Musical scale rationalization – a graph-theoretic approach

Albert Gräf
Institute of Musicology, Dept. of Music-Informatics
Johannes Gutenberg-University Mainz, Germany

December 2002

Abstract

While most Western music today uses the well-established equal-tempered 12 tone scale, there are many reasons why composers and music theorists are still interested in “just” scales in which the scale tones are simple integer ratios. Therefore the rationalization of musical scales is an important problem in music theory. The first general scale rationalization algorithm based on a complete enumeration approach was proposed by the influential contemporary composer Clarence Barlow. This paper gives a general formulation of the problem and proves that the problem is NP-complete. The paper also describes new algorithms for scale rationalization based on clique search in graphs, and a method to draw rational scales using multidimensional scaling. A scale rationalization and visualization software based on these algorithms is discussed as well.

1 Introduction

It is a well-known fact that pure musical intervals can be described in terms of simple frequency ratios, such as 1/1 (the *unison* or *prime*), 2/1 (the *octave*), 3/2 (the *perfect fifth*) or 5/4 (the *major third*). It has also been observed that the "pleasantness" or "consonance" of an interval is somehow correlated with the complexity of its frequency ratio. Thus, e.g., in most contexts the perfect fifth is recognized as the most consonant interval besides the prime and the octave. An abundance of different consonance measures have been proposed, such as Euler's *gradus suavitatis* [7], Helmholtz's harmoniousness measure [9], James Tenney's harmonic distance [4], Paul Erlich's harmonic entropy [6] and, last not least, Clarence Barlow's harmonicity function [1]. An interesting point about Barlow's harmonicity function is that it can be employed to *rationalize* a musical scale. That is, given an arbitrary (not necessarily rational) scale, it is possible to transform the scale into a rational form in which each scale tone is close to the corresponding tone in the original scale while keeping to simple integer ratios. For instance, if the input is the usual equal-tempered 12 tone scale then the result will be a just 12 tone scale. The rationalized scale can then be used, e.g., for compositional purposes or to study harmonic relationships in the original scale. So this method is valuable for music theorists and composers interested in just intonation.

Unfortunately, however, scale rationalization is computationally very intensive; as we prove in this paper, the problem is in fact NP-complete. We treat the general case of "subadditive" consonance measures which yield mathematical metrics on scales. This allows us to both visualize rational scales using multidimensional scaling (another idea proposed by Clarence Barlow), and to formulate the scale rationalization problem as a clique problem on an underlying "harmonicity graph". The latter approach leads to a variety of improved rationalization algorithms employing variations of the standard clique search algorithm by Carraghan and Pardalos [3], which often help to solve the problem much more efficiently than with the "brute force" method which requires a complete enumeration of all potential solutions. These methods are applicable to a wide range of possible consonance measures, including the ones proposed by Barlow, Euler and Tenney. The algorithms have actually been implemented in a graphical program suitable for experimenting with just and microtonal tunings.

The paper is organized as follows. We first give a general definition of subadditive consonance measures and the corresponding metrics. We then show how to employ these functions to formulate the scale rationalization problem as a clique problem, prove that the problem is NP-complete, and propose some heuristic branch and bound algorithms for solving the problem. We also discuss how to visualize the harmonicity graph of a scale using multidimensional scaling. Finally, we briefly describe a scale rationalization and visualization software based on our methods, and take a look at some simple examples. In the conclusion we summarize our results and point out some directions for further research.

2 Harmonicity

The term "consonance" is somewhat troublesome, since it is used to denote several different phenomena in music theory and psychoacoustics [11]. Hence we adopt Barlow's terminology and use the artificial terms *harmonicity* and *disharmonicity* throughout this paper.

Barlow's harmonicities, as well as Euler's gradus suavitatis and several other consonance measures are defined in terms of sums of elementary disharmonicities (called "indigestibilities" by Barlow) of the prime factors of a rational interval. In the following we give a generic definition which covers all these approaches. So let the *prime disharmonicities* $g(p) > 0$ for all prime numbers p be given. (We assume throughout that $g(p)$ can be computed in polynomial time with respect to $\log p$.) We extend g to all positive rational numbers x as follows. If $x = p_1^{a_1} p_2^{a_2} \cdots p_n^{a_n} > 0$ is a rational number, where $p_1 < p_2 < \cdots < p_n$ are the prime factors of x and $a_1, a_2, \ldots, a_n$ their (positive or negative) multiplicities, then the *disharmonicity* $g(x)$ of x (with respect to the given prime disharmonicities) is defined as:

$$g(x) = |a_1|g(p_1) + |a_2|g(p_2) + \cdots + |a_n|g(p_n).$$

The *harmonicity* $h(x)$ of x is then defined as the inverse of its disharmonicity, that is,

$$h(x) = 1/g(x).$$

Note that, in particular, $g(1) = g(1/1) = 0$ and hence the unison has infinite harmonicity for any given prime disharmonicities. We also remark that the disharmonicity function is *additive* for integer arguments, i.e., $g(xy) = g(x) + g(y)$ if x and y are positive integers. Moreover, the disharmonicity of an interval x/y given in its cancelled-down form (i.e., x and y integer, $\gcd(x, y) = 1$), is always the sum of the disharmonicities of the numerator and the denominator:

$$g(x/y) = g(x) + g(y) = g(xy).$$

It is also easy to see that in the general case, for arbitrary *rational* values $x, y > 0$, the disharmonicity function is *subadditive*:

$$g(xy) \leq g(x) + g(y).$$

Using the generic definition above, we can obtain various different harmonicity measures such as Barlow's disharmonicities and Euler's gradus function by just plugging in suitable prime disharmonicities. For instance:

- Barlow disharmonicities: $g_{\mathrm{B}}(p) = 2(p-1)^2/p$
- Euler disharmonicities: $g_{\mathrm{E}}(p) = p - 1$

The Barlow and Euler disharmonicities for some common intervals are shown in Fig. 1. (The meaning of the "cent" values is explained below.) Note that the factor 2 in the definition of the Barlow disharmonicities is just a normalization factor which makes the octave 2/1 have a value of 1. We also remark that Euler's original definition of the *gradus*

Interval	Ratio	Cents	g_{B}	g_{E}
unison	1/1	0.00	0.00	0
minor semitone	16/15	111.73	13.07	10
minor whole tone	10/9	182.40	12.73	9
major whole tone	9/8	203.91	8.33	7
minor third	6/5	315.64	10.07	7
major third	5/4	386.31	8.40	6
perfect fourth	4/3	498.04	4.67	4
tritone	45/32	590.22	16.73	13
perfect fifth	3/2	701.96	3.67	3
minor sixth	8/5	813.69	9.40	7
major sixth	5/3	884.36	9.07	6
minor seventh	16/9	996.09	9.33	8
major seventh	15/8	1088.27	12.07	9
octave	2/1	1200.00	1.00	1

Figure 1: Barlow and Euler disharmonicities for some common intervals.

suavitatis actually adds an extra 1 term after summing up the prime disharmonicities, i.e., if we denote Euler's gradus function as Γ then $\Gamma(x) = g_{\mathrm{E}}(x) + 1$. But in the following we stick to our definition because it makes the measure subadditive which is not the case for the original gradus function.

3 Harmonic distance

Next we show how to derive a scale metric from a disharmonicity function. For our purposes a *scale* is simply a finite set S of positive real numbers. The members of S are called *tones* or *pitches*. We also allow pitches to be specified in *cents*, a logarithmic measure which divides the octave into 1200 equidistant logarithmic steps, 100 cents for each equal-tempered semitone. That is, given a frequency ratio x, the corresponding cent value is $1200 \log_2 x$ where $\log_2$ denotes the base 2 logarithm. Thus, e.g., the unison equals 0 cents, the octave is 1200 cents and the equal-tempered fifth (which corresponds to the frequency ratio $2^{7/12}$) is exactly 700 cents, whereas the perfect fifth 3/2 is about 701.96 cents.

In this section we only consider *rational* scales, i.e., scales whose pitches are all rational. Given two scale tones $x, y \in S$, we define the *harmonic distance* $d(x, y)$ as the disharmonicity of the interval between the pitches, i.e.,

$$d(x, y) = g(x/y).$$

Note that to calculate this value we have to cancel common factors in x and y. For instance, if $x = 3/2$ (the fifth over the base tone) and $y = 5/4$ (the major third) then

$$d(x, y) = g(3/2 \times 4/5) = g(6/5),$$

which, not very surprisingly, is the disharmonicity of the minor third. In this manner we can calculate harmonic distances between all pairs of scale members.

Now it is important to note that, for *any* choice of positive prime disharmonicities, the harmonic distance function d thus defined is indeed a *metric* in the mathematical sense. That is, it obeys the following rules:

- $d(x, y) = 0$ if and only if $x = y$, for all $x, y \in S$
- $d(x, y) = d(y, x)$, for all $x, y \in S$
- $d(x, z) \leq d(x, y) + d(y, z)$, for all $x, y, z \in S$

The first two conditions are easily verified. The third condition, which is also known as the *triangle inequality*, follows from the subadditivity of the disharmonicity function.

4 The scale rationalization problem

In the following we generally assume a disharmonicity function g and the corresponding harmonic distance metric d as defined in the preceding section. We now take a closer look at the scale rationalization problem. Generally speaking, we are given an arbitrary (not necessarily rational) scale S and ask how we can transform S into a "similar" rational scale S' in which the intervals between scale pitches are as "simple" as possible. Of course, the devil lies in the details, and so we now have to specify what exactly we mean by "similar" and "simple".

Obviously, in order to get a scale S' which sounds similar to the orginal scale S, we have to match the pitches of S as closely as possible. But since there are infinitely many rational numbers in even the smallest positive range around a real value we need a secondary criterion which enables us to choose the "best" among those. This can be accomplished with Barlow's method [1] which always selects a given number of alternatives x' for a given scale tone $x \in S$ with a given minimum harmonicity, $h(x') \geq h_{\min}$, and the highest *weighted* harmonicity

$$h_w(x') = w(x')h(x'),$$

where w is a weight function which weights pitches according to their offsets from the original scale tone. Barlow defines the weights as

$$w(x') = \exp(-\Delta(x')^2 \ln(1/a)/t^2),$$

where $\Delta(x') = 1200|\log_2(x'/x)|$ is the absolute offset of x' from x in cents, t is the tolerance in cents and a is the desired *attenuation* factor which determines the weight of the harmonicity values at the edges of the tolerance range. Thus the harmonicities are modified by superimposing the usual kind of bell-shaped Gaussian curve, which has the effect that those pitches will be preferred which either have a high harmonicity or are close to the original scale tone. We refer the reader to the cited reference [1] for an example.

So in the following we may assume that a (nonempty, finite) set of alternative rational "candidate" pitches $C(x)$ is given for each $x \in S$. Next we have to decide which combination of candidate pitches is considered to form a "simple" scale. For this purpose we employ the harmonic distance metric. We will be interested in solutions in which the harmonic distances for all pairs of rational pitches are as small as possible. To accomplish this, we could start out with a global optimization method, as proposed by Barlow, which attempts to minimize the total (or average) harmonic distance of all intervals in the resulting scale. However, this approach effectively lumps all intervals together, which is often a bad idea because some intervals will be inherently more disharmonious than others. For instance, an interval of about 600 cent (a tritone) will almost surely have a higher disharmonicity than an interval of about 700 cent (a fifth), in *any* reasonable rationalization of the scale. Therefore we adopt an approach which enables us to specify different disharmonicity bounds for the individual intervals in the scale. We thus formulate the scale rationalization problem as follows:

Problem 1 (Scale rationalization) INSTANCE: A collection of pitch candidate sets C_i, $i = 1, \ldots, n$, and disharmonicity bounds b_{ij} for $i, j = 1, \ldots, n$, $i \neq j$.
QUESTION: Are there pitches $x_i \in C_i$ for $i = 1, \ldots, n$ such that $d(x_i, x_j) \leq b_{ij}$ for $i, j = 1, \ldots, n$, $i \neq j$?

(In the following we generally assume that the disharmonicity bounds are *symmetric*, i.e., $b_{ij} = b_{ji}$ for all $i \neq j$. We can do so without loss of generality since d is a metric and hence $d(x_i, x_j) = d(x_j, x_i)$ for all i, j.)

Note that we have stated the problem in the form of a decision problem, namely the problem to decide whether *any* rationalization exists which satisfies the given disharmonicity bounds. Of course, if the answer is affirmative then we will still be interested in obtaining the "best" such solution according to certain optimization criteria; we return to this question in the following section.

Let us now see how the scale rationalization problem relates to the clique problem on graphs. First we recall some terminology: A *graph* is a pair $G = (V, E)$ where V is a finite set, the set of *nodes* or *vertices* of the graph, and E is a set of unordered pairs of nodes vw, the *edges* of G. Two nodes v and w connected by an edge vw are called *adjacent*. In this paper, all graphs are *simple* (they do not contain multiple edges between the same pair of nodes), and *loopless* (they do not contain edges connecting a node to itself). A *clique* of a graph is a subset of nodes $U \subseteq V$ such that every two distinct nodes $u, v \in U$ are connected by an edge uv. The *clique problem* can be stated as follows:

Problem 2 (Clique) INSTANCE: Graph $G = (V, E)$, $k \leq |V|$.
QUESTION: Does G contain a clique of size k?

The clique problem is "difficult" in a precise mathematical sense: the problem is *NP-complete*. The class of NP-complete problems comprises many important practical problems which do not appear to have an "efficient", i.e., polynomial-time solution. For the definition and the many ramifications of this concept we refer the reader to Garey and

Johnson's classic book on the subject [8]. To see how scale rationalization can be formulated as a clique problem we need the following definition.

Definition 1 (Harmonicity graph) *Let pitch candidate sets C_i, $i = 1, \dots, n$ and disharmonicity bounds b_{ij} for $i, j = 1, \dots, n$, $i \neq j$ be given. Then the* harmonicity graph $G(C, b)$ *has $|C_1| + \cdots + |C_n|$ nodes v_i^x, $i = 1, \dots, n$, $x \in C_i$, and the edges $v_i^x v_j^y$ for which $d(x, y) \leq b_{ij}$, $i, j \in \{1, \dots, n\}$, $i \neq j$, $x \in C_i$, $y \in C_j$.*

We have defined the harmonicity graph in such a manner that $\{x_1, \dots, x_n\}$ is a solution to the scale rationalization problem if and only if $\{v_1^{x_1}, \dots, v_n^{x_n}\}$ is a clique of the harmonicity graph. Hence:

Theorem 1 *An instance of the scale rationalization problem is solvable if and only if the corresponding harmonicity graph has a clique of size n.*

We will explore this relationship to develop different variations of a scale rationalization algorithm in the following section. We conclude this section with a proof that the scale rationalization problem, like the clique problem, is NP-complete. For this purpose we employ a reduction from the following *3-satisfiability* problem ("3SAT") of propositional logic; see [8, p. 46]. 3SAT is yet another example of an NP-complete problem.

Problem 3 (3SAT) INSTANCE: Set U of variables, and a set K of clauses where each clause consists of three literals of the form u or $\bar{u}$, $u \in U$.
QUESTION: Is there a truth assignment $T : U \mapsto \{\text{true}, \text{false}\}$ which satisfies all clauses?

(Note that in order for all clauses to be satisfied, each clause must contain a literal v such that $T(v) = \text{true}$, where $T(v) = T(u)$ if $v = u$ and $T(v) = \neg T(u)$ if $v = \bar{u}$.)

Theorem 2 *The scale rationalization problem is NP-complete, for any given harmonic distance metric d, even if the disharmonicity bounds are all the same and each candidate set contains at most three pitches.*

Proof. The problem clearly belongs to NP since we can verify in polynomial time whether a given selection $x_i \in C_i$, $i = 1, \dots, n$, satisfies the disharmonicity bounds. We reduce 3SAT to the scale rationalization problem. Let an instance $U = \{u_1, \dots, u_m\}$, $K = \{K_1, \dots, K_n\}$ of 3SAT be given. We show how to transform this instance into a corresponding instance of the scale rationalization problem which has a solution if and only if the original 3SAT instance has one. We assume without loss of generality that $m \geq 3$ (otherwise our instance of 3SAT can be solved in linear time, by simply enumerating all truth assignments).

We first choose mutually distinct prime numbers $p_1, p_2, \dots, p_m$. (This can be done in polynomial time with respect to m.) Now for $j = 1, \dots, m$ let

$$x_j = p_1 \cdots p_m / p_j^2.$$

Then for each $j, j_1, j_2 \in \{1, \ldots, m\}$, $j_1 \neq j_2$, we have:

$$g(x_j^2) = 2 \sum_{j=1}^{m} g(p_j) \qquad g(x_j/x_j) = 0$$

$$g(x_{j_1} x_{j_2}) = 2 \sum_{j \neq j_1, j_2} g(p_j) \qquad g(x_{j_1}/x_{j_2}) = 2g(p_{j_1}) + 2g(p_{j_2})$$

Thus, since $m \geq 3$, it is possible to choose a bound B such that $g(x_{j_1} x_{j_2}) > B$ if $j_1 = j_2$ and $g(x_{j_1} x_{j_2}), g(x_{j_1}/x_{j_2}) \leq B$ otherwise. We now construct an instance of the scale rationalization problem as follows. For each clause K_i let C_i be the set of pitches x_j for which $u_j \in K_i$ and pitches $1/x_j$ for which $\bar{u}_j \in K_i$. Furthermore, let $b_{i_1 i_2} = B$ for all $i_1, i_2 \in \{1, \ldots, n\}$, $i_1 \neq i_2$. Now it is easy to verify that $S = \{y_1, \ldots, y_n\}$, $y_i \in C_i$, is a solution for the scale rationalization instance if and only if T is a satisfying truth assignment for the 3SAT instance, where $T(u_j) =$ true iff $x_j \in S$, $j = 1, \ldots, m$. Conversely, if T is a satisfying truth assignment then we can pick a literal $v_{j_i} \in K_i \cap \{u_{j_i}, \bar{u}_{j_i}\}$ such that $T(v_{j_i}) =$ true for $i = 1, \ldots, n$. We then obtain a solution $S = \{y_1, \ldots, y_n\}$ for the scale rationalization instance, where $y_i = x_{j_i}$ if $v_{j_i} = u_{j_i}$ and $y_i = 1/x_{j_i}$ otherwise. q.e.d.

5 Scale rationalization algorithms

Using the relationship between scale rationalization and the clique problem established in the previous section, we now show how to solve the rationalization problem with Carraghan and Pardalos' branch and bound procedure for enumerating cliques in a graph [3]. The basic algorithm is shown in Fig. 2. The algorithm starts out with an initial solution C (the empty clique) and the set V of all nodes of G. It then adds nodes from V to the current clique C, one at a time, checks whether the new configuration can still lead to a clique of the desired size, and invokes itself recursively on the new partial solution. The invariant maintained during execution of the algorithm is the fact that C always is a clique of G and the current set of candidate nodes V consists of all nodes which are adjacent to all nodes in C. The algorithm terminates when all cliques have been enumerated. (In a concrete implementation we will of course stop the algorithm as soon as the desired number of cliques has been produced.)

If you compare the above algorithm to the original one in [3], you will notice some slight modifications. First, the node selection strategy (step 7 of the algorithm) is adaptable rather than using a fixed node order determined before execution; this allows us to apply different search strategies as described below. Second, the recursion is terminated as soon as a clique of the given size is found; this accounts for the fact that we know in advance how large our cliques must be.

To apply the algorithm to the scale rationalization problem, it is invoked on the harmonicity graph for the given problem instance (cf. Definition 1) and with the desired clique size $k = n$. If we just want to take a quick look at some (not necessarily optimal) solutions, the algorithm can be run until the desired number of alternative solutions is obtained. But

```
1  Inputs: Graph G = (V, E), desired clique size k.
2  Outputs: All cliques C of G with |C| = k.
3  Method: begin clique(∅, V) end
4  proc clique(C, V) ≡
5    if |C| ≥ k then output C
6              else while V ≠ ∅ do
7                        Choose a node v ∈ V;
8                        C' := C ∪ {v}; V := V \ {v};
9                        V' := {w ∈ V : uw ∈ E ∀ u ∈ C'};
10                       if |C'| + |V'| ≥ k then clique(C', V') fi
11                  od
12   fi
13 end
```

Figure 2: Carraghan/Pardalos clique algorithm.

we can also use it to enumerate *all* solutions and list the "best" among them. Following Barlow, a suitable measure for the quality of a solution $S = \{x_1, \ldots, x_n\}$ is the average harmonic distance $\bar{d}(S)$ between all selected pitches $x_i \in C_i$, which is to be minimized:

$$\bar{d}(S) = \frac{\sum_{i \neq j} d(x_i, x_j)}{n(n-1)}.$$

Equivalently, we can also maximize the inverse of $\bar{d}(S)$ which Barlow calls the *specific harmonicity*:

$$\bar{h}(S) = 1/\bar{d}(S).$$

The crucial step in the algorithm is step 7 in which the next candidate node is selected. We can use different node selection schemes in this step to implement alternative search heuristics. The chosen search heuristic determines which solutions will be enumerated first and how long the search takes. In our computational experiments we found that the search can be sped up considerably by selecting an appropriate search heuristic for the type of search (full or partial enumeration) and given problem parameters (e.g., depending on whether the harmonicity graph is "sparse" or "dense"). Here are some search heuristics we found to be useful:

- The *first-first* strategy. Here we simply select nodes in the "natural" order $v_1^{x_{11}}, v_1^{x_{12}}, \ldots, v_2^{x_{21}}, v_2^{x_{22}}, \ldots$, where the different pitches x_{ij} of each candidate set C_i are sorted in some given order (e.g., by decreasing weighted harmonicities). This strategy is useful if we want the candidate pitches to be considered in the prescribed order.

- The *hardest-first* strategy. In this strategy we always pick a node of smallest degree in the graph induced by the current node set V. That is, we take a node $v \in V$

which minimizes the value $|\{w \in V : vw \in E\}|$. The rationale behind this strategy is that we want to do "the hardest nodes first" in order to reduce the number of alternatives to consider in later stages of the algorithm. It has been observed by Carraghan and Pardalos that this approach in fact tends to reduce the overall running time if the input graph is dense. Thus this strategy is appropriate if there are many "harmonious" intervals in the harmonicity graph. We also found this strategy helpful when enumerating all solutions in order to find an optimal solution.

- The *random-first* strategy. Here we always select the next node at random (each node $v \in V$ is selected with the same probability). This strategy is useful if one wants to have a quick look at the average solution quality.

- The *best-first* strategy. In this strategy we always select a node $v \in V$ which minimizes $\sum_{w \in C}^{vw \in E} d(v, w)$ where $d(v, w)$ denotes the harmonic distance between the pitches represented by the nodes v and w. This strategy tends to enumerate solutions first which have a lower average harmonic distance; it effectively turns the algorithm into a "greedy" heuristic. This is useful if we want to find some "good" (but not necessarily optimal) solutions quickly.

Note that all described variations of the algorithm take exponential time in the worst case. As we have pointed out in the previous section, the scale rationalization problem is NP-complete already in its decision form and hence one cannot hope for a polynomial-time solution. However, we have found the procedures based on the Carraghan/Pardalos algorithm to be a good solution method for the problem, since they enable us to experiment with different search strategies and algorithm parameters until a good solution is obtained in a reasonable amount of time. The algorithm seems to be practical for the usual kinds of scales composers work with, which rarely have more than a few dozen pitches.

We remark that other, more advanced algorithmic approaches to the scale rationalization problem seem possible. As we point out in the following section, one can often "embed" scale metrics in the Euclidean plane or space. This fact could lead to more efficient algorithms for some special cases of the problem, by exploiting geometric structure. For instance, it is known that for a certain class of "geometric" graphs, namely the "unit disk" graphs, the clique problem can actually be solved in polynomial time using bipartite matching techniques [5]. Geometric graphs have their edges defined in terms of Euclidean distances between the nodes, just like we defined the edges of the harmonicity graphs in terms of harmonic distances. Thus it might be possible to apply similar techniques to solve some special cases of the scale rationalization problem as well.

6 Drawing a scale

We now discuss how to visualize a rational scale by drawing its harmonicity graph in 2- or 3-dimensional space in such a manner that the visible (Euclidean) distances between the scale tones provide a good approximation for the actual harmonic distances. (The harmonicity graph $G(S)$ of a rational scale $S = \{x_1, \dots, x_n\}$ is defined analogously to Definition 1, but now there is only one node v_i for each scale tone x_i.) Such visualizations make it easy to spot the harmonic relationships inside a scale and to compare different rationalizations of a scale by just taking a look at the corresponding pictures.

The method we employ for this purpose is called *multidimensional scaling* (henceforth abbreviated as "MDS"). Whenever we have a metric d on a finite set $S = \{x_1, \dots, x_n\}$, MDS can be used to *embed* the members of the set into m-dimensional Euclidean space, by assigning a point u_i to each $x_i \in S$ such that the Euclidean distances $d_2(u_i, u_j) = ||u_i - u_j||_2$ match the metric distances $d(x_i, x_j)$ as closely as possible. Of course, if we want to draw a real picture, say, on a computer screen, we better find an embedding in low-dimensional space. Hence for our purposes we concentrate on 2- and 3-dimensional embeddings. In our software we also perform a *principle axis* rotation of the resulting embedding, so that the coordinates with the greatest amount of variation are on the x and y axes. Another point that deserves mentioning is that the embeddings produced with MDS are usually not determined uniquely; depending on the choice of parameters, the MDS algorithm may produce distinct embeddings which correspond to different local minima of the "stress" function explained below.

MDS is routinely used in the social sciences to analyze statistical data involving similarity measurements. A fortunate consequence of this situation is that an abundance of different MDS methods is readily available, like, e.g., Torgerson's "classical" algorithm [12] and Kruskal's gradient method [10]. The method we mainly employ in our software is a modern MDS algorithm due to De Leeuw and Heiser, called "SMACOF" (the curious acronym stands for "scaling by majorizing a convex function"), see [2] for details. This algorithm, like most others, attempts to minimize the so-called *stress* of the embedding, which is a measure for the error in the representation, defined as:

$$\sigma = \sum_{i<j} (d_2(u_i, u_j) - d(x_i, x_j))^2.$$

If the stress is zero then the embedding provides a perfect visualization. This is rarely achieved since many interesting metrics are not Euclidean at all (this condition can be checked using Torgerson's algorithm), and even if a metric is Euclidean then it might not be perfectly embeddable in low-dimensional space. This also happens with harmonic distance metrics, but we have found that the embeddings produced for many interesting scales provide fairly good representations of the harmonic distances, at least for the Barlow and Euler metrics (we have not tried other metrics with a substantial number of examples yet). To determine the relative error in the representation one usually works with a relative kind of stress measure, called *stress-1*, which is defined as the ratio between σ and the total of the squared metric values, $\sum_{i<j} d(x_i, x_j)^2$. A rule of thumb says that an MDS solution is acceptable if the stress-1 value, as a percentage, does not exceed some 10%.

As an example, Fig. 3 shows the Barlow harmonicity graph of an Indian shruti scale which exhibits a fairly regular structure. The edges of the graph are for a harmonicity threshold of 0.1 (i.e., edges are shown for all intervals with a maximum Barlow disharmonicity of 10). The picture makes it easy to spot the chains of fifths in the scale, as well as the thirds and sixths between the central chain and the two "side cords". Note that this is just a side view of a 3-dimensional embedding; the chains of fifths are actually curved in the direction of the z axis which is invisible in the figure.

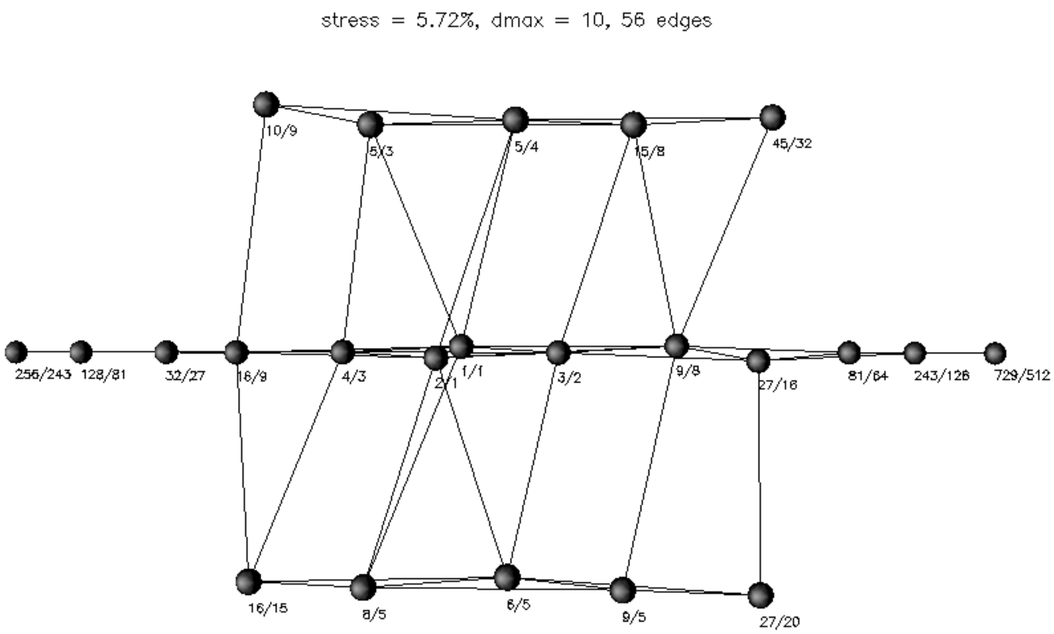


Figure 3: Indian shruti scale.

7 The scale program

The algorithms described in this paper have actually been implemented in the *scale* program, an interactive, graphical software for scale rationalization and visualization. This program enables the user to enter a scale of pitches specified either as integer ratios or cent values, rationalize the scale using the algorithms described in Section 5, and draw a 2- or 3-dimensional image of the rationalized scale using the method sketched out in Section 6. Scale images can be saved in most popular image formats (this is how Fig. 3 was obtained). The program also includes a simple "MIDI tuner" with which the user can tune a connected MIDI synthesizer to the current scale, in order to actually listen to the scale being worked upon.

The scale program is freely available for download from the author's website.[1] It supports the popular scale file format of the *scala* program by Manuel Op de Coul. An

[1]See `http://www.musikwissenschaft.uni-mainz.de/~ag/q/`. To run the scale program, you need a Linux system with Q, Octave, OpenDX and Tcl/Tk. The latter three items are included in most recent Linux distributions. Download links for all required software can be found on the aforementioned website.

extensive archive with more than 2800 different scales in scala format is available from Op de Coul's website.[2]

A screenshot of the program's main window, showing a rationalized chromatic scale rendered using the Barlow metric, is depicted in Fig. 4. In the following we walk through a typical session with the program, in order to obtain this picture.

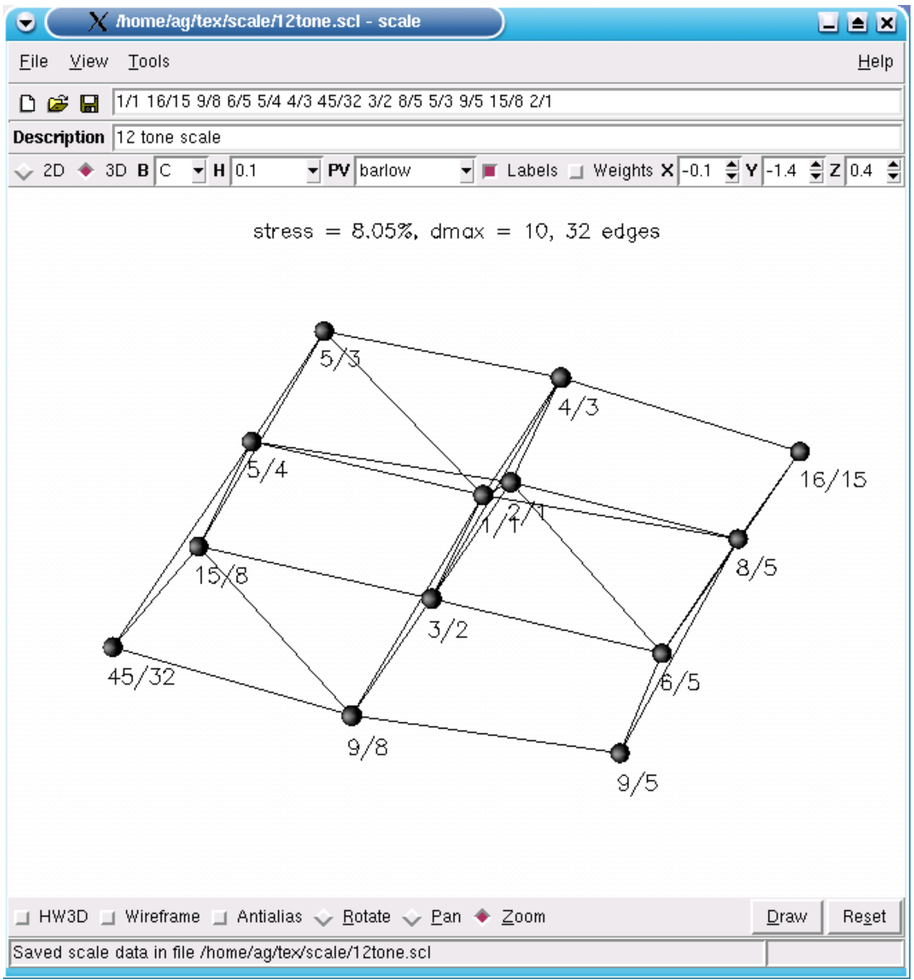
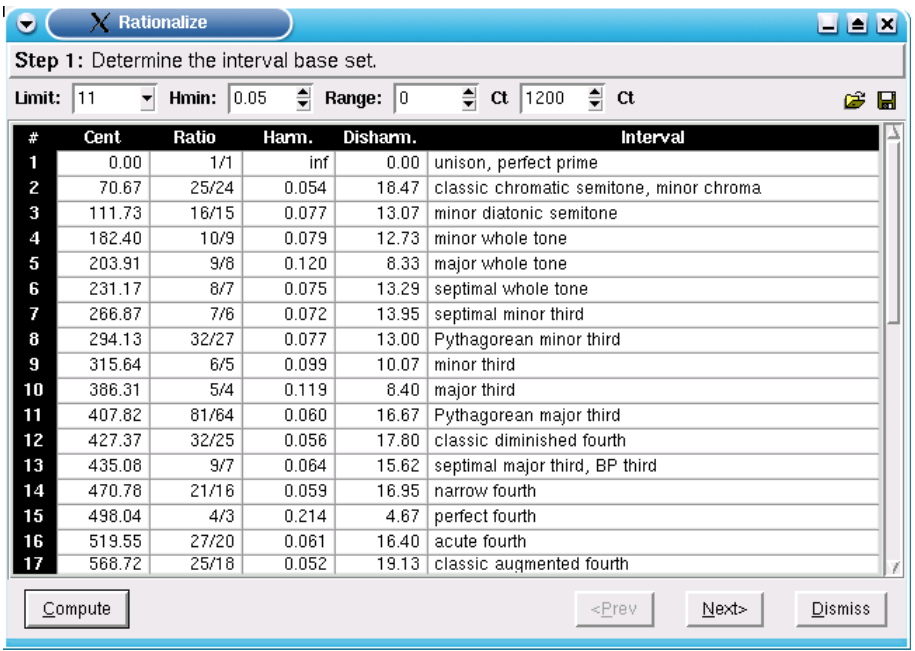


Figure 4: Scale program with rationalized 12 tone scale.

We start out with the equal-tempered 12 tone scale. For this purpose we enter the scale pitches as cent values into the line at the top of the window, right below the menubar. We enter the following values: 0.0 100.0 200.0 300.0 400.0 500.0 600.0 700.0 800.0 900.0 1000.0 1100.0 1200.0. We can also enter a brief description of the scale in the *Description* field. First we will rationalize the scale using the Barlow metric (as specified with the *PV* field in the main window). For this purpose we invoke the *Rationalize* operation in the *Tools* menu. A dialog appears with which we perform the following three steps:

[2]See `http://www.xs4all.nl/~huygensf/scala/`.

1. *Determine the interval base set.* Here we enter the desired (prime) limit, minimum harmonicity and cent range of the rational pitches to consider as possible tunings for the scale tones. In our example we use the default limit (11) and cent range (one octave) and we set the minimum harmonicity to 0.05. (The default harmonicity value of 0.06 would only yield the 24 most harmonious intervals which do not include the tritone.) After hitting *Compute* we obtain a list of 38 intervals (cf. Fig 5). We then hit the *Next* button to proceed with the next step.



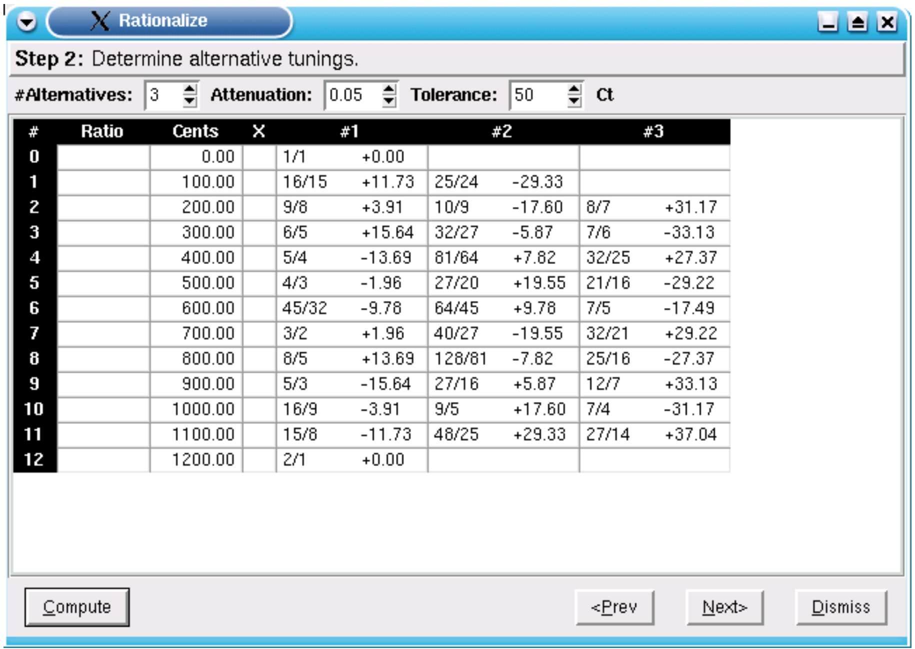
#	Cent	Ratio	Harm.	Disharm.	Interval
1	0.00	1/1	inf	0.00	unison, perfect prime
2	70.67	25/24	0.054	18.47	classic chromatic semitone, minor chroma
3	111.73	16/15	0.077	13.07	minor diatonic semitone
4	182.40	10/9	0.079	12.73	minor whole tone
5	203.91	9/8	0.120	8.33	major whole tone
6	231.17	8/7	0.075	13.29	septimal whole tone
7	266.87	7/6	0.072	13.95	septimal minor third
8	294.13	32/27	0.077	13.00	Pythagorean minor third
9	315.64	6/5	0.099	10.07	minor third
10	386.31	5/4	0.119	8.40	major third
11	407.82	81/64	0.060	16.67	Pythagorean major third
12	427.37	32/25	0.056	17.80	classic diminished fourth
13	435.08	9/7	0.064	15.62	septimal major third, BP third
14	470.78	21/16	0.059	16.95	narrow fourth
15	498.04	4/3	0.214	4.67	perfect fourth
16	519.55	27/20	0.061	16.40	acute fourth
17	568.72	25/18	0.052	19.13	classic augmented fourth

Figure 5: Scale rationalization, step 1.

2. *Determine alternative tunings.* We now select the number of desired tuning alternatives for each scale tone, as well as the harmonicity attenuation and the tolerance range (cf. Section 4) which determine which alternatives from the base set computed in step 1 will be selected. For our example we simply accept the default values (at most 3 tuning alternatives, attenuation factor 0.05, tolerance 50 cents), hit *Compute* to calculate the candidate pitches and then *Next* to go to the next step. See Fig. 6.

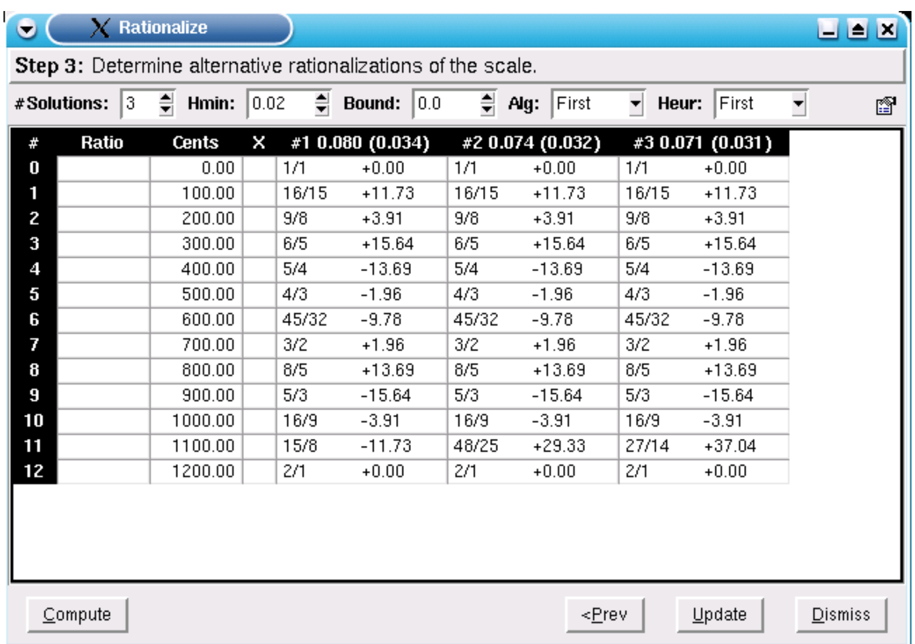
3. *Determine alternative rationalizations of the scale.* Now the stage is set to actually compute different rationalizations of the scale. The configurable parameters in this step are the following (cf. Fig. 7):

   - the number of alternative solutions solutions we want to compute;
   - a global minimum harmonicity for all intervals (this value, inverted, provides a default value for the maximum disharmonicity bounds);
   - a lower bound for the specific harmonicity of the solution;



#	Ratio	Cents	X	#1		#2		#3	
0		0.00		1/1	+0.00				
1		100.00		16/15	+11.73	25/24	-29.33		
2		200.00		9/8	+3.91	10/9	-17.60	8/7	+31.17
3		300.00		6/5	+15.64	32/27	-5.87	7/6	-33.13
4		400.00		5/4	-13.69	81/64	+7.82	32/25	+27.37
5		500.00		4/3	-1.96	27/20	+19.55	21/16	-29.22
6		600.00		45/32	-9.78	64/45	+9.78	7/5	-17.49
7		700.00		3/2	+1.96	40/27	-19.55	32/21	+29.22
8		800.00		8/5	+13.69	128/81	-7.82	25/16	-27.37
9		900.00		5/3	-15.64	27/16	+5.87	12/7	+33.13
10		1000.00		16/9	-3.91	9/5	+17.60	7/4	-31.17
11		1100.00		15/8	-11.73	48/25	+29.33	27/14	+37.04
12		1200.00		2/1	+0.00				

Figure 6: Scale rationalization, step 2.



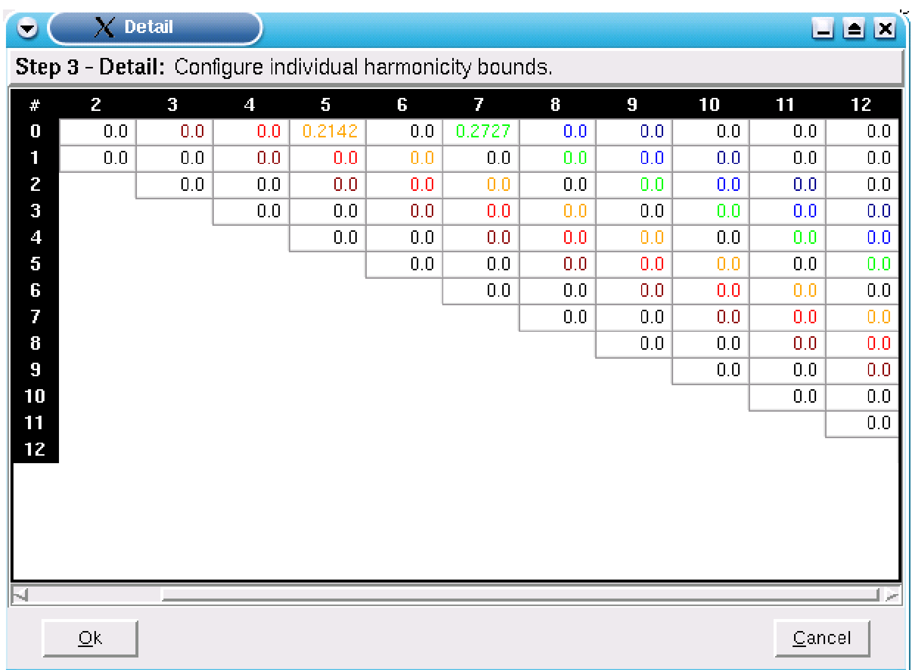
#	Ratio	Cents	X	#1 0.080 (0.034)		#2 0.074 (0.032)		#3 0.071 (0.031)	
0		0.00		1/1	+0.00	1/1	+0.00	1/1	+0.00
1		100.00		16/15	+11.73	16/15	+11.73	16/15	+11.73
2		200.00		9/8	+3.91	9/8	+3.91	9/8	+3.91
3		300.00		6/5	+15.64	6/5	+15.64	6/5	+15.64
4		400.00		5/4	-13.69	5/4	-13.69	5/4	-13.69
5		500.00		4/3	-1.96	4/3	-1.96	4/3	-1.96
6		600.00		45/32	-9.78	45/32	-9.78	45/32	-9.78
7		700.00		3/2	+1.96	3/2	+1.96	3/2	+1.96
8		800.00		8/5	+13.69	8/5	+13.69	8/5	+13.69
9		900.00		5/3	-15.64	5/3	-15.64	5/3	-15.64
10		1000.00		16/9	-3.91	16/9	-3.91	16/9	-3.91
11		1100.00		15/8	-11.73	48/25	+29.33	27/14	+37.04
12		1200.00		2/1	+0.00	2/1	+0.00	2/1	+0.00

Figure 7: Scale rationalization, step 3.

- the type of algorithm and search heuristic to invoke.

For a first run, we can simply keep the default settings which will invoke the scale rationalization algorithm described in Section 5 to compute the first three solutions using the “first-first” strategy. A short while after hitting the *Compute* button the solutions will be displayed. Except for the first solution with a specific harmonicity of about 0.08, these solutions are not very good, so let’s try the “best-first” strategy instead (we select *Best* in the *Heur* field and push *Compute* again). Now we already have two solutions with a specific harmonicity of 0.08.

But we are still not satisfied; now we want to go for a (nearly) optimal solution, by selecting the *Best* algorithm in the *Alg* field. This requires some careful tuning of the algorithm parameters since the real optimum might take hours to compute. Taking a look at the best solutions from the second run we see that all these solutions have a minimum interval harmonicity (the values in parentheses behind the specific harmonicities in each column header) of more than 0.033. So we decide to increase the value in the *Hmin* field to 0.033; this will reduce the number of edges in the harmonicity graph and hopefully speed up the algorithm.



#	2	3	4	5	6	7	8	9	10	11	12
0	0.0	0.0	0.0	0.2142	0.0	0.2727	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3			0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4				0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5					0.0	0.0	0.0	0.0	0.0	0.0	0.0
6						0.0	0.0	0.0	0.0	0.0	0.0
7							0.0	0.0	0.0	0.0	0.0
8								0.0	0.0	0.0	0.0
9									0.0	0.0	0.0
10										0.0	0.0
11											0.0
12											

Figure 8: Scale rationalization, step 3, detail settings.

Next, using the *Detail* button in the right upper corner of the rationalization dialog we can bring up a second dialog which allows us to choose minimum harmonicity values (the inverse of the disharmonicity bounds) for all intervals in the scale; see Fig. 8. The dialog has the form of an upper triangle matrix (only these values need to be specified since the disharmonicity bounds are symmetric). Some entries are colored so that one can easily spot the "main" intervals (thirds, fourths, fifths and sixths). The harmonicity bounds can be selected from a context menu which pops up when pressing the right mouse button inside a cell of the matrix; the menu displays the harmonicity values in decreasing order for the corresponding interval combinations, given the available candidate pitches. For our example, we just selected the largest possible harmonicity values for the fourth and fifth in the first row of the matrix. This effectively fixes these two intervals over the base tone at 4/3 and 3/2, respectively. By pressing *Ok* we accept these values and return to the rationalization dialog.

To further speed up the search we finally select the "hardest-first" strategy in the *Heur* field. After hitting *Compute* and waiting for some time (on an Athlon 1.4GHz PC this computation takes some 5 minutes) the program has computed the solutions shown in Fig. 9 which are optimal for the minimum harmonicity values we specified. Each column lists one solution, and in each row we find the ratio for one scale tone as well as the corresponding tuning offset, i.e., the difference in cents from the original pitch. You will notice that the computed solutions actually use the common just intervals for the twelve semitones. The three solutions only show some minor differences in the tuning of the tritone and the minor seventh.

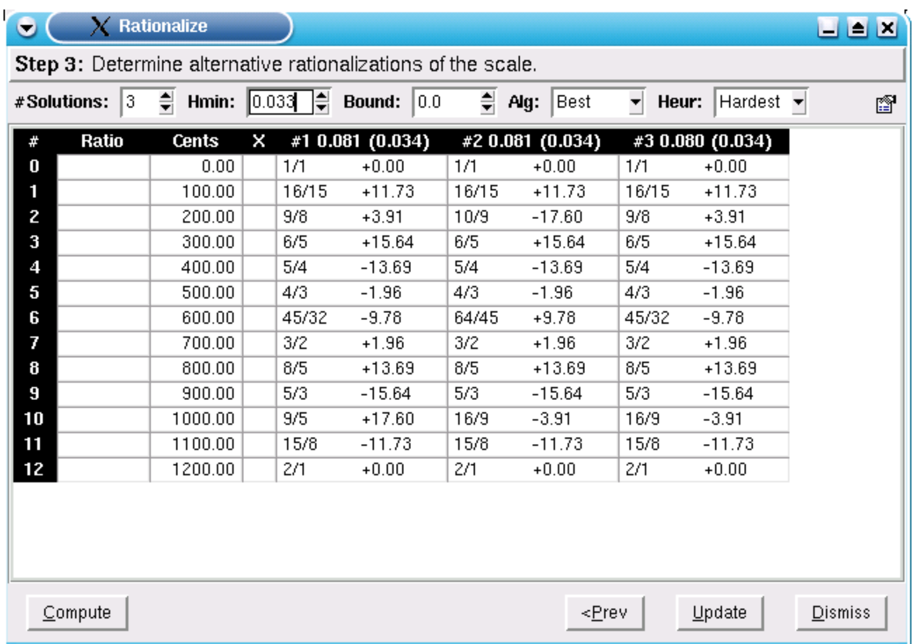


Figure 9: Scale rationalization, final result.

To finish the rationalization, we select one of the solutions (say, the first one) by clicking on the corresponding column header and press *Update* to update the scale in the main window accordingly. Next we want to draw a harmonicity graph of the scale. We enter a minimum harmonicity value for the edges to be drawn (say, 0.1) in the *H* field of the main window and push the *Draw* button to embed the scale using MDS and draw the picture.

The result is shown in Fig. 4. One can rotate the picture to see the spatial structure of the embedding by dragging the mouse pointer in the image window. Finally, we might wish to tune our synthesizer to the rationalized scale. For this purpose we invoke the MIDI tuner (operation *Tune* in the *Tools* menu), which displays a dialog for selecting the tunings for the 12 MIDI notes in the octave; see Fig. 10.

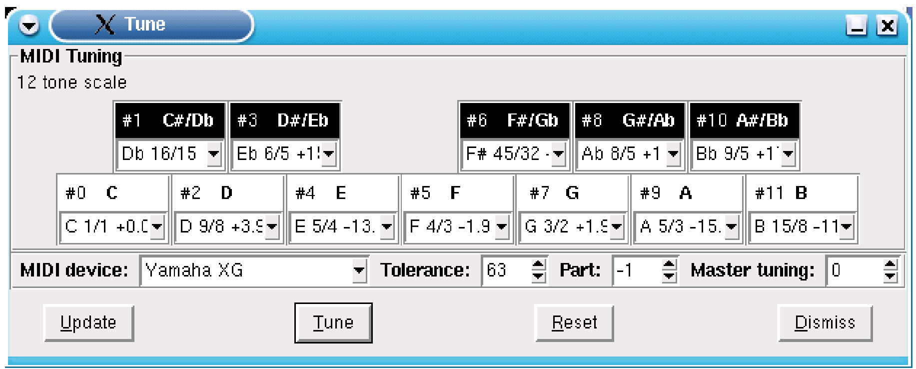


Figure 10: MIDI tuner dialog.

In a similar manner the scale program can be used to rationalize any scale one might think of. There is a practical limit on the size of the scale, however, since the rationalization algorithm takes exponential time in the worst case. For large scales consisting of several dozen pitches the complete enumeration procedure probably needs ages to finish, but a reasonably good rationalization might still be obtainable using the "best-first" heuristic. Another way to reduce the running time for large scales is to perform the rationalization in an incremental fashion, by first rationalizing the most prominent pitches of the scale which then remain fixed while the remaining pitches are considered.

As another example, Fig. 11 shows a rationalization of the 1/4 comma meantone scale, a kind of temperament which was very popular in Europe from approximately 1500 to 1700. This rationalization of the scale plays a role in one of Barlow's most recent compositions. Starting from the cent values 0.0 76.0 193.2 310.3 386.3 503.4 579.5 696.6 772.6 889.7 1006.8 1082.9 1200.0, the rationalization is easily obtained with the scale program as the best solution for a global minimum harmonicity value of 0.033 as above, and maximum harmonicities for the minor third and perfect fourth and fifth over the base tone. (The other solution parameters are as in the preceding example.)

8 Conclusion

We can summarize the main results of this paper as follows:

- The consonance of rational intervals can be measured in terms of subadditive harmonicity functions which give rise to corresponding harmonic distance metrics.

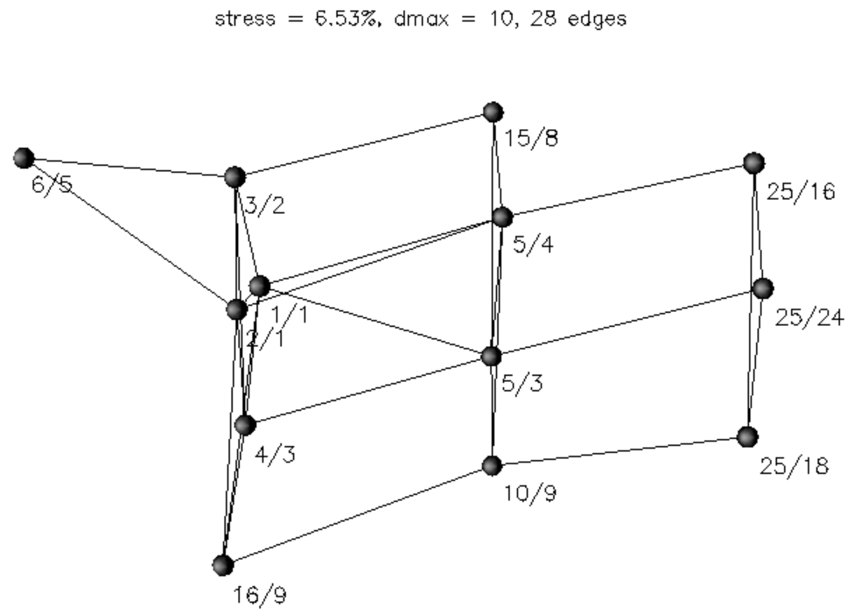


Figure 11: Rationalization of the 1/4 comma meantone scale.

- This approach leads to a formulation of the scale rationalization problem as a clique problem on a harmonicity graph constructed from the different rational candidate pitches for each scale tone and the given disharmonicity bounds.
- The problem is NP-complete.
- Nevertheless the problem can often be solved in a reasonable amount of time using branch and bound search heuristics for the clique problem.
- Harmonic distance metrics can also be used to draw the harmonicity graph of a scale in such a manner that the visible Euclidean distances provide a good approximation of the harmonic distances.

As we stated in Section 4, the scale rationalization problem turns out to be NP-complete already if at most three candidate pitches are given for each scale tone. This raises the question whether the problem becomes polynomial-time solvable if we only consider two alternatives per pitch. Also, the reduction from 3SAT in our NP-completeness proof employs big prime numbers and thus does not rule out the possibility of a "pseudo-polynomial" algorithm (cf. [8, Section 4.2]).

Another interesting direction for further research are more advanced algorithms which exploit the geometric structure of the harmonicity graph. In particular, as we already mentioned in Section 5, it should be interesting to study the cases in which the problem can be solved using matching techniques. Unfortunately, it does not seem that these techniques will help to solve the optimization version of the problem in which, e.g., the specific harmonicity is to be maximized.

Yet another topic for future research are other potential applications. For instance, it has been pointed out by Barlow that scale rationalization techniques can also be used to tune instruments in realtime and to perform rhythmic quantization. Our methods should be applicable to these problems as well, provided that the number of pitches or time intervals to be considered at the same time is not too large.

Acknowledgements

I would like to thank Clarence Barlow for introducing me to the topic of scale rationalization, for many enlightening discussions, and for commenting on earlier drafts of this paper.

References

[1] C. Barlow. On the quantification of harmony and metre. In C. Barlow, editor, *The Ratio Book*, Feedback Papers 43, pages 2–23. Feedback Publishing Company, Cologne, 2001.

[2] I. Borg and P. Groenen. *Modern Multidimensional Scaling*. Springer Series in Statistics. Springer, New York, 1997.

[3] R. Carraghan and P. M. Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9:375–382, 1990.

[4] J. Chalmers. *The Divisions of the Tetrachord*. Frog Peak Music, 1993.

[5] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[6] P. Erlich. On harmonic entropy. Mills College Tuning Digest, 1997. See `http://sonic-arts.org/td/entropy.htm`.

[7] L. Euler. Tentamen novae theoriae musicae ex certissimis harmoniae principiis dilucide expositae. St. Petersburg, 1739.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[9] H. v. Helmholtz. *Die Lehre von den Tonempfindungen als physiologische Grundlage für die Theorie der Musik*. Vieweg, Hildesheim, 6th edition, 1983.

[10] J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29:115–129, 1964.

[11] J. Tenney. *A History of 'Consonance' and 'Dissonance'*. Excelsior Music Publishing Company, New York, 1988.

[12] W. S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17:401–419, 1952.