The Four-Part Harmonisation Problem: A comparison between Genetic Algorithms and a Rule-Based System

Somnuk Phon-Amnuaisuk; Geraint A. Wiggins
Music Informatics Research Group,
Division of Informatics, University of Edinburgh,
80 South Bridge, Edinburgh, EH1 1HN

Abstract

Problem solving can be compared to a search in a state space. The description of the domain knowledge must be encoded as a knowledge base, then the problem solver searches through the state space for an appropriate answer. The experiment reported here is a comparison between GAs and a rule-based system searching the same space: that of four-part harmonisations of chorale melodies. The experiment leads to the conclusion that the quality of the solution fundamentally depends on the knowledge the system possesses and that the KBS is better suited for this task. We then outline an alternative approach of *explicitly structuring control knowledge* to solve this problem.

1 Introduction

The purpose of this paper is to give a comparison of results between *Genetic Algorithms (GA)* and a rule-based technique in the music problem domain. Our previous experiments in generating traditional musical harmony using GA are surprisingly successful given the limited amount of knowledge in our system (Phon-Amnuaisuk et al., 1999), but there are limitations in two issues:

- There is no guarantee of an optimum solution from a GA.
- The GA is a very attractive weak method search; but as one puts more knowledge in the system, it becomes computationally very expensive.

It is inevitable that the system must be rich in knowledge in order to be able to produce quality outputs consistently. Being aware of computational expense in a knowledge-rich GA, we seek an alternative approach. We go back to the conventional rule-based system as a starting point. This experiment is a step towards another alternative which we call the *explicitly structured knowledge* paradigm, which will be discussed later.

The experiment is carried out with a chorale style four-part harmonisation. In order to compare the behaviour of two different search techniques, both systems have been constructed with exactly the same knowledge.

2 Knowledge engineering

When writing a small-scale program for a particular task, one often does not think much about the knowledge engineering issue (*i.e.*, an explicit separation of domain knowledge from the structure of the program) and one often liberally mixes domain knowledge into the program structure. As the program grows bigger, one wants to make the program extendable and easy to maintain. This requirement leads to a distinction of two knowledge components in a system; the knowledge base and the inference engine which exploits the knowledge. To achieve an extendable and modifiable system, the system should be constructed with a separation between domain knowledge part and the exploitation of the domain knowledge part (knowledge base and inference engine) (Davis, 1980; van Harmelen, 1989).

The knowledge can be organised in different ways. Generally, one talks about knowledge in terms of *meta-level knowledge, object-level knowledge, domain knowledge and control knowledge*. In this paper, we will discuss and compare the knowledge content of the GA system and the rule-based system with reference to the above classification terms. These terms sometimes carry an ambiguous meaning; here we use them as follows:

- Object-level and Meta-level knowledge: Davis and Buchanan (1985) define the concept of meta-level knowledge as:

  ‘In the most general terms, meta-level knowledge is knowledge about knowledge. Its primary use here is to enable a program to “know what it knows”, and to make multiple uses of its knowledge. That is, the program is not only able to use its knowledge directly, but may also be able to examine it, abstract it, reason about it, or direct its application.’

  “Meta-level” and “object-level” are used to express the relationship between two knowledge levels. Normally the meta-level is the knowledge about the object level and the terms could be used to express the relative relationship; a meta level can be an object level of the higher meta-level.

Way (1994) suggests that although all the work on knowledge representation is interested in knowledge about knowledge, when meta-level knowledge is explicitly discussed, it is usually referring to the concept that object-level knowledge refers to domain knowledge and meta-level knowledge is the knowledge about the form and structure of the object-level knowledge.

- Domain knowledge and Control knowledge: Generally, domain knowledge is a section of world which is captured in the system and control knowledge is the knowledge that the system uses for exploiting the domain knowledge. The domain knowledge describes what the system knows about the problem. The control knowledge describes how the system makes use of the domain knowledge. The control knowledge can be domain specific or domain dependent.

To compare the results of two different approaches is difficult. The difficulty comes not directly from comparing the results but from constructing two systems which encode the same knowledge. We will discuss this issue next.

2.1 Where does knowledge reside in our GA?

In the Genetic Algorithms (GAs) method (Goldberg, 1989), solutions in a state space are represented as a population of *chromosomes*. The search is controlled by *reproduction operators* and a *selection process*.

One of the strengths of a GA system is that it can be constructed as a general-purpose search engine (a weak method approach) due to the advantage of the reproduction operators being able to operate without the domain knowledge information. In our GA, we have decided, a little unusually, to use knowledge-rich structures. The domain knowledge resides in:

- Chromosomes: Musical information (*e.g.,* pitch, interval, time, duration) is represented in choromosomes.
- A selection process: A *fitness function* judging the fitness of each chromosomes and deciding the reproduction opportunity of the chromosomes.
- Reproduction operators: Search control knowledge is embedded in the reproduction operators. Applying these operators ensures that all points in the state space can be reached.

More details can be found in Phon-Amnuaisuk et al. (1999).

2.2 Where does knowledge reside in our rule-based system?

In this experiment, the rule-based system is implemented (in Prolog) with a logic-based knowledge representation. Both the GA system and the rule-based system employ the same representation structure for musical information. Since both systems have the same granularity in their fundamental knowledge representation level, there is no difference in implicit knowledge in the knowledge representation at this level. We argue that both systems have the same amount of explicitly coded domain knowledge.

In contrast, the rule-based system is coded to solve the four-part harmonisation in a structured way. The search control knowledge is embedded in the program structure. The system uses chronological backtracking in the search process. It solves the problem in this order:

- Fill in the cadence section
- Fill in the precadence section
- Fill in the body section
- Fill in the introduction

2.3 Mixed knowledge

Both systems have their domain knowledge built up from the same basic building blocks, but there is a difference in their meta-level knowledge due to their fundamental architectures. The meta level in both systems is mixed within the object level knowledge.

2.3.1 Basic building blocks

The logic based knowledge representation is the selected representation scheme. The musical knowledge is built up with knowledge of:

- Pitch and Duration
- Harmony of the combined pitches
- Structure of the combined harmony (*e.g.* cadence)

2.3.2 Meta level in both systems

Both systems have their object-level knowledge and meta-level knowledge mixed together. In the GA system, the meta-level knowledge is mixed in the GA structures and GA operations (*e.g.,* chromosome representations, selection process and reproduction process). On the other hand, in the rule-based system, meta-level knowledge is mixed in the program structure. It is the characteristic of the GA system to perform an unstructured search because there is no explicitly structured plan in their search mechanism. The rule-based system, however, performs a structured search because the search plan is explicitly coded in the program. The explicitly structured search, an extra implicitly mixed meta-level knowledge, yields a better result in the rule-based system.

3 Experimental Results

3.1 Comparison criteria

The harmonisation rules employed by both systems are the standard four part harmonisation rules found in literature on harmony. In the GA system, these rules are employed to judge the quality of the output by rating how good *(fit)* the solution is. To compare the result of a rule-based search with the GA search, the result of a rule-based search is rated with the same fitness function from the GA system.

The fitness function is constructed from the harmonisation knowledge extracted from standard four-part harmony writing procedures. The evaluation is carried out at each vertical, and between each pair of adjacent horizontal, positions. The list below summarises all the requirements and the corresponding penalties if the requirements are not met. The fitness function evaluates the results according to these requirements. The penalties are, in some cases, imposed into multiple levels of punishment:

Requirement	**Penalty**
No hidden unison	10
All voices are within their own ranges	10
Intervals between Soprano-Alto, Alto-Tenor are not more than 1 octave	10
Interval between Tenor-Bass is not more than 12 diatonic degrees	10
No 2nd inversion chord unless it's a cadential 64	10
No doubling of leading note, prefer root and fifth doubling	10
No crossing between voices	10
No tritone leap	10
Alto, Tenor should not leap greater than perfect fourth	10
Bass should not leap greater than perfect fifth, but can leap 1 octave	10
Seventh from dominant seventh must resolve step down	10
No parallel unison, fifth, octave between two voices	10
Harmonic progression preference (Progression, Retrogression, Repetition)	10,1[a]
Opening the phrase with root position tonic on strong beat	100
End the phrase with a cadence	100,20[b]

Remarks

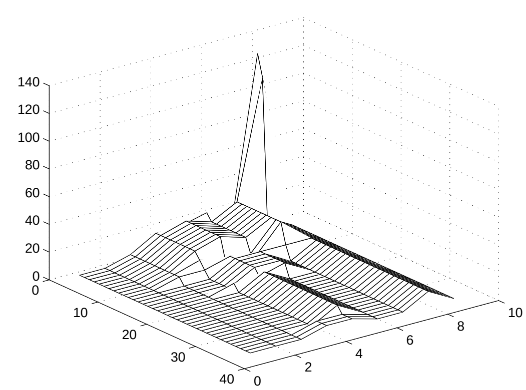
- a: Strong progression (*e.g.* fifth descent) would be free from punishment. Here, we have decided to punish retrogression and repetition by 1 point punishment and others with a 10 point punishment.
- b: Wrong cadence pattern would be punished as high as 100 points. Right pattern but with improper inversion or doubling would be punished at 20 points. The punishment in cadence is higher than others since we want to get the cadence in the right context before other parts.

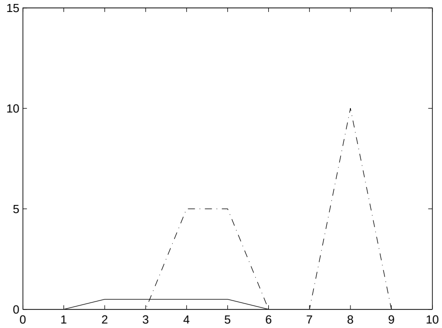
3.2 Results

We now present the results from both systems with the harmonisation example of the first lines of *Joy to the world*, *Epiphany* and *Bach's Chorales*. The fitness profile of the best chromosome in each GA generation for 250 generations is plotted. We decided to run the GA for 250 generations since at this stage, the GA is already converged and it is unlikely that it would produce a better answer if we let it run longer. We create a three dimensional plot of penalty value, generation and the position in the chromosome (*i.e.*, position of each melody), showing the positions which get punished by the evaluation function. It is also clearly shown that the GA tends to fall into a local optimum and it is very hard for the present system to get out of this local optimum. One may argue that the nature of harmonisation problem requires a much more sophisticated and highly knowledge rich GAs system to accomplish the task. However, once we have a GA system at that level, the system may not be recognisable as a GA at all. The fundamental issue here, we believe, is that the nature of the problem is not compatible to the GA approach.

Both the GA's solution and the rule-based system's solution are presented in short score for comparison. In order to give a clear comparison between a GA search and a rule-based search, the result from the rule-based search has also been fed to the GA's fitness evaluation function so that both results (from the GA—*broken line* and the rule-based system—*solid line*) could be compared in terms of penalty value.

3.2.1 Joy to the world





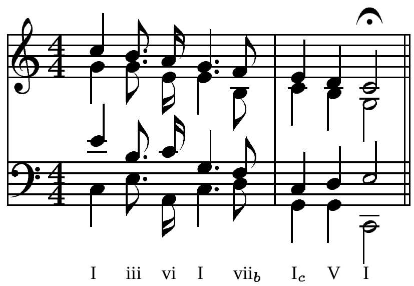


Figure 3: Result from the GA (Joy to the world)

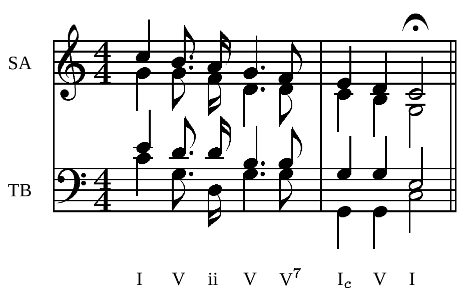


Figure 4: Result from the rule-based system (Joy to the world)

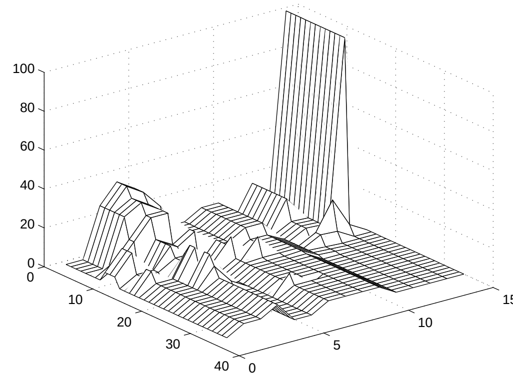
comes from:

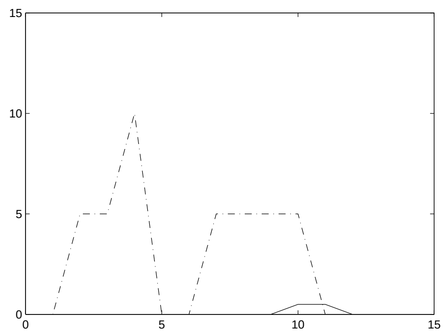
- The parallel octave between I-vii$_b$ transition in the first bar.
- The last pitch in the bass part is out of range.

The solution from the rule-based system has produced a much better output. In this case the penalty is only 2 and comes from soft constraint harmonic progression (*i.e.*, progress from V to ii, and progression from V to V^7).

3.2.2 Epiphany

The best solution from the GA scores 40 penalty points from:





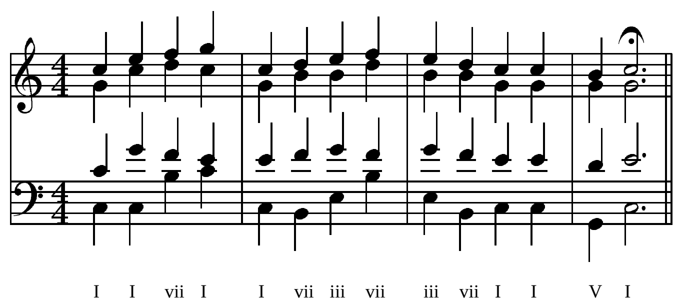


Figure 7: Result from the GA (Epiphany)

- Big root bass leap from I to vii progression in the first bar.
- Bass voice on the fourth beat of the first bar is considered as too close to the tenor voice.
- Progression iii to vii is not considered as an appropriate progression.

The solution from the rule-based system has produced a much better output. In this case the penalty is only 1 which comes from soft constraint harmonic progression (*i.e.*, progress from V to vi).

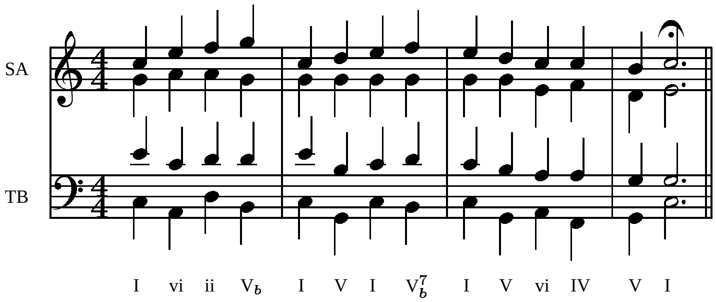
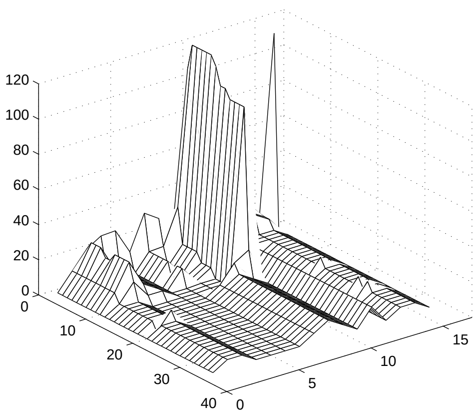
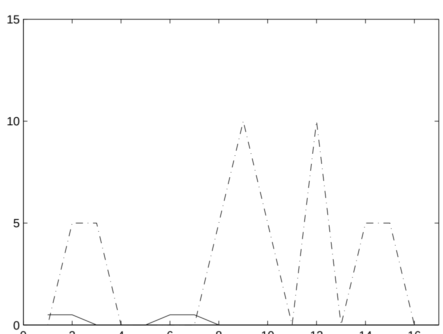


Figure 8: Result from the rule-based system (Epiphany)

3.2.3 Bach Chorale





- Big alto leap in the first bar.
- progression ii to I before cadence (first phrase) is considered inappropriate.
- Parallel fifth at first phrase cadence.

Figure 11: Result from the GA (Bach's Chorale)

Figure 12: Result from the rule-based system (Bach's Chorale)

- Bass g on the second beat of the second phrase is considered as too close to the tenor voice.
- progression iii to V before cadence(second phrase) is considered as inappropriate.

The solution from the rule based search scores 2 penalty points. It is the penalty from soft constraint harmonic progression (results from progression IV-I).

4 Towards an explicit control of the harmonisation problem

It is clear from these results that our conventional rule-based approach delivers a much better result (judged by the lower penalty values). This means that the rule-based search is able to find a better solution with the same explicitly coded domain knowledge. The results should not be a surprise: it is as expected from this experiment. However, this does not imply that the rule-based approach is an ultimate panacea. As a matter of fact, the rule-based system delivers a better result because it possesses extra implicit knowledge. The implicit part is from the fact that, here, the rule-based system is coded to solve the harmonisation problem with a structured search mechanism, but the GA system solves the problem with an unstructured search mechanism (Phon-Amnuaisuk et al., 1999). The GA system searches through the state space randomly by means of reproduction operators. The reproduction operators introduce changes randomly and locally in genes or group of genes in a chromosome. These unstructured local changes are prone to lead the GA system to a non-optimal solution (Wiggins et al., 1999).

In order to have explicit control of the knowledge, the control knowledge must be explicitly structured. The system with explicitly structured knowledge would let us take advantage of the implicit meta-level knowledge in the structure of the program (as appears in our experiments). The explicitly and separately represented control knowledge also gives the system modularity, efficiency, ease of developing, debugging and maintaining. This idea is not new: Bundy (Bundy, 1987; Bundy and Welham, 1979) has devised many systems with explicit control knowledge. Our ideas are influenced by this line of thought.

We coin the term *the explicitly structured knowledge paradigm* to represent the mental strategy for solving a musical composition problem. In this paradigm we view domain knowledge from two dimensions:

- Musical knowledge: The knowledge in this dimension is mostly naturally described declaratively.
- Musical processes: The knowledge in this dimension is mostly naturally described procedurally.

The philosophy behind this idea is grounded on the symbolic-AI approach and the belief that all knowledge has a hierarchical structure, so it is natural to deal with it in a hierarchical manner. Hierarchical representation of musical knowledge is general and powerful enough to capture the structural knowledge for computer composition and analysis purposes (Smaill et al., 1993).

In the explicitly structured knowledge paradigm, the musical knowledge dimension is hierarchically constructed. For example, the structure of a sonata may be represented as:

- Sonata → Exposition, Development, Recapitulation
- Exposition → 1st group, 2nd group
- Development → 1st group, 2nd group
- Recapitulation → 1st group, 2nd group

The musical knowledge dimension is represented by means of a *score* which represents the traditional musical notation (musical score) and the interpretation of that notation.

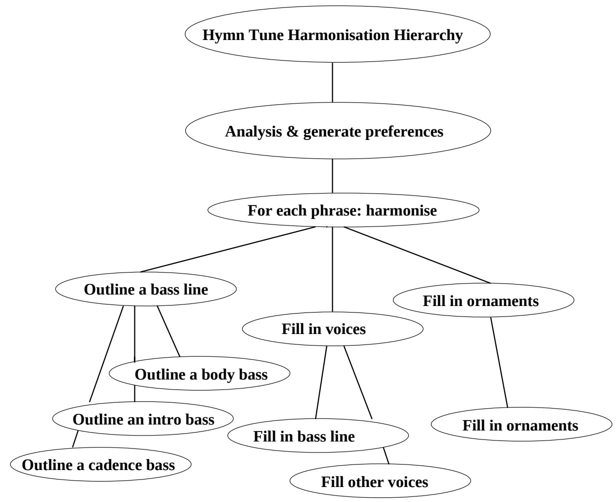


Figure 13: The hierarchical structure of a harmonisation process

The musical process dimension is represented by means of *methods* which encode a hierarchical structure of musical processes. Each method describes particular problem solving procedures. The method embodies object-level knowledge and meta-level knowledge of other methods. At the very top level, an inference engine controls the application of these methods.

One may point out that our *method* shares similar ideas with an object-oriented approach [1] (*i.e.*, our method is a concept which may be based on other concepts, and so on). One might say that both paradigms share the same intuition in a global view: looking at the problem as a chunk of composite smaller problems. However, the usage of our methods is more specifically defined than the use of object in object-oriented programming in general. We intend in future work to develop these ideas further.

5 Conclusion

It is quite clear from the experiments that the implicit knowledge in the program structure plays a crucial role in contributing to the quality of the harmonisation output.

[1] But do not get confused with the word "method" used in object-oriented programming, they denote different concepts

The rule-based system delivers much better output. However, this does not mean that the GA is an inferior system in comparison with the rule-based system. The conclusion from this experiment is that the quality of the output of any system is fundamentally dependent on the overall knowledge that the system (explicitly and implicitly) possesses.

A system may be constructed without clearly structured knowledge. Our rule-based system in this experiment is an example. The CHORAL expert system (Ebcioglu, 1993) is another example of a complex and unstructured knowledge system. The drawback of systems in this category is that they are very hard to extend, modify and understand.

To address this problem, we propose the explicitly structured knowledge paradigm. As the knowledge is explicitly structured, we can explicitly control it, and hence achieve a more effective and powerful means of search control.

References

A. Bundy. The use of explicit plans to guide inductive proofs. Dai research paper no.349, Dept Artificial Intelligence, University of Edinburgh, 1987.

A. Bundy and R. Welham. Using meta-level descriptions for selective application of multiple rewrite rules in algebraic manipulation. Dai research paper no.121, Dept Artificial Intelligence, University of Edinburgh, 1979.

R. Davis. Meta-rules: Reasoning about control. In *Journal of Artificial Intelligence*, volume 15, 1980.

R. Davis and B. G. Buchanan. Meta-level knowledge overview and applications. In Brachman R.J. and Levesque H.J., editors, *Reading in Knowledge Representation*, chapter 22, pages 390–397. Morgan Kaufmann Publishers, Inc., 1985.

K. Ebcioglu. An expert system for harmonizing four-part chorales. In S. M. Schwanauer and D. A. Levitt, editors, *Machine Models of Music*, chapter 17, pages 385–402. The MIT Press, 1993.

D. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Addison-Wesley, 1989.

S. Phon-Amnuaisuk, A. Tuson, and G. Wiggins. Evolving Musical Harmonisation. In *ICANNCA*, 1999.

A. Smaill, G. Wiggins, and M. Harris. Hierarchical music representation for composition and analysis. In *Computing and the Humanities Journal*, 1993.

F. van Harmelen. A classification of meta-level architecture. In P. Jackson, H. Reichgelt, and F. van Harmelen, editors, *Logic-based knowledge representation*, chapter 2, pages 13–36. The MIT Press, 1989.

E. C. Way. *Knowledge representation and metaphor.* Intellect Books, 1994.

G. Wiggins, G. Papadopoulos, S. Phon-Amnuaisuk, and A. Tuson. Evolutionary Methods for Musical Composition. In *International Journal of Computing Anticipatory Systems*, 1999.