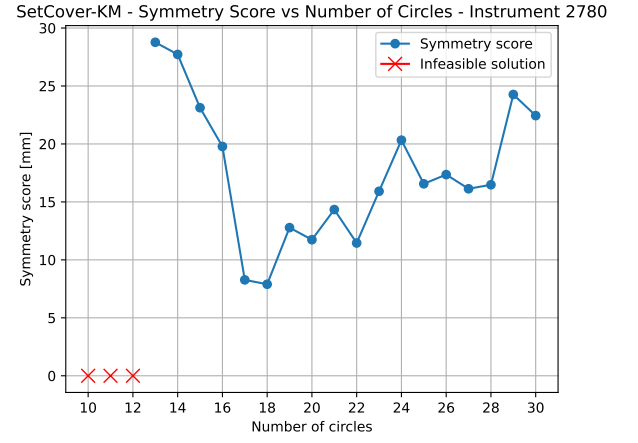


(b) Number of circles vs Symmetry score

Figure E.5: Number of circles study – SetCover-KM – Instrument 2778

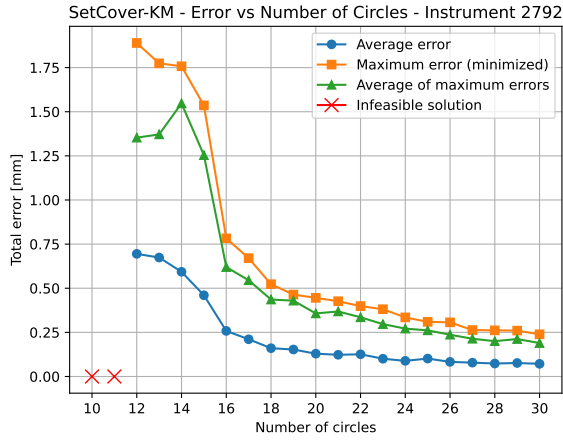


(a) Number of circles vs Error Metrics

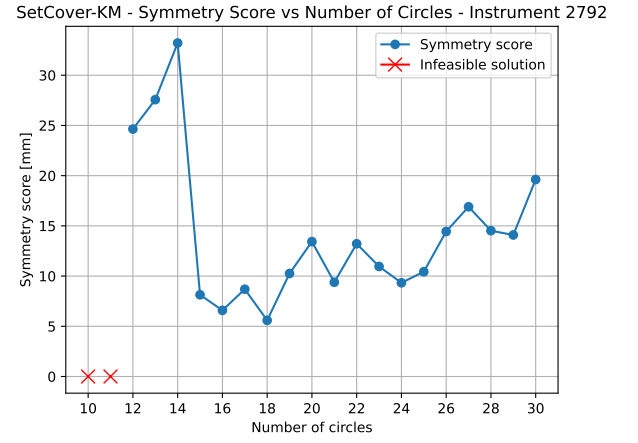


(b) Number of circles vs Symmetry score

Figure E.6: Number of circles study – SetCover-KM – Instrument 2780

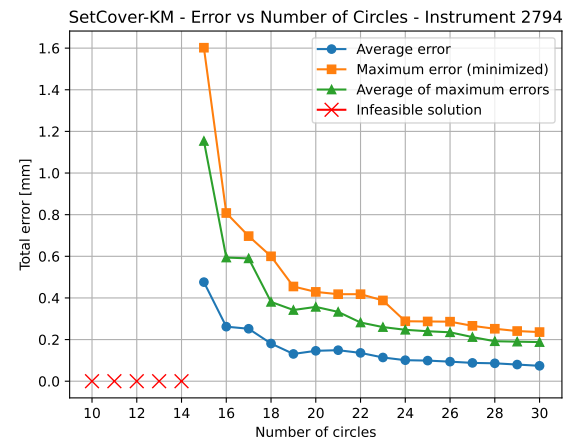


(a) Number of circles vs Error Metrics

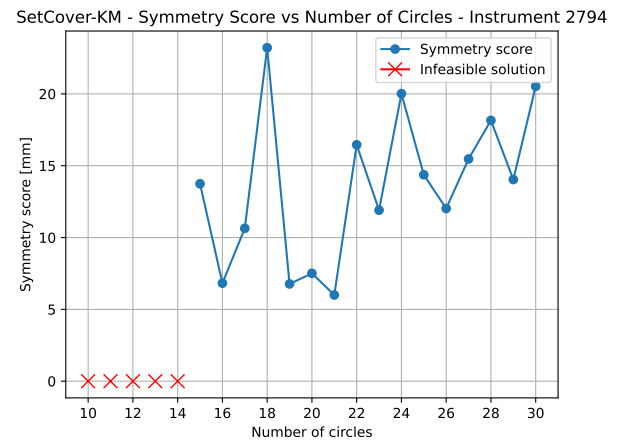


(b) Number of circles vs Symmetry score

Figure E.7: Number of circles study – SetCover-KM – Instrument 2792

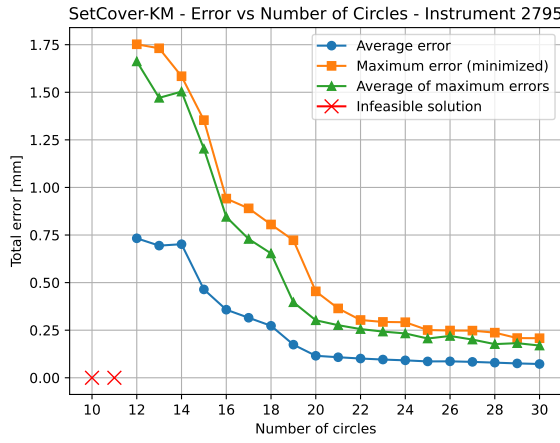


(a) Number of circles vs Error Metrics

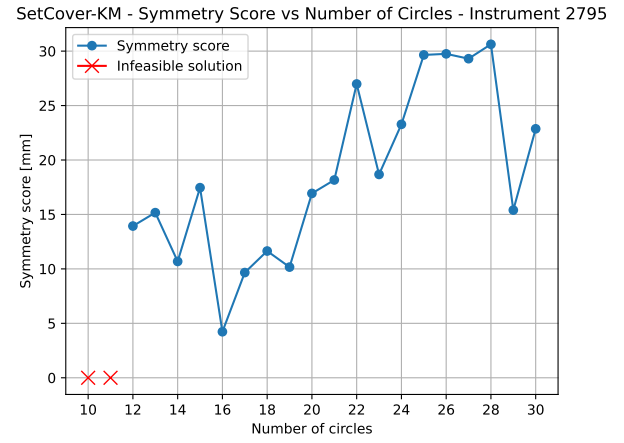


(b) Number of circles vs Symmetry score

Figure E.8: Number of circles study – SetCover-KM – Instrument 2794

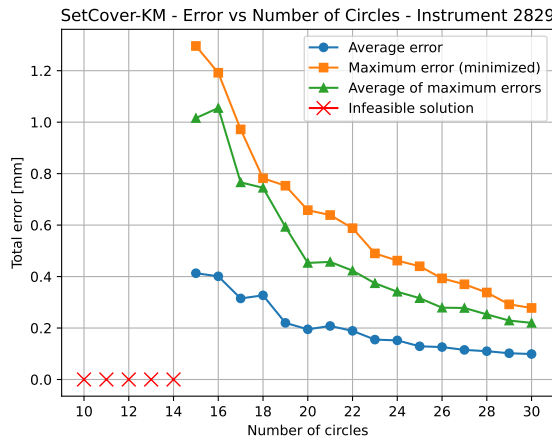


(a) Number of circles vs Error Metrics

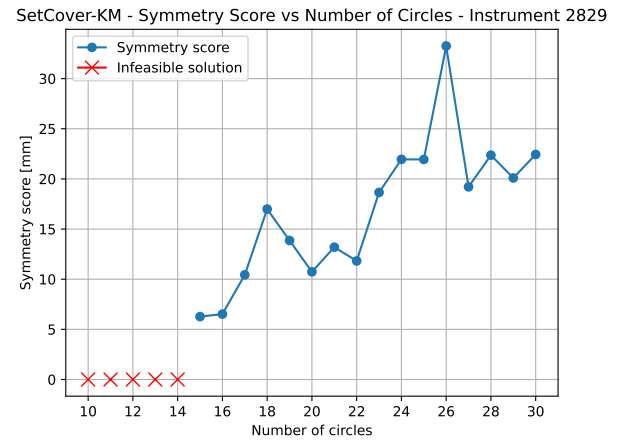


(b) Number of circles vs Symmetry score

Figure E.9: Number of circles study – SetCover-KM – Instrument 2795

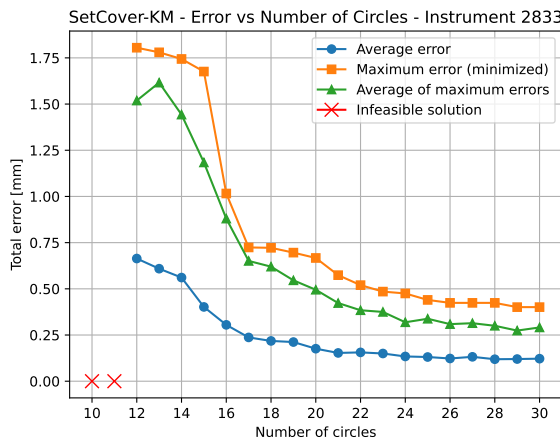


(a) Number of circles vs Error Metrics

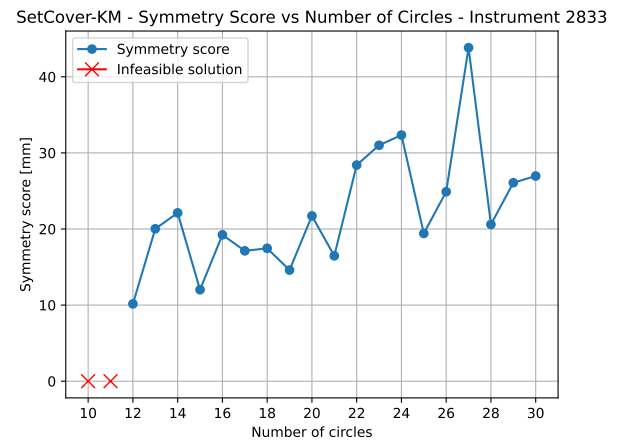


(b) Number of circles vs Symmetry score

Figure E.10: Number of circles study – SetCover-KM – Instrument 2829

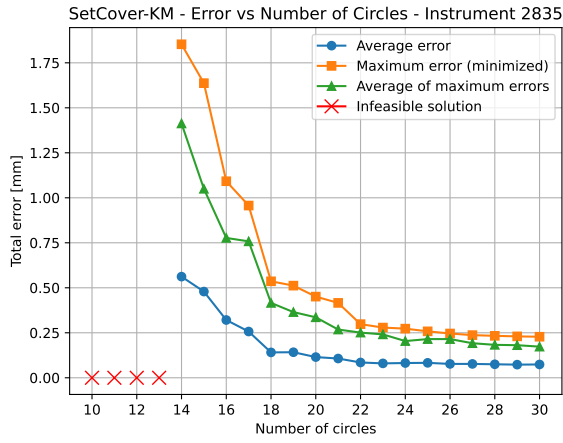


(a) Number of circles vs Error Metrics

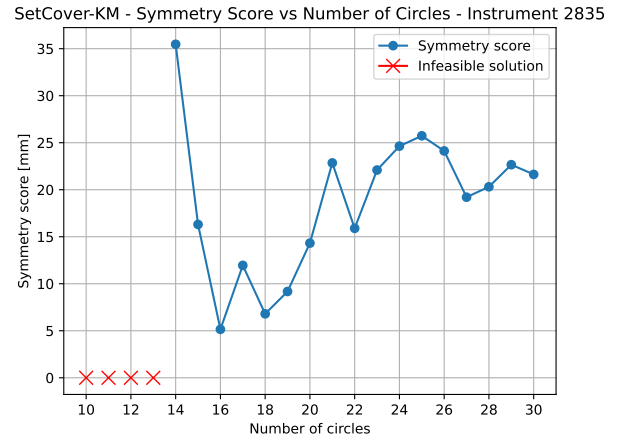


(b) Number of circles vs Symmetry score

Figure E.11: Number of circles study – SetCover-KM – Instrument 2833

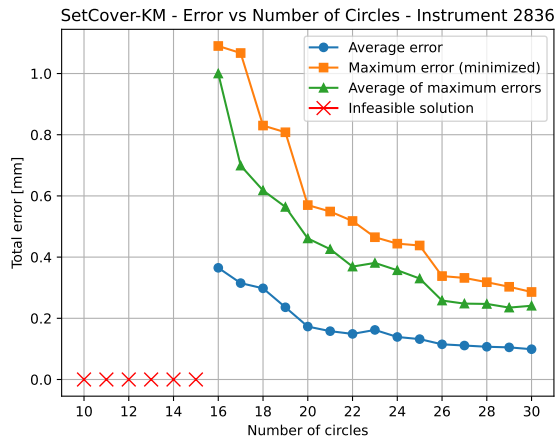


(a) Number of circles vs Error Metrics

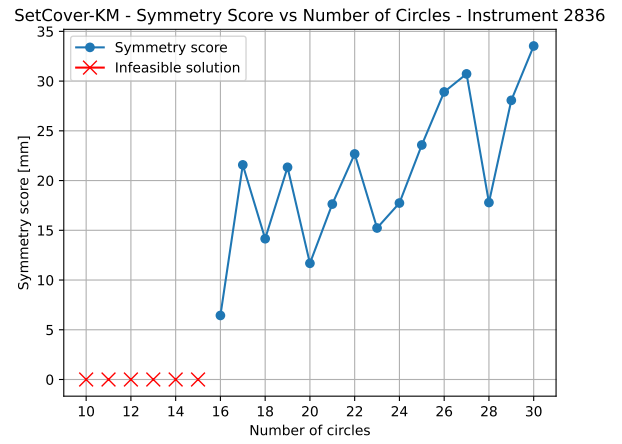


(b) Number of circles vs Symmetry score

Figure E.12: Number of circles study – SetCover-KM – Instrument 2835

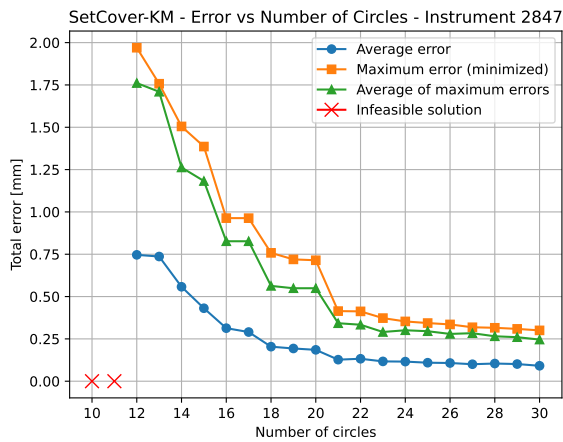


(a) Number of circles vs Error Metrics

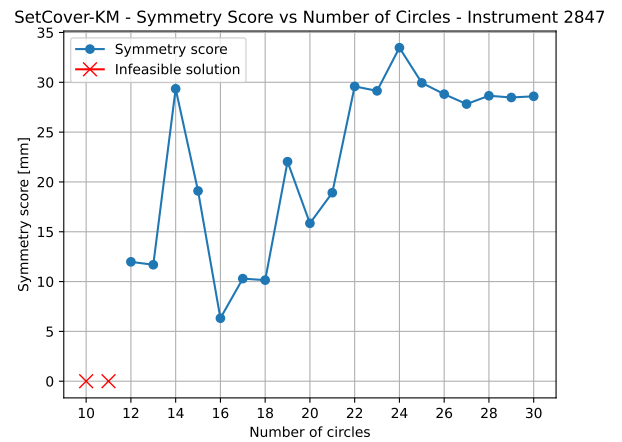


(b) Number of circles vs Symmetry score

Figure E.13: Number of circles study – SetCover-KM – Instrument 2836



(a) Number of circles vs Error Metrics



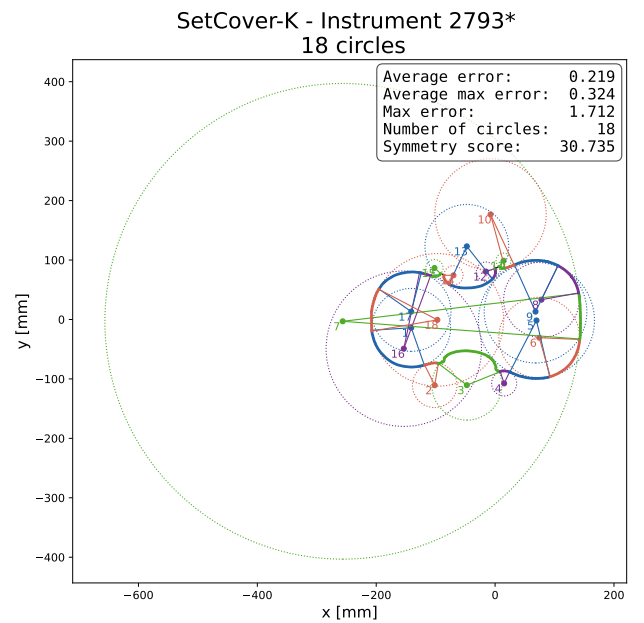
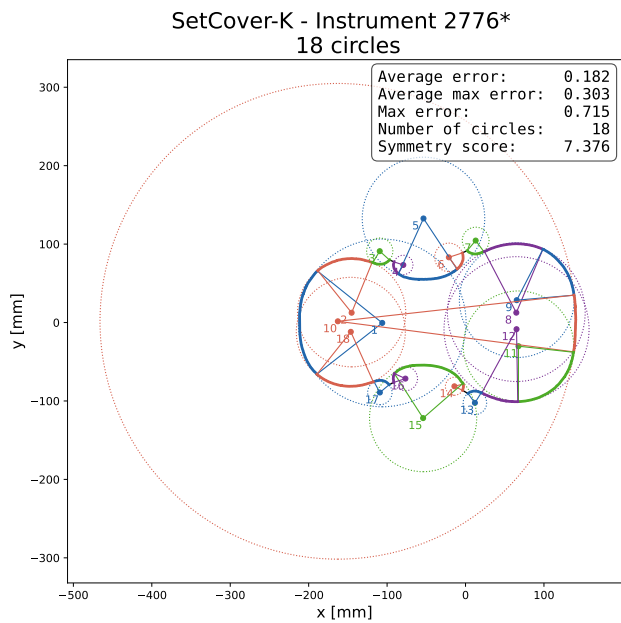
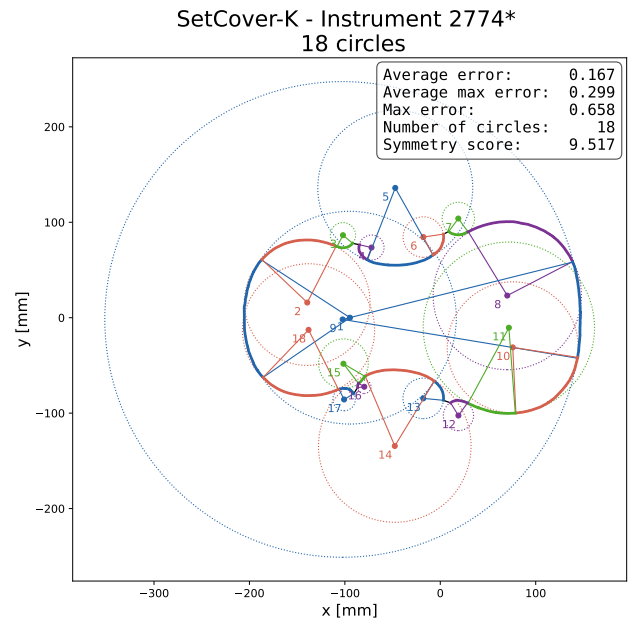
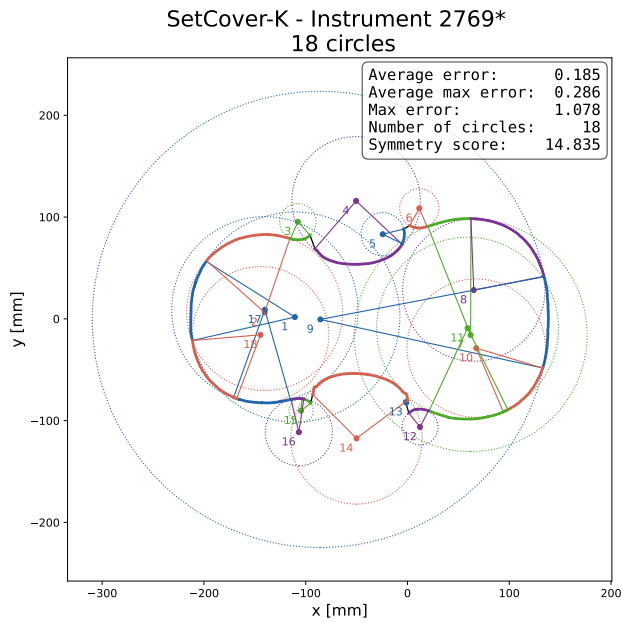
(b) Number of circles vs Symmetry score

Figure E.14: Number of circles study – SetCover-KM – Instrument 2847

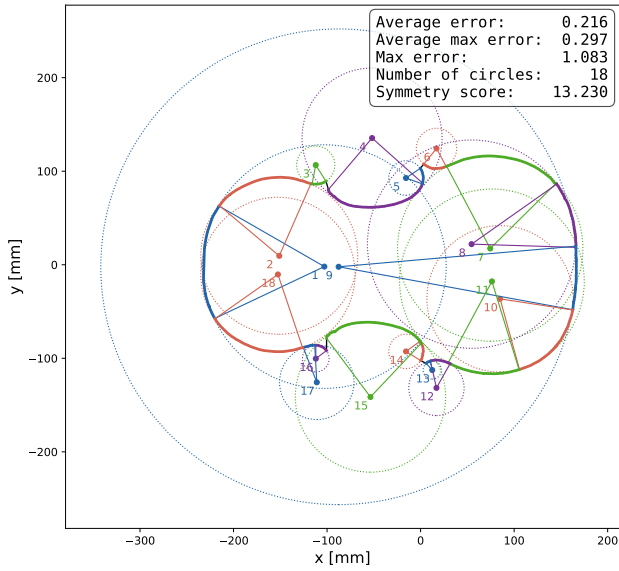
Appendix F

Reduced Instruments – All Results

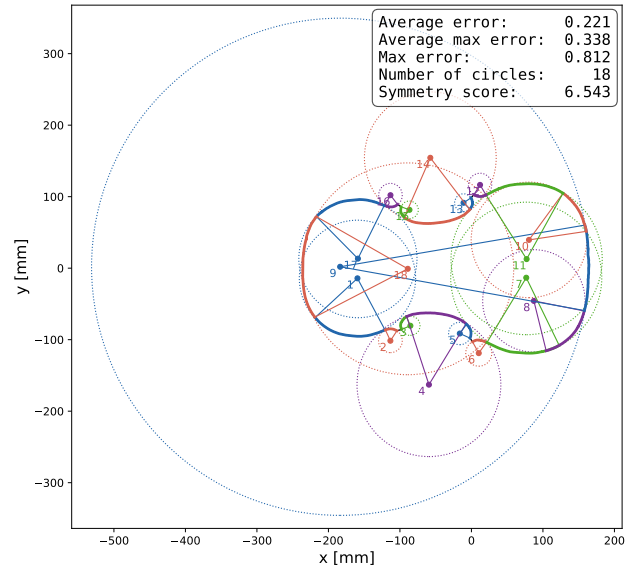
F.1 SetCover-K – $K = 18$



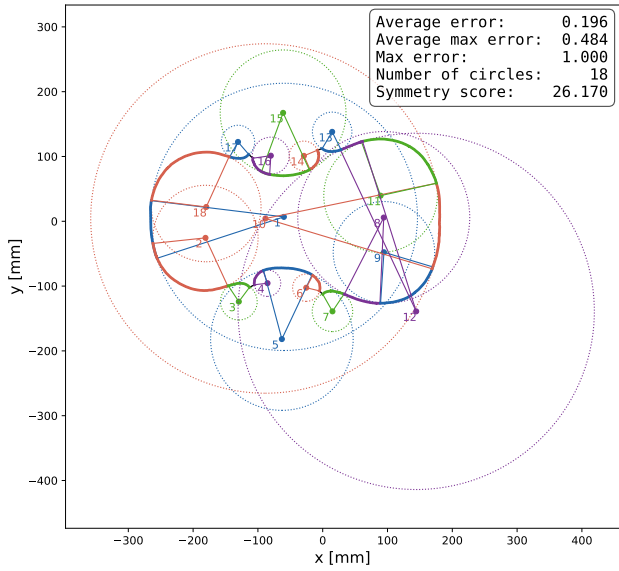
SetCover-K - Instrument 2828*
18 circles



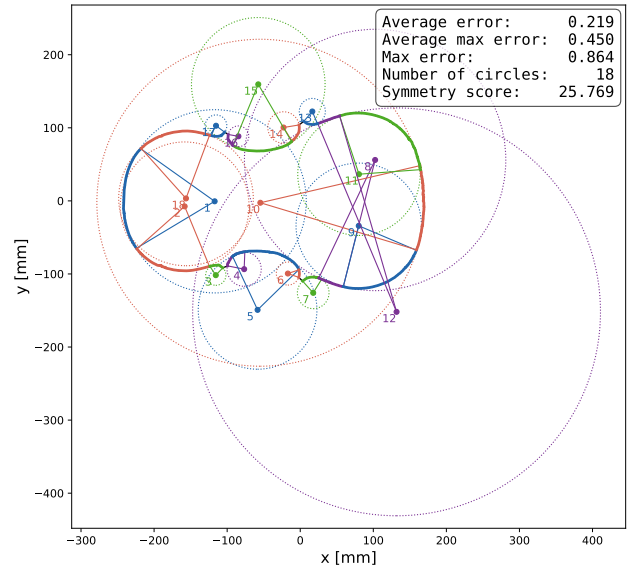
SetCover-K - Instrument 2831*
18 circles



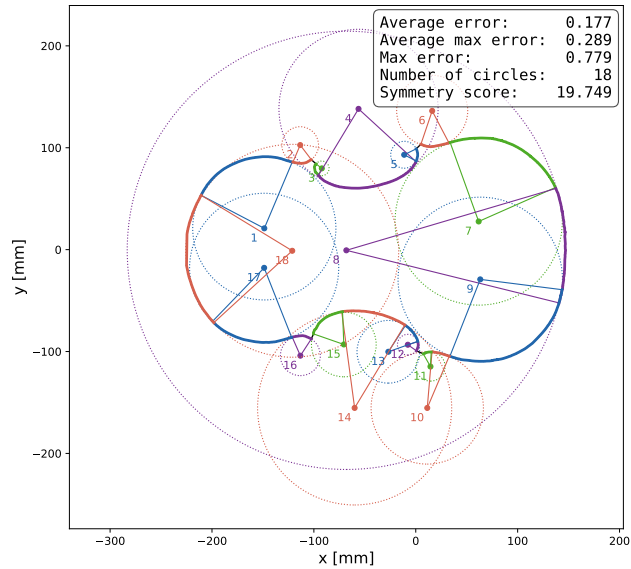
SetCover-K - Instrument 2840*
18 circles



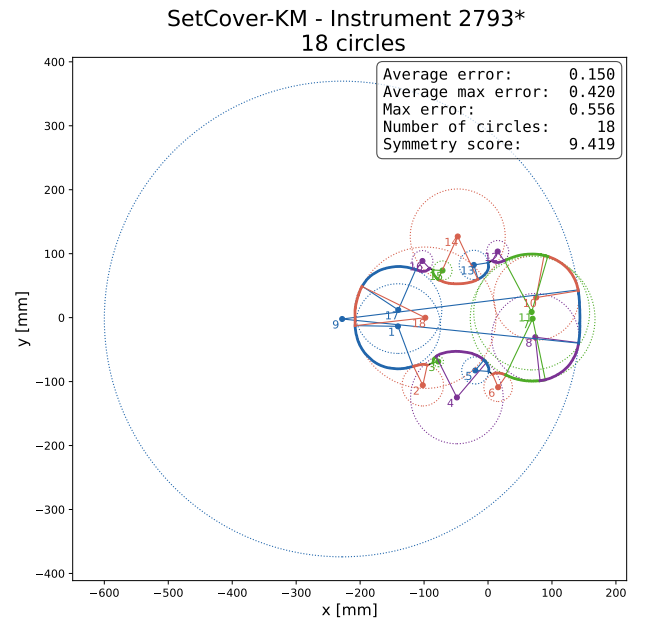
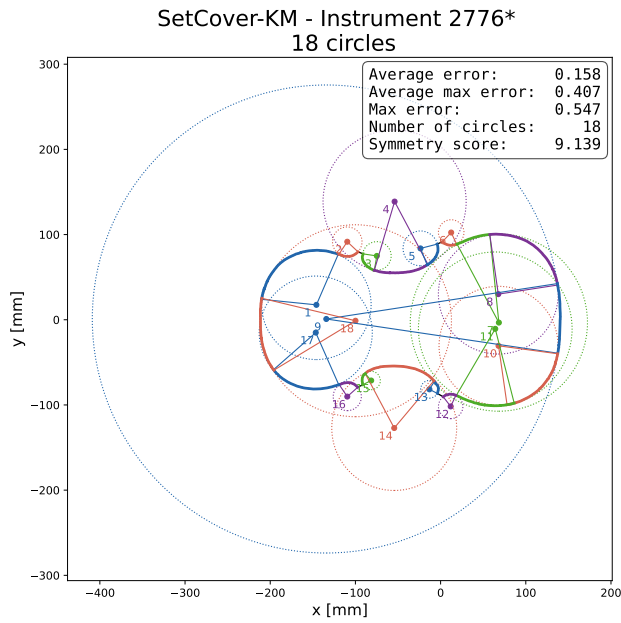
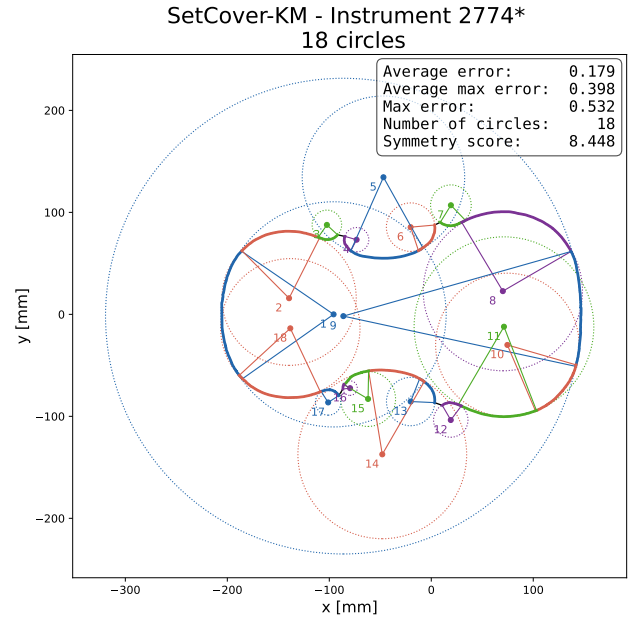
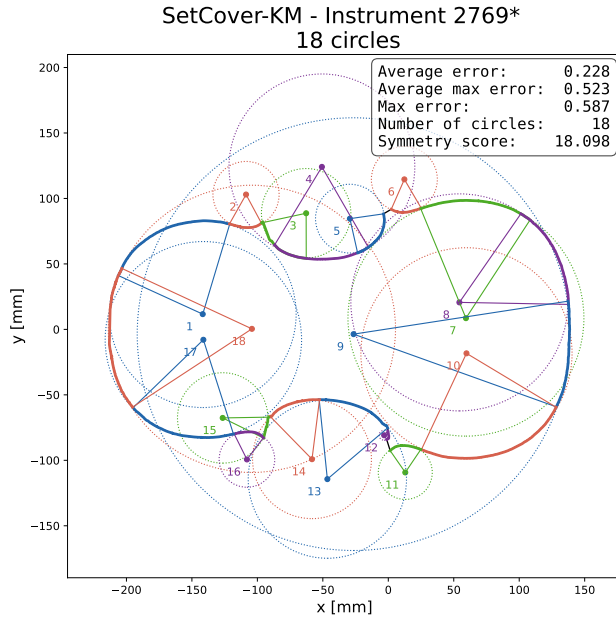
SetCover-K - Instrument 2846*
18 circles



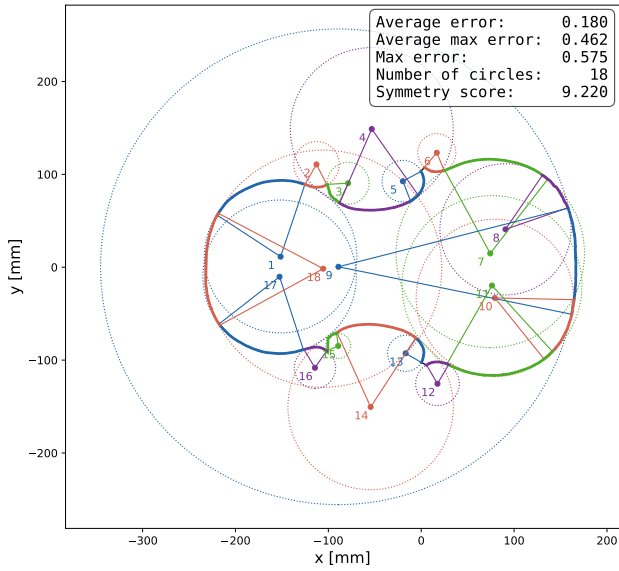
SetCover-K - Instrument 2848*
18 circles



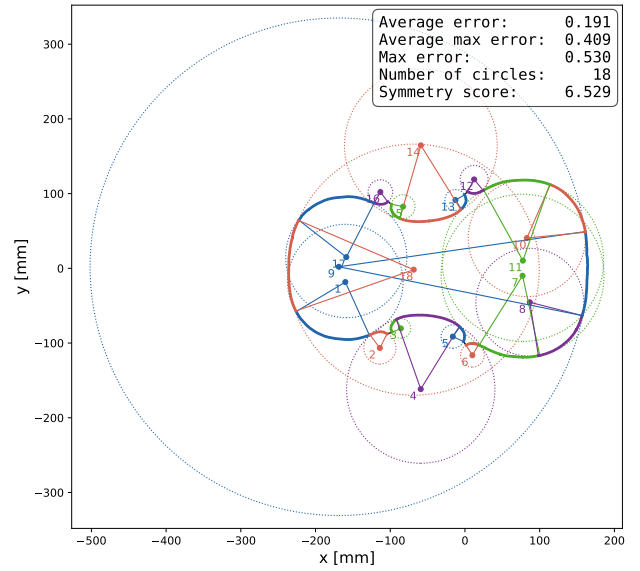
F.2 SetCover-KM – $K = 18$



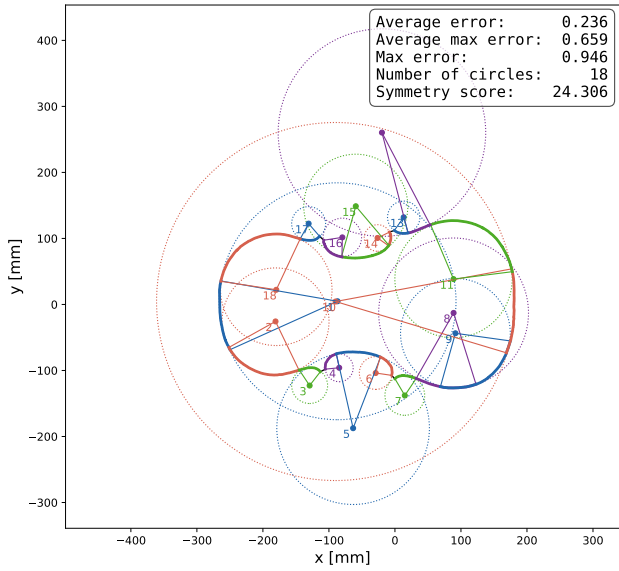
SetCover-KM - Instrument 2828*
18 circles



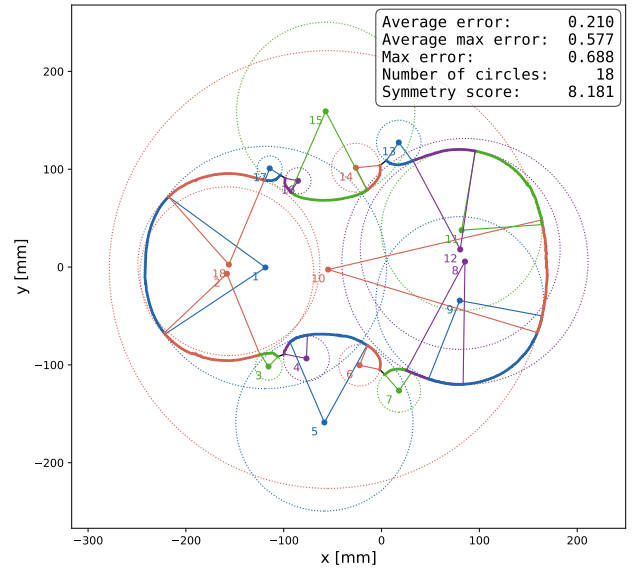
SetCover-KM - Instrument 2831*
18 circles



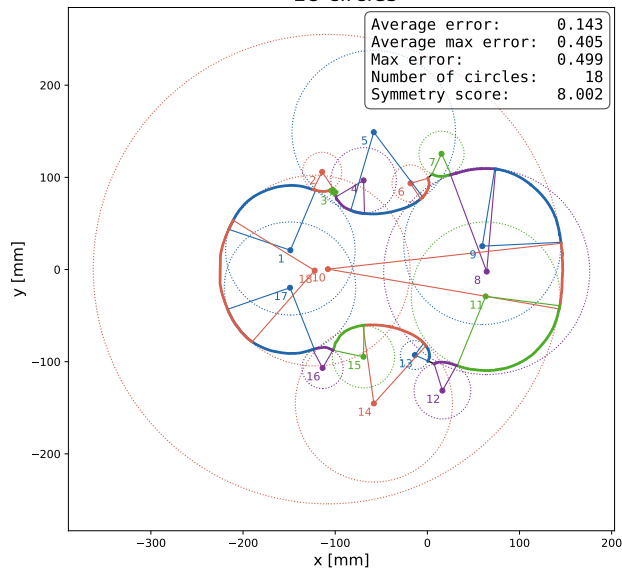
SetCover-KM - Instrument 2840*
18 circles



SetCover-KM - Instrument 2846*
18 circles



SetCover-KM - Instrument 2848*
18 circles



UNIVERSITÉ CATHOLIQUE DE LOUVAIN

École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl