Presented at JIM 2023 (Journées d'Informatique Musicale), Paris, France, May 24-26, 2023

A CONSTRAINT FORMALIZATION OF FUX'S COUNTERPOINT

Damien Sprockeels, Thibault Wafflard, Peter Van Roy
Université catholique de Louvain
{damien.sprockeels, peter.vanroy}@uclouvain.be
thibault.wafflard@gmail.com

Karim Haddad
IRCAM
karim.haddad@ircam.fr

RÉSUMÉ

La composition musicale assistée par ordinateur est un domaine en plein essor depuis plusieurs décennies maintenant, particulièrement dans le domaine de l'intelligence artificielle appliquée à la composition musicale. Cet article vise à contribuer à ce domaine en offrant un outil mathématique et informatique formalisant le contrepoint à deux voix selon Johann Joseph Fux à l'aide de la programmation par contraintes. Cet outil s'inscrit dans un projet plus ambitieux qui a pour but de créer un outil complet basé sur la programmation par contraintes afin d'assister les compositeurs dans leur processus de création, en ne nécéssitant aucunes notions en informatique.

1. INTRODUCTION

The last decades have seen a great increase in the use of computers for musical composition. Applications are endless : from software aiming to facilitate the composer's work by providing an intuitive interface to write scores or combine musical samples into a piece (LogicPro [16], Sibelius [25], OpenMusic [20]), to using machine learning to generate music fitting a given description or a certain style (MusicLM [1]). Our work uses an approach based on constraint programming. We observe that there is a close correspondence between the theory of musical harmony and the abilities of constraint programming. In this way, we aim to build a tool that allows the composer to express musical ideas which are then implemented in terms of constraints. The tool uses a constraint solver to generate musical solutions that satisfy the constraints. This allows the composers to think exclusively about musical ideas as the tool is in charge of expressing these ideas as a score.

Constraint programming Constraint programming is a field in computer science that allows for the specification of a set of rules, called constraints, to be applied to a set of variables modelling a problem. A solver then searches for solutions for the problem that respect the constraints specified. Ever since the invention of constraint programming, it has been an appealing idea to encode musical concepts as constraints. Many concepts can be expressed in this way, as previous work has shown ([28], [29], [5]). Constraint programming has the ability, given an appropriate interface, to allow the composer to specify their musical ideas at a high level of abstraction and it allows an iterative process where the composer can refine his or her musical ideas depending on the musical solution given by the constraint solver.

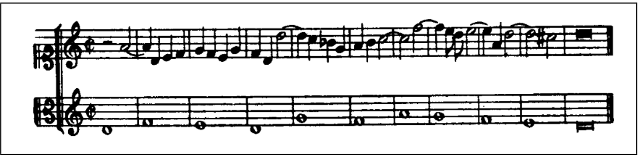
Figure 1. Example counterpoint in the style of Fux [14].

Music composition using constraints Constraint programming has often been applied to composition [23, 19, 17, 21], but this has not yet led to a practical tool usable by composers. There are two main reasons for this : first, past research was mainly focused on investigating how to express musical rules as constraints and second, the software tools that were developed required skills in programming and mathematics. A third problem that we have recognized in our own work is that in order to be practically useful to a composer, the tool must incorporate knowledge of the desired musical style. If it does not have such knowledge, the composer must first create it from scratch which makes it much harder to use. That is the main drawback of our own previous work ([2], [6], [3]). Indeed, the prototypes we developed are limited to a small set of constraints and do not support a coherent style. Despite this, they give useful proofs of concept and they can give interesting solutions to musical problems. The present work goes beyond these tools and is a first step towards a tool that supports a coherent musical style.

Choice of the Fux theory Musical styles are complex and few have been analyzed to a degree that would allow to easily formalize them as constraints. As our starting point, we have therefore chosen a relatively simple style that has been defined thoroughly, namely musical counterpoint. We have chosen the theory of counterpoint as presented by Johann Joseph Fux in his classic work Gradus Ad Parnassum, which was first published in 1725 [13]. Figure 1 gives an example of fifth species counterpoint for two voices in the Fux theory. We have chosen the Fux theory for three main reasons. First, it is presented in a clear and progressive way that steps from simple to more complex forms of counterpoint. Second, it is defined with a strict and comprehensive set of rules that make it straightforward to formalize in a mathematical language. Third, it is considered as a classic work and many of its rules are still present in various musical styles that are popular today. We have chosen the Fux counterpoint theory over other counterpoint theories, such as Jeppesen [15], Dubois [26], and Kœchlin [4], because of its simplicity, clarity, and completeness. Mikael Laurson gives a formalization of counterpoint based on the Bach Chorals in the context of the PWConstraints library of PatchWork [19].

Previous attempts at modeling Fux counterpoint Several earlier attempts at modeling Fux counterpoint have been made, some even using constraint programming. The first attempt we could find was Schottstaedt [24], who used a series of conditional statements to ensure a generated counterpoint would satisfy Fux's rules. His work provides a preference system for each rule as well as a relatively good efficiency (however, the fifth species could in some cases be computationally heavy). The main drawbacks were that the search for a solution was not customizable and that his work required computer science knowledge to be useable by a composer. Ovans and Davidson [22] provide an interactive interface where the composer can select a note and observe dynamic changes in the available possibilities for the remaining notes such that the result satisfies Fux's rules for the first species of two voice counterpoint. It does not however allow to generate counterpoint for other species. Cope [7] used a learning approach to generating first species counterpoint. He does not implement every rule for this species that Fux gives, but rather uses an evaluation based on six criteria representing the main rules as well as the examples from Fux's treatise which are used to train the model. Torsten Anders [28], in his Ph.D. work on Strasheela presents a constraint formalization of the first species of the Fux theory for two voices. We go beyond this to formalize all five species of Fux two-voice counterpoint. More recently, Herremans and al. [8] developed a tool allowing to generate fifth species counterpoint using a variable neighbourhood search algorithm to try and maximize an objective function. Our work goes a step further by analysing different translations of Fux's treatise and by allowing complete freedom for the composer when it comes to how strictly they want to follow the rules.

Ongoing project This paper is one step of a long-term collaborative project on computer-aided composition using constraint programming between UCLouvain and IRCAM. We briefly summarize our past work. In 2020, Baptiste Lapière [2] implemented Rhythm-Box, a proof of concept tool to explore constraints for rhythm. Rhythm-Box was implemented in the OpenMusic visual programming platform for music composition [20] using the Gecode constraint solver [10]. This work was accompanied by the development of GiL [11], an interface between Gecode and OpenMusic. In a next step, Damien Sprockeels developed Melodizer [6], a proof of concept tool applying constraint programming towards the modeling of simple musical rules for pitch. This work was presented at the IRCAM Forum in March 2022. Finally, we extended Melodizer to combine rhythm and pitch and to support polyphony, giving Melodizer 2.0 [3]. Melodizer 2.0 has an improved software architecture that supports a hierarchical approach to create music by combining constraint blocks. Future work aims to develop practical tools that target different musical styles, as explained in Section 5.

Contributions This paper makes three contributions :

— A mathematical formalization of the Fux theory of counterpoint for two voices. For brevity, the present paper presents the full formalization of the third species and explains how the other species are formalized. The full formalization is expected to be available by June 2023 in the master's thesis of Thibault Wafflard [27].
— An implementation of this formalization in OpenMusic using the Gecode constraint solver. This implementation is available on github as explained in Section 4.1.
— An evaluation of this implementation. We compare solutions found by the constraint solver with the counterpoints suggested by Fux in his treatise. We also show how it is possible to generate new counterpoints using the formalization.

Structure of the paper The paper is organized as follows :

— Section 2 summarizes the Fux theory of two-voice counterpoint.
— Section 3 gives the formalization of the third species of the Fux theory for two-voice counterpoint.
— Section 4 explains the implementation and evaluates it with respect to the Fux treatise.
— Section 5 explains how we intend to continue this work towards a full-fledged tool for music composition.
— Section 6 recapitulates the work and gives some conclusions.

2. FUX THEORY OF COUNTERPOINT

The Gradus Ad Parnassum by Johann Joseph Fux is considered as a classic reference for counterpoint [13]. Famous composers such as Haydn, Beethoven, and Mozart are all said to have studied it. The treatise is organised in a didactical and progressive manner. It is presented as a conversation between a master, Aloys, and his student, Joseph, where Joseph is discovering and applying the rules of counterpoint aided by Aloys. The treatise starts with two-voice counterpoint, which is presented stepwise in five progressive forms, called "species". The elaboration of a counterpoint starts with a theme given by the composer, called the cantus firmus. The rules of counterpoint are then used to generate additional voices, called counterpoints. In the first species, both the cantus firmus and

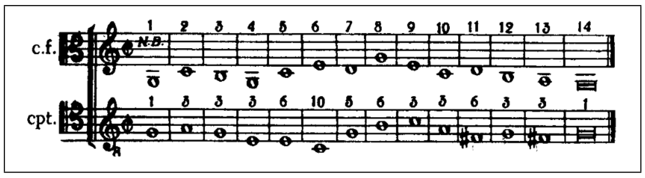

Figure 2. Example counterpoint for the first species [14].

counterpoint consist of measures of whole note against whole note. After a didactic conversation between master and student, the treatise introduces the second species, where the counterpoint consists of two half-notes per measure against a whole note cantus firmus. This leads to the third species of counterpoint, where we have four quarter notes against a whole note, the fourth species which is essentially the first species with syncopation, and finally into the fifth species which is called "florid counterpoint" and is the result of the four other species combined. The treatise then adds a third voice and repeats the same process, and does it again for four voices. We now briefly summarize the rules for each species of two-voice counterpoint. Section 3 then gives the formalization of the third species.

Terminology To define his rules, Fux occasionally uses terms that are no longer used today. He uses the terms "thesis" and "arsis", the former referring to the downbeat, and the latter referring to the upbeat. In $\frac{4}{4}$ signature, they correspond to the first and third beat respectively. He also uses the term "diminution" to refer to a melodic interval of a third that is filled by a diatonic note. When using the term "perfect consonance", he refers to unisons, fifths and octaves. When he uses the term "imperfect consonance", he refers to thirds and sixths.

First species For the first species, the rules are as follows. All intervals between the counterpoint and the cantus firmus must be consonances, and imperfect consonances should be preferred ; The first and last harmonic intervals of the counterpoint must be a perfect consonance ; The key tone is tuned according to the first note of the cantus firmus ; A unison is only allowed on the first and last notes ; In the next to last measure, a major sixth or a major third must exist depending if the cantus firmus is in the lower or upper part ; The melodic intervals can not be greater than a minor sixth ; There can not be direct motion to reach a perfect consonance ; There cannot be a *battuta* octave, i.e., an octave reached by a lower voice going up and an upper voice going down more than a third skip ; Contrary motion is always preferred to oblique or parallel motions. Figure 2 shows an example of two voice first species counterpoint from Fux's treatise.

Second species The rules from the first species regarding harmony are applied on the notes on thesis, while the rules regarding melody are applied to all notes. Rules regarding motion are either modified or more complex. Additionally, the following rules are added : Notes on arsis

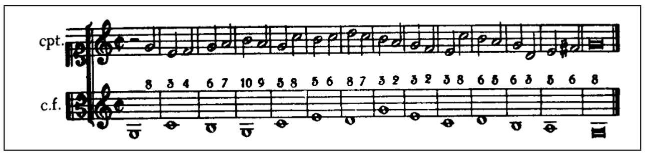

Figure 3. Example of counterpoint (second species) [14].

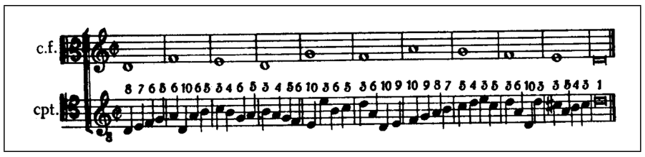

Figure 4. Example counterpoint for the third species [14].

cannot be dissonant unless there is a diminution ; In addition to the rules for the next to last note from the first species, the thesis note of the next to last measure should be an interval of a perfect fifth, and if that is not possible, an interval of a sixth must be used ; If the two voices are getting so close that there is no contrary motion possible without crossing each other, then the melodic interval of the counterpoint can be an octave leap ; Two consecutive notes cannot be the same ; If the melodic interval of the counterpoint between the thesis and the arsis is larger than a third, then the motion is perceived on the basis of the arsis note. Figure 3 shows an example of two voice second species counterpoint.

Third species For the third species, the rules are as follows. If five notes follow each other by joint degrees in the same direction, then the third note must be consonant ; If the third note of a measure is dissonant then it must be a diminution and the second and the fourth notes must be consonant ; It is best to avoid the second and third harmonies of a measure to be consonant with a one degree melodic interval between them ; In addition to the rule of the next to last note of the first species, if the cantus firmus is in the upper part, then the first note of the next to last measure should be a minor third above the cantus firmus ; Each note and its two beats further peer are preferred to be different ; The motion is perceived on the basis of the fourth note. Figure 4 shows an example of two voice third species counterpoint.

Fourth species For the fourth species, dissonance can appear in the thesis but the arsis must be consonant. Indeed this species inherits most of the rules from the first species but shifted by half a measure, corresponding to the

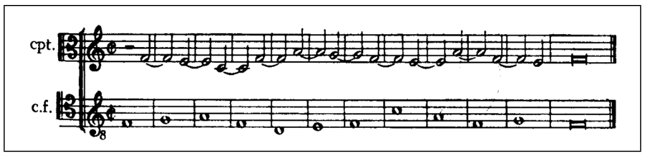

Figure 5. Example counterpoint (fourth species) [14].

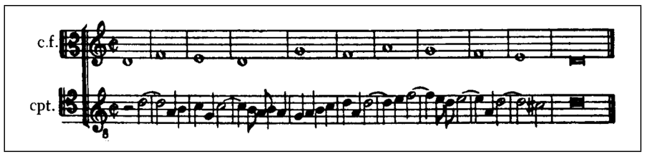

Figure 6. Example counterpoint for the fifth species [14].

Interval	Unison/Octave	Second		Third		Fourth
Type	Perfect	Minor	Major	Minor	Major	Perfect
Value	0	1	2	3	4	5

Interval	Tritone	Fifth	Sixth		Seventh	
Type	♯4 / ♭5	Perfect	Minor	Major	Minor	Major
Value	6	7	8	9	10	11

Table 1. MIDI values of the intervals over an octave range.

delay resulting from syncopation. The rules for this species are as follows. Arsis harmonies must be consonant; Dissonant harmonies must be followed by the next lower consonant harmony; If the cantus firmus is in the lower part then no second harmony can be preceded by a unison/octave harmony; If the cantus firmus is in the upper part, then no harmonic seventh interval can occur; For the rule of species 1 about the next to last measure to be satisfied, if the cantus firmus is in the lower part, then the harmonic interval of the thesis note must be a seventh, and if the cantus firmus is in the upper part, then the harmonic interval of the thesis note must be a second; Thesis half notes should be the same as the preceding arsis half note and be linked to it; Each arsis note and its two measures further peer are preferred to be different. Figure 5 shows an example of two voice fourth species counterpoint.

Fifth species The fifth species is the result of the combination of the previous four species and more particularly species 3 and 4. It can be seen as small pieces of these species combined into a single counterpoint. It is then no longer a question of new rules but rather of the previous rules so that they can all merge and apply to the notes depending on the species from which they come. Figure 6 shows an example of two voice fifth species counterpoint.

3. FORMALIZATION OF THE THIRD SPECIES FOR TWO-VOICE COUNTERPOINT

Presenting the formalization for all species would be too long for this paper. We choose to detail the formalization of the third species because it is more interesting than the others and in addition it reuses rules from the first species. We exclude the fifth species since it is essentially just the union of all the rules of the other species. The complete formalization will be available in June 2023 as a master's thesis [27]. To make it easier to represent intervals mathematically, we will be using MIDI notation. Table 1 presents the main intervals together with their MIDI value.

It is important to realize that Fux's treatise is a natural-language text that does not always present rules in a way that is easy to transform into mathematical expressions. Indeed, some rules are considered as obvious by Fux and are not mentioned explicitly but all the examples present in his treatise seem to follow them. Some rules deal with composer preferences, which requires a cost function in the formalization. Assigning values to those costs can alter the accuracy of the representation of Fux's ideas, especially because preferences are subjective, which is why in the implementation discussed in section 4 the costs are adjustable by the user and not considered to be fixed. Finally, some rules differ depending on the edition and translation of the treatise. For these rules, we cross-referenced three different versions (French [12], English [14], and Latin [13]) and used a critical analysis of Fux's examples to determine the most appropriate formulation. For example, rule 9 in section 3.3 varies depending on the translation.

In order to express the musical rules mathematically as clearly as possible, we first define some common constants and variables.

3.1. Constants

m is the number of notes in the cantus firmus. Since the cantus firmus consists exclusively of whole notes, it is also the number of measures of the cantus firmus and by extension of the counterpoint.

$$m \in \mathbb{N} \tag{1}$$

n is the number of notes in the counterpoint.

$$\begin{gathered} n \in \mathbb{N} \\ n = 4 * (m-1) + 1 \end{gathered} \tag{2}$$

Since the counterpoint in the last measure is a whole note.

Cons$_p$ is the set of perfect consonances in the form of MIDI values representing the interval between two notes.

$$\mathit{Cons}_p = \{0, 7\} \tag{3}$$

Cons$_{imp}$ is the set of imperfect consonances, also in the form of MIDI values representing the interval between two notes.

$$\mathit{Cons}_{imp} = \{3, 4, 8, 9\} \tag{4}$$

Cons is the set of all consonances.

$$\mathit{Cons} = \mathit{Cons}_p \cup \mathit{Cons}_{imp} \tag{5}$$

Cf is an array of size *m* representing the cantus firmus.

$$\mathit{Cf} = (a_1, a_2, \ldots, a_m) \tag{6}$$

For all notes a_i we have $a_i \in \{0, 1, \ldots, 127\}$. a_i corresponds to the MIDI value of note i of the cantus firmus.

j is the index of a measure in the cantus firmus.

$$j \in [1, m] \tag{7}$$

l is an index excluding the last measure of the cantus firmus.

$$l \in [1, m-1] \tag{8}$$

3.2. Variables

Cp is an array of size n representing the notes of the counterpoint.

$$Cp = (b_1, b_2, \ldots, b_n) \tag{9}$$

For all notes b_i we have $b_i \in \{0, 1, \ldots, 127\}$. b_i corresponds to the MIDI value of note i of the counterpoint. It is possible to access the values of the counterpoint on the basis of the index of the measure. For example, for measure j, the notes of the counterpoint are

$$\begin{gathered}\{Cp[4*(j-1)+1], Cp[4*(j-1)+2],\\ Cp[4*(j-1)+3], Cp[4*(j-1)+4]\}\end{gathered} \tag{10}$$

H is a $m * 4$ array where m is the number of measures and the number of notes in the cantus firmus. It represents the harmonic interval between the counterpoint and the cantus firmus in absolute value.

$$\forall i \in \{1,2,3,4\} \quad H[j,i] = |Cp[4*(j-1)+i] - Cf[j]| \tag{11}$$

M is an array of size *n-1* representing the melodic interval between notes of the counterpoint.

$$\forall i \in [1, n-1] \quad M[i] = |Cp[i+1] - Cp[i]| \tag{12}$$

P is an array of size *m-1* representing the relative motion of the counterpoint and the cantus firmus between two consecutive measures.

$$P = (c_1, c_2, \ldots, c_{m-1}) \tag{13}$$

For all elements c_i, we have $c_i \in \{0, 1, 2\}$ where a value of *0* represents contrary motion, a value of *1* represents oblique motion and a value of *2* represents parallel motion.

3.3. Rules

It is important to mention that some rules are considered as obvious by Fux and are not mentioned explicitly in his treatise. Nevertheless, they are important and will be marked with a star * in the rest of this section. The following rules are valid for all species :

- The number of measures of the counterpoint is the same as for the cantus firmus*.
- The counterpoint must have the same time signature and tempo as the cantus firmus. The time signature must be $\frac{4}{4}$*.
- The counterpoint must be in the same key and mode as the cantus firmus*. Fux determines the key with the first note of the cantus firmus, and the mode is such that there are no accidentals. However, depending on the key, some notes can be borrowed from the major or minor mode of that key for the counterpoint.
- Chromatic melodies are forbidden* :

$$\begin{gathered}\forall k \in [1, n-2]\\ \neg(M[k] = 1 \land M[k+1] = 1 \land Cp[i] \neq Cp[i+2])\end{gathered} \tag{14}$$

Rules inherited from the first species We now give the rules specific to the third species. The first eight rules are inherited from the first species and are only applied to the first note of the measures, except for the melodic rules (rule 6 and preference 2) that are applied to all notes.

Rule 1 All harmonic intervals must be consonances.

$$H[j,1] \in Cons \tag{15}$$

Rule 2 The first and last harmonic intervals of the counterpoint must be a perfect consonance.

$$H[1,1] \in Cons_p \land H[m,1] \in Cons_p \tag{16}$$

Rule 3 The key tone is tuned according to the first note of the cantus firmus such that the first and last notes in the bass must be the tonic.

$$\begin{aligned}Cp[1] < Cf[1] &\implies H[1,1] \mod 12 = 0\\ Cp[m] < Cf[m] &\implies H[m,1] \mod 12 = 0\end{aligned} \tag{17}$$

Rule 4 The counterpoint and the cantus firmus cannot play the same note at the same time except in the first and last measure.

$$\forall i \in [2, m-1], \forall k \in [1,4] \quad H[i,k] \neq 0 \tag{18}$$

Rule 5 If the cantus firmus is in the lower part, the harmonic interval of the penultimate note must be a major sixth ; if the cantus firmus is in the upper part, the harmonic interval of the penultimate note must be a minor third.

$$\begin{aligned}Cp[4*(m-1)] > Cf[m-1] &\implies H[m-1,4] = 9\\ Cp[4*(m-1)] < Cf[m-1] &\implies H[m-1,4] = 3\end{aligned} \tag{19}$$

Rule 6 Melodic intervals cannot exceed a minor sixth interval unless they are an octave.

$$\forall i \in [1, n-1] \quad M[i] \notin \{9, 10, 11\} \tag{20}$$

Rule 7 Perfect consonances cannot be reached by direct motion.

$$H[l+1,1] \in Cons_p \implies P[l] \neq 2 \tag{21}$$

Rule 8 In the start of any measure, an octave cannot be reached by the lower voice going up and the upper voice going down more than a third skip*.

$$\begin{aligned} \mathit{UpperFirst} &:= \mathit{max}(\mathit{Cp}[4*(l-1)+4], \mathit{Cf}[l]) \\ \mathit{LowerFirst} &:= \mathit{min}(\mathit{Cp}[4*(l-1)+4], \mathit{Cf}[l]) \\ \mathit{UpperSecond} &:= \mathit{max}(\mathit{Cp}[4*l+1], \mathit{Cf}[l+1]) \\ \mathit{LowerSecond} &:= \mathit{min}(\mathit{Cp}[4*l+1], \mathit{Cf}[l+1]) \\ \mathit{UpperSecond} - \mathit{UpperFirst} &\geq 4 \\ \wedge \mathit{LowerFirst} - \mathit{LowerSecond} &\geq 4 \\ \implies H[l+1,1] &\neq 12 \end{aligned} \tag{22}$$

This equation seems quite complicated but is actually straightforward. The first four lines of the equation aim to identify the upper and lower notes of the voices for the end of the previous measure and the beginning of the current measure. The remaining two lines say that if, between the fourth beat of measure *l* and the first beat of measure *l+1*, the upper voice goes up by a major third or more and the lower voice goes down a major third or more, the harmonic interval between the cantus firmus and the counterpoint on the first beat of measure *l+1* can not be an octave.

Rules specific to the third species The following rules are specific to the third species. Some rules for this species are subject to interpretation. For example, some rules are stated differently depending on the language the treatise is translated in. When there is room for interpretation, our decision will be clearly explained.

Rule 9 If five notes follow each other by joint degrees in the same direction, then the harmonic interval of the third note must be consonant.

$$\begin{aligned} &\forall k \in [1, n-4] \\ \mathit{beat} &:= ((k+2)+3) \mod 4+1 \\ \mathit{measure} &:= ((k+2)-1) \div 4+1 \\ \mathit{isJoint}(&\mathit{Cp}[k], \mathit{Cp}[k+1], \mathit{Cp}[k+2], \\ &\mathit{Cp}[k+3], \mathit{Cp}[k+4]) \\ &\implies H[\mathit{beat}, \mathit{measure}] \in \mathit{Cons} \end{aligned} \tag{23}$$

Beat and measure are just mathematical expressions to get the beat and measure corresponding to the note *k+2* of the counterpoint. The function *isJoint* has been introduced to make the equation easier to read. This function returns true if the difference between adjacent arguments is smaller than or equal to 2 and if they are all in increasing or decreasing order. This rule differed among the three translations [14, 13, 12], so we used common sense and logical deduction to interpret it as best we could.

Rule 10 If the third note of a measure is dissonant then it must be a diminution and the second and the fourth notes must be consonant.

$$\begin{aligned} H[j,3] \notin \mathit{Cons} &\implies \\ H[j,2] \in \mathit{Cons} &\wedge H[j,4] \in \mathit{Cons} \end{aligned} \tag{24}$$

Interval	Unison/Octave	Second		Third		Fourth
Type	Perfect	Minor	Major	Minor	Major	Perfect
Value	1	0	0	1	1	2

Interval	Triton	Fifth	Sixth		Seventh	
Type	♯4 / ♭5	Perfect	Minor	Major	Minor	Major
Value	forbidden	2	2	forbidden	forbidden	forbidden

Table 2. Cost values for melodic intervals

$$\begin{aligned} H[j,3] \notin \mathit{Cons} &\implies \\ |\mathit{Cp}[4*(j-1)+4] - \mathit{Cp}[4*(j-1)+3]| &\leq 2 \end{aligned} \tag{25}$$

$$\begin{aligned} H[j,3] \notin \mathit{Cons} &\implies \\ |\mathit{Cp}[4*(j-1)+3] - \mathit{Cp}[4*(j-1)+2]| &\leq 2 \end{aligned} \tag{26}$$

$$\begin{aligned} &H[j,3] \notin \mathit{Cons} \implies \\ \mathit{Cp}[4*(j-1)+4] &> \mathit{Cp}[4*(j-1)+3] \\ &> \mathit{Cp}[4*(j-1)+2] \\ \vee \mathit{Cp}[4*(j-1)+4] &< \mathit{Cp}[4*(j-1)+3] \\ &< \mathit{Cp}[4*(j-1)+2] \end{aligned} \tag{27}$$

The first equation ensures that the first and third note are consonant, the second and third equations ensure that the intervals are seconds, and the fourth equation ensures that the intervals are in the same direction. The second and third equation together make sure that it is a diminution.

Rule 11 In the penultimate measure, if the cantus firmus is in the upper part, then the harmonic interval of the first note should be a minor third.

$$\mathit{Cp}[4*(m-2)+1] > \mathit{Cf}[m-1] \implies H[m-1,1] = 3 \tag{28}$$

3.4. Preference modelled with costs

In addition to the rules presented previously, Fux gives a series of preferences both explicitly and implicitly. We model preferences mathematically using a cost function. This section gives the mathematical expressions for the preferences and then gives a value for the cost function to evaluate each preference. The solver then minimizes the sum of the costs. Since Fux does not give values for the preferences and does not provide a hierarchy between them, the values presented here are an attempt at modelling Fux's preferences as closely as possible. However, Fux often decides to break the stated preferences to create more original counterpoints.

Preference 1 Imperfect consonances are preferred to perfect consonances for the first note of each measure.

$$\begin{aligned} \mathit{cost}(H[j,1] \in \mathit{Cons}_{imp}) &= 0 \\ \mathit{cost}(H[j,1] \in \mathit{Cons}_{p}) &= 1 \end{aligned} \tag{29}$$

Preference 2 Tritone intervals are forbidden.

$$\forall i \in [1, n-1] \quad \mathit{cost}(M[i] = 6) = \infty \tag{30}$$

Preference 3 Contrary motions are preferred to other types of motion.

$$\begin{aligned} cost(P[j] = 0) = 0 \\ cost(P[j] = 1) = 1 \\ cost(P[j] = 2) = 2 \end{aligned} \tag{31}$$

As a reminder, $P[j] = 0$ means the motion of the voices getting to measure j is a contrary motion, $P[j] = 1$ means it is oblique motion and $P[j] = 2$ means it is parallel motion. It is important to note that Fux does not explicitly make the distinction between oblique and parallel motion, in fact this omission is one of the main criticisms of his work. We decided to make such a distinction because it is generally observed in other counterpoint treatises, and because these costs are adjustable so the distinction can be ignored if desired simply by giving oblique and parallel motion the same cost.

Preference 4 The second and third harmonies of a measure are preferred to not be consonant with a one degree melodic interval between them*.

$$\begin{aligned} cost(H[j,2] \in Cons \wedge H[j,3] \in Cons \\ \wedge |M[4*(j-1)+2]| < 2) = 2*m \end{aligned} \tag{32}$$

Again, the cost is high to represent a "last resort".

Preference 5 Each note and its two beats further peer are preferred to be different*.

$$\begin{aligned} \forall j \in [1, n-2] \\ cost(Cp[j] = Cp[j+2]) = 1 \end{aligned} \tag{33}$$

Preference 6 Melodic intervals are preferred to be small. This preference is not clearly explained, so the way we interpreted it was to give a lower cost to smaller intervals (i.e. seconds and thirds) compared to the other intervals. Table 2 gives the preference we assigned for each interval based on the example that Fux provides.

4. IMPLEMENTATION AND EVALUATION

4.1. Implementation

The mathematical formalization detailed in section 3 as well as the other species for two voice counterpoint were implemented using the Gecode constraint solver [10] so that we could compare the results of our mathematical model with the examples given by Fux in his treatise. This implementation was done inside of OpenMusic [20] in the form of a library. The final version of the software will support a simple user interface to avoid the need to write Common Lisp code. This library can be found here : `github.com/sprockeelsd/Melodizer`. This code is available under the free license GPL-3.0. At the moment, counterpoint can only be generated based on $\frac{4}{4}$ time signature cantus firmi.

To use our implementation, one has to download Gecode[10], GiL [11] and OpenMusic [20]. All details regarding installation are on the github page. To use the mathematical functions representing Fux's counterpoint, one has to create a Voice object in OpenMusic to represent the cantus firmus. Then, a fux-cp function must be called with this cantus firmus as a first attribute and the number representing the species as a second optional attribute. The default value will generate a counterpoint of the first species. Evaluating the fux-cp function will only create the problem, so in order to search for solution one has to use the search-next-fux-cp function after locking the evaluation (press b) of the fux-cp function.

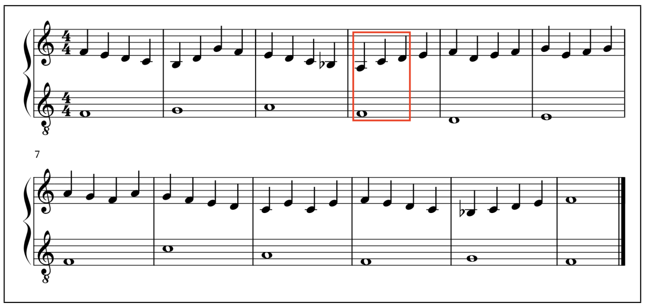

Figure 7. Cantus firmus and corresponding counterpoint as presented by Fux.

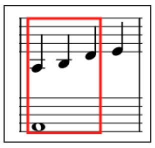

Figure 8. Measure 4 of the counterpoint from figure 7 respecting preference 4.

4.2. Comparison with Fux

In this section we compare the results produced by our tool to the counterpoints that Fux himself presents in his treatise for the same cantus firmi. We only present the counterpoints for one cantus firmus in this paper since they all lead to the same conclusions. The cantus firmus we chose is in the "mode of Fa" and is twelve measures long. Figure 7 shows this cantus firmus together with the counterpoint that Fux gives in his treatise. In order to make sure our solver indeed considers this counterpoint as a solution, a constraint saying that the solution must be equal to Fux's counterpoint was added to the set of constraints detailed in section 3. For all cantus firmi we tested, the solver could indeed find Fux's solution.

However, the solution given by Fux is usually far from the first solution suggested by our solver. Indeed, for the cantus firmus in figure 7, the solver finds that this counterpoint has a cost of 56 using the values described in section 3. Figure 9 presents the first solution found by the

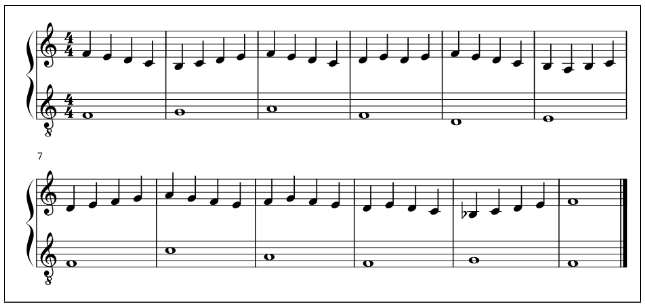
Figure 9. Cantus firmus from Fux's treatise and corresponding counterpoint generated by our solver.

solver [1]. It is the one with the lowest cost (though other solutions might have an equivalent cost) and has a cost of 24. This can be explained by the fact that Fux establishes preferences but does not seem to follow them closely. Preference 4 is a good example. In measure 4, the harmonic interval goes from a third to a fifth to a sixth (highlighted in red in figure 7), when according to his preference the skip of a third should have been between the second and third note of the measure, hence the harmonic interval should have moved from a third to a fourth and then to a sixth. Figure 8 shows this measure when the intervals respect preference 4. In his treatise, Fux does not provide a hierarchy between the different preferences that he mentions, and neither does he provide cost values for them as his treatise does not aim to produce a mathematical formalization. The costs we decided to assign to different rules were therefore done in a best effort approach, resulting in a high cost for this specific scenario.

For the solution generated by our solver, the main factor increasing the cost is due to preference 5 being "broken" multiple times. As we do not provide a hierarchy between the different costs, since Fux himself doesn't provide one, we do not have control on what rules are prioritized over others. A solution to this problem we implemented is that all costs are adjustable by the user. This is in our opinion the best approach, as Fux himself changes his preferences for each cantus firmus.

A final observation is that the last measure of the counterpoint the solver generates is often very similar to the one from Fux. That is because the penultimate measure has strict rules, meaning that fewer possibilities remain.

4.3. Examples of new compositions using the default costs inferred from Fux's examples

Let us now use our solver to generate counterpoints to a brand new cantus firmus that we create ourselves using the costs inferred from the examples. In this section, we present the first counterpoint generated by the solver, hence the one with the smallest cost. Figure 10 starts with a cantus firmus in the mode of G to generate

1 . This example can be listened to here : `https://www.youtube.com/watch?v=nDt060Os3VI`

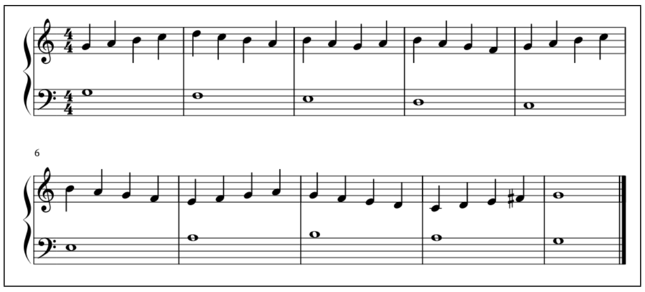
Figure 10. Counterpoint generated based on a cantus firmus in the mode of G (not from Fux's treatise).

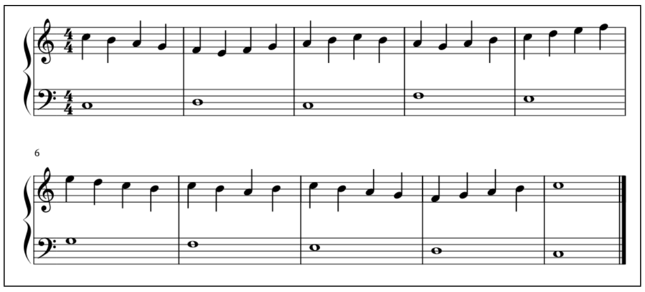
Figure 11. Counterpoint generated based on a cantus firmus in the mode of C (not from Fux's treatise).

a counterpoint [2]. This counterpoint has a cost of 20 using the values presented in section 3. We can see that a lower cost tends to limit the originality in the counterpoint, since it tries to respect the preference rules as closely as possible. Indeed, all melodic intervals in figure 10 are seconds resulting in an exclusively stepwise melodic progression, which technically respects Fux's preferences but one could argue that the melody lacks character. Similarly, the notes tend to stay in the typical mode of G and no accidentals are used except for the penultimate measure where the seventh of the scale of G major is present due to the constraints on that specific measure.

Figure 11 shows another counterpoint generated based on a cantus firmus we wrote ourselves [3]. The cost for this one is even lower, and is equal to 14. We can see very clearly how melodies respecting the preferences as much as possible tend to have similar traits and can be seen as unoriginal. Two solutions are offered to counter this problem : the first is that costs are adjustable by the user, meaning that one can influence how important each customizable aspect of the melody are. The second, which is still a work in progress, is to use a branch-and-bound solver to allow for adaptation for the search of solutions. A branch-and-bound solver allows to add an additional rule every time a solution is found. Thus, if the user is not satisfied by the solution the solver offers, in the future they will be able to select a degree of difference that the next solution will have with respect to the previous solution.

2 . This example can be listened to here : `https://www.youtube.com/watch?v=zB69kJr6oA0`

3 . This example can be listened to here : `https://www.youtube.com/watch?v=Jjya5IpCnTw`

Parameter	step-cost	third-cost	fifth-cost	min-skip-percentage
Value	0	0	1	0.3

Table 3. Modified parameters used to generate the counterpoint in figure 12.

Parameter	step-cost	third-cost	fourth-cost	fifth-cost	min-skip-percentage
Value	0	0	0	1	0.2

Table 4. Modified parameters used to generate the counterpoint in figure 13.

4.4. Experimenting with costs to generate more interesting compositions

In order for our solver to generate more interesting melodies, we implemented a couple of optional rules as well as some computational optimizations. First, we added a constraint to force a percentage of intervals to be skips, to encourage the solver to find counterpoints that step away from the original preferences. This already made the solutions suggested by the solver more interesting to us because it was no longer just conjoint intervals. We also added the rule that each disjoint interval must be followed by a conjoint interval in the opposite direction. This rule comes from Gallon and Bitsch [18] and not from Fux's treatise, but we thought it would make sense to add a rule for skips since we were allowing for more of them. This percentage can be adjusted by the user. Finally, we allowed the user to select a cost for each interval to guide the solver towards what they want. Adding those rules, especially the first one, increased the time for the solver to find a solution since it was going against its heuristic. The solver aims to minimize the costs as close as possible to Fux's preference, and using more skips increases the cost.

To remedy this problem, we decided to compute the minimal cost that the forced skips were inducing. This allows the solver to avoid searching for solutions with a cost that is lower than this minimal cost. As a result, the solver can find solutions relatively quickly for a reasonable number of skips (around 25% of skips). Figure 12 gives the solution for the cantus firmus taken from Fux that figure 9 shows [4]. Table 3 gives the modified costs and parameters we used to produce this counterpoint. It is clearly different than the counterpoint generated without playing with the costs, using a lot more skips than the previous solution. Thirds are more present than other disjoint intervals because of the costs we assigned to each of them. Though it can be said that it does not follow Fux's preferences as closely, it is in our opinion more interesting and it still follows Fux's rules. Figure 13 shows another example of a counterpoint in the mode of A with modified costs [5]. Table 4 shows the costs used for this example. The presence of fourth intervals gives the melody yet another flavour.

4 . This example can be listened to here : `https://www.youtube.com/watch?v=SQkblMr4yNA`

5 . This example can be listened to here : `https://www.youtube.com/watch?v=96NunBGW38I`

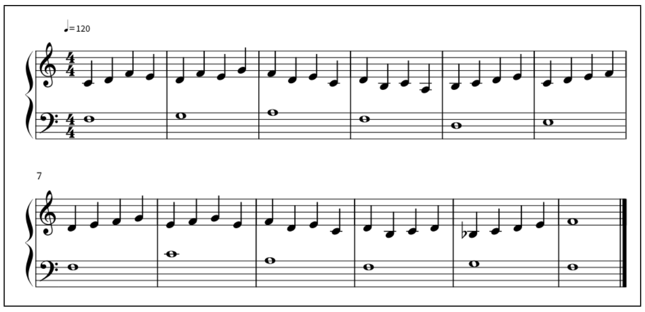

Figure 12. Counterpoint generated by the solver using the costs discussed in section 4.4 based on the cantus firmus in figures 7 and 9.

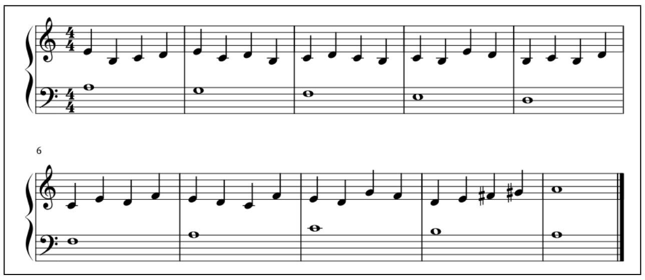

Figure 13. Counterpoint generated by the solver using the costs displayed in table 4.

5. FUTURE WORK

The ultimate goal of our project is to develop a practical tool for music composition based on constraint programming. We consider that a practical tool needs to satisfy three properties : (1) it allows a composer to express musical ideas without expertise in mathematics or programming, (2) it embodies specific knowledge about a desired musical style, and (3) it is computationally efficient for the desired compositions. The formalization and solver presented in this paper are a step towards the efficient embodiment of musical knowledge for one style. Our next steps will extend the formalization to other musical styles and will add an interface that requires only musical knowledge on the part of the composer. One concrete step that we are currently working on is to formalize the rules applicable to popular rock music in the period from the 1960s to the 1990s, based partly on the analysis done in the Ph.D. thesis of Drew Nobile [9].

6. CONCLUSION

This work presents a mathematical formalization of the Fux theory of two-voice counterpoint and an implementation of this theory as a constraint satisfaction problem. The implementation is done in the OpenMusic platform by means of an interface with the Gecode constraint solver. We have chosen the Fux theory because of its simplicity and relative importance. This work also provides a critical analysis of this implementation, by comparing its results to the counterpoints presented by Fux himself in his treatise, and by looking at counterpoints generated from brand new cantus firmi. This work is part of a long-term project in computer-aided music composition with constraint programming. The ultimate goal is to make a practical tool that can be used by composers targeting one or more important musical styles.

7. REFERENCES

[1] Andrea Agostinelli, Timo Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. "MusicLM : Generating Music From Text", *arXiv*, 2023.

[2] Baptiste Lapière. "Computer-Aided Musical Composition". Master's thesis. UCLouvain, June 2020.

[3] Chardon Clément, Diels Amaury, and Gobbi Federico. "Melodizer 2.0 : A Constraint Programming Tool For Computer-aided Musical Composition". Master's thesis. UCLouvain, June 2022.

[4] Charles Kœchlin. "Précis des règles du contrepoint". Heugel & Cie. Paris, 1926.

[5] Charlotte Truchet. "Contraintes, Recherche Locale et Composition Assistée par Ordinateur". PhD thesis. Université Pierre et Marie Curie, 2004.

[6] Damien Sprockeels. "Melodizer : A Constraint Programming Tool For Computer-aided Musical Composition". Master's thesis. UCLouvain, Jan. 2022.

[7] David Cope. "A Musical Learning Algorithm". Computer Music Journal 28(3), 2004, pages 12-27.

[8] Dorien Herremans and Kenneth Sörensen. "Composing fifth species counterpoint music with a variable neighbourhood search algorithm". Expert Systems with Applications 40(16), 2013.

[9] Drew Nobile. "A Structural Approach to the Analysis of Rock Music". PhD thesis. Graduate Faculty in Music, City University of New York, 2014.

[10] Gecode : Generic Constraint Development Environment. `https://www.gecode.org`. Visité le 19/02/2023.

[11] GiL = Gecode interface Lisp. `https://github.com/sprockeelsd/GiL`

[12] Johann Joseph Fux. "Gradus ad Parnassum". Johann Joseph Fux. French. Ed. by Gabriel Foucou. Trans. Latin by Simone Chevalier, 2019.

[13] Johann Joseph Fux. "Gradus ad Parnassum". Latin. Ed. by Johann Peter van Ghelen. 1966 New York Broude Bros reprint. Vienna, 1725.

[14] Johann Joseph Fux. "The study of counterpoint : from Johann Joseph Fux's Gradus ad Parnassum". English Trans. by Alfred Mann. Norton & Company. New York - London, 1965.

[15] Knud Jeppesen. "Counterpoint : The polyphonic style of the sixteenth century". Dover Publications, Inc. New York, 1992.

[16] LogicPro - Apple, `https://www.apple.com/fr/logic-pro/`. Visité le 19/02/2023.

[17] François Pachet and Pierre Roy. "Musical Harmonization with Constraints : A Survey". Constraints Journal, Kluwer Publisher, 6(1) :7-19, 2001.

[18] Noël Gallon and Marcel Bitsch. "Traité de contrepoint". Durand & Cie, 1964.

[19] Mikael Laurson. "PatchWork PWConstraints". IRCAM Documentation, Oct. 1996.

[20] OpenMusic - IRCAM, `http://repmus.ircam.fr/openmusic/home`. Visité le 19/02/2023.

[21] Örjan Sandred. "Constraint-Solving Systems in Music Creation", Chapter 12, Handbook of Artificial Intelligence for Music, 2021, pages 327-344.

[22] Russell Ovans and Rod Davison. "An Interactive Constraint-Based Expert Assistant for Music Composition", Ninth Canadian Conference on Artificial Intelligence, 1992.

[23] Camilo Rueda. Magnus Lindberg, Mikael Laurson, Georges Bloch, and Gérard Assayag. "Integrating Constraint Programming in Visual Musical Composition Languages". ECAI 98 Workshop on Constraints for Artistic Applications, Brighton, 1998.

[24] Bill Schottstaedt. "Automatic species counterpoint". Report No. STAN-M-19, Center for Computer Research in Music and Acoustics, May 1984.

[25] Sibelius - Avid, `https://www.avid.com/fr/sibelius`. Visité le 19/02/2023.

[26] Théodore Dubois. "Traité de contrepoint et de fugue". Heugel & Cie. Paris, 1901.

[27] Thibault Wafflard. "Formalizing Fux's Theory of Musical Counterpoint Using Constraint Programming". Master's thesis. UCLouvain, June 2023 (to appear).

[28] Torsten Anders. "Composing Music by Composing Rules : Design and Usage of a Generic Music Constraint System". PhD thesis. School of Music & Sonic Arts, Queen's University Belfast, 2007.

[29] Torsten Anders and Eduardo Miranda. "Constraint Programming Systems for Modeling Music Theories and Composition". ACM Computing Surveys 43 (Oct. 2011), 30 :1–30 :38.