To appear in *Evolutionary Programming VI: Proceedings of the Sixth Annual Conference on Evolutionary Programming*, 1997.

Musica ex Machina: Composing 16th-Century Counterpoint with Genetic Programming and Symbiosis

John Polito, Jason M. Daida, Tommaso F. Bersano-Begey

The University of Michigan
Artificial Intelligence Laboratory and Space Physics Research Laboratory
2455 Hayward Ave
Ann Arbor, Michigan 28109-2143
office (313) 647-4581; fax (313) 764-5137
jptwo@umich.edu, daida@umich.edu, tombb@engin.umich.edu

Abstract. GPmuse is software which explores one connection between computation and creativity using a symbiosis-inspired genetic programming paradigm in which distinct agents collaborate to produce 16th-century counterpoint.

1. Introduction

In a recent popular article on the science of creativity [3], Margaret Boden describes two well-known computer programs that produce music in the styles of different composers. That a computer can at least appear to be creative suggests that there may exist profound connections between computation and creativity.

Consideration of such connections involves the asking of what Boden has called "Lovelace-questions," (see [2]) one of which has bearing on this paper. This "Lovelace-question" asks whether computational ideas can help us understand how human creativity is possible. The authors of this paper answer with Boden in the affirmative.

Music of the 16th-century owes much to computation: the rules and conventions of the pre-Baroque may seem suffocatingly restrictive. However, the rules of 16th-century counterpoint helped to shape much of the "classical" music heard today. In one sense, then, we have not questioned whether computers can create but rather how rules might facilitate creativity and if it might be possible to characterize a space of music in terms of solutions to these rules.

Our investigations also touch upon two difficult issues in genetic programming. The first involves the creation of lengthy, coherent, highly-structured objects. Music can be thought of as a type of programming code: just as many 16th-century works have a clear and coherent structure, so does well-written code. While genetic programming does produce nontrivial code, it does not readily produce code that is coherently structured.

The second issue involves computational hybridization. Composition of 16th-century counterpoint involves several types of tasks, each of which suggests a distinct type of programming solution. Intuitively, one might envision a multiple-agent genetic programming kernel; however, the ramifications of introducing a level of organization above automatically defined functions (ADFs) are not well-known.

This paper introduces our current software program, GPmuse, and focuses on both the implementation of a symbiosis-inspired multiple-agent genetic programming architecture and the design of GPmuse's musical agents. The remainder of Section 1 provides

further background material. Sections 2 through 4 describe our implementation. Sections 5 and 6 feature one result from our program. Section 7 highlights a few conclusions.

1.1 Previous Work

Genetic algorithms and genetic programming have been applied to several areas of investigation in computer-generated music. A first area of investigation has explored the auralization of processes and data that focus less upon any particular musical tradition and more upon computation (e.g., L-systems). One implementation [18] has resulted in music that is reminiscent of the randomness employed by Cage [4] or the stochastic music of Xenakis [20]. At time of writing, this implementation is perhaps the only other compositional software that uses genetic programming.

Two other investigations implement genetic algorithms: one explores computer-generated music using rules and conventions from 20th-century musical traditions (e.g., Second Viennese School, jazz), while another has examined what most people consider to be Classical music (e.g., [13][14]).

Examination of the musical output of these works reveals that, where applicable, musical variations or developments are tightly linked in style to their "seed" theme or generative musical material.

1.2 Background: Music

Johann Joseph Fux's treatise *Gradus ad Parnassum* [9] has been perhaps the most widely studied text of 16th-century music theory.[3] Published in 1725, this treatise presents the study of polyphonic counterpoint, a style of "note against note" composition that reached its height of popularity in the 16th century. Polyphonic compositions of the period are of interest to musicologists in that such pieces must be examined from a horizontal perspective: a composer might have written the complete tenor line of a work, then the soprano, then the bass and finally the alto (see [1][19][21]).

Rules of counterpoint are presented by Fux in a logical order of mastery: after the reader of the treatise has presumably mastered simple rules, more complex restrictions are introduced. A hierarchy of rules can be extracted from Fux's text: some rules apply in only a handful of cases while others must be observed scrupulously.

Like most theory texts, [9] includes a large number of homework problems (with solutions) for the prospective student. To solve such a problem, one begins with a core line of music called a *cantus firmus*.[4] One then composes lines around the *cantus firmus* such that the relationships within and between the lines embody rule-specific properties.

One of the most common musical devices in sacred and secular polyphonic music of the 16th-century is imitative counterpoint. Perhaps beginning in the early 16th century in the Franco-Flemish region of Europe (see [10]), imitative counterpoint is a style of polyphonic counterpoint in which two voices containing the same musical material are separated in time and transposed such that the vertical intervals formed by the original line and its counterpart obey the rules of polyphonic counterpoint.

The set of pitches available to composers of the 16th century was smaller than the 12-note octave pitch set of the common practice period.[5] Music was based on 7-note modes rather than the keys of later musical periods. Notes with chromatic inflection (sharps or flats) only occurred in special musical situations.

1.3 Background: Symbiosis

GPmuse contains three distinct agents that use genetic programming as a generative engine. Each agent acts upon a different musical task: the polyphony agent focuses on the use of harmony in a composition, the selection agent on the generation of new *canti firmi*, and the imitation agent on maximizing the contrapuntal potential of a theme. Sections 4.1 through 4.3 further describe these agents.

Recent work (see [6][7]) outlines the use of symbiosis as a metaphor for computational hybridization. In particular, [6] suggests the addition of a layer of organization and fitness functions above automatically-defined functions (see [16]); that is, a multiple-agent genetic programming implementation in which agents are evaluated on both individual performances and joint operation on a shared "object."

Each agent generates instruction sets rather than musical output. Borrowing from Gruau's work on cellular encoding (see [11]), we generate musical output with the help of a seeded *cantus firmus*; this output is a direct manifestation of the agents' joint operation. Section 4.4 further details the interaction between agents.

2. Architecture

2.1 Individual and Subpopulation Overview

Each agent consists of one result-producing branch (RPB) and several automatically-defined functions (ADFs). An individual carries code for all three agents; that is, each individual has three RPBs and enough ADFs for all three agents. Since each agent has a unique fitness function, care must be taken that each agent is ranked solely on its performance and not that of other agents carried on the same individual.

To assure that each pool of agents evolves and is evaluated separately, three subpopulations are formed and one agent is bred in each subpopulation: the polyphony agent uses subpopulation 0; the selection agent, subpopulation 1; the imitation agent, subpopulation 2. Code for the agents not being bred in each subpopulation is ignored.

2.2 Fitness Overview

In every generation, each subpopulation's individuals are evaluated according to a fitness function unique to that subpopulation; however, only the result-producing branch (RPB) corresponding to the subpopulation number currently being evaluated is called. Each subpopulation's fitness case is a homework problem in the style of Fux's homework problems that uses a 14th-century plainchant, "Pange Lingua Gloriosi," as the *cantus firmus*. To each subpopulation's fitness case corresponds a distinct fitness function.

After all subpopulations' individuals have been evaluated, they are ordered by "homework-based" adjusted fitness. Starting with the best individual of each subpopulation, next taking the second-best of each subpopulation, etc., trios of individuals, one representing each agent, act in concert to generate a piece of music that results from the action of the agents' rule sets upon a musical embryo. Each generated piece of music is given a "piece-based" adjusted fitness by a fitness function distinct from the fitness functions of each subpopulation. Each individual of the trio of individuals takes the mean of its agent-specific, homework-based adjusted fitness and the piece-based adjusted fitness as its new adjusted fitness.

After all individuals have taken a new adjusted fitness, each subpopulation is then ranked by adjusted fitness by the GP kernel for fitness-based reproduction [15]. At the end of each generation, the new best individuals from each subpopulation work in concert upon an embryo to generate a musical score as mentioned above. This piece is ranked as before with the musical score fitness function.

3. Representation of Music

3.1 Note Representation

GPmuse represents a note as a structure. A note structure contains an integer value, `pitch`, which is mapped bijectively from the set of "white keys"; that is, pitches without accidentals. A = 440 Hz maps to 0 and the bijection continues in the obvious fashion. The second value in a note structure is a flag, `chrom`, which indicates chromatic inflection (flat, sharp, or natural). The last value in a note structure, `dur`, holds durational information.

Rests are stored as notes with pitch REST, a constant defined so that it is ignored in most computations.

3.2 Score Representation

A piece of 16th-century music for n voices[6] that is m eighth-notes long is seen by GPmuse as an ordered n-collection of m-arrays of note structures where each structure contains pitch, duration and chromatic inflection information for a single note as described in Section 3.1. Each voice in the collection is strictly ordered in time: for voice α, structure $\alpha[i-1]$ holds information for voice α's activity at eighth note i of the array. This method of storage allows for computation and retrieval at a note-by-note level; it is the musical equivalent of watching a movie frame by frame. Note structures that translate as rests are included in the length of an array.

4. Agent and Embryo Implementation

4.1 Polyphony Agent

The polyphony agent of GPmuse generates n-voice (polyphonic) counterpoint given a *cantus firmus*. The agent was designed by following the course of study outlined in Fux's text; each problem set from Fux became a set of fitness cases. The polyphony agent was first developed to follow the simplest, strongest rules of polyphonic counterpoint: those rules introduced initially with topmost hierarchical significance.

After the initial problem set was solved, this agent was tested against increasingly complex counterpoint problems, moving through problems of moderate complexity. As criteria became more exacting and problems more difficult, additional problem-specific data was made available to the polyphony agent in the form of a larger function set.

Though the initial function set could have in theory provide the polyphony agent with all data necessary to complete the initial problem set, it is crucial to note that the number of solvable problem sets and consistently well-evaluated fitness-function conditional statements grew more quickly than the cardinality of the function set.

After the polyphony agent was able to solve problems of moderate complexity, its function and terminal sets were fixed as the respective unions of all terminals and functions found sufficient to solve the tested problem sets. The "Pange Lingua Gloriosi" *cantus firmus* then became the only fitness case during the homework portion of GPmuse's operation as described previously.

The fitness function for the polyphony agent was fixed as a large subset of the conditional decompositions of the rule sets presented by Fux.

The fitness function for the individuals of subpopulation 0 recasts the rules that are to be followed in each problem set as comparative conditional functions. The raw fitness of an individual is incremented by a pre-defined constant if a conditional evaluates correctly; such evaluation occurs at each eighth note for each voice or pair of voices.[7] Hits, a parallel fitness measure, are similarly determined for a core subset of conditionals. Standardized fitness is a linear combination of hits and raw fitness; homework-based adjusted fitness is 1 / (1 + standardized fitness).

To give an example of problem formulation, one task in Fux's earliest problem set is to take a given *cantus firmus* and generate one voice above the *cantus firmus* such that each resulting vertical interval is consonant. Consonant intervals [21] are the unison, major and minor third, perfect fourth, perfect fifth, major and minor sixth and intervals that differ from those listed by one or more perfect octaves.

The terminal set is the set of all legal horizontal motions within one voice. This is the set of consonant intervals with the addition of the major and minor second.[8]

All but one function in the function set are comparative conditional operators that yield problem-specific data to the agent: for instance, whether the pitch resulting from a selected horizontal motion is higher than the simultaneously-sounding pitch of the *cantus firmus*. The remaining function adds chromatic functionality.

Actual notes are never received by the polyphony agent through its function operations or terminal selections; instead, an initial correct pitch for each voice over a given *cantus firmus* is selected at random and subsequent pitches are the result of horizontal motions selected by the RPB's evaluations. Once one voice is composed through all the places of a block, another voice is evaluated. Once a voice is fully composed, it cannot be altered by the agent.

4.2 Imitation Agent

The agent affiliated with subpopulation 2 takes the given *cantus firmus* and produces two-voice imitative counterpoint. Like the polyphony agent, the imitation agent was also initially trained against Fux's *canti firmi* but when fully developed began to use "Pange Lingua Gloriosi" as its sole fitness case.[9] The fitness function remained unaltered after the training period on the Fux *canti firmi* ended.

Unlike the task of generating polyphonic counterpoint, the imitation agent's task is largely an optimization problem: find a horizontal offset h and a vertical offset v such that there are a maximum number of consonant vertical sonorities. Figure 1 is an example with $(h, v) = (4,-3)$.

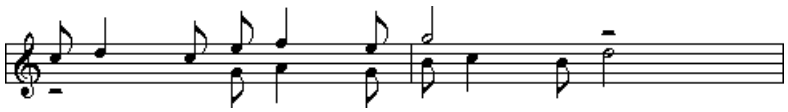

Figure 1. An example of imitation.

The homework-based raw fitness is composed of the sum of the squares of the differences between the number of elements at which a conditional is addressed correctly and total number of elements in the array. Hits is the number of elements where one high-importance vertical and horizontal motion criterion is fulfilled. Standardized fitness is set equal to raw fitness and homework-based adjusted fitness is 1 / (1 + standardized fitness).

The agent consists of an RPB and two ADFs; ADFs choose values for *h* and *v*, respectively.

The function set for the RPB holds prog2, prog3 and information about location within the array. The function sets for the ADFs consist of +, - and a function that sets the variable specific to the ADF (*h* or *v*) to its argument.

The RPB's terminal set is a set of actions that navigate within the array. The ADFs' terminal sets contain constants, user-defined parameters and array elements.

4.3 Selection Agent

The selection agent, affiliated with subpopulation 1, is related in function to the imitation agent but has a more complex task. The selection agent takes a *cantus firmus* and selects a portion of the *cantus firmus* to pass on as a new *cantus firmus* to other agents. In addition to horizontal and vertical offsets (*h* and *v*) there is a beginning element in the array, *b*, and a length-of-selection, *l*.

Characteristics of musical themes are left to a composer's individual tastes; lacking such recourse, the selection agent was trained on the assumption that portions of a *cantus firmus* that harmonized well with the whole had merit as subjects for imitative counterpoint and polyphonic composition.

Fitness, hits, standardized fitness and homework-based adjusted fitness are calculated as with the imitation agent save that any instance of dissonance results in a zero fitness.[10]

This theme-selecting agent has the most involved architecture, consisting of an RPB and four ADFs. As well, agents control two four-tuples (*h, v, b, l*): temporary variables, which are set by ADFs, and permanent variables, which are updated from temporary variables via a specific terminal.

The terminal set for the agent's RPB consists of the array-navigating actions also found in the selection agent's RPB terminal set. The ADFs' terminal sets are also similar to those of the imitation agent's ADFs, containing values of constants, user-defined parameters and array elements as well as setting temporary variable values.

The function set for the RPB of the selection agent holds conditional comparative information, `prog2`, `prog3` and variable-comparing functions. Function sets for the ADFs include +, - and variable-specific variable-setting functions.

As with all other agents, after initial testing and development of sufficient function and terminal sets, the "Pange Lingua Gloriosi" plainchant became the only fitness case.

4.4 Embryo

Gruau [11] has described a method in which genetic programming specifies the graph-constructing operations necessary in growing an embryonic neuron into a full neural network. Koza *et al.* [17] broadened Gruau's concept so that genetic programming specifies graph-constructing operations necessary in growing embryonic circuits. Works by these authors often describe a single genetic programming agent operating on an embryo represented as some type of graph. This paper describes several distinct agents that operate on an embryo not dependent on a graph representation.[11] Instead, we have treated the embryo in a manner outlined by [6] as described in Section 1.3. In particular, [6] suggests that GPmuse's agents operate on some portion of an "object" that is shared between all agents—this "object" being an embryo that maps the agents' instructions to music scores. Furthermore [6] indicates the possibility for a hierarchy of fitness functions: fitness functions that correspond to each agent may be loosely coupled to the fitness function that correspond to a music score.[12]

After trios of individuals are selected according to homework-based fitness, a piece of music unfolds by applying the instructions of the three agents in the trio in a predetermined order resulting in a specific musical form. The musical form is based upon Josquin des Pres' "Kyrie" from *Missa Pange Lingua* [8] in which he excerpted portions of the "Pange Lingua Gloriosi" for use as *canti firmi*. The actions upon the embryo proceed as follows (see Figure 2):

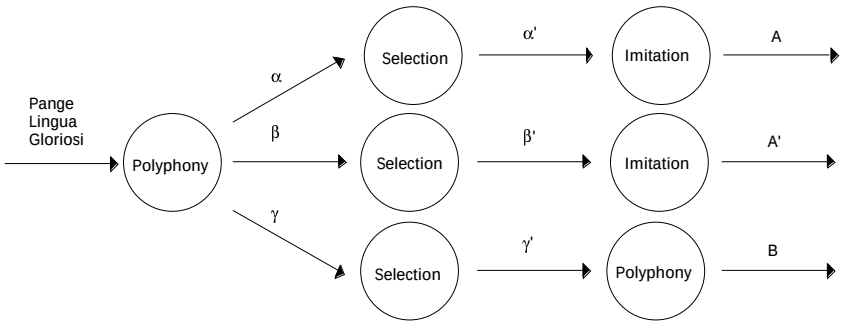

Figure 2. Process flow in creating a piece of music.

"Pange Lingua Gloriosi" is given to the polyphony agent of the trio, which creates a four-voice polyphony by composing three voices (α, β, γ) around the *cantus firmus*. Each composed line is passed to the selection agent, which selects a subset of each line as a new *cantus firmus*, totaling three new *canti firmi* (α', β', γ'): The first *cantus firmus* generated by the selection agent (α') is passed to the imitation agent, which creates a two-voice imitative counterpoint (A). Likewise, the second such *cantus firmus* (β') is treated by the imitation agent's rule set but has its output expressed in the two voices not used in the first case (A'). The last *cantus firmus* generated by the selection agent (γ')is returned to the polyphony

agent, which generates a four-voice polyphony (B) with which to close the piece .

The form roughly described by the above parameters is an A A' B form.[13] However, it is possible for any section to have a length of zero.

Pieces generated in this fashion are scored by summing the number of elements at which conditionals evaluate correctly and then dividing by the length of the entire array to generate the piece-based adjusted fitness.[14]

5. Results

All training runs as well as GPmuse runs used run parameters corresponding precisely to those found in [15]: probability of reproduction: 10%; probability of crossover: 90%; maximum depth of an individual: 6; method of generation for initial population: ramped half-and-half; maximum depth of an individual: 17; selection method: fitness.

The discussed result (Figure 3) was generated by the reranked best individuals in the 40th generation of a run of 50 generations of 500 individuals per subpopulation. The piece's adjusted fitness was a perfect 1.0.

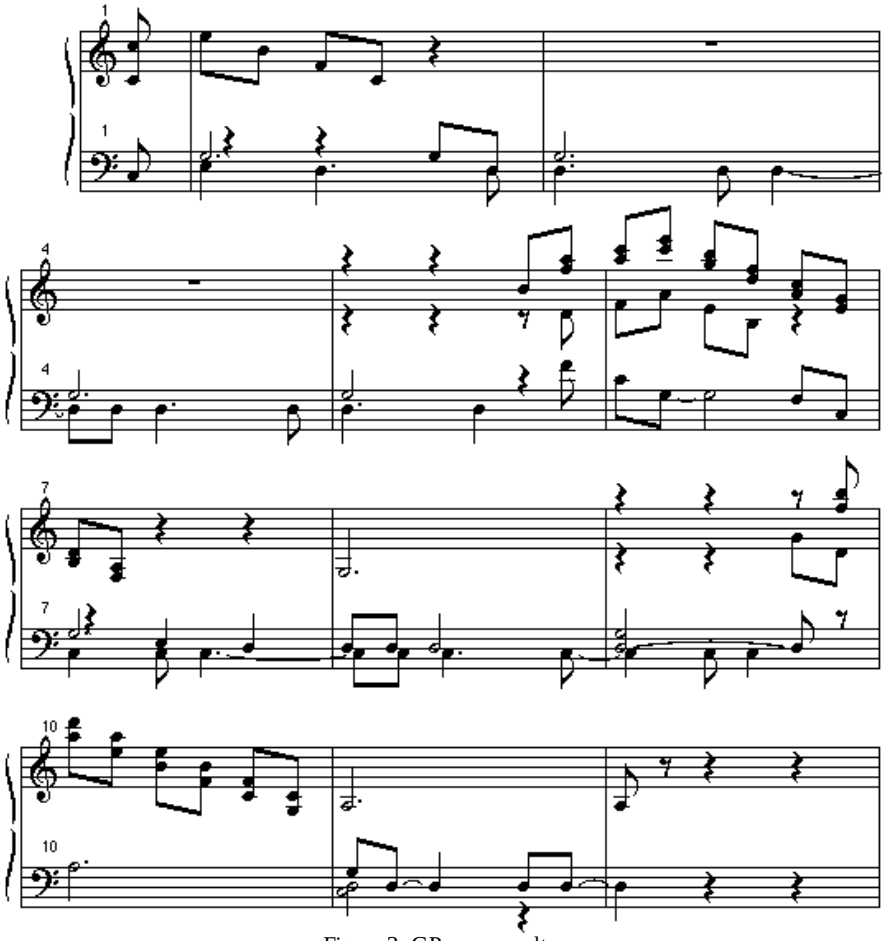

Figure 3. GPmuse result.

6. Discussion

In creating the result in Figure 4, the A and A' sections described previously were set to a length of zero. However, if the form of the generated material is analyzed in the same fashion as one would analyze the form of a Western European-derived art music piece, the generated music seems to have the form: A B A' B' A'' B'', corresponding to measures 1-2, 2-5, 5-7, 7-9, 9-11 and 11-13.

These sections can be distinguished by their rhythmic activity and their span of pitches as well as their melodic contour. The A sections[15] all contain what appears to be a melodic motive in which the highest voice rises to an E and then moves rapidly downward through a sequence of fourths. B sections are concentrated in narrow bands of low pitches; each contains the same rhythmically diverse pattern of unison pitch repetitions.

The presence of easily distinguishable thematic material in A and B sections has been a surprising result because such small forms within the larger blocks (A A' B)[16] mandated by GPmuse were neither induced by direct action of any of the three agents nor explicitly rewarded by any of GPmuse's fitness functions.

The reappearance, slightly altered, of previous musical material withstands comparison to the development of musical ideas as might be done by composers in the Western European-derived tradition. The falling fourths sequence first appears in section A as an active line above fairly static lower voices. In the A' section, it has been harmonized in full triads, creating a descending sequence of vertical sonorities. In the closing A section, A'', the repeated pitch pattern, consisting largely of fourths, is accompanied by a parallel line a fourth away.

The example is transcribed in 3/4 time with an eighth-note pick-up largely due to the near-constant presence of a pedal tone dotted-half note that is articulated at the beginning of each measure. The combination of the bell-like pedal tones on downbeats and the occasional burst of falling "chords" with a non-transposed motive yields music with an aural resemblance to the piano music of Claude Debussy, an early 20th-century Impressionist composer.

Even with strict adherence to the rules of [9], GPmuse has developed a few idiosyncrasies. Fux, for instance, never insisted that music be structured such that its performers be able to breathe, as vocal music was naturally performed by humans with finite lung capacity. GPmuse, however, had no such requirements, and on occasion has composed near-endless musical phrases that unfold for hundreds of eighth notes without pause.

There have been other idiosyncrasies: In the discussed example, GPmuse expressed the fourth voice (traditionally the bass line) with notes ranging from bass to high soprano range. Fux gives scant indication that voices must be constrained in their natural ranges; such constraints were so obvious to Fux that they were not worthy of mention. For a computer, however, such constraints must be explicitly programmed.

7. Conclusion

GPmuse has generated output that resembles the musical output of a specific niche of Western European-derived musical literature. That a group of agents working with a subset of 16th-century music theory produce music whose phenotype fits quite strongly into a particular portion of 20th-century musical literature rather than the literature of the 16th-

century has been surprising. That simple explorations of 16th-century music theory rule set space resulted in a different yet viable pre-existing musical type may support the argument that not only can rules facilitate creativity but that creativity may adhere to rules.

This paper has also described the hybrid architecture and implementation of multiple-agent GP system that was designed to produce highly-structured, coherent output (for this application, musical scores). We implemented an embryo that works with diverse GP agents; by doing so, we introduce additional organizational levels in genetic programming beyond that implied by ADFs.

Further information and a recording of the score described in this paper are available at the following URL: http://www-personal.engin.umich.edu/~daida/

Acknowledgments

We thank James Borders and Edward Parmentier for their musical expertise; Robert Bertram for implementation of a "score"; Catherine Grasso, Steven Ross, Mary Simoni and Stephen Stanhope for thought-provoking discussion; Donald Higman and William Bolcom for the impetus to begin this exploration; the reviewers for their helpful comments. We finally acknowledge the following people who have contributed to GPmuse: Amber Pewe, Chau Doan, and Joshua Bueller.

Bibliography

[1] Aron, P., 1524. *Toscanello in musica*, Book 2, Ch. 16, trans. P. Bergquist, 1970, Colorado Springs: Colorado College Music Press.

[2] Boden, M.A., 1992. *The Creative Mind: Myths and Mechanisms*, New York: BasicBooks.

[3] Boden, M.A., 1996. "Artificial genius," *Discover*, 17(10), pp.104–113.

[4] Cage, J., 1973. *Silence: Lectures and Writings*, Middletown: Wesleyan University Press.

[5] Cope, D., 1991. *Computers and Musical Style*, A-R Editions, Inc.

[6] Daida, J.M., S.J. Ross, B.C. Hannan, 1995. "Biological symbiosis as a metaphor for computational hybridization," *Proceedings of the Sixth International Conference on Genetic Algorithms*, L.J. Eshelman (ed.), San Francisco: Morgan Kaufmann Publishers, Inc., pp. 328–335.

[7] Daida, J.M., C.S. Grasso, S.A. Stanhope, S.J. Ross, 1996. "Symbionticism and complex adaptive systems I: Implications of having symbiosis occur in nature," *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, Cambridge: The MIT Press, in press.

[8] Josquin des Pres. Early 1500s. ed. 1979. London: Chester Music J. W. C.

[9] Fux, J.J., 1725. *Gradus ad Parnassum*, trans. Alfred Mann. 1971, *The Study of Counterpoint*. New York: W.W. Norton & Company, Inc.

[10] Grout, D.J., C.V. Palisca, 1996. *A History of Western Music*, 5th edition., New York: W.W. Norton & Company, Inc.

[11] Gruau, F., D. Whitley, 1993. "Adding learning to the cellular development of neural networks: Evolution and the Baldwin Effect," *Journal of Evolutionary Computation*, 1(3), pp. 213–233.

[12] Holtzman, S.R., 1994. *Digital Mantras: The Languages of Abstract and Virtual Worlds*, Cambridge: The MIT Press.

[13] Jacob, B.L., 1995. "Composing with genetic algorithms," *Proceedings of the International Computer Music Conference*.

[14] Jacob, B.L., 1996. "Algorithmic composition as a model of creativity," *Organised Sound*, 1(3).

[15] Koza, J.R., 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge: The MIT Press.

[16] Koza, J.R., 1994. *Genetic Programming II: Automatic Discovery of Reusable Programs*, Cambridge: The MIT Press.
[17] Koza, J.R., F.H. Bennett III, D. Andre, M.A. Keane, 1996. "Automatic WYWIWYG design of both the topology and component values of electrical circuits using genetic programming," *Genetic Programming 1996: Proceedings of the First Annual Conference*, J.R. Koza, D.E. Goldberg, D.B. Fogel, and R.L. Riolo (eds.), Cambridge: The MIT Press, pp. 123–131.
[18] Putnam, J., 1996. "A grammar-based genetic programming technique applied to music generation," *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, Cambridge: The MIT Press, in press.
[19] Soderlund, G.F. 1947. *Direct Approach to Counterpoint in the 16th Century Style*, New York: F. S. Crofts & Co.
[20] Xenakis, I., 1992. *Formalized music: Thought and Mathematics in Composition*, revised edition. Pendragon Press.
[21] Zarlino, G., 1558. *Le istitutioni harmoniche*, Part 3, trans. G.A. Marco and C.V. Palisca, 1968, appears in *The Art of Counterpoint*, New Haven: Yale University Press.

Endnotes

[1] The contrapuntal rules fill a small book. See [9].

[2] In the larger field of artificial intelligence and computer music, D. Cope's seminal work with EMI (Experiments in Musical Intelligence) does a noteworthy job of mimicking different styles, although it is arguable that it can create new styles. See [5] and [12].

[3] Exercises from Fux's book and discussions of his work are found in the writings of Haydn, Mozart and Beethoven, among others. See [9], p.xi-xiii.

4 In the plural, *canti firmi*.

[5] Common practice period begins roughly with the work of J. S. Bach. See [19].

[6] For this paper, *n* usually equal to 4. *n* strictly greater than 2.

[7] A conditional's constant reflects the importance attributed to it by Fux and peer theoreticians. Vertical-interval conditional criteria require pairs of voices.

[8] Zarlino [21] here is more liberal than Fux [9]. GPmuse follows Zarlino's consonance classification.

[9] When training on Fux's *canti firmi*, the agent regularly achieved optimal solutions with its best individual. In 20 runs with parameters as described in section 5, individuals converged towards optimal solutions between generations 4 and 15.

[10] When trained on the Fux *canti firmi*, the selection agent never performed at better than 75-80% of optimal performance.

[11] In the language of [6], an embryo is a shared artifact *B*.

[12] "Loosely coupled" means that the fitness pertaining to a music score does not necessarily determine the fitness of each individual in an agent's subpopulation.

[13] A A' B means that there are three sections to the piece. The first two are similar in form and/or content while the last is dissimilar.

[14] If the resulting score is greater than 1.0, it is rounded to 1.0.

[15] A sections are A, A', A". Similarly, B sections are B, B', B".

[16] Here, the A A' B refers back to the form of Josquin's "Kyrie." See [8].