

Andrew Sorensen

Queensland University of Technology &
Australasian CRC for Interaction Design
Kelvin Grove, 4059
Australia
a.sorensen@qut.edu.au

Andrew R. Brown

Queensland University of Technology &
Australasian CRC for Interaction Design
Kelvin Grove, 4059
Australia
a.brown@qut.edu.au

Abstract

In this article we report on progress at the Australian CRC for Interaction Design investigating the computational generation of orchestral music based in the Germanic Symphonic tradition. We present an introduction to the project including a brief overview of our intended methods and some guiding principles for the project. We then outline the current state of the project and introduce our initial algorithmic system with a special emphasis on an implementation of Paul Hindemith's harmonic system. We conclude with some initial findings and future goals. We provide an extensive selection of audio examples online that accompany, verify and enhance information provided in this report.

Introduction

Starting in 2008 the Australian CRC for Interaction Design began an exploration into the computational generation of orchestral music based loosely around a mid to late romantic aesthetic. There are several reasons for our current interest in computational approaches to the generation of romantic orchestral music. Firstly, there is a commercial aspect to this investigation. A significant amount of the music currently used in mainstream media is orchestral; feature films, documentaries, television dramas/sitcoms and computer games all make significant use of orchestral soundtracks. Surprisingly, perhaps, the computer plays a disproportionately small role in the creation of orchestral music beyond its role as a notation and recording device. Arguably this is due to the structural complexity of orchestral music of the mid-late romantic period and a cultural gap between electroacoustic and orchestral musical aesthetics.

Secondly, the generative agenda of the project builds on a long tradition in computer music research, but still seems to be a contentious area of investigation in the boarder music community, and with some in the computer music community. While we acknowledge the significant efforts of projects such as Cope's EMI software (Cope 1992, 2001), there remain many problems with the application of such research within a real-world context. A cursory survey of the lack of generative music processes in current professional music software is all that is required to see that this is the case. Thirdly, we have an interest in music of the Germanic Symphonic tradition that has had an influence on music for film and, more recently, computer games and the rich musicological tradition

A Computational Model For The Generation Of Orchestral Music In The Germanic Symphonic Tradition: A progress report

that surrounds it. Although our own musical practice extends well beyond this genre, there is a fundamental musical vocabulary here that we perceive as integral to our growth as computational musicians and we believe is generally applicable to many musical styles. This paper outlines some of our recent attempts to apply musical theories from the orchestral tradition to generative computational techniques that can be applicable in practical contemporary contexts such as cinema and computer games.

Background

We acknowledge that this is an ambitious project and our hopes for success are largely based on the hypothesis that many of the problems inherent in algorithmic composition are implementation issues. In other words, we do not believe that a more detailed specification of the problem is required, but instead a more practical solution for its implementation. Our hope is that we have reached a turning point where the technology is capable of adequately supporting the vast wealth of musical and computational theory available to us. In particular we hope that our use of Impromptu (Sorensen 2005), a real-time dynamic system, will facilitate our exploration of this highly temporal medium.

After many years of building computational music systems the authors are now guided by two beliefs. Firstly, the seemingly obvious belief that algorithms derived from musical analysis are far more likely to provide effective musical solutions than those from formalisms outside the music domain. Having investigated Neural Nets, Genetic Algorithms, Genetic Programming, Cellular Automata and the like (Towsey et al. 2000; Brown 2005, 2005a; Wooller et al. 2005; Gerber and Brown 2006, Brown et al. 2008) the authors have returned to a simple set of principles based around probability, linearity, periodicity, set theory and recursion (Sorensen and Brown 2007). This is not to say that there is not or will not be a role for the other processes, only that we believe sonifications resulting from them tend to produce music outside the bounds of the Romantic orchestral style that is our focus here. Our goals are first and foremost to achieve a musical outcome and although we may stumble across issues related to machine or human

creativity these are not primary goals of the project.

Secondly, and very much in line with the first statement, we are interested in holistic solutions to our problem. We question the usefulness of a reductionist approach that focuses on one musical element, e.g., pitch or rhythm, in isolation when dealing with real world compositional problems. We have found that it is only by experimenting with musical results that attempt to simultaneously address all dimensions of the musical puzzle, even if to differing degrees, that generative musical process can succeed. The musical whole is always more than the sum of the parts in our experience. We anticipate that we will be better able to address sub problems while experimenting with the whole and in this paper focus on one sub-problem, harmonic tension, by producing works that also include generative melody, rhythms, form, dynamics and so on.

Research design

At this stage of the project we have been proceeding using a pseudo-scientific method whereby various hypothesis, regarding music theories and computational implication, are tested through experiment for observable results. Indeed the datum of our project, are the musical compositions generated by our system. These data, as with all experimental research, are the core outcome of our work and the final and authoritative guide to the success of our project. We therefore provide numerous examples online to accompany this paper. Where doubt arises about information in this paper the musical examples should be considered the canonical source.

Hindemith's Harmonic System

We began by searching for a suitable harmonic system that accommodated the diatonic and chromatic content of the Germanic Symphonic music of the later half of the nineteenth century. Not only would the chosen harmonic system need to be musically applicable but also able to be implemented within the constraints of a real time system. There are many harmonic systems that could provide suitable stepping off points but based on the success of early experimentation with Hindemith's system we did not consider an exhaustive survey of alternative harmonic systems and techniques necessary before moving forward.

Paul Hindemith's "The Craft of Musical Composition - Book 1" appeared to fit our purposes (Hindemith 1970). It outlines a harmonic system based around intervallic relationships and the tensions inherent in these relationships. While certainly not the only comprehensive theory based around intervallic relationships, Hindemith's is simple, elegant and suitable for fairly direct computational implementation. Because Hindemith's tension relationships are subjective some have argued about the theoretical soundness of his choices. Also

some suggest that its reliance on harmonic relationships may only formally be applicable to music using just intonation. They argue that an equally tempered system blurs functional distinctions between major and minor triads (Clark 2007). Despite these reservations Hindemith's system presented a good starting point for our investigations because it afforded an opportunity for rapid implementation and subsequent aural experiment. Another strong validation of Hindemith's harmonic system is Hindemith himself who made extensive use of the system for his own compositions, going so far as to rewrite many of his early works so that they would conform to this system which he developed later in life.

It is important to clarify that our intentions were never to test Paul Hindemith's system as originally conceived but to use it as a starting point to further our own harmonic enquiry for generative purposes. We therefore took the liberty to add, modify and remove aspects of Hindemith's system as our test results dictated. Usually these modifications were made to support an autonomous version of the system never intended by Hindemith himself.

Hindemith's System Of Chord Qualities

Hindemith devised a system of chord qualities that grouped all possible chords into one of six chord-groups, numbered I-VI. Hindemith's system segregated chords based upon their intervallic relationships between chord tones. The set of internal intervals identify a chord-group to which each chord is assigned. Chords in a particular group are assumed to share a similar pureness, stability or harmonic tension. The six groups are defined as follows:

- i. No tritone, no seconds, no sevenths
- ii. No minor seconds, no major 7ths, with tritone
- iii. No tritones
- iv. Contains one or more tritones
- v. Indeterminate
- vi. Indeterminate (with tritone predominating)

An example may help to clarify how this works in practice. Consider the dominant 7th chord C7 [C E G Bb] containing the following set of intervals [M3 P5 M6 m6 m7 m3 M2 tritone]. This interval set places C7 in chord-group II. Cmaj7 [C E G B] containing the intervals [M3 M6 m6 P5 m7 m3 m2] would be placed in chord-group III. Hindemith makes additional sub-segregations based on root position but we ignore this addition in our current work.

Hindemith also provided a guide for the creation of root progressions also based on intervallic relationships. Hindemith defined two interval series (series I and II) that he uses for the creation of root progressions and other melodic materials. We do not provide further explanation here as our pre-

sent system makes only indirect use of these series for the creation of root progressions.

In a simple sense, Hindemith's theory progresses by assigning chord quality to root notes where the choice of chord quality is based upon the amount of harmonic tension required at any given point in the composition. All changes in tension or resolution are relative to the current chord quality, rather than measured against an absolute scale. It should be pointed out that Hindemith never intended for the system to be autonomous and as such it provides a general, but not absolute, guide for us to follow.

Investigation and Modification

Our initial investigation of Hindemith's system indicated that it would provide musical benefits including; chromatic flexibility, relatively direct control of musical tension, the ability to cleanly separate chord quality from root progression and the trivial variation of existing root progressions. Some of the technical advantages of the system are its suitability for real time work because it works within limited temporal restraints (i.e., There is no need to look ahead or behind) and because of the efficiency of the algorithm.

During the development cycle there have been a number of technical additions that provide constraints over Hindemith's system. Firstly, there is a Pitch Class Set (PCS) parameter that offers the opportunity to help constrain chordal choice. For example, Hindemith's system places no constraints over the choice of major or minor chord qualities, leaving this instead as a subjective choice for the composer. For example, both Cmaj [C E G] and Cmin [C Eb G] are contained within chord-group I. Our implementation uses the PCS parameter as a means for weighting these types of choices as a way of maintaining "obvious" constraints that Hindemith left to composers. Additionally, the current implementation of the algorithm only makes use of Hindemith's first three chord groups (I-III) leaving out the indeterminate choices (V-VI) and the multiple tritones subordinate group (IV). This was an early choice to impose limits on the range of harmonic variety to a practical, as opposed to theoretically complete, range. To date, our limited subset has proven to provide adequate variation. We may choose to add the further three groups at a later stage or for use in different musical contexts. Additionally, we do not currently implement inversions as distinct chordal subgroups, as Hindemith does. This allows us to more simply manage the construction of musically meaningful root movement and voice leading.

Before beginning a detailed investigation of the algorithmic implementation we will describe in brief the overall design of the music production system.

System Overview

The generative process begins with harmonic selection. Even though there is no requirement to begin this way, we feel that this is reasonable given the dominance of harmonic considerations in late romantic music and the emphasis on thematic rather than more traditionally melodic movement.

Our harmonic selection process begins by generating a root progression including both pitch and rhythm dimensions. The root progression is generated to fit a given time period; eight bars for example. The algorithm then progresses linearly through the root progression. For each root note several activities take place. Firstly, a chord quality is selected for the given root. The harmonic tension of this chord quality is substantially influenced by the Hindemith chord group value (I, II, or III) passed to the chord quality generator. Once the chord quality is selected a score is constructed based on current settings for orchestration, pitch range, harmonic accompaniment and voice leading. Finally, a bass line is added and thematic material is selected and manipulated to fit the current harmonic context.

Generative processes for aspects other than harmonic considerations are relatively rudimentary placeholders. We have already started to reap benefits from the approach of implementing many aspects of the system in parallel, even if not with equal sophistication. One interesting result of this experience is just how much is possible with a very superficial implementation provided that coverage is broad. We will now discuss each aspect of the overall system in more detail.

Rhythm Generator

As the rhythm generator is discreet and is used by many parts of the system we will begin by describing its implementation. The rhythm generator is a simple stochastic function providing control over duration, tactus, level of syncopation and rhythmic value list (herein called the RVL parameter). The generator initially selects a rhythm value at random from a user provided RVL. The generator will continue to select rhythm values at random from the RVL until either (a) the maximum duration is reached or (b) the maximum duration is surpassed at which point the algorithm will backtrack to a point at which it can successfully continue forward.

Implementing gestalt laws of proximity and similarity that evaluate the results of randomly chosen rhythm patterns provides the musical effectiveness of this algorithm. The weighting of this gestalt selection is made through a combination of the tactus and syncopation parameters. At a tactus point a stochastic selection will always be made, at all other times a percentage choice, the syncopation value, will determine a gestalt or stochastic choice. Random reselection is forced until patterns pass evaluation. This simple algorithm has proven very effective at handling all of our existing rhythmic generation requirements.

Root Selection

The root selection algorithm currently operates using a relatively simple point-to-point style approach. There is no explicit cadential knowledge built into the root selection algorithm. The first part of the root selection process defines a harmonic rhythm selected using the stochastic rhythm generator described above. The point-to-point process begins with parameters for a starting Pitch Class (PC), ending PC and interval value list (IVL). The algorithm selects N-2 (i.e., minus starting PC and ending PC) values from the IVL that when combined move from starting PC to ending PC.

Cadential knowledge can be added by setting an appropriate PC ending sequence. A half cadence for example requires no modification while a perfect cadence is trivially implemented by appending a resolving I chord. This simple process provides the basis for implementing all major cadential forms. The critical roll that bass movement plays in cadential movement is acknowledged and is discussed later.

The harmonic complexity of the root progression is controlled by the contents of the IVL, however two further Pitch Class Set (PCS) quantization methods are available. Firstly, the output of the root progression generation can be quantized against a PCS. Secondly, the IVL can be implemented as a step value list (SVL) that is applied against a PCS.

In conclusion, our control over the root selection process is fairly fine grained. We can choose to quantize to a PCS or not, we can choose to work within a traditional system of cadences or not and we can control the degree of harmonic complexity in our root progression by adjusting the values in the IVL.

Chord Quality

A chord quality is defined for each root in the root progression as it is played. The chord quality is selected using Hindemith's chord quality system. Our initial implementation of Hindemith's algorithm used a simple iterative stochastic process. A random selection of PC's is made and the complete set of intervals joining these PC's is calculated. A random selection of the PCS [C E G] would produce the interval list [M3 m3 P5 M6 m6]. This set of intervals is then tested against a user specified Hindemith chord group (i.e. I, II or III) for validation. Successful validation of the random selection results in it's being returned to the user as appropriate. At this stage it is important to realize that the chord is a set of interval relationships free of any definite pitch classes. As a final step in the process the user provides a root for the chord and an appropriate PCS is generated. The PCS does not designate an inversion nor octave displacement; rather, it is an unordered set of pitch classes.

As an example let us consider a group I

chord that can be either a major or minor triad. If the user asks for a group I chord with root 2 then the algorithm will probabilistically return either a Dmaj or Dmin chord with no extensions. Chord groups II and III provide a far greater range of chord qualities and the distribution of type I, II and III chord qualities is, along with the root progression itself, the greatest source of control over harmonic tension.

Bass Note

After assigning a chord quality the next stage of the process assigns one or more bass tones for the current chord duration (the bass may move). For cadential purposes the generator attempts to accentuate the current musical key while avoiding strong cadential reference. Our initial implementation achieves this by using a simple weighting to reject root and bass correlation (i.e., we use inversions instead of root positioning) away from phrase boundaries. At this stage we freely invert using any option available within the context of the current chord quality. The generator also uses root position chords where possible, and attempts to maintain minimal linear voice movement of the bass.

Scoring

The next phase in our process involves the voicing and orchestration of the chord. We now have a fairly complete harmonic picture; we know the root, quality and bass of the chord. Using this information we proceed to orchestrate the chord using a further range of system parameters including the current instrumentation, number of active voices, and lower and upper pitch bounds. Additionally the scoring algorithm uses the previously scored harmonic block as a reference point for smooth linear part movement.

Our linear part movement algorithm follows a simple point-to-point approach searching for the shortest path between the two chords while maintaining complete chord coverage (i.e., making sure each chord tone is represented). Finally, the algorithm makes decisions about the style of accompaniment to apply. At present we have only implemented two accompaniment styles, an arpeggio style and a homophonic style. A choice about whether or not to perform the bass part is also made at this stage and this decision depends upon the instrumentation, range and accompaniment style chosen.

Thematic Material

The last stage in the process involves generation of a thematic fragment to fit the current harmonic context and duration. First we generate a rhythm equal to the duration of the current chord, and then select intervals at random from a weighted list to determine pitch contour. The generated theme can then be transposed to commence on any degree of the

currently active chord. The theme is quantized to a musical scale that agrees with the current chord quality, chord root and current key. We choose a scale by correlating the pitches from the chord with the standard church modes, rooted against the current key. There is an additional option to select a scale simply from the chord root; this is primarily applied against chordal roots outside of the current key.

With a series of simple extensions this trivial thematic generator can be tuned to provide passable melodic material. In part this is due to the dominance of thematic rather than melodic invention in romantic music.

We provide the ability to choose not to play a melodic fragment and provide the option to accompany the chosen thematic fragment with a delayed copy played on a second instrument and possibly in a separate register. Thirdly, we make extensive use of themes previously generated during the course of performing the piece using random repetition and recapitulation to provide some global coherence. The final stage of thematic generation is to assign an instrument for the performance of the theme.

Orchestral Performance

Following the generation of the thematic material our musical data is ready to be performed. We use the Vienna Symphonic Library for sample playback.

We have designed a quasi agent-based approach for instrumental playback to take advantage of the low level control we have over sample playback. Instruments act semi-independently in their responses to musical note information choosing a sample patch based on a notes volume, duration, articulation style, and so on. For example, a trumpet knows which sample bank it should play from given a heavy attack, loud volume and short duration, as well as providing some simple range checks and instrument specific performance options (muted options for brass, pizzicato for strings etc.). We also apply a variety of gestural control mechanisms at this level, such as multi layered oscillators for dynamic modulation, control of legato performance parameters, fine grained volume control and alike as detailed previously (Sorensen and Brown 2007). These performance details greatly enliven performance.

Additionally the playback system supports multiple independent metronomes allowing us to modulate tempo for each individual part if required. This allows us to fairly trivially add rubato playing where instruments are linked to a single metronome or move independently to their own individual metronome.

The process described above is rapid enough to be calculated and performed in real-time for a relatively complete orchestration of flutes, oboes, clarinets, bassoons, trumpets, horns, trombones, tuba, violins, violas, cellos, basses and per-

cussion. A detailed discussion of all aspects of our performance system would require another full paper of equal length, so we will defer that report for another time.

Shortcomings

Before we begin a discussion about the results of our current research we would like to point out some of the known shortcomings of our current system so that we can take account of them when discussing our results.

We make very little allowance for good voice leading beyond our simple shortest path solution. Voice leading is an area in which computational study has arguably had its greatest successes (Ponsford 1999, Huron 2001, Hömel 2004). Given that we pay only superficial regard to voice leading we can expect to suffer from many of the classic part writing concerns, parallel movement, consecutive fifths and octaves etc. Another concern of our current shortest path algorithm is our lack of control over the distribution of voices over the complete pitch range. In other words there is little to stop bunching of parts at the top, centre or bottom of the pitch range. Given previous research in this area we feel confident that voice leading can be easily improved in the future.

There is no explicit shaping of the point-to-point root progression that should result in non-directed root progressions and therefore non-directed harmonic movement. Additionally, our current cadential support is superficial and we would expect this to manifest in contrived and stilted phrase and section boundaries.

We use no melodic shaping whatsoever. In fact, aside from simple pitch constraints and basic repetition, we provide no melodic support at all. One might expect this to result in generally worthless melodic material. A related shortcoming is the absence of any counter-melodic material aside from trivial thematic canoning that is occasionally employed.

The lack of higher-level structure is a common weakness in generative music systems. Our system is currently limited to the manual enveloped control of some global parameters in the shaping of overall form and structure. We expected that this would provide reasonable control over musical intensity as dictated by harmonic tension, dynamic and orchestration for example, but provide no value for other structural features, such as repetitions and variations that require more than local memory.

We currently used structured accompaniments. This is actually not as unusual as many people would expect for either manual or generative compositional systems. There are a range of accompaniment patterns that are found time and time again in orchestral music and programs such as Band-In-A-Box rely almost entirely on them. We currently provide no mechanism for escaping beyond the bounds of the preconceived accompani-

ment patterns. We anticipated that this would severely limit the range of stylistic output that the system was capable of, but this limit is not difficult to extend and will certainly be a focus of further developments.

Results and Future Work

Here we outline our subjective reactions to the musical output of our system (the data) and sincerely hope that interested readers will also review the musical examples made available here:

http://impromptu.moso.com.au/acmc08_examples.html

Listening back to the music generated by our system included a happy surprise for us that local cohesion operated surprisingly well. With only a few simple operations, such as thematic repetition and re-use of root progressions we feel the music has reasonable local structure without being overly restrictive. Most importantly we feel that our harmonic implementation is currently operating well enough to provide a balance between harmonic novelty and structural cohesion. Overall we are happy with our current implementation of Hindemith's chord quality system. We do however need to make improvements to the sophistication of our cadential treatment in order to help drive the music at section boundaries.

When listening to the music generated by our system there are many obvious deficiencies in the music currently produced. Most pronounced is the lack of meta-structure. This is hardly surprising as it is a common complaint about generative music systems and we foreshadowed this in our own shortcomings section above. We are not overly concerned about this at this stage, however, as we hope that introducing a "memory" into our system will help to introduce opportunities for global cohesion. We are also working with various user interface tools to allow higher-level human control over a parameter automation process that will assist to provide meta-structure. The development of these interfaces is directly in line with our desire to produce usable tools for working professionals in film, TV and computer game environments, and early prototypes of these interfaces are discussed in an accompanying publication.

Our voice leading, as we speculated earlier, does indeed cause some concern, although only partially. Our concerns over common voice leading issues such as parallel movement seem to be largely misplaced. In general we have no major complaints about the shortest path approach, which provides relatively smooth voice movement. Of greater concern however is the arbitrary wandering of this movement, which can lead to voice bunching and unnecessary voice crossover. We will need to address this by forcing some form of inner voice boundary checking and possible use of medium scale melodic contour control.

Our minimal set of accompaniment styles is a severe limitation on the stylistic variety possi-

ble with the current system. Indeed, our requirement that accompaniments be somewhat predefined is an inherent weakness in the system more generally. In future work we would like to investigate extending the instrument agent definition to provide some form of environmental listening, enabling each instrument "agent" to have the ability to perceive it's relationship to surrounding material. In this way we may be able to provide automatic accompaniment based around various gestalt principles of grouping, symmetry and self-similarity.

One of the great surprises for the work so far is how well completely random thematic material can work. This is not to say that it is reliable, indeed thematic materials can be as bad as they can be good. However, given that our thematic implementation is a placeholder only, we find its operation surprisingly serviceable. Our intention is to extend this current system to provide greater local scope. For example, we currently only support short thematic fragments, we would like instead to support melodic material that could be sub-divided into smaller thematic components. This would maintain the interesting expositional features that we currently enjoy in the system while enabling longer, more cohesive, melodic passages. We hope this might also provide us with more cohesive phrase level operation, a current deficiency linked to the aforementioned cadential weakness.

This brings us to our most significant insight. Overall we are convinced that the success of the system to date is very largely due to our holistic approach to implementation. In other words, we feel that the output of our present system is far greater than the sum of its often greatly limited parts. This is in line with our initial expectations, and on the surface would appear to support our approach. However, we are increasingly concerned that a by-product of this success is a severe limitation to the range of stylistic output the system is capable of. In other words, we are concerned that our approach may be producing output closer to a single "work" when we would hope for a system capable of producing significantly varied "works". At this early stage it is difficult to know if the system's aesthetic limitations are due to its partial implementation or whether there is a more serious methodological problem with our approach. We will need to keep this strongly in mind in the future.

Conclusion

This has been an introduction to the ongoing work at the Australian CRC for Interaction Design into the generation of orchestral music. This is a report on early outcomes and we have much work still to do ourselves and in partnership with others.

Acknowledgments

The work reported in this paper has been sup-

ported by the Australasian CRC for Interaction (ACID) through the Cooperative Research Centre Programme of the Australian Government's Department of Education, Science and Training.

References

- Brown, A. R. (2004). An aesthetic comparison of rule-based and genetic algorithms for generating melodies. *Organised Sound*, 9(2): 191-198.
- Brown, A. R. (2005). Exploring Rhythmic Automata. Proceedings of the *Applications of Evolutionary Computing: EvoWorkshops 2005*, Lausanne, Switzerland. Springer, pp. 551-556.
- Brown, A. R. (2005a). Generative Music in Live Performance. Proceedings of the *Australasian Computer Music Conference*, Brisbane, Australia. ACMA, pp. 23-26.
- Brown, A. R., Wooller, R. and Miranda, E. R. (2008). Interactive Evolutionary Morphing as a Music Composition Strategy. In *Music As It Could Be: New Musical Worlds from Artificial Life*. E. R. Miranda, Ed. Madison, Winconsin, A&R Editions.
- Clark, J. W. (2007). *Hindemith's Analyses in The Craft of Musical Composition*. Retrieved 7 April, 2008, from <http://jasonwclark.com/hindemith.aspx>.
- Cope, D. (1992). Computer Modeling of Musical Intelligence in EMI. *Computer Music Journal*, 16(2): 69-83.
- Cope, D. and Hofstadter, D. R. (2001). *Virtual music : computer synthesis of musical style*. Cambridge, Mass., MIT Press.
- Gerber, A. and Brown, A. R. (2006). Visualising Music with Impromptu. Proceedings of *Medi(t)ations: The Australasian Computer Music Conference*, Adelaide. ACMA, pp. 38-43.
- Hindemith, P. (1970). *The Craft of Musical Composition: Theoretical Part 1*. Mainz, Germany, Schott.
- Hömel, D. (2004). ChordNet: Learning and Producing Voice Leading with Neural Networks and Dynamic Programming. *The Journal of New Music Research*, 33(4): 387-397.
- Huron, D. (2001). Tone and Voice: A Derivation of the Rules of Voice-Leading from Perceptual Principles. *Music Perception*, 19(1): 1-64.
- Ponsford, D., Wiggins, G. and Mellish, C. (1999). Statistical learning of harmonic movement. *The Journal of New Music Research*, 28(2): 150-177.
- Sorensen, A. (2005). Impromptu: An interactive programming environment for composition and performance. Proceedings of the *Australasian Computer Music Conference 2005*, Brisbane. ACMA, pp. 149-153.
- Sorensen, A. and Brown, A. R. (2007). aa-cell in practice: an approach to musical live coding. Proceedings of the *International Computer Music Conference*, Copenhagen. ICMA, pp. 292-299.
- Towsey, M., Brown, A. R., Wright, S. and Diederich, J. (2000). Towards Melodic Extension Using Genetic Algorithms. Proceedings of *Interfaces: The Australasian Computer Music Conference*, Brisbane. Australasian Computer Music Association, pp. 85-91.
- Wooller, R., Brown, A. R., Miranda, E. R., Berry, R. and Diederich, J. (2005). A framework for comparison of processes in algorithmic music systems. Proceedings of the *Generative Arts Practice*, Sydney, Australia. Creativity and Cognition Studios Press, pp. 109-124.