An application of evolutionary algorithms and machine learning in four-part harmonization*

Mikołaj Sikora[0009−0004−7483−0370] and Maciej Smołka[0000−0002−3386−0555]

Institute of Computer Science
AGH University of Krakow, Kraków, Poland
https://www.informatyka.agh.edu.pl/en/
{mikolaj.sikora, smolka}@agh.edu.pl

Abstract. The task of four-voice harmonization of a given melody is one of the most fundamental, but at the same time the most complex problems in functional harmony. This problem can be formulated as a discrete optimization problem with constraints using a set of rules coming from the theory of music. Unfortunately, a straightforward solution of such a problem, i.e., a mere fulfillment of the rules, ensures only the formal correctness of the obtained chord sequences, which does not necessarily imply overall musical quality as perceived by humans. Trying to catch some non-formalized factors of this quality we have decided to utilize artificial intelligence methods with some 'creative' potential that can provide solutions at acceptable level of formal correctness. In this paper we perform the harmonization using a genetic algorithm, an algorithm based on a Bayesian network, as well as a hybrid of these. In a series of experiments we compare the performance of the three algorithms with each other and with a rule-based system that provides chord sequences at a high level of formal correctness. Besides the formal evaluation all obtained solutions were rated by musical experts. The results show that the studied algorithms can generate solutions musically more interesting than those produced by the rule-based system, even if the former are less formally correct than the latter.

Keywords: algorithmic composition · constrained discrete optimization · evolutionary algorithms · machine learning

1 Introduction

Harmony is one of the most important branches of music theory. It governs the co-sounding of several melodies by arranging simultaneous sounds into chords and setting the rules of chord structure and succession. Therefore, it is one of the key factors of musical composition. The foundations of modern harmony date back to the Baroque musical period, i.e., the 17th and the first half of 18th centuries. The main assumptions of the Baroque approach, called the *functional*

* This research was supported in part by the funds of Polish Ministry of Education and Science assigned to AGH University of Krakow.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

harmony [11] (a modern description can be found in [13]), were in the mainstream till the beginning of the 20th century and still are foundations of many modern musical genres. In this approach melodies are composed using seven-note *major* or *minor* scales depending on their global key. At every step (i.e., note) of the scale we can build a chord. Basic chords are three-note *triads*, more sophisticated can be built by adding appropriate extra elements, omitting or altering some of existing ones. A key notion in the functional harmony is *harmonic function* of a chord, which determines the role of the chord in a musical composition and in consequence sets the rules for allowed connections with other chords. The basic harmonic functions are: *tonic (T)* based on the first note of the scale, *subdominant (S)* based on the fourth note and *dominant (D)* based on the fifth note. Each of the remaining functions is in a way related to T, S or D, which is then its *base* function, e.g., the chord built on the second note is a kind of subdominant.

The rules of functional harmony can be divided into two groups. The first group determines the validity of connections between subsequent chords and the second one governs the chord structure. In both groups there are *hard rules* which cannot be broken and *soft rules* which can be broken, but breaking these rules degrades the musical quality of the composition.

Four-part harmonization is a problem where the input single-voice melody is transformed into a four-voice choir score using the harmony rules. The score consists of four separate voices (i.e., melodies): the highest soprano with the input melody, alto, tenor and the lowest bass. The harmonization problem is considered fundamental in the musical education and in the professional composition. At the same time it is a complex task, because often it is hard (or even impossible) to satisfy all the rules. Moreover, even compliance with all the rules is not sufficient to obtain a composition that would be satisfactory from the aesthetic point of view.

1.1 State of the art

The soprano harmonization problem has been deeply studied by computer scientists for the last 30 years. Rule-based expert systems are the most classical method of solving this problem with computers. One of the most famous is *CHORAL* [4] which models 350 functional harmony rules. It was tested on Johann Sebastian Bach's chorales. Rule-based systems perform deterministic solution space search and they are a practical way to satisfy all fundamental rules. In this paper we use a rule-based system described in [2] for three purposes. First, it serves as a reference level for studied stochastic algorithms. Second, it detects broken rules in any solution and provides an evaluator of objectives in our optimization task. Third, it supports the Bayesian network (see below).

The first genetic algorithm solving the harmonization problems was created by McIntyre [7]. It was only able to generate solutions in C-major key. It used the single-point crossover and the binary tournament selection. It turned out that a major challenge is the definition of appropriate genetic (especially mutation) operators. Phon-Amnuaisuk and Wiggins [9] proposed a genetic algorithm with

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

a fitness function based on functional harmony rules. The more rules were broken, the greater (i.e., worse) the fitness value was. The main problem with this solution was the fact that when one connection was fixed, another connection became illegal. De Vega [14] also showed that *divide and conquer* approach does not work in this case. It should be noted that the notion of harmonic function is rarely used in the construction of harmonization algorithms. Paper [8] shows one of notable exceptions.

Neural networks has become very popular machine learning model over the recent years, also in the domain of soprano harmonization. Probably the first neural network to solve the problem was *HARMONET* [5]. Since soprano harmonization can be formulated as a time series prediction task, Long Short-Term Memory networks (LSTMs) and their variations, such as BiLSTMs [6, 16], are currently the most common approaches. For training researchers usually use J. S. Bach's chorales.

Yet another approach to the harmonization is based on Markov models [16, 17]. Some researchers use Markov Decision Processes [17] with awards related to functional harmony rules. Others tried to solve the multi-criteria problem in the case of melodies, chords and tonality [10]. Finally, there exist approaches which use N-gram models, which are solved by Prediction By Partial Match algorithm [15].

1.2 Contribution

As said before, the most of the existing approaches do not use the *harmonic function* term. The main contribution of this paper is the use of this term to model functional harmony rules and create abstraction over the key of the musical pieces, which is useful to model tensions and release them. We propose a model of a chord and of a harmonic function (described in the next section) that covers the whole domain of the Baroque harmony, i.e., allows us to algorithmically handle all chords of this era. This approach can then be used in the construction of algorithms that generate results in more aesthetically pleasing solutions of the soprano harmonization problem. We study three such algorithms: a genetic algorithm, an algorithm based on the Bayesian network and a hybrid of these. We aimed at aesthetically satisfying harmonizations, which is hard to rate automatically. Therefore, the performance of the algorithms was evaluated by human experts, mostly teachers of the Academy of Music in Kraków.

A minor but important contribution concerns the second and the third of the above-mentioned algorithms. Namely, they both make use of machine learning, so they require an appropriate training data. Unfortunately, to the best knowledge of the authors, there are no musical data sets labelled with harmonic functions of chords. Therefore we needed to prepare such a set based on a set of J. S. Bach chorale scores.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

2 Soprano harmonization problem

In this section we provide a mathematical formulation of the soprano harmonization problem that we shall make use of in the sequel. We put the problem into the framework of discrete optimization with constraints.

We model a chord as a tuple (s, a, t, b, hf, loc), where s, a, t, and b are, respectively, soprano, alto, tenor and bass notes, hf is a harmonic function and loc is the information about the location inside the bar: downbeat, on-beat and off-beat. A harmonic function is a tuple $(base, deg, inv, e, o, down, pos, sys, key)$, where $base$ is the base function (T, S or D), deg is the degree (a note of the scale on which the chord is built), inv is the inversion (chord component in the bass voice), e is a list of extra components, o is a list of omitted components, $down$ is an indication if the degree is lowered by a semitone, pos is the position (a note in the soprano voice), sys is the chord *system* (describing distances between neighboring voices, can be *open*, *close* or *undefined*) and key is chord's key, which is necessary to model some special chords (it is related to the specific step of the scale or its alteration, e.g., a secondary dominant to D has the key equal to the second step of scale). For the attributes of a chord or of a function we adopt the functional notation, e.g., $hf(c)$ shall denote the harmonic function of chord c, $s(c)$ the soprano note of c and $key(f)$ the key of function f.

Let us now make some observations on the set of functional harmony rules. Let C be the set of all chords, let F be a set of all possible harmonic functions. We divide the set R of harmonic rules into disjoint subsets R_c and R_f of rules concerning chords and harmonic functions, respectively. As said before, $R_c = H_c \cup S_c$ and $R_f = H_f \cup S_f$, where H_c, H_f are sets of hard rules and S_c, S_f are sets of soft rules with $H_c \cap S_c = \emptyset$ and $H_f \cap S_f = \emptyset$. Furthermore, we observe that $R_c = R_{c,1} \cup R_{c,2} \cup R_{c,3}$ and $R_f = R_{f,1} \cup R_{f,2}$, where rules $R_{t,i}$ are applied to i consecutive chords or functions for $t \in \{c, f\}$ and $i \in \{1, 2, 3\}$. The final observation is that there are no hard rules applicable to 3 chords, i.e., $H_c \cap R_{c,3} = \emptyset$.

To formulate the harmonization as an optimization problem we introduce the following penalty function.

$$p\colon \left(\bigcup_{i=1}^{3} R_{c,i} \times C^i\right) \cup \left(\bigcup_{i=1}^{2} R_{f,i} \times F^i\right) \longrightarrow \mathbb{R}_+ \cup \{+\infty\}. \tag{1}$$

We assume that

$$p(r, x_1, \ldots, x_k) = 0 \tag{2}$$

if $(x_1, \ldots, x_k) \in C^k \cup F^k$ comply with $r \in R$,

$$p(r, x_1, \ldots, x_k) = +\infty \tag{3}$$

if $(x_1, \ldots, x_k)$ break hard rule $r \in H_c \cup H_f$ and

$$0 < p(r, x_1, \ldots, x_k) < +\infty \tag{4}$$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

if $(x_1, \ldots, x_k)$ break soft rule $r \in S_c \cup S_f$. The penalty function allows us to define two objectives: the one involving rules related to chords

$$\begin{aligned} V_c(c_1, \ldots, c_n) = \sum_{r \in R_{c,1}} \sum_{j=1}^{n} p(r, c_j) + \sum_{r \in R_{c,2}} \sum_{j=1}^{n-1} p(r, c_j, c_{j+1}) \\ + \sum_{r \in R_{c,3}} \sum_{j=1}^{n-2} p(r, c_j, c_{j+1}, c_{j+2}) \end{aligned} \tag{5}$$

and the one involving rules related to harmonic functions

$$V_f(f_1, \ldots, f_n) = \sum_{r \in R_{f,1}} \sum_{j=1}^{n} p(r, f_j) + \sum_{r \in R_{f,2}} \sum_{j=1}^{n-1} p(r, f_j, f_{j+1}). \tag{6}$$

Finally, the harmonization problem is as follows: given a sequence of soprano voice notes $S = (s_1, s_2, \ldots s_n)$, a sequence of their locations inside bars $(l_1, \ldots, l_n)$ (that is connected to the *time signature* of a piece) and a key of the whole melody k we seek for a sequence of chords $(c_1^*, c_2^*, \ldots c_n^*)$ and a sequence of harmonic functions $(f_1^*, f_2^*, \ldots f_n^*)$ that are a Pareto solution of two-criteria problem

$$(V_c(c_1^*, \ldots, c_n^*), V_f(f_1^*, \ldots, f_n^*)) = \min (V_c(c_1, \ldots, c_n), V_f(f_1, \ldots, f_n)) \tag{7}$$

where the minimum is taken over all $(c_1, \ldots, c_n) \in C^n$, $(f_1, \ldots, f_n) \in F^n$ that satisfy constraints

$$hf(c_i) = f_i, \quad s(c_i) = s_i, \quad loc(c_i) = l_i, \quad key(f_i) = k, \tag{8}$$

for $i = 1, \ldots, n$. The input of the soprano harmonization problem consists of a soprano melody, a time signature and a key. The former two are first transformed into sequences $(s_1, \ldots, s_n)$ and $(l_1, \ldots, l_n)$. The main objective is to find the remaining three voices, where each note of every voice corresponds to a single note in the soprano, such that the resulting sequence of chords optimizes total penalties V_c and V_f, i.e., it minimizes the number of broken rules for connections between chords (5) as well as the number of broken rules for connections between harmonic functions (6). As for the constraints, they guarantee that the output soprano melody is the same as the input melody, the output chord c_i^* has function f_i^* and that the output sequence of chords complies with the time signature and the key. Additionally, we would like to achieve solutions that are pleasing and interesting for the human ear. Some of the soft rules are aimed at meeting such expectations, but it is hard (probably even impossible) to cover all aesthetic aspects in a mathematical way.

Below we list soft and hard rules used in this paper. They are collected from the music theory literature [11–13] and discussed with domain experts.

– **Hard rules for chord connections** prohibit: parallel octaves, parallel fifths, overlapping voices, putting alto above soprano, tenor above alto or bass above tenor, exceeding prescribed voice ranges, motion of all voices in one direction, certain jumps in the same voice, false relation, harmonic function repetition without a motion of voices.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

- **Soft rules for chords connections** concern: avoiding sum of jumps resulting in a forbidden jump in the same voice, constructing alto, tenor and bass melodies using the smallest possible intervals, preferring some standard movement of voices in particular connections, preferring doubling of the soprano component for simple triad chords, preferring putting the fifth in soprano when there is also the fifth in bass.
- **Hard rules for harmonic function connections** prohibit: (major D)-S connection, placing an unrelated chord after a secondary dominant, using the fifth in bass on an off-beat location, putting together a harmonic function with same function lowered, changing base harmonic function when the previous harmonic function has the same degree, changing the inversion when the degree is increasing by one, doubling the third of a chord unless it is the Neapolitan chord or a VI degree chord following a dominant.
- **Soft rules for harmonic function connections** concern: preferring D-T, S-T and secondary dominants, promoting using non-basic chords, promoting large T-S-D-T connections, discouraging from changing harmonic functions on off-beats, promoting changing functions on every downbeat, preferring the Neapolitan chord over the simple second degree function, preferring T, S and D in meaningful places in harmonization, preferring dominants with the seventh over simple dominants.

As said before, some soft rules are designed to meet the aesthetic expectations of human ears. Their aim is to make final harmonizations more interesting and expressive, but they are not strictly connected to the formal correctness of the resulting sequence of chords. Putting particular non-zero penalty values on such rules can be seen as a mathematical way of preference expression. It is worth noticing that some of the soft rules are conflicting and cannot be fully satisfied simultaneously. In consequence, obtained harmonizations hardly ever can have zero values of the objectives.

3 Algorithmic approach

In this section we shall describe the algorithms used to solve the harmonization problem.

3.1 Genetic algorithm

To apply genetic algorithms one needs to define the form of an individual (i.e., a potential solution), a fitness function and kinds of genetic operators transforming sets of individuals. In our case, an individual is simply a sequence of chords, where a component chord fills the role of chromosome with the chord attributes as genes.

Then, we define function g transforming a soprano note into a set of chords harmonizing it. Such a construction is possible and independent on a particular problem, because set F of all harmonic functions is well-known and finite. For

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

every melody note we know the key, its location in the bar, so we can find the subset of F where every function harmonizes the melody note. Finally, we can create all instances of chords for these harmonic functions, where every chord contains the considered melody note in the soprano voice. The construction of g allows us to efficiently compute the fitness (see below) and ensures that all chords generated by genetic operators are valid, which solves a common problem depicted in existing studies [1].

According to equations 5 and 6 we define fitness as the pair (V_c, V_f). We reuse the definition of p from equation 1, relaxing it only for broken hard rules, i.e., changing (3) into

$$p(r, x_1, \ldots, x_k) = 1000 \tag{9}$$

for $r \in H_c \cup H_f$. For soft rules we use penalties less than 70, so the above penalty acts as 'soft infinity' and is appropriate for the genetic algorithm used.

We consider two types of single-individual genetic operators, called here *mutators*. The first type is formed by **repair operators**, which are applied with 100% probability and correct errors in specific places where a given rule has been broken. The second type consists of **classic mutation operators** that explore the solution space and are applied with significantly smaller probability.

We use the following repair operators.

i) *DSConnectionSMutator*: in the S-D connection it changes S to T and replaces the chord with a random one with this harmonic function.
ii) *DSConnectionDMutator*: similar as above, but changes D to T.
iii) *SingleThirdMutator*: when a chord contains too many thirds it changes one of them to a different chord component (e.g., fifth).
iv) *DTMutator*: it ensures correctness of the connection D-T.
v) *SeventhToThirdDTMutator*: in the connection of (D with seventh in bass) - T, changes bass chord component in T chord to the third.

Classic mutation operators with probability of their application are as follows.

i) *ChangeBaseFunctionMutator* (0.1): changes base function (T, S, D) to another.
ii) *SwapComponentsMutator* (0.15): swaps alto chord component with tenor chord component.
iii) *ExpandToQuadrupleMutator* (0.1): appends extra chord component to a simple triad chord (e.g., seventh to a pure dominant).
iv) *ChangeInversionMutator* (0.15): changes the inversion of a chord.
v) *ChangeBassOctaveMutator* (0.25): moves bass note an octave up or down.
vi) *AddOmit1ToDominantMutator* (0.05): omits the root in the dominant chord,
vii) *AddOmit5ToDominantMutator* (0.05): omits the fifth in the dominant.
viii) *ChangeSystemMutator* (0.1): changes harmonic function system from open to close or vice versa.
ix) *ChangeDegreeMutator* (0.125): changes the degree of harmonic function (without changing the base harmonic function),
x) *IntroduceModulationMutator* (0.1): introduces a modulated chord.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

When a mutator is applied, a new chord is selected randomly from the set generated by g using a probability distribution based on a *chord distance*. The *chord distance* is a sum of many components: difference between MIDI numbers of every voice's note, difference of degrees, difference of base harmonic functions, difference of keys, difference of number of different chord components. Numerous experiments showed that using this metric improves the performance of the genetic algorithm and the mutation process. Moreover, the mutators related to harmonic function changes are disabled after a half of genetic epochs. Therefore, in the final half of computations we focus on the improvement in the correctness of chord connections.

An initial population is chosen randomly. We use a one-point crossover on the bar line with probability 0.2. As the selection operator we use the binary tournament based on scalarized fitness $V_c + V_f$. For the succession we use *NSGA-II* operator [3] choosing non-dominated survivors according to (V_c, V_f) vector value.

3.2 Bayesian network

Bayesian networks (BNs) are of course machine learning tools and not global optimization methods. Therefore, it cannot be directly applied in the solution of problem (7)–(8). Instead, we use a BN to predict harmonic functions matching given soprano notes. This way we obtain a sequence of harmonic functions that are then passed to the rule-based system, which solves the easier harmonization problem with additional harmonic function labels.

As a training set we used chorales composed by Johann Sebastian Bach. Its usage is very common in learning models for soprano harmonization. Furthermore, these chorales have a homophonic texture with four voices and are excellent examples of functional harmony. They can be found in the *music21* library by MIT[1].

To provide sufficient information to the Bayesian network we need more information than just four melodies. To achieve this, we preprocess chorales with some simple operations to obtain full chordal homophony, where every voice has the same rhythm. Afterwards, the choral contains only chords with no additional notes between them. Then, the harmonic functions of the chords can be identified. Finally, we add remaining information, e.g., about special chord components. The final model of each chorale contains the global key, time signature and the sequence of chords. Each chord contains every voice's note, length of a chord, harmonic function, information about the location inside a bar where the chord is placed, the degree of melody note in the global key. Every note is described by base note (C, D, E, F, G, A, B), MIDI note number and chord component. For example $A\flat 4$ in F minor chord has a MIDI note number equal to 68, base note A and the chord component is the minor third.

After applying the preprocessing algorithm to Bach's chorales we receive the final training set, which contains 168 major and 165 minor chorales[2]. This set

[1] `https://web.mit.edu/music21`
[2] `https://github.com/miksik98/Bach-Chorales-Dataset`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

was then used to train the BN. The training was conducted using the maximum likelihood algorithm.

The target of BN is the prediction of a harmonic function based on a specific soprano note and the information about the place of a note inside the bar. The latter is inspired by some rules related to this property. Moreover, we add information about the previous harmonic function, note and place in the bar, as well as information about the next note and place. It is necessary to prevent our model from making invalid predictions such as D-S connection. Figure 1 shows

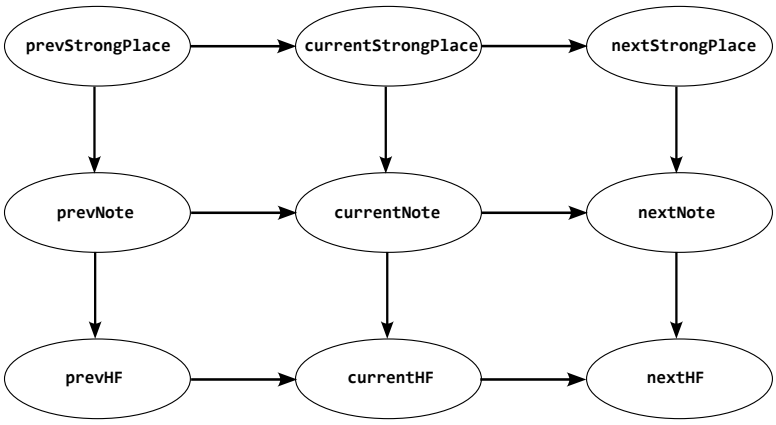


Fig. 1: General concept of the Bayesian network.

the general concept of a Bayesian network. Notes are processed one by one except the first: we assume that the first harmonic function is T. Then for each note we already know *prevStrongPlace* (flag indicating if note is placed on the on-beat), *prevNote* (degree of the input melody note in the global key), *prevHF* (previous harmonic function), *currentStrongPlace*, *currentNote*, *nextStrongPlace*, *nextNote*. We predict *currentHF* and *nextHF*.

Due to the fact that every node of a trained BN has an a posteriori distribution of probability over its values we propose two variants of prediction: *maximum a posteriori (MAP)* (we choose the value with highest probability) and *stochastic (S)* (we draw a random value using a posteriori probability distribution).

Figure 1 does not show the whole picture. Firstly, every harmonic function node is represented by additional nodes: *IsMajor* (flag indicating if chord is major, otherwise it is minor), *Key* (for chords from global key it is null, for others it indicates the key of the modulation), *Base* (base harmonic function - T, S or D), *Extra* (additional components), *Omit* (omitted components), *Degree*, *Inversion*, *IsDown* (indication if chord's degree is lowered), *Position* (chord component in the soprano voice). Moreover, we create two separate networks for problems with a major key and for problems with a minor key.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

Secondly, there are connections between nodes related to two consecutive functions. As the connections between *prev* (previous) and *current* are identical to those between *current* and *next* here we describe only the former.

i) *currentStrongPlace* - is an observation, no edges incoming.
ii) *currentNote* - is an observation, no edges incoming.
iii) *currentPosition* - is dependent on *currentNote*, *currentKey*, *currentDegree*, because knowing melody note and the root of the chord we can find what kind of chord component is in the soprano.
iv) *currentInversion* - is dependent on *currentStrongPlace* (prevents some rules violation), *currentInversion*, *prevPosition*, *prevInversion* (e.g., prevents parallel fifths), *currentNote*, *currentDegree*.
v) *currentIsDown* - is dependent on *prevDegree*, *currentNote*, *currentBase* (there are few functions which can be lowered and soprano note can imply it).
vi) *currentIsMajor* - is dependent on *currentIsDown* (if chord is lowered, then its mode is minor), *currentNote* (can be the third of chord), *currentDegree*, *currentBase* (there can be different modes, e.g., minor S).
vii) *currentKey* - is dependent on *prevKey* (can imply current), *currentBase*, *currentDegree*, *currentIsMajor*, *currentNote* (can imply modulation).
viii) *currentExtra* - is dependent on *currentPosition*, *currentNote*, *currentInversion* (can be one of extras), *currentDegree*, *currentKey*.
ix) *currentOmit* - is dependent on *currentExtra* (if extra is not empty there should be maybe omit component), *currentPosition*, *currentNote*, *currentInversion* (we cannot skip bass and soprano components), *currentDegree*.
x) *currentBase* - is dependent on *prevBase* (to avoid D-S connections), *prevKey* (if it is nonempty, *currentBase* could be D), *currentDegree*.
xi) *currentDegree* - is dependent on *prevKey*, *prevExtra*, *prevDegree* and *currentNote*.

All edges and dependencies on harmonic function properties are based on functional harmony rules.

We propose a sequential algorithm for predicting the properties of a sought harmonic function. After every step we update the probability distribution for every unobserved node. Firstly we start with *currentDegree (MAP)*, then *currentKey (S)*, *currentisDown (MAP)*, *currentIsMajor (S)*, *currentBase (MAP)*, *currentPosition (MAP)*, *currentInversion (S)*, *currentExtra (MAP)*, *currentOmit (MAP)*. With the above schema we are sure that every predicted harmonic function is valid because the training set contains only valid harmonic functions.

If the training set covers the domain of harmony, thus defined Bayesian network can generate interesting harmonic function sequences. To solve the whole soprano harmonization problem based on these sequences we use a rule-based system that produces chords complying with the functions generated by the BN.

3.3 Hybrid algorithm

The hybrid algorithm is a combination of genetic and Bayesian network algorithms. A part of the initial population of the genetic algorithm is created using

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

the network, and a second part is created randomly. Then we proceed as in the genetic algorithm. This approach mixes already correct solutions from the network with the random, which produces new higher-quality results faster. In our tests, we used an initial population with a probability of 0.5% of taking a solution from a Bayesian network algorithm.

4 Test results

For the implementation of algorithms we used the Scala language. For the genetic algorithm we used additional *jenetics* library[3] and for Bayesian network we used *SMILE* library[4]. All details can be found in the open-source code repository[5].

The main objective of the tests is to compare the performance of all three algorithms, i.e., genetic algorithm (GA), algorithm using Bayesian network (BN) and hybrid algorithm (HA), with the rule-based system (RS). The tests were carried out on a machine with Intel Xeon CPU E5-2680 v4 (2.40GHz) processor with 28 cores and 64 GB RAM. We selected 10 input melodies from the Polish harmony book by Targosz [12]. Four of them have major keys, another four have minor keys and the remaining two contain alterations. The average length of tasks is 9 bars. Moreover, these melodies are varied in terms of keys (keys with flats, keys with sharps) and time signature ($\frac{2}{4}$, $\frac{3}{4}$, $\frac{4}{4}$, $\frac{6}{8}$). Every solution obtained with a specific algorithm was judged in two categories:

1. evaluation of broken rules using scalarized fitness from GA, i.e., $V_c + V_f$,
2. evaluation of musical quality by domain experts using scale $\{1, 2, 3, 4, 5, 6\}$, where 6 is the highest mark.

Every algorithm was run on each melody, so the domain experts (professional musicians, harmony teachers) received 10 input melodies and 40 harmonizations as MP3 files[6]. For nondeterministic algorithms we chose one of the results with the lowest value of rule metric, i.e., total penalty (scalarized fitness).

We started tests with the parametrization of the genetic algorithm. After a set of tests we found final parameters: 2000 epochs, 2000 individuals in population, probability of crossover - 0.2, mutation probability - proposed in the previous chapter, survivors percentage - 30%. In the hybrid algorithm we altered the above parameters with a population size of 1000 and 1500 epochs. Every task was solved 30 times. For the genetic algorithm we found confirmation of the mutators disabling strategy, because the final solution was always found in the second half of epochs. As shown in the Table 1 the best performance by the rule metric received the rule-based system. It confirmed the RS ability to generate formally valid harmonizations as expected. The second algorithm in most tasks was the hybrid. In the Figure 2 we can see that HA was very stable for

[3] `https://jenetics.io`
[4] `https://www.bayesfusion.com/smile`
[5] `https://github.com/HarmonySolverTeam/HarmonySolverBackend/tree/mgr`
[6] Available at `https://youtube.com/playlist?list=PLysSbLi9j6PjOqNIzTdekEowPFix2bOlU`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

Table 1: Rule metric results.

Group	Melody	GA		HA		RS	BN
		Mean	IQR	Mean	IQR		
Major	1	2703.5	1001.5	1994.5	2064.5	390	3970
	2	3650.5	1399.5	3095.5	1869.5	479	3369
	3	1676	1018	700.5	952.5	479	2834
	4	858.5	1044	930.5	963	370	1956
Minor	5	851.5	768	804	31.25	379	811
	6	522	68.75	619	201.5	379	4846
	7	5465.6	1227.75	3353.5	947.5	644	5505
	8	760	262.75	756	1041.75	472	1820
With alterations	9	7605	2022.5	4416	48	914	5624
	10	7350	1195.25	6925	1962.75	1013	12544

the minor key group (tasks 5-8). The poorest results were received for the group with alterations: sometimes more than 8 broken hard rules. What is surprising, BN's performance received the inter-quartile range (IQR) close to 0. RS is deterministic and we always get the same harmonization for a given melody. For these reasons, in Table 1 IQR for RS and BN is not reported and the only value shown is the mean result for the rule metric.

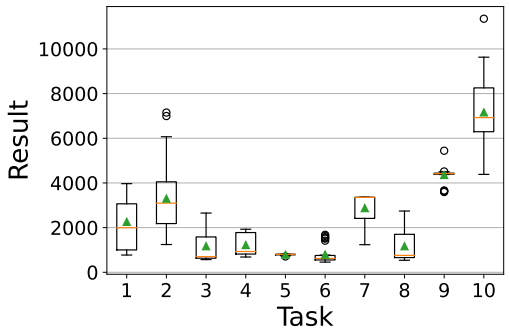


Fig. 2: Rule metric results for HA.

The average expert grades are aggregated in the Table 2. The extreme ratings number per algorithm is shown in the Figure 3. The highest number of 6 grades received a hybrid algorithm and it was twice better than the rule-based system, which seemed to be the worst algorithm in the group with alterations. The genetic algorithm and rule-based system received the highest number of 1 and 2 grades. RS generated simple harmonic structures in harmonizations, which was not appreciated by the experts. In contrast, GA generated too bizarre harmonic structures. BN and HA were more consistent in this aspect. What is more, BN

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

received the second lowest number of 1 and 2 grades, because experts appreciated Bach's harmonization style, which was used to train the network and could be observed in resulting harmonizations.

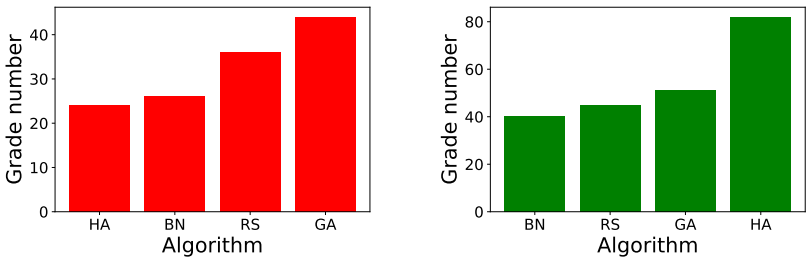


Fig. 3: Number of lowest (left) and highest (right) experts' grade.

Example solutions for the tenth task are shown in Figure 4. In the first bar in the GA and HA solutions there are consecutive secondary dominants, which build tension at the beginning of the final harmonization. RS is very typical and not advanced. BN uses some chord inversions as well as interesting static connections, like in the second bar. The solutions generated by genetic algorithms are more colorful than those of other algorithms. It was sometimes viewed negatively by experts.

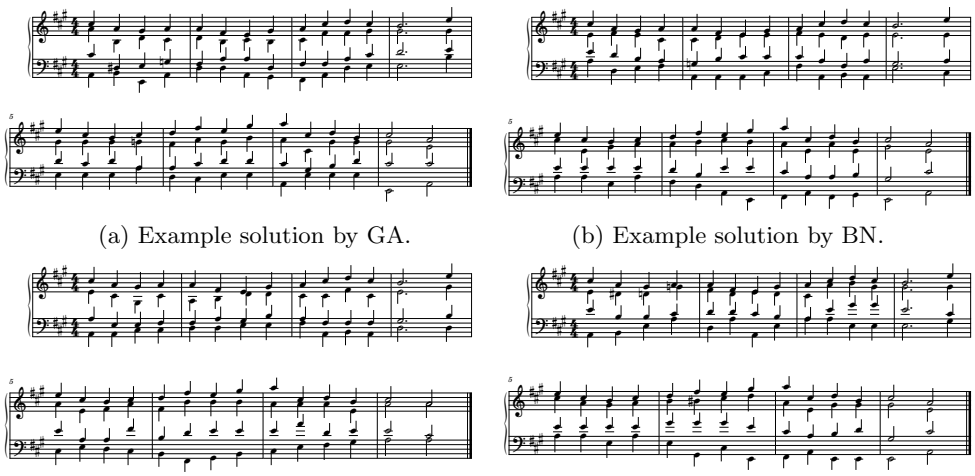
Table 2: Comparison of average expert metric results of algorithms.

Group	Melody	GA	BN	RS	HA
Major	1	4.40	4.05	4.13	4.25
	2	3.73	3.40	4.60	4.83
	3	4.28	4.75	4.43	4.85
	4	4.53	4.18	4.60	4.33
Minor	5	4.68	4.43	4.10	4.90
	6	3.88	4.68	4.13	4.50
	7	3.454	3.80	4.95	4.35
	8	4.70	4.05	4.43	4.20
With alterations	9	4.35	4.35	3.43	4.83
	10	3.83	4.25	3.80	4.10

5 Conclusions

The results of experiments shown in the previous section and comments of domain experts confirm that a rule-based system produces harmonizations which

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16



(a) Example solution by GA.

(b) Example solution by BN.

(c) Example solution by RS.

(d) Example solution by HA.

Fig. 4: Example solutions for 2nd test melody.

are the best in terms of rules of functional harmony, but it is achieved by using uncomplicated harmonic structures. These results were rated as safe and boring by experts. The Bayesian network algorithm reached low ranks in both metrics, but this is probably because the test melodies were not similar to chorales that the network was trained on. Maybe if the melodies had been taken from Bach's chorales the algorithm would perform better. The domain experts said there were many interesting harmonic connections, but they did not have any sensible consequences. They also noticed many typical Baroque harmonic functions (e.g., minor S in major key), so we can assume training was successful. The genetic algorithm generated the most interesting harmonic connections related with melody phrases, especially creating tensions and releasing them. On the other hand there were many random parts. The bass line was illogical and there were many jumps inside its melody. The most common mistakes were false relation and repetition of the same function through the bar line. The hybrid algorithm generated the best solutions. It received twice as many *6 grades* as the rule-based system, and the highest average grade. It contains elements of both algorithms it combines, so we can see features of Bach's style which were enriched by the genetic algorithm.

To sum up, the results confirm that the use of evolutionary algorithms and machine learning can produce more interesting harmonizations than rule-based systems. They can also satisfy most of the functional harmony rules. A straightforward future research should probably focus on the hybrid algorithm, which uses domain knowledge for learning and mutation and which was highly rated by the experts. Another promising option is the use of Bayesian networks trained on a more general musical data and with a more complex structure, e.g., considering sequences of chords with length greater than 3. Finally, there are some other promising methods for soprano harmonization to explore like LSTM and

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16

BiLSTM neural networks. A more comprehensive idea is the composition of algorithmic harmonizers with techniques based on the counterpoint theory that would result in more complex polyphonic musical compositions.

References

1. Allan, M., Williams, C.: Harmonising chorales by probabilistic inference. In: Advances in Neural Information Processing Systems. vol. 17. MIT Press (2004)
2. Dajda, J., et al.: Current trends in software engineering bachelor theses. Computing and Informatics **40**(4), 930—956 (2021)
3. Deb, K., et al.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)
4. Ebcioğlu, K.: An expert system for harmonizing four-part chorales. Computer Music Journal **12**(3), 43–51 (1988)
5. Hild, H., Feulner, J., Menzel, W.: Harmonet: A neural net for harmonizing chorales in the style of J. S. Bach. In: Advances in Neural Information Processing Systems. vol. 4. Morgan-Kaufmann (1991)
6. Lim, H., Rhyu, S., Lee, K.: Chord generation from symbolic melody using BLSTM networks. In: Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017, pp. 621–627. ISMIR (2017)
7. McIntyre, R.A.: Bach in a box: the evolution of four part Baroque harmony using the genetic algorithm. In: Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, vol. 2, pp. 852–857. IEEE (1994)
8. Mycka, J., Żychowski, A., Mańdziuk, J.: Human-level melodic line harmonization. In: Computational Science – ICCS 2022, pp. 17–30. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-08751-6_2
9. Phon-Amnuaisuk, S., Wiggins, G.A.: The four-part harmonisation problem: A comparison between genetic algorithms and a rule-based system. In: Proceedings of the AISB'99 Symposium on Musical Creativity. pp. 28–34. AISB London (1999)
10. Raczynski, S., Fukayama, S., Vincent, E.: Melody harmonisation with interpolated probabilistic models. Journal of New Music Research **42**, 223–235 (2012)
11. Rameau, J.P.: Treatise on Harmony (1722). Dover Publications, Inc. (1971)
12. Targosz, J.: Podstawy harmonii funkcyjnej. PWM (2004), in Polish
13. Toutant, W.: Functional Harmony. Wadsworth Publishing Company (1985)
14. de Vega, F.F.: Revisiting the 4-part harmonization problem with GAs: A critical review and proposals for improving. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 1271–1278. IEEE (2017)
15. Whorley, R.P., et al.: Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonisation problem. Journal of New Music Research **42**, 237–266 (2013)
16. Yeh, Y.C., et al.: Automatic melody harmonization with triad chords: A comparative study. Journal of New Music Research **50**(1), 37–51 (2021)
17. Yi, L., Goldsmith, J.: Automatic generation of four-part harmony. In: Proceedings of the Fifth UAI Bayesian Modeling Applications Workshop, CEUR Workshop Proceedings, vol. 268, pp. 81–86. CEUR-WS.org (7 2007)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_16