Wiley
Complexity
Volume 2025, Article ID 3086287, 16 pages
https://doi.org/10.1155/cplx/3086287

WILEY

Research Article

Advancing 4-Part Evolutionary Harmony Through Analysis of Human–Machine Approaches to Teaching–Learning

Elia Pacioni [1,2] **and Francisco Fernández De Vega** [1]

[1] *Department of Computer and Communications Technology, Mérida University Center, University of Extremadura, Mérida, Spain*
[2] *Institute of Informatics, University of Applied Sciences and Arts of Western Switzerland (HES-SO Valais-Wallis), Sierre, Switzerland*

Correspondence should be addressed to Elia Pacioni; eliapacioni@unex.es and Francisco Fernández De Vega; fcofdez@unex.es

Received 17 January 2025; Revised 22 October 2025; Accepted 28 October 2025

Guest Editor: Penousal Machado

This work explores the challenges related to the 4-part harmony problem, addressing both the computational complexity of the search space and the benefits of integrating human teaching/learning processes into evolutionary problem-solving approaches. From a computational perspective, we analyze strategies to enhance algorithm efficiency, including parallelization, precomputation of fitness values, directed mutation, and adaptive directed mutation, which collectively reduce the time required to find solutions. Synthetic harmonic models are employed to validate these techniques. Complementing this, we investigate the role of human expertise, emphasizing the synergy between expert teaching and the learning processes of novice students. By examining how human teaching and learning paradigms can inspire innovative problem-solving techniques, we draw on the concept of evolutionary machine teaching, which reduces the search space, applied here to a standard harmonic model. Our findings highlight the potential of integrating computational advancements with methodologies driven by human learning. Specifically, the search space produced by Sharpmony students accounts for less than 1% of the total space. Using this approach, we have achieved a fourfold speedup over previous results of the same quality. Moreover, longer runs of the new approach have provided solutions with an average fitness of less than 1 error, considering the complete set of 50 rules and exceptions.

Keywords: 4-part harmonization; adaptive directed mutation; anticipated fitness evaluation; directed mutation; evolutionary machine teaching; human teaching; search space reduction

1. Introduction

SATB (soprano, alto, tenor, bass) harmonization is a widely practiced technique in choral music. It peaked during the Baroque era, particularly with J.S. Bach, who greatly enhanced this harmonization style in his compositions—this period marked the foundation of key principles for choral composition. These principles include chord progressions, the restriction of certain dissonant combinations, melodic movements, and the sound balance among different voice parts. Lopez-Rincon et al. [1] outline a classification of AI methods used in algorithmic music composition, covering heuristic methods such as evolutionary algorithms (EAs) and dynamic programming; deep learning techniques like convolutional and recurrent neural networks; stochastic approaches; and symbolic AI methods including agents, declarative programming, and grammatical representation. Additionally, Liu and Ting's research evaluates AI techniques in music composition and highlights the most effective methods for specific tasks [2]. Notably, neural networks demonstrate superior performance in imitative systems, while Markov models excel in predicting musical notes based on prior inputs. Genetic algorithms (GAs) are particularly adept at generating chord progressions.

Many studies have been carried out throughout the years, yet the issue remains unresolved since satisfactory outcomes with accurate scores are still lacking. Moreover, running times for real 4-part harmony exercises may take days to be approximately solved. Simulations that fail to mirror reality have thus typically relied on a limited and simplified range of problems. As a result, this research area continues to present a considerable challenge today.

In this paper, we focus on the harmonization of sheet music based on a provided melody. The primary task is improving the quality of the solution while reducing computing time. We consider new perspectives on incorporating expert information into evolutionary approaches to 4-part harmonization. Including domain experts in research teams has positive effects but also some drawbacks: for instance, the many constraints and rules that experts have defined to determine the quality of solutions may lead to excessively reducing the number of high-quality solutions to be considered within huge search spaces. On the other hand, algorithms that do not involve experts risk oversimplifying the problem and leading to the identification of a large number of qualitatively poor solutions. Therefore, finding a balance is critical and poses a significant challenge.

This paper extends our previous work [3], revisiting and extending its three pillars—anticipation of fitness evaluation, directed mutation, and synthetic harmonic models. A Bayesian ranking analysis (Plackett–Luce) is employed to validate the results and understand which approach provides better performance. At the same time, moving from synthetic setups to the whole real search space exposes a practical ceiling: precomputation becomes prohibitive, and any rule update entails large recomputation. To address this, we reduce the search space using harmonizations produced by Sharpmony students. Within this reduced space, we assess directed and adaptive directed mutation; short-run experiments, together with the Bayesian analysis, progressively narrow the choices deployed in the long runs.

Our study progresses in stages—each step prepares the next: we first validate feasibility on synthetic models; then, the following experiment calibrates the population size; so, the mutation strategies (directed, random, hybrid) is defined. Then, we introduce a machine–teaching inspired reduction of the solution space using Sharpmony data; we compare directed and adaptive directed mutation within this reduced space; finally, we run long runs with the configurations selected based on the short runs and the Bayesian analysis.

Summarizing, this paper extends previous results obtained by means of anticipation of fitness evaluation over synthetic harmonic models as follows:

- A machine teaching–inspired approach of local search is designed to improve directed mutation in order to reduce the search space when trying to apply the evolutionary approach to 4-part harmony exercises.
- The new solution space is based on data produced by students of the Sharpmony[1] application, which provides a concrete example of how the human teaching and learning process can be leveraged to refine the problem representation [4].
- Two mutation operators are compared on the newly produced search space: directed mutation and adaptive directed mutation.
- Short and long runs are shown to understand the potential of the approach.

The experiments and results are presented in stages, following the time sequence used to arrive at each point.

The paper is structured as follows: Section 2 presents the literature review and the techniques employed to address the problem, Section 3 describes the incremental methodology used to conduct and validate the research, and Section 4 presents the experiments performed and the results obtained. Finally, Section 5 shows the conclusions.

2. 4-Part Harmonization and EAs

2.1. SATB Harmonization Problem. Art and computer science are increasingly connected, and over the years, they have given rise to numerous projects and areas of interest ranging from image creation or editing to text writing to the field of music. In particular, since the 1990s, various approaches to computer-assisted music composition have been presented.

This paper focuses on 4-part harmonization, an introductory composition exercise designed for music students. Typically, the teacher provides a melody assigned to the soprano, and the student must complete the other three voices: alto, tenor, and bass. This exercise requires following specific rules developed over the centuries to encourage the creation of beautiful choral harmonies. For those who wish to learn more, the Sharpmony Project offers a list of more than 50 commonly adopted rules, along with their exceptions.

Several evolutionary approaches have been developed in recent decades to address this problem. However, many of these solutions tend to simplify the real complexity of the issue, as is the case with the method proposed by Horner and Goldberg in 1991 [5]. In 1994, McIntyre [6] introduced a technique that used a predefined melody and key as the basis for harmonization. However, even just harmonizing nine notes required thousands of individuals across hundreds of generations.

In 2014, Kaliakatsos et al. [7] experimented with a method to create chord progressions by assigning scale degrees to each note and then selecting chords and notes for each voice based on those degrees. During the process, they highlighted the difficulties associated with the many rules required to ensure the quality of the generated scores.

In 2017, Fernández proposed a new approach based on GAs, making generation of a complete score possible [8]. This method used only 11 rules during the evolutionary process. The starting melody consisted of 29 notes divided into 8 bars. The harmonization result contained 10 chords with errors and required as many as 24 h to complete.

More recently, EvoComposer [9] was developed, an EA designed to automate 4-part harmonization. The method was based on a multiobjective approach to optimize the choice of chords for harmonization and the melodic quality of individual vocal lines. To meet melodic and harmonic rules, the system used a hybrid evaluation function, which combined theoretical principles from classical music with weights derived from a statistical analysis of Bach's chorales, ensuring a balance between stylistic consistency and creativity. However, this method had some limitations: it lacked sufficient granularity in detecting errors in the final score. In addition, the statistical approach based on Bach's chorales did not consider the changes in musical rules from the Baroque period to the present.

This work builds on earlier versions of the GA described in [3, 4], in which the number of rules and exceptions was greatly expanded, exceeding 50 rules and exceptions applied. This increase led to a drastic reduction in the available solutions.

Before describing the approach, we show below a review of the main ideas related to the applied methodology and the main tool used in this research.

2.2. Sharpmony. Sharpmony is the starting point of our research on 4-part harmonization; it is the first AI-based tool specifically designed to assist students and teachers of harmony [10, 11]. The ecosystem consists of (i) the core, developed in LISP, including algorithms to detect tonality and modulations within scores, the EA, together with the fitness function that includes the rules, and modules to process and produce MusicXML files; (ii) web portal; and (iii) Android mobile application. On the one hand, students can choose to use the web portal or the mobile application to create their exercises (Figure 1), submit them for correction, and examine the results produced by the AI. On the other hand, teachers can manage their conservatory students, assign them exercises, and monitor progress through corrections made by the system.

To support the students' study process, Sharpmony classifies errors by color, applying the colors directly to the affected notes. In this way, the student can read the type of error made and, through Sharpmony's harmony rules manual[2], can access a brief explanation of the error.

2.3. Anticipating Fitness Evaluation. In GAs, the fitness evaluation step is typically the most computationally expensive component, as it requires repeatedly assessing candidate solutions. To address this issue, researchers have proposed various approaches aimed at anticipating or approximating fitness evaluations to save computational resources without significantly sacrificing the quality of solutions.

These approximation techniques can be categorized into two main types: problem approximation and function approximation [12]. Problem approximation methods involve transforming the original problem statement into a simpler, easier-to-solve variant. Conversely, function approximation methods seek to replace the exact fitness function with a computationally cheaper surrogate, which can estimate fitness values with acceptable accuracy. Shi presents various statistical approximation methodologies including polynomial models [13], Kriging models, and support vector machines [14]. Inheritance-based approaches have also been explored, determining a child's fitness from a weighted combination of parental fitness values, thus reducing the number of direct evaluation [14].

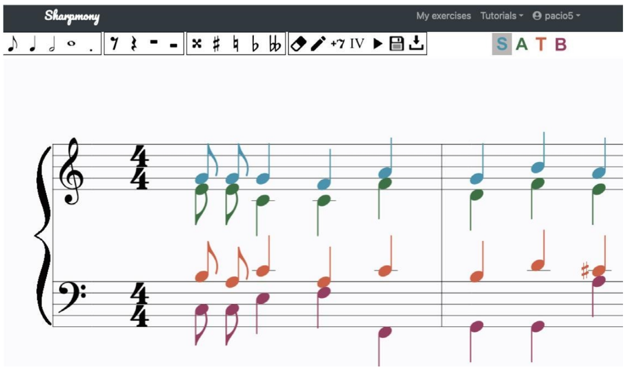



Figure 1: Sharpmony 4-part harmony editor.

Additionally, surrogate models such as radial basis function networks have shown potential. They reduce the number of evaluations required by approximately 70%. Gaussian-based Surrogate [15] models and neural networks models [16] are other notable examples in this category.

Although our research shares the common goal of reducing computing effort associated with fitness evaluation, our approach aims to retain exact and precise fitness values. Instead of approximation, we employ precomputed partial fitness values, inspired by educational methodologies involving interaction between human teachers and students during exercise solving activities. This strategy allows instant retrieval of fitness evaluations, ensuring computational efficiency without compromising accuracy. The details of our approach are elaborated in subsequent sections.

2.4. Directed Genetic Operators. In biology, changes in individuals are not always random; sometimes, they are induced by the evolutionary process to fill weaknesses or solve problems [17]. In this case, we can speak of direct or induced mutation. In 1994, Bhandari et al. proposed that this approach effectively reduced the solution space and accelerated algorithm convergence [18]. This paper explores two induced operators: direct mutation and adaptive directed mutation.

A recent study highlighted the effectiveness of the adaptive directed mutation operator, showing that it significantly outperforms five traditional mutation operators. This approach improved the convergence, accuracy, and reliability of the algorithms examined [19].

Direct and adaptive mutation are distinguished by their ability to adapt to changes in the problems faced, allowing for better fitness values even in complex contexts. However, overusing directed mutation can cause convergence to suboptimal solutions. For this reason, it is crucial to carefully configure the GA parameters.

A novel method involves applying directed mutation by establishing the mutation probability for each gene in the chromosome and evaluating its effect on the individual's improvement. Then, this probability is modified according to the results obtained from the applied mutation [20].

In 2023, Carvalho et al. introduced a similar approach in genetic programming (GP), specifically evolutionary grammar [21].

In our case, we implement both directed mutation and adaptive directed mutation and compare the two mutation operators.

2.5. Analyzing Solution Space. It is crucial to consider the dimension of the search space of the 4-part harmonization problem. Starting with the scale degrees (I, II, III… VII) and including different types of chords, such as triads (e.g., C-E-G for degree I in C major) and quatriads (e.g., C- E-G- B), as well as the inversions available for a single chord (such as C-E-G-B, E-C-G-B, … and all possible permutations), and note positions within score (for instance, octave 4 or 5 for soprano, 3 or 4 for alto…), we come to calculate that there are 2855 valid combinations for a given key in the current implementation of Sharpmony [3], including all possible kind of chords, such as secondary dominants, sixth chords, and so on. This also includes variations such as repetitions of notes to complete the 4 voices required in SATB.

Moreover, if we consider a pair of consecutive chords in a single key and given that we have 2855 possibilities for assigning notes to the voices, the search space increases exponentially, reaching over 8 million combinations. With three consecutive chords, the possibilities exceed 23 billion. Since melodies often contain many notes, it is evident that the number of combinations grows rapidly, affecting subsequent choices. We need thus to explore this search space to find scores with as few errors as possible.

This analysis, however, is only an initial estimate and does not consider modulation, that is, the change of tonality within the same score. When modulations are introduced, the search space widens considerably: with 24 possible tonalities (12 major and 12 minor), the number of available consecutive chord pairs exceeds 4 billion, while for trios, it comes to about $(2855 \times 24)^3 = 3.22 \times 10^{13}$. These numbers demonstrate the complexity of the problem and the enormous size of the solution space, which makes it challenging to find good solutions.

2.5.1. Reducing the Search Space. In EAs, the size of the solution space is crucial. A very large space is hard to explore and incurs high costs to identify optimal solutions, while excessive restrictions can lead to poor quality results. Numerous approaches have been developed to address this problem.

For example, Das and Pratihar [22] introduced a method that balances population diversity and selection pressure in GAs by using local searches in specific subspaces. Another progressive approach, proposed by Orito and Hanada [23], involves progressively fixing some variables in the problem, shrinking the search space until a more manageable subset containing optimal solutions is obtained. Other studies have focused on reducing the space during the allocation phase, as in the case of algorithms for VLSI cell placement [24] or in the adaptive control of micro-aircraft [25], exploiting symmetries and structural redundancies to limit the combinations to be explored. These approaches fall mainly into two categories: space reduction in the allocation phase or progressive elimination of variables. Such techniques find application in various fields, such as finance [23] and agent-based systems [26].

One aspect that has yet to be explored concerns how learners cope with and reduce the search space during the learning process. This ties in with the way teachers structure exercises, gradually introducing more complex problems to guide students. In the following sections, we explore how these principles can be applied to EAs and how to leverage these strategies to improve the effectiveness of solution search.

2.6. Machine Teaching. MT represents an innovative approach to artificial intelligence, distinguishing itself from traditional techniques by its emphasis on the role of the teacher over that of the student. This methodology focuses on how the teacher transmits knowledge to the system [27]. In contrast to machine learning (ML), which relies on processing large amounts of data to train a model, MT reverses conventional roles: the teacher's expertise in a specific domain becomes the focus of the process, guiding the artificial intelligence algorithm through optimal dataset selection [28].

MT favors a strategic design of the data provided to the system, with the goal of guiding and optimizing the algorithm's behavior, reducing execution and training time without sacrificing the quality of the solutions produced. It also emphasizes the teacher's responsibility in transferring knowledge through the use of practical examples [29].

Ramos describes interactive machine teaching (IMT), which centers on the human role through an iterative method that generates semantically verifiable models [30]. Ng and colleagues elaborate on knowledge decomposition in the context of MT, highlighting how teachers can effectively decompose and transfer their knowledge to students in human-centered computing contexts [31].

The relevance of MT goes beyond evolutionary ML techniques, such as GP. It can also significantly contribute to other evolutionary approaches, integrating learning and teaching processes between humans and machines. Recently, a correlation between ML and combinatorial optimization problems has also been highlighted, suggesting the possibility of translating such problems from one domain to another [32].

In this context, the present work is inspired by the MT model, which aims to collect teaching and learning experiences to narrow the research space. This has been possible by the introduction of Sharpmony as an AI-assisted 4-part harmony and counterpoint education tool in music conservatories and universities, which ultimately led to collecting and analyzing more than 19,000 exercises.

3. Methodology

3.1. The Problem. To fully understand the problem, it is essential to clarify how the GA is used to generate SATB musical scores works from a given melody. The process is organized into two independent and sequential phases, each governed by a dedicated GA. The hardware used is presented at the beginning of the results section and is the same for all experiments presented in this article and previous ones.

3.1.1. Phase 1—Chord Progression Generation. The first GA is responsible for generating a harmonically correct sequence of chords based on a given melody, which, in this research, is 8 bars long (as shown in Figure 2), and the rest of the voices are initially silent. The melody provided represents the default voice (can be Soprano, alto, tenor, and bass). The given melody may belong to a single key, such as C major, or involve modulations between different keys, such as from C major to A minor or G major. However, in the context described, modulations are excluded, limiting the research to the key of C major. However, if no possible arrangements are found, modulation is allowed as an exceptional case.

In this phase, each melodic note is mapped. We first consider available chords including that note, and then analyze their corresponding scale degree, such as II, V, or VII for a D note in the key of C major. For convenience, we assign the degree to the note since the chord consists of that note, and the others are rests. Therefore, assigning it to the chord means verifying that note can exist in the chosen degree. An individual in this GA is thus a sequence of chord degrees, as shown at the top in Figure 3.

When we decide not to evolve the chord progressions, we exclude the initial GA. In this scenario, we have two options: (i) we can utilize a random population, evaluate it, and select the best progression; or (ii) the user can provide a specific progression to use. In both instances, the evolutionary process of the first GA is not applied. In experiments, it is referred to as Evolving Chord Progression = NO.

3.1.2. Phase 2-Score Harmonization. Once the chord progression is established, the second GA takes over to generate the full SATB harmonization. Here, each individual represents a complete four-voice score. This GA is responsible for distributing the chord notes among the other voices (e.g., if degree II is applied and the melody contains a D, F, and A must be present in the other voices, in the case of a triad chord). In this GA, an individual is represented by a sequence of chords. Figure 3 shows, in the bottom part, the chromosome structure of the second GA. Each column represents a chord, and each row represents an entry. For example, the first chord is E, C, G, C and correctly belongs to the associated degree I. Only the chords conforming to harmony rules are considered, including triads and diatonic seventh chords, augmented sixths (Italian, French, German, and Neapolitan), and secondary dominants.

When chord notes are distributed among voices according to evolved degrees, the complete score is verified through a fitness function that applies harmony rules. This function evaluates about 50 types of errors, including: Parallel 5th, Parallel 8th, Overlapping voices, …, (up to 50 rules, as described above).

However, the fitness function of the second phase does not include rules related to degree progression, such as cadences or phrase conclusions. These aspects are handled by the first phase, which determines an appropriate chord sequence for use in the next phase. Adding new types of chords increases the search space and the number of rules to be verified, making the process more complex. In addition, the more rules that need to be verified, the longer it takes to analyze a pair of chords: each pair takes 22.5 s to evaluate. As a result, analysis of a complete chromosome can take about 10 min.

3.2. Parallelization, Precalculation of Partial Fitness Values, and Synthetic Model Creation. Parallel computation is useful when computational loads are high; otherwise, the system risks worsening because of the cost of parallelism. In this case, the total time required to evaluate an individual is about 10 min, which translates into hours of computation in the whole evolutionary process. Therefore, parallel computation can be of paramount importance if adequately implemented.

The proposed approach concerns the precomputation of partial fitness values, handling the computation of errors in parallel, and consequently, the execution of the fitness function. This choice is based precisely on the cost of the fitness function and its impact on our problem.

Given that the fitness value of a given score is computed by adding the number of errors found in every consecutive pair of chords, the idea is to precompute errors in every possible pair of chords we may use in a given tonality and store those values in a database. Afterward, we could simply check and add the number of errors corresponding to a pair of chords in a given candidate solution.

We use a relational database, specifically MariaDB, to store all the data. The estimated size of the entire database including the optimization process is almost 2 GB, a size that is perfectly manageable with MariaDB.

The process presented is valid if we precompute all possible pairs of chords available. Unfortunately, a sequential approach would take about 5.8 years in our computing facilities, considering 22.5 s per chord pair and about 8 million pairs available in a single key. Although the computation time may be reduced due to the advantages of parallelism, it is not enough to make the computation usable in the experiments we tried to perform.

Yet, as a first step to validate the methodology, we decided to test our approach using synthetic harmonic models. By synthetic harmonic model, we refer to an artificially constructed model where no real-world harmonic rules are applied. Instead, we generate a simplified database by arbitrarily defining which chord pairs are considered acceptable or not, without relying on any traditional music theory. In practice, this means that for each possible pair of chords, we directly assign an error value—such as labeling

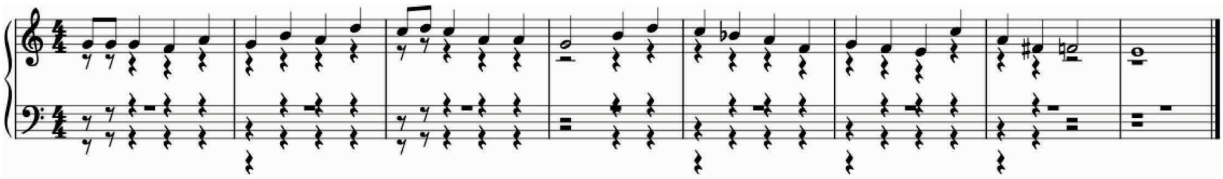
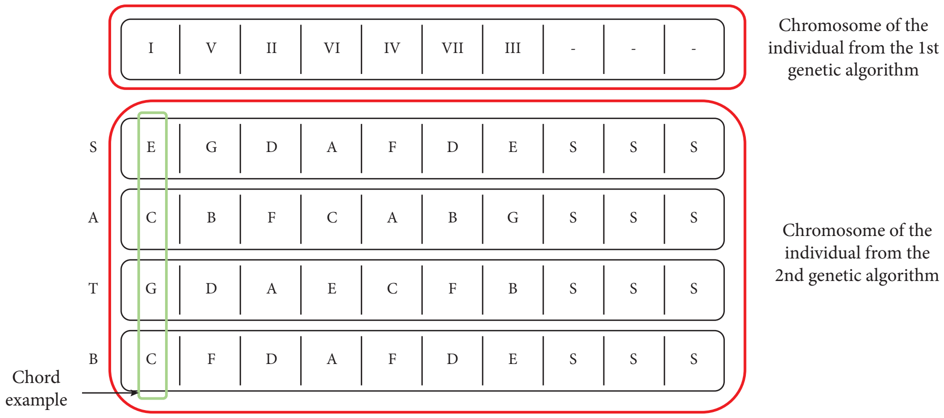



Figure 2: Starting melody provided to the EA.



	I	V	II	VI	IV	VII	III	-	-	-
S	E	G	D	A	F	D	E	S	S	S
A	C	B	F	C	A	B	G	S	S	S
T	G	D	A	E	C	F	B	S	S	S
B	C	F	D	A	F	D	E	S	S	S

Figure 3: The figure shows the chromosomes of the two GAs. The top one is an individual from the first GA, considering the key of C major. Below is the chromosome of the individual from the second GA.

a certain percentage of chord pairs as error-free and distributing errors across the rest according to a desired pattern.

This approach offers two major advantages. First, it drastically reduces computation time: unlike real harmonic models, no rule-based analysis is required to determine errors between chord pairs. The database can be generated immediately, simply by deciding which transitions should be penalized. Second, it allows full control over the difficulty and structure of the fitness landscape, since we can create harmonic models of arbitrary complexity or simplicity.

Using synthetic models, the EA is tested in a controlled environment, where the behavior of the search process can be evaluated. While these models are purely artificial and do not represent any real musical style, they allow us to assess whether the algorithm can converge toward optimal solutions when provided with a well-defined yet simplified fitness function.

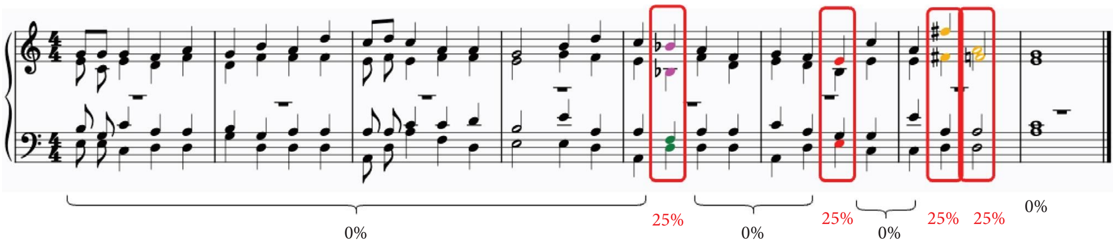
Moreover, although not explored in this work, synthetic models could in principle be designed to generate entirely novel harmonic systems, leading to innovative musical scores that explore unconventional sonorities or compositional techniques.

Therefore, in this research, we first employ a synthetic harmonic model to verify the feasibility of our methodology and assess whether the EA can successfully converge in this artificial context. In the second stage, we apply additional techniques to build the database based on the harmonic model traditionally taught in music conservatories.

3.3. Directed Mutation. Once the database with chord pair data, whether synthetic or real, is available, we can use this data in our fitness function to evaluate EA individuals. This information also enables the implementation of directed mutation, a strategy where mutation is applied selectively to problematic areas of each candidate solution, rather than randomly across the entire genome.

In detail, the mutation operator utilized the feedback provided by the fitness function, which not only returns the total number of errors but also identifies the exact positions of chords that generate those errors. During mutation, instead of selecting a random position, only the chord-involved errors are eligible for mutation (errors are expressed between pair of chords). Each of these positions has equal mutation probability $100/N_{\text{list items}}$, while error-free chords have a mutation probability of 0%, as shown in Figure 4.

Once a problematic chord position is selected, the actual mutation remains stochastic: the chord at that position is replaced by a new chord, chosen randomly from the database of possible chord configurations. This selection process is conditioned only by the available chord options stored in the database, typically considering the previous chord for consistency with chord-pair sequences, but without guaranteeing harmonic correctness or error-free outcomes. Therefore, while the selection of mutation targets is guided (directed), the choice of the new chord is random within the space of harmonically valid successors, maintaining exploratory capacity.



Probability of being selected in the mutation process

Figure 4: Selecting mutation points in directed mutation.

This approach increases the likelihood that each mutation will contribute to reducing the overall error count, focusing evolutionary effort on the problematic regions of each individual.

However, this strategy also carries a potential risk of compromising genetic diversity. Since mutation is systematically concentrated on the same error-generating chords, the algorithm may repeatedly generate similar or even identical individuals, especially when the number of alternative chords available for certain positions is limited. Moreover, although the replacement chord is always selected from the database of valid chord configurations, this does not guarantee that the new chord will resolve the existing error: it could still result in a harmonically incorrect solution or even introduce new errors in subsequent chord transitions.

It is essential to note that directed mutation does not dictate how each gene is modified, but rather where the mutation is applied. This strategy can be combined with any mutation technique: whether the mutation affects entire chords, single notes, or applies specific musical criteria, the key aspect of directed mutation is that mutations are preferentially applied to positions currently responsible for errors, as identified by the fitness function.

3.4. *Taking Inspiration From MT.*

Once the directed mutation technique has been validated on synthetic harmonic models (see results below), it is important to apply this approach to the standard harmonic model as well, and we have been inspired by MT to do this.

In this paper, we explore the possibility of reducing the solution space by using the knowledge acquired by students during the learning process, which is directly contained in the exercises performed on the Sharpmony platform.

The idea is to consider the subset students use as the EA search space. The hypothesis is that the teaching and learning process leads students to focus on a small area of the search space while still obtaining quality solutions. If this hypothesis were validated, we could focus exclusively on this reduced subset, allowing us to improve the computational efficiency and quality of the solutions obtained.

Then, the exercises produced by Sharpmony students from which the chord pairs are extracted and analyzed, and the errors are precomputed and saved using the encoded rules, thus obtaining the database of the reduced real harmonic model.

We thus analyzed 13,000 students' exercises from the database, and 130,000 chord pairs of the standard harmonic model were checked, which corresponds to 1.6% of the total. This process is helpful in defining the distribution of errors, and we observed that error-free chords account for approximately 8% of the total checked and stored. Therefore, we decided to evenly distribute the chord pairs according to the number of errors, from 0 to 9, with a rate of 10%. However, since this is a preliminary evaluation aimed at validating the methodology rather than analyzing the actual error distribution and considering that the whole database is still under processing, with the percentage of error-free chords increasing, we opted for a simplified and balanced approach. Specifically, we decided to evenly distribute the chord pairs across error classes, ranging from 0 to 9 errors, with 10% assigned to each class. This controlled sampling allows for a uniform and unbiased evaluation.

3.5. *Directed Genetic Operators: Directed Mutation vs. Adaptive Directed Mutation.*

Finally, the database produced from Sharpmony's student data is used to test the directed mutation and compare it with the directed adaptive mutation.

The directed mutation is presented in Section 3.3. To develop the adaptive directed mutation, we start with the directed mutation itself. The major difference is that in directed mutation, the chords with errors all have the same probability of being chosen for mutation. In contrast, in adaptive directed mutation, the probability that a chord with errors is chosen for mutation is proportional to the number of errors caused by the chord relative to the total number of errors in the musical score.

This new approach makes it possible to increase the probability of mutating the chord that generates the most errors within the score, reducing the number of total errors more quickly.

3.6. *Statistical Validation of Results.*

To assess the comparative analysis of the algorithm configurations, a Bayesian framework based on ranking models was adopted, explicitly using the Plackett–Luce model. This methodological choice was motivated by the need to directly quantify the relative probabilities of each configuration outperforming the others, incorporating the variability observed across multiple runs.

Each execution was treated as an independent ranking over the configurations, constructed from their observed performance metrics. These rankings were collectively analyzed through Bayesian inference to obtain posterior distributions over the strength parameters associated with each configuration. Unlike classical significance tests, this approach does not yield binary accept/reject decisions but instead provides interpretable posterior probabilities that express the likelihood of each configuration being the best.

Rojas-Delgado et al. [33] advocate for the use of Bayesian ranking models in algorithm comparison, highlighting their ability to account for uncertainty while avoiding multiple-testing corrections. Their framework supports probabilistic interpretation as a practical alternative to significance-based hypothesis testing, particularly in benchmarking scenarios.

4. Results and Discussion

All tests are run on a Dell M1000e cluster with 15 M600/M610 blade. The processors present are all Intel Xeon of the following models: E5506, E5507, E5640, E5520, and X5670. The cluster has a total of 136 cores and 376 GB of RAM. A virtual machine with 8 vCPUs and 8 GB of RAM is used for each run, and the Debian 12 operating system without GUI is used to minimize the interference of other software on CPU usage.

4.1. Performance Analysis. Here, both local and global performance aspects of the GA are analyzed. Specifically, two distinct measurements are considered:

- Chord-pair error computation: the time required to compute the error between two consecutive chords, a fundamental operation in the fitness evaluation of individuals.
- Complete execution of the GA: the total runtime of the evolutionary process under different configurations.

To enhance the efficiency of error computation, a precomputed database containing chord-pair fitness values has been used. As shown in Table 1, we present the average computation times for this operation, comparing the traditional sequential approach with a parallel implementation that leverages the precomputed database. Each result is averaged over 100 independent tests. The sequential computation of errors takes 22.5 s on average, whereas using the parallel version with the database reduces this time to only 0.001 s, resulting in a speedup of 22,500.

Table 1: Error calculation time between two consecutive chords in seconds.

Sequential	Parallel + DB	Speedup
22.5	0.001	22,500

However, chord pair errors computation represents only one component of the total computational load. To assess the impact at the algorithm level, performance was measured by considering the execution of the entire evolutionary process, which consists of two GAs. Three different configurations were tested for the entire process: (1) sequential execution, (2) execution with parallelism but without using the database, and (3) parallel execution with the precomputed fitness database. Figure 5 shows the total execution times in these three configurations. Compared to the basic sequential version, the parallel version achieves a 1.5-fold increase in speed, while the combination of parallelism and precomputed fitness evaluations results in a 12.8-fold increase in speed. This confirms that although parallelism alone provides a moderate improvement, the use of the precomputed database has the most significant impact on the total execution time of the evolutionary system.

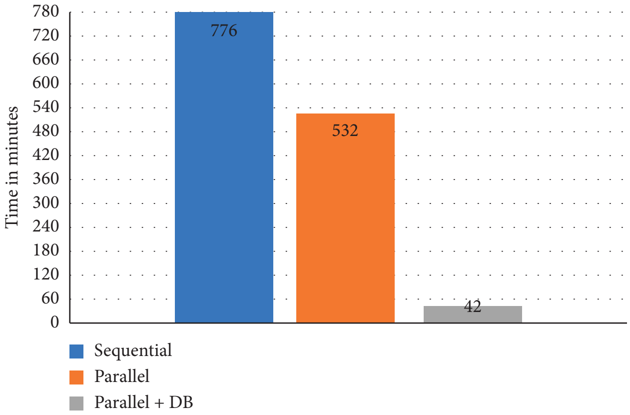


Figure 5: Comparison of execution times of the two GAs (whole algorithm) between sequential and parallel versions with database in minutes.

4.2. Results Over Synthetic Models. Before analyzing the size of the search space, we first consider the time savings from using precomputed fitness values when available. As mentioned before, this can be analyzed considering a synthetic harmonic model: it is enough to fill the database with fake fitness values for any possible pair of chords that could be updated with real ones when available. We check the potential of this idea in combination with a directed mutation operator.

The first test involves trying three harmonic patterns with different percentages of error-free chords. This gives us an idea of how the algorithm may behave when we encode patterns of different styles, such as Jazz or classical. We consider three models with 1%, 10%, and 20% error-free chords, each runs 5 times.

In this case, the whole GA is tested using directed mutation with 4 individuals over 100 generations. Different database configurations, each representing a distinct harmonic, were used across separate runs. This approach demonstrates that the EA can consistently continue evolving solutions, regardless of the specific harmonic model employed. Figure 6 illustrates how the results improve progressively across generations in all tested cases, regardless of the model employed or the initial percentage of error-free chords, thereby supporting the initial hypothesis regarding the approach's adaptability. Bayesian tests are not necessary here because the interest is to see whether the GA can work

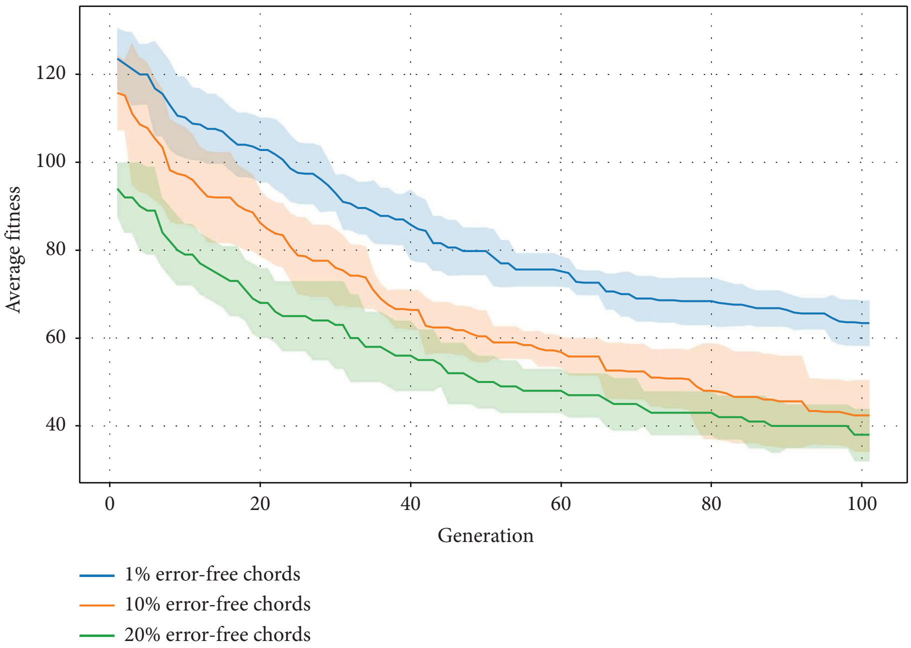


Figure 6: Tested synthetic harmony models.

with different synthetic harmonic models and whether it improves across generations, and it is not intended to show which model performs better.

4.2.1. Considering Population Sizes. The above-described experiments, although different from the point of view of the synthetic models, were exactly the same from the point of view of the GA applied. Thus, the three experiments using 4 individuals in the population showed modest improvement in the 100 generations computed.

We thus decided to test larger values for the population size in order to decide an appropriate value for the rest of the experiments. Thus, two experiments were conducted: one with 20 individuals and another with 8 individuals, reaching a maximum of 1000 individuals evaluated during the evolutionary process.

Figure 7 compares the number of individuals analyzed with the average fitness (here, average fitness means the average fitness values of the best individual in each experiment) value and standard deviation obtained over the runs. The experiment with 8 individuals showed significantly better results than the one with 20. To validate the results, a Bayesian test is performed with the Plackett–Luce model with a Dirichlet prior. The results indicate that the configuration using 8 individuals is strongly favored over the one with 20 individuals. Specifically, the estimated strength parameter (θ) assigned to the 8 individuals' setup is 0.86, with a 95% credible interval (CI) of [0.55, 0.99], while the 20-individuals configuration receives a much lower strength of 0.14, CI [0.00, 0.46]. Furthermore, the posterior probability that the 8-individuals configuration outperforms the 20-individuals one is approximately 98.5%. These results reflect a consistent and statistically meaningful superiority of the 8-individuals setup across the considered runs. Consequently, we decided to use 8 individuals with a larger number of generations for the next experiments.

4.2.2. Analyzing Synthetic Harmonic Models and Mutation Operators. Once the appropriate population size is decided, 8 individuals, we decided to run the experiment with synthetic models using this size before addressing the problem of using real harmonic models.

The general configuration of the algorithm is presented in Section 3.6, and the population size is 8 individuals. Table 2 lists the parameters used to test each configuration of the GA, varying according to the experiment performed. For instance, 50% DIRECTED 50% RANDOM means that half of the time directed mutation is applied, and half of the time random one. For each configuration, 10 runs were performed, and the average of the fitness values obtained was used for comparison.

Figure 8 illustrates the results, showing the average fitness over the generations.

Most of the experiments follow a similar trend except the EV0 experiment. To validate the results, a Bayesian test is performed with the Plackett–Luce model with a Dirichlet prior. The results indicate that configuration EV0 shows the highest strength (mean 36.07% and CI [18.36%, 55.60%]), followed by EV3 (21.16%, CI [10.53%, 36.37%]) and EV4 (21.61%, CI [10.67%, 35.00%]). Configurations EV1 (4.61%, CI [1.29%, 10.31%]), EV2 (7.50%, CI [2.92%, 14.48%]), and EV5 (9.05%, CI [3.86%, 16.73%]) show significantly lower performance, with strengths consistently below 10%. Notably, configuration EV0 exhibits the highest posterior probability of being the best among all candidates (77.5%), further confirming its leading performance. This indicates that the application of directed mutation, coupled with a preliminary step of chord degree evolution, contributes to

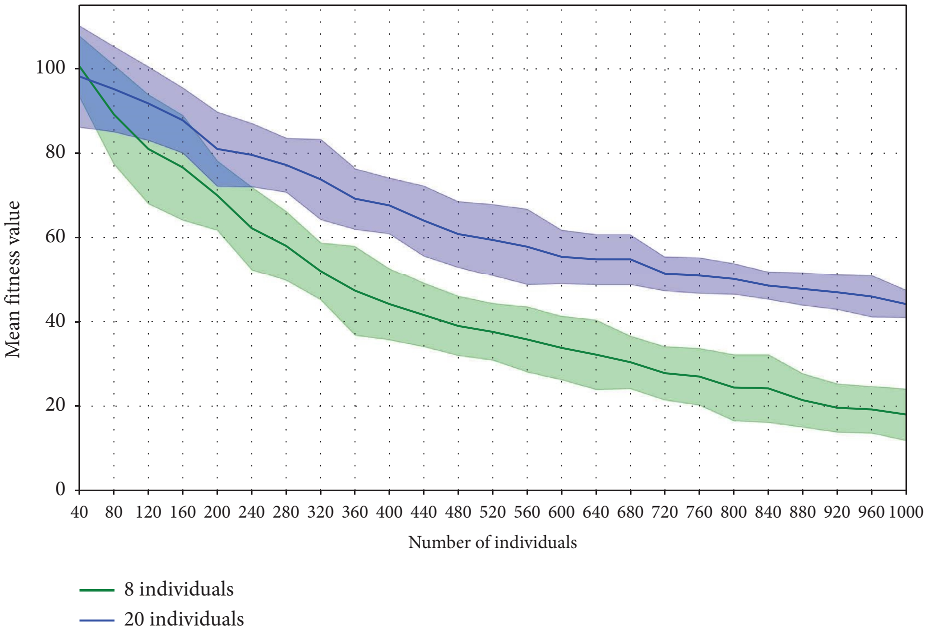


Figure 7: Comparison of the total number of individuals evaluated and best fitness. Directed mutation was applied, and chord progression evolved first.

Table 2: Genetic algorithm configuration used for experiments.

Exp.	Evolving chord progression	Mutation
EV0	Yes	100% directed
EV1	Yes	100% random
EV2	Yes	50% dir. 50% rand
EV3	No	100% directed
EV4	No	100% random
EV5	No	50% dir. 50% rand

improving the obtained solutions, promoting faster improvement and more efficient results.

4.3. How the Human Teaching–Learning Process Allows to Narrow the Search Space. Once we have validated the effectiveness of anticipating the fitness evaluation, synthetic harmonic models, and directed mutation, the natural next step is to use the standard harmonic model.

Unfortunately, as described earlier in Section 2.5, it is impossible to precompute the entire solution space, so we were inspired by a machine teaching and human learning approach and analyzed Sharpmony students' exercises. About 13,000 of them were extracted and analyzed, and chord pairs errors were computed and saved in the database. This was possible because the number of different chord pairs ever employed was quite small when compared with the whole search space. This is quite noticeable: although a huge search space is available, the teaching–learning process allows them to focus on a very narrow area, as shown below.

The analysis thus produced a database with just 67,240 chord pairs, which represents less than 1% of the possible pairs in a single key; 43.41% of the chord pairs ever used by students are error-free, as shown in Figure 9(a), and 17, 296 chord pairs have only one error, while only one pair records the maximum number of errors, which is 13.

The reduction in frequency is particularly strong between 0 and 4 errors, which together account for 97.80% of the chord pairs. As a result, only 2.20% of the pairs have more than 4 errors, indicating that combinations with a high number of errors are rare and rarely used by students.

Figure 9(b) elaborates on the most frequent types of errors. Among these, the most common are incorrect duplication of notes, omission of the third in a chord, use of an incorrect chord or chord not belonging to the key, and direct fifths or octaves. These errors together account for 53.2% of the total.

It is useful to point out that the database includes the work of students with different proficiency levels, from the earliest exercises on fundamental rules to the more advanced ones tackled in Harmony 1 and 2 courses. For this reason, the database also includes complex chords, such as the Neapolitan sixth and key changes.

Analysis of key changes between chord pairs (Figure 10) shows that these are present in 4798 pairs, of which 2052 are error-free. Although the GA does not allow modulation for now, it is important to study this distribution. These data make us realize that Sharpmony users make limited use of modulation.

Analysis shows that the data obtained from students' exercises differ significantly from the overall space of solutions, accounting for only 0.83% of the total available. This is the result of the learning path: students gradually assimilate the rules and arrangements, starting from the simplest to the most complex ones.

Moreover, the hypothesis formulated is confirmed: students manage to significantly narrow the search space, identifying a high percentage of error-free chord pairs and,

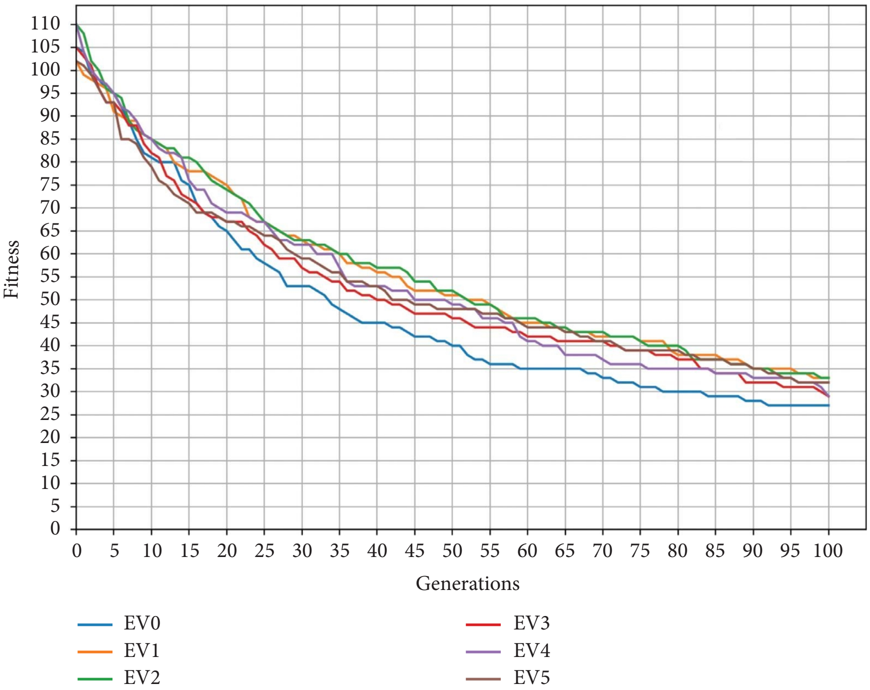
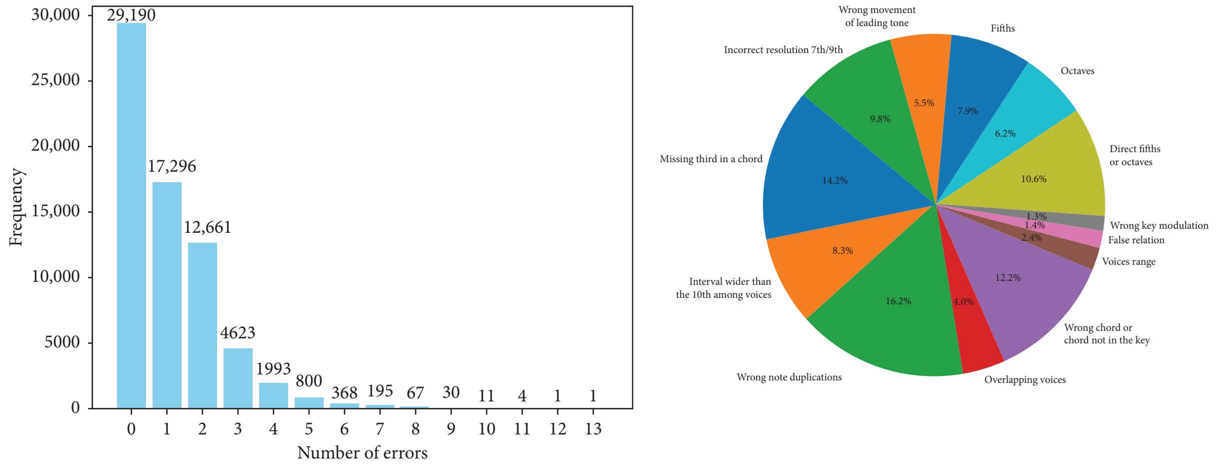


Figure 8: Average fitness values for different GA configurations over synthetic models of harmony.



(a)

(b)

Figure 9: (a) Error frequency and (b) error type.

consequently, correct solutions for their exercises. This allows both the educational method teachers adopt and the student's learning process to be indirectly incorporated into the algorithm, effectively exploiting this narrower search space.

4.4. Comparison Between Directed Mutation and Adaptive Directed Mutation. Only the chord pairs effectively used by the students were considered to evaluate the proposed approach to reducing the solution space, with the number of errors already calculated and recorded in the database. All rules defined in Sharpmony for error checking were applied, except for dissonant movements involving more than 3 consecutive chords. Consequently, the fitness values reported do not include any penalties for dissonant movements.

To ensure the statistical validity of the results, each configuration was repeated 35 times: (i) algorithm with directed mutation and (ii) algorithm with adaptive directed mutation. Figure 11 shows the comparison between directed mutation (in green) and adaptive directed mutation (in purple).

We can see that adaptive directed mutation gives similar results to directed mutation. In addition, the standard deviation for directed mutation is slightly lower. The Bayesian Plackett–Luce model was employed to estimate the relative

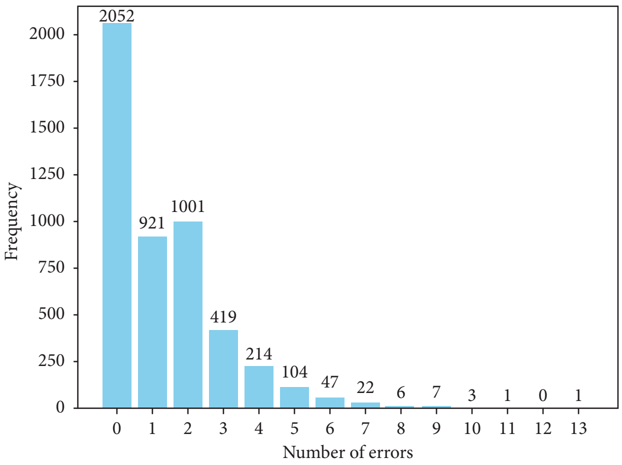


Figure 10: Error frequency with a key change.

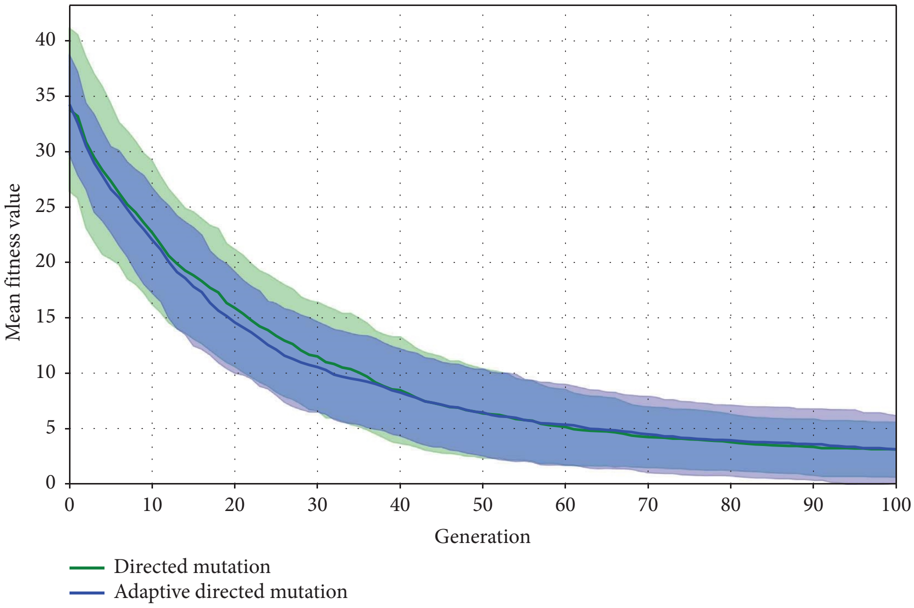


Figure 11: Comparison of directed mutation and adaptive directed mutation.

strengths (θ) of the two mutation strategies based on their ranking performance across 35 independent runs. The analysis reveals that Directed Mutation and Adaptive Directed Mutation exhibit nearly identical mean posterior strengths ($\theta = 0.50$, 95% CI [0.34, 0.66] for both). Furthermore, the posterior probability that Directed Mutation outperforms Adaptive Directed Mutation is approximately 49.6%, confirming their statistical equivalence in terms of ranking performance. Given their similar effectiveness, it is worth noting that Directed Mutation is computationally less demanding. This is because it does not require calculating the probability of each chord being selected for mutation when errors occur; instead, it applies the same probability to all chords with errors.

Furthermore, to evaluate the effectiveness of the process, we compare the results obtained with the initial 2017 implementation [8]. The original version of the algorithm employed only 11 rules, as compared to the 50 rules and exceptions used by the current version, making it more difficult to search within the solution space.

The experiment performed in 2017 employing 50 individuals for 100 generations lasted 24 h and obtained the best fitness of 10, that is, 10 errors, by applying the 11 rules implemented. The current version, which takes about the same amount of time over 100 generations, produced solutions with 5 errors on average and used more than 50 rules. Thus, the new version represents a significant leap in quality over the previous one.

4.4.1. *Analyzing Longer Runs.*

To test the real convergence of the algorithm, we run longer tests with 200 generations. Figure 12 shows how the algorithm starts with a mean fitness of about 33.3 and ends, at generation 200, with a mean fitness of 0.7, and standard deviation of 1.57. In the 2017

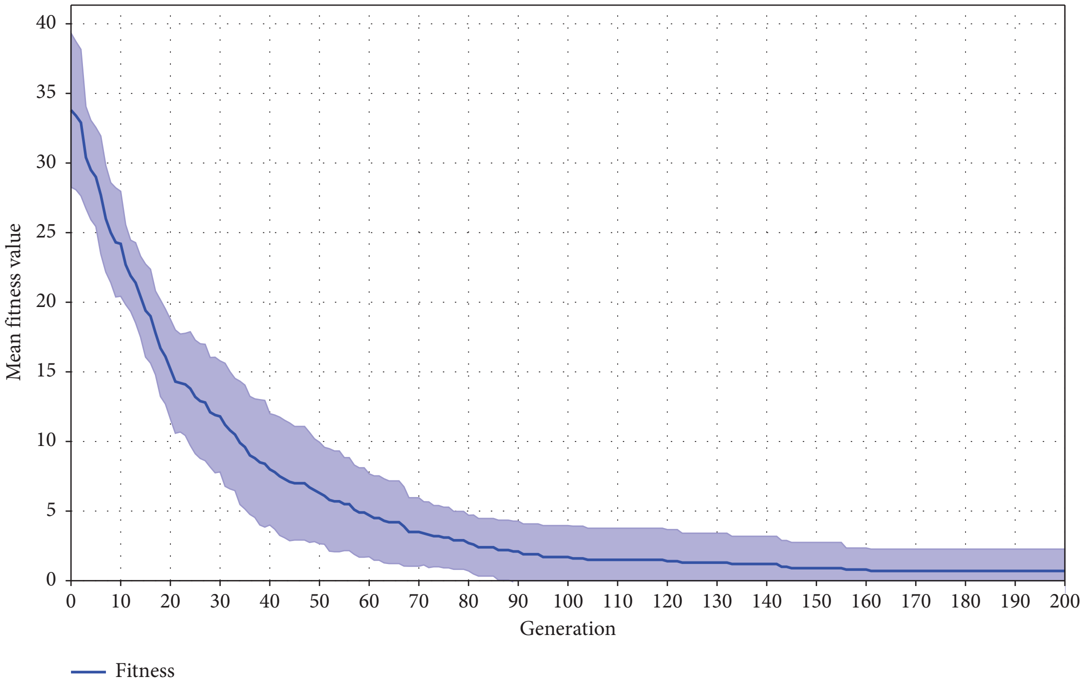


FIGURE 12: Long runs with directed mutation over student's search space.

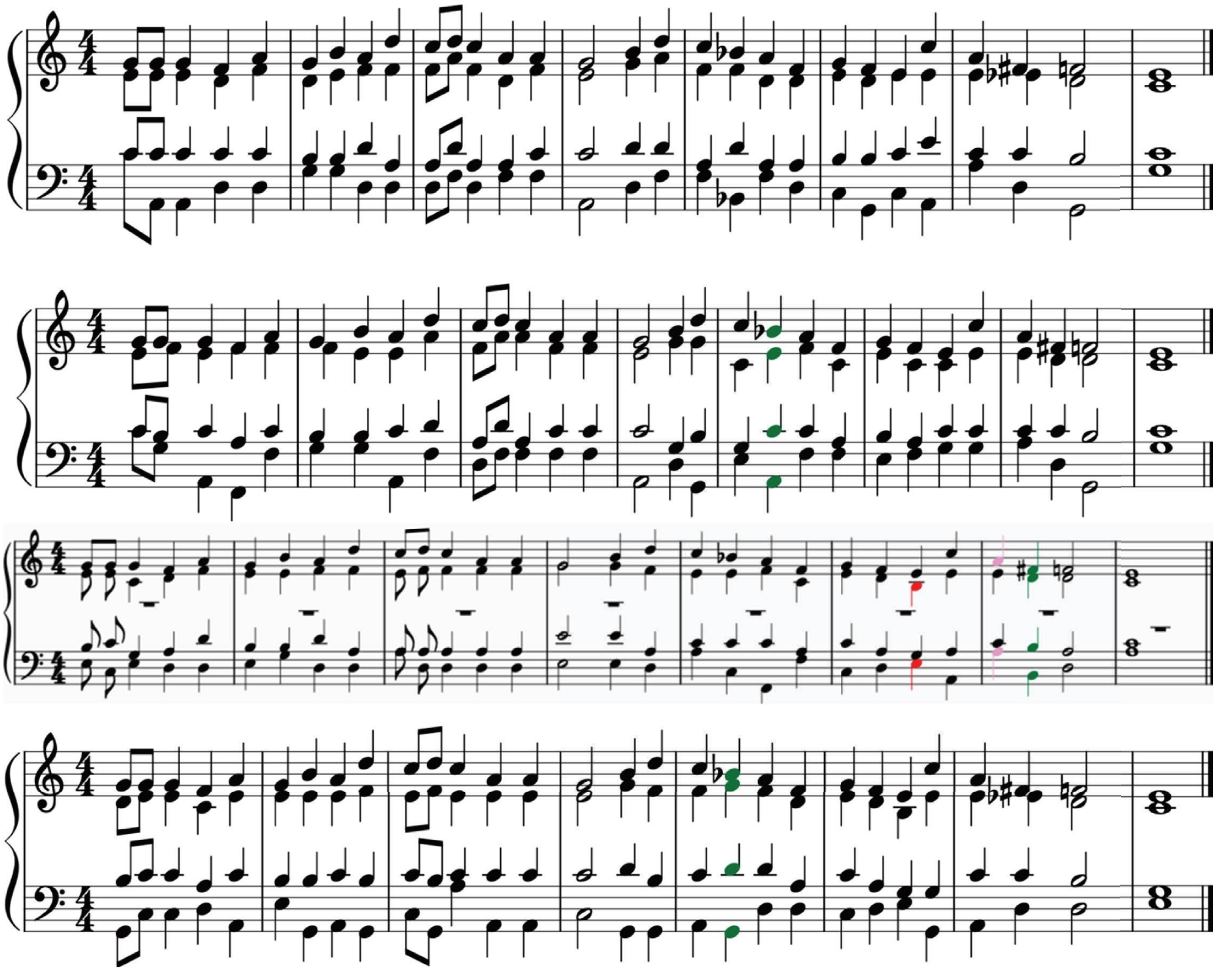



FIGURE 13: Some scores result from the evolutionary process, checked by Sharpmony.

experiments, 50 individuals were used over 100 generations, with 11 rules, resulting in a total of 5000 individuals analyzed. In these new experiments with directed mutation, we employ 8 individuals with 200 generations, with more than 50 rules and exceptions. The total individuals analyzed is 1600. Comparing the results, we go from an average fitness of 10 to 0.7. Therefore, the use of Directed Mutation, combined with research space based on student learning, led to better results than previous approaches. To the best of our knowledge, these are the best results found by evolutionary approaches.

Figure 13 shows some examples of scores obtained by the algorithm at the end of the evolutionary process. We show one example with no errors and the others with some errors to highlight how the system provides corrections. The colors in the notes mark errors found.

Thus, reducing the solution space by exploiting the knowledge acquired and used by the students, combined with the use of directed mutation, significantly improved the whole evolutionary process.

To summarize, the MT model–inspired approach, together with the precomputation of fitness values and directed mutation, made it possible to (i) reduce the space of solutions to be explored, (ii) improve the quality of results for the same execution time, and (iii) obtain solutions with a minimal number of errors, even when applying a more extensive set of rules for the verification of exercises.

5. Conclusions

This paper presents a significant advance in the approach to the 4-part harmonization problem using artificial intelligence, particularly EAs. Unlike approaches already in the literature, which are often carried out with simpler problems, this study employs over 50 rules and exceptions developed with the support of conservatory professors. Moreover, for the first time, an analysis of the search space is shown from the point of view of students addressing the problem.

Synthetic models are used for testing the main parameters for the algorithm, but they also allow us to envision the usefulness of the approach for any harmonic model we may apply in the future.

Numerous tests were conducted to evaluate the use of parallelism, the anticipation of the calculation of fitness values, and the directed mutation operator. Tests were applied over the synthetic models considered. The results show that parallelism and precomputing partial fitness values provide a speedup of 12.8 when compared with the standard sequential approach.

We also considered the machine teaching point of view, taking into account the way humans teach and learn in this specific context. In this case, precomputation of fitness values is also applied over the chord pairs that students have ever applied when solving exercises using Sharpmony.

The number of chord pairs used by students to complete their exercises was thus analyzed and compared with the total number of theoretically available combinations of chord pairs. The results show that the teaching–learning process allows students to drastically reduce the search space, considering less than 1% of the total possibilities. Therefore, we focused on this narrow subset of partial solutions.

Next, we compared the use of directed mutation and adaptive directed mutation. The results show that at the performance level, the directed mutation and adaptive directed mutation methods are similar. However, directed mutation is preferable because its computational cost is lower, unlike adaptive directed mutation, which requires calculations to assign the mutation probability to each agreement with error.

Finally, we compared the new approach with the algorithm presented in 2017. In the previous version, the algorithm produced an individual with 10 errors by applying 11 rules. The new version provides error-free sheet music on long runs. We must remember that 50 rules are applied now instead of just 11. It marks a significant improvement and demonstrates the validity of the new approach.

In the roadmap for future work, we want to implement local search to optimize the selection of the new chords.

Finally, we want to connect the database to the Sharpmony application and feed it directly with the students' data.

Data Availability Statement

The research data are not shared as we are unable to share the data publicly currently.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

We acknowledge support from the Spanish Ministry of Economy and Competitiveness under projects PID2020-115570GB-C21 and PID2023-147409NB-C22 funded by MCIN/AEI/10.130\-39/501100011033, and cofinanced 85% by the European Union, European Regional Development Fund, and the Regional Government of Extremadura. Managing Authority. Ministry of Finance (GR24142).

Endnotes

[1] https://sharpmony.unex.es.

[2] https://sharpmony.unex.es/index.php?r=tutorials%2Fharmonic-manual.

References

[1] O. Lopez-Rincon, O. Starostenko, and G. Ayala-San Martín, "Algoritmic Music Composition Based on Artificial Intelligence: a Survey," in *International Conference on Electronics, Communications and Computers (CONIELECOMP)* (Veracruz, Mexico, May 2018), 187–193, https://doi.org/10.1109/CONIELECOMP.2018.8327197.

[2] C.-H. Liu and C.-K. Ting, "Computational Intelligence in Music Composition: a Survey," *IEEE Transactions on*

Emerging Topics in Computational Intelligence 1, no. 1 (2017): 2–15, https://doi.org/10.1109/TETCI.2016.2642200.

[3] E. Pacioni and Francisco Fernández de Vega, "On the Impact of Directed Mutation Applied to Evolutionary 4-Part Harmony Models," in *Artificial Intelligence in Music, Sound, Art and Design* (EvoMUSART, 2024), https://doi.org/10.1007/978-3-031-56992-0_20.

[4] E. Pacioni and F. F. de Vega, "Paving the Way Towards Evolutionary Machine Teaching," in *Evolutionary Algorithms* (2024).

[5] A. Horner and D. E. Goldberg, "Genetic Algorithms and Computer-Assisted Music Composition," in *International Conference on Mathematics and Computing* (Jaipur, India, January 1991), 437–441.

[6] R. A. McIntyre, "Bach in a Box: the Evolution of Four-Part Baroque Harmony Using the Genetic Algorithm," in *Evolutionary Computation, 1994* (IEEE World Congress on Computational Intelligence, 1994), 852–857, https://doi.org/10.1109/ICEC.1994.349943.

[7] M. Kaliakatsos-Papakostas and E. Cambouropoulos, "Probabilistic Harmonization with Fixed Intermediate Chord Constraints," in *ICMC* (2014), https://doi.org/10.13140/2.1.3079.5526.

[8] F. Fernández de Vega, "Revisiting the 4-Part Harmonization Problem with GAs: a Critical Review and Proposals for Improving," in *IEEE Congress on Evolutionary Computation (CEC)* (2017), 1271–1278, https://doi.org/10.1109/CEC.2017.7969451.

[9] R. De Prisco, G. Zaccagnino, and R. Zaccagnino, "EvoComposer: an Evolutionary Algorithm for 4-Voice Music Compositions," *Evolutionary Computation* 28, no. 3 (2020): 489–530, https://doi.org/10.1162/evco_a_00265.

[10] F. de Vega, Francisco, J. Alvarado, A. Sánchez, M. Serrano, and E. Pacioni, "Evolutionary Algorithms: a New Hope for the Future of Music Teaching," in *Proceedings of the Conference on Genetic and Evolutionary Computation (ACM GECCO'23)* (Lisbon, Portugal, July 2023), 65–66, https://doi.org/10.1145/3583133.3596945.

[11] F. de Vega, Francisco, J. Alvarado, and M. Morita, "CAEPIA-APP Competition: Sharpmony, a Computational Intelligence Based Tool for 4-Part Harmony," in *Proceedings of the XIX CAEPIA* (Coruña, Spain, October 2021), 1009–1012.

[12] Y. Jin, "A Comprehensive Survey of Fitness Approximation in Evolutionary Computation," *Soft Computing* 9, no. 1 (2005): 3–12, https://doi.org/10.1007/s00500-003-0328-5.

[13] S. Hosder, L. T. Watson, B. Grossman, et al., "Polynomial Response Surface Approximations for the Multidisciplinary Design Optimization of a High-Speed Civil Transport," *Optimization and Engineering* 2, no. 4 (2001): 431–452, https://doi.org/10.1023/a:1016094522761.

[14] L. Shi and K. Rasheed, "A Survey of Fitness Approximation Methods Applied in Evolutionary Algorithms," *Evolutionary Learning and Optimization* 2 (2010): 3–28, https://doi.org/10.1007/978-3-642-10701-6_1.

[15] D. Buche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating Evolutionary Algorithms with Gaussian Process Fitness Function Models," *IEEE Transactions on Systems, Man, and Cybernetics-Part C* 35, no. 2 (2005): 183–194, https://doi.org/10.1109/tsmcc.2004.841917.

[16] H. Ulmer, F. Streichert, and A. Zell, "Model-Assisted Evolution Strategies," in *Knowledge Incorporation in Evolutionary Computation* (2005), 333–355.

[17] D. Melamed, Y. Nov, A. Malik, et al., "De Novo Mutation Rates at the Single-Mutation Resolution in a Human HBB Gene-Region Associated with Adaptation and Genetic Disease," *Genome Research* 32, no. 3 (2022): 488–498, https://doi.org/10.1101/gr.276103.121.

[18] D. Bhandari, N. R. Pal, and S. K. Pal, "Directed Mutation in Genetic Algorithms," *Information Sciences* 79, no. 3–4 (1994): 251–270, https://doi.org/10.1016/0020-0255(94)90123-6.

[19] P.-H. Tang and M.-H. Tseng, "Adaptive Directed Mutation for Real-Coded Genetic Algorithms," *Applied Soft Computing* 13, no. 1 (2013): 600–614, https://doi.org/10.1016/j.asoc.2012.08.035.

[20] B. Rodríguez-Puerta, F. Díaz-Barrancas, F. Chávez, and F. F. de Vega, "Un Análisis Preliminar De Nuevos Modelos De Mutación Dirigida En Algoritmos Genéticos," *Conferencia De La Asociación Española Para La Inteligencia Artificial (CAEPIA 2018)* (Granada).

[21] P. Carvalho, J. Mégane, N. Lourenço, and P. Machado, "Context Matters: Adaptive Mutation for Grammars," in *Genetic Programming: 26th European Conference, EuroGP 2023, Held as Part of EvoStar 2023* (Brno, Czech Republic, April 2023), https://doi.org/10.1007/978-3-031-29573-7_8.

[22] A. K. Das and D. K. Pratihar, "A New Search Space Reduction Technique for Genetic Algorithms," *Advances in Intelligent Systems and Computing* 812 (2019): 111–119, https://doi.org/10.1007/978-981-13-1540-4_12.

[23] Y. Orito and Y. Hanada, "Search Space Reduction Approach in Evolutionary Algorithms: the Case of High-Dimensional Portfolio Replication Problem," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (Vienna, Austria, October 2017), 554–559, https://doi.org/10.1109/SMC.2017.8122664.

[24] Y. Shiraishi, T. Ono, and M. A. E. Dahb, "Solution Space Reduction of Simulated Evolution Algorithm for Solving Standard Cell Placement Problem," in *Fifth International Conference on Natural Computation* (Tianjin, China, August 2009), 420–424, https://doi.org/10.1109/ICNC.2009.681.

[25] K. E. Duncan, S. K. Boddhu, M. Sam, and J. C. Gallagher, "Islands of Fitness Compact Genetic Algorithm for Rapid In-Flight Control Learning in a Flapping-Wing Micro Air Vehicle: a Search Space Reduction Approach," in *IEEE International Conference on Evolvable Systems* (Canberra, Australia, December 2014), 219–226, https://doi.org/10.1109/ICES.2014.7008743.

[26] A. S. S. M. B. Ullah, R. Sarker, and D. Cornforth, "Search Space Reduction Technique for Constrained Optimization with Tiny Feasible Space," in *Proceedings of the Conference on Genetic and Evolutionary Computation (ACM GECCO'08)* (Lisbon, Portugal, July 2023), 881–888, https://doi.org/10.1145/1389095.1389268.

[27] X. Zhu, "An Overview of Machine Teaching" (2018), https://doi.org/10.48550/arXiv.1801.05927.

[28] X. Zhu, "Machine Teaching: an Inverse Problem to Machine Learning and an Approach Toward Optimal Education," *Proceedings of the AAAI Conference on Artificial Intelligence* 29 (2015): 1, https://doi.org/10.1609/aaai.v29i1.9761.

[29] P. Y. Simard, "Machine Teaching: a New Paradigm for Building Machine Learning Systems" (2017), https://doi.org/10.48550/arXiv.1707.06742.

[30] G. Ramos, C. Meek, P. Simard, J. Suh, and S. Ghorashi, "Interactive Machine Teaching: a Human-Centered Approach to Building Machine-Learned Models," *Human-Computer Interaction* 35, no. 5–6 (2020): 413–451, https://doi.org/10.1080/07370024.2020.1734931.

[31] F. Ng, J. Suh, and G. Ramos, "Understanding and Supporting Knowledge Decomposition for Machine Teaching," in

Proceedings of the 2020 ACM Designing Interactive Systems Conference (DIS'20) (Eindhoven, Netherlands, July 2020), 1183–1194, https://doi.org/10.1145/3357236.3395454.

[32] Y. Bengio, A. Lodi, and A. Prouvost, "Machine Learning for Combinatorial Optimization: a Methodological Tour D'Horizon," *European Journal of Operational Research* 290, no. 2 (2021): 405–421, https://doi.org/10.1016/j.ejor.2020.07.063.

[33] J. Rojas-Delgado, J. Ceberio, B. Calvo, and J. A. Lozano, "Bayesian Performance Analysis for Algorithm Ranking Comparison," *IEEE Transactions on Evolutionary Computation* 26, no. 6 (2022): 1281–1292, https://doi.org/10.1109/tevc.2022.3208110.