**Nick Collins**
Department of Music
Durham University
Palace Green,
Durham DH1 3RL, UK
nick.collins@durham.ac.uk
www.composerprogrammer.com

# A Funny Thing Happened on the Way to the Formula: Algorithmic Composition for Musical Theater

**Abstract:** Algorithmic composition methods must prove themselves within real-world musical contexts to more firmly solidify their adoption in musical practice. The present project is an automatic composing program trained on a corpus of songs from musical theater to create novel material, directly generating a scored lead sheet of vocal melody and chords. The program can also produce output based upon phonetic analysis of user-provided lyrics. The chance to undertake the research arose from a television documentary funded by Sky Arts that considered the question of whether current-generation, computationally creative methods could devise a new work of musical theater (the research described here provides but one strand within that project). Allied with the documentary, the resultant musical had a two-week West End run in London and was itself broadcast in full. Evaluation of the project included both design feedback from a musical theater composer team, and critical feedback from audiences and media coverage. The research challenges of the real-world context are discussed, with respect to the compromises necessary to get such a project to the stage.

Academic algorithmic composition projects treating popular music are historically rarer than those investigating such domains as species counterpoint or bebop jazz, though there is a new wave of contemporary activity, perhaps best exemplified by algorithmic methods for electronic dance music (Eigenfeldt and Pasquier 2013; Collins and McLean 2014). The earliest computer music research in automatic composition includes generation of the 1956 pop song "Push Button Bertha" (Ames 1987), or generation of nursery rhymes based on information theory (Pinkerton 1956). Yet the predominant investigative domain, as exemplified by the careers of those most famous of algorithmic composers, Lejaren Hiller and David Cope, has been classical art music, and, in research terms, published work is often restricted to classical training exercises such as chorale harmonization. Opposing this trend, Cybernetic Composer, by Ames and Domino (1992), was a museum project for a Kurzweil synthesizer able to generate output in four styles of popular music. More recent manifestations of algorithmic composition within popular culture frequently incorporate interactive control. The 1990s saw the Koan software, by Pete and Tim Cole, and Brian Eno's spearheading of the promotion of generative music (Eno 1996). More recently, manifestations include the mobile apps Noatikl by the Cole brothers and Eno's collaboration with Peter Chilvers on Bloom. Algorithmic procedures have become more visible within digital audio workstations, such as Max for Live projects or Logic's MIDI Scripter, and they appear as the basis of the JukeDeck startup company (jukedeck.com), which aims to provide royalty-free generative music for the masses. Such recent work, in the domain of bedroom enthusiasts and corporations as much as academics, has not received much attention in terms of published studies.

Even while acknowledging a gathering research impetus into algorithmically generated popular music, prior work on the automatic creation of musical theater is nonexistent. The absence of previous work in automatic generation of musical theater may be due to a critical rejection of the area as supposedly lacking academic kudos, and to a lack of opportunity to get involved with real productions (which are rather high-budget enterprises). The present project was motivated by involvement in the TV documentary series Computer Says Show, funded by Sky Arts (www.wingspanproductions.com), whose premise was the research question of whether computational methods could devise a successful stage musical. Teams of academics analyzed existing musicals in terms of setting, plot, and audience emotional response; and considered automatic book and lyrics generation, audio analysis of cast recordings through

Computer Music Journal, 40:3, pp. 41–57, Fall 2016
doi:10.1162/COMJ_a_00373

music information retrieval (MIR), and, in the present case, symbolic composition of song lead sheets (Colton et al. 2016). The enclosing project provided real-world constraints and deadlines, and it promised the ultimate test of a real theatrical West End run.

This article describes the core algorithms for lead-sheet generation, both for generating pure song material and, when further constrained, to set lyrics. In terms of the taxonomy developed by Pearce, Meredith, and Wiggins (2002), this is computational modeling of musical style, to stand or fall on critical reception; evaluation included within-design-cycle feedback from the close involvement of a musical theater director, composers, and TV production staff, and eventually from critics and audiences for the real production run. Working towards the ecologically valid final show compromised purity of evaluation that might otherwise have been found in more controlled (and contrived) laboratory circumstances, and raises methodological issues in reaching beyond pure computer music research. It was, however, too good an opportunity to miss, revealing alternative public perspectives on musical algorithms. This article has a further contribution as a cautionary tale for researchers who follow in moving out of the safety of the laboratory.

## The Lead-Sheet Generation Algorithm and Its Parameters

The software rests upon corpus analysis of existing musical theater material as well as upon hard-coded rules providing generative constraints, thus combining corpus- and rules-based work. Corpus-based work included an automatic chord-detection analysis of a large set of musical theater cast recordings informing a harmony-generation model, and a custom corpus of musical theater songs in a novel format that favored analysis, and thus subsequent synthesis, of musical phrases. Phrase materials were subject to Markovian modeling, and analysis statistics were fed into particular production rules. Refinement of the algorithms was chiefly motivated by feedback from the primary documentary participants, two music theater specialists (Benjamin Till and Nathan Taylor). This process was seen as necessary to constrain the domain of permissible generation to favor a higher proportion of effective outputs. Initial representational and modeling decisions required in the application of machine learning to any corpus are themselves hard-coded impositions by the systems designer, and so taking a pragmatic middle way utilizing both corpus- and rules-based techniques was not seen as compromising the project's research.

The code was written in SuperCollider, generating text files in FOMUS score format (Psenicka 2009) and, in parallel, MIDI files. The MIDI files could be imported into Sibelius, and the FOMUS software acted as the interface to automatic generation of final PDF scores within Lilypond (MIDI and PDF files were supplied for each lead sheet). Additional external callouts for the lyrics analysis were made to Python and the Natural Language Toolkit (NLTK; see Bird, Klein, and Loper 2009). To give a taste of the software's generativity, multiple score examples are given at points throughout the article, although such illustrations remain snapshots of the true large output space.

### Chord Sequence Model

A parallel project, undertaken by Bob Sturm, Tillman Weyde, and Daniel Wolff, applied MIR analysis to a large corpus of musical theater cast recordings (from *A Chorus Line* to *Wicked*); the most reliable MIR features for the purposes of training an algorithmic composition system were provided by chord detection. Chords were extracted throughout using the Chordino VAMP plug-in (Mauch and Dixon 2010) with Sonic Annotator for batch extraction (www.vamp-plugins.org/sonic-annotator). Fifty-three shows had been marked as "hits" in an analysis of economic and critical factors by James Robert Lloyd, Alex Davies, and David Spiegelhalter, leading to 1,124 analyzed audio files totaling around 53 hours of audio (Colton et al. 2016).

The chord data is not absolutely reliable, in that the plug-in itself is not as good a listener as an expert musicologist, but it does provide a large data source

*Figure 1. Six examples of generated sequences of 24 chords each. The first three examples are in the home key of C major (major-key chord transition model); the second three are in C minor (minor-key chord transition model).*

```
[C, G, G6, F6, Am7, Cmaj7, G, G, Dm, G, C, Cmaj7, Am7, G7, Cmaj7, F, Dm, Em, G, C, G,
Fmaj7, Em, C]
[C, G, C, F, G, Cmaj7, C, Em, F, G, C, G, C, G7, C, G7, C, Am7, G, C, G6, Am, Em, F]
[C, F6, Bm7♭5, C, F6, C, F, C, F, C, F, G, C, G, C, G, F6, F, G, F, Cmaj7, F, C, C]
[Cm, Bdim7, Cm, Bdim7, Cm, Bdim7, Cm, Bdim7, A♭, A♭, A♭maj7, Cm, Bdim7, A♭6, Cm, Bdim7, A♭,
G7, Cm, Fm, Cm, G7, Cm, Fm]
[Cm, Fm, Cm, A♭, G7, Cm, A♭6, Fm6, G7, Baug, G7, Cm, Baug, G7, Cm, A♭6, A♭, G,
Bdim7, G, A♭maj7, G, A♭m, G]
[Cm, Bdim7, Fm, G, Cm, A♭, A♭m, A♭m6, A♭maj7, A♭m, Bdim7, G7, Cm, G7, Cm, A♭6, Bdim7, G7, A♭,
A♭6, Bdim7, Fm6, G7, Baug]
```

otherwise unobtainable with the human resources at hand. A parsing program was written to translate the textual chord shorthand from the Chordino plug-in into pitch-class note information. Data were cleaned up by removing any chord changes considered to be too fast (i.e., quicker than a half a second, corresponding to one beat at 120 bpm), and by ignoring any results labeled as "N" (i.e., where no chord had been found in a given section of audio). Sequences of chords were only considered complete when at least three chords were detected in a row without any "N" intervening.

Having obtained a large set of chord sequences representing hit musical theater, two chord generators were obtained. In the first case, no attempt was made to impose a home key. In the second, only relative motion between chords fitting within a single major or minor key was permitted to train the model; separate major- and minor-key models were created. The machine-learning algorithm was a prediction-by-partial-match, variable-order Markov model (up to order 3, cf. Pearce and Wiggins 2004). Its application requires integers, so an encoding from chords to integers was created, where ten chord types and twelve chromatic pitches translate to one of 120 possible integers. Figure 1 provides three examples of generated sequences of 24 chords in C major and another three examples in C minor, created with the major- and minor-key models and constrained to start with the root chord of the home key. Certain loops are evident in the statistics of chord transition; for example, the third minor-key example includes a case of major-to-minor chord alteration (on A♭) temporarily stuck in repetition. Chord types are sometimes altered, for example, from a major chord on a particular root to a major chord with added sixth on the same root, potentially lifted from a harmonic sequence or vamping pattern in source material. The chord sequences are generally musical and in character with musical theater, though without any innovative individual style.

A further chord model was obtained by taking the chord-transition table data from Declercq and Temperley (2011), which correspond to a corpus of 100 successful songs from popular music. This model was eventually not used for the musical, as it lacked the specificity of the musical theater, although it did provide a useful comparator.

### Representation and Analysis of the Melody Corpus

Although some musical theater MIDI files are available online, the reliability and consistency of the data are too variable for immediate corpus work (files are often created by amateur enthusiasts, without any standard track arrangement and often as nonquantized renditions). Because song creation in a passable musical theater style was the most essential compositional task, requiring stylistically appropriate vocal melody at core, the decision was therefore made to encode a central corpus of musical theater songs as prime exemplars for system training. The encoding fundamentally respected musical phrasing, marking up all melodic phrases explicitly, to have a corpus innately centered on vocal melody. The two musical theater experts, who were allied with the documentary team, advised on a subset of songs to encode from musicals that had been denoted "hits." (These musicals included such well-known shows as *Cats*, *The Lion King*, and *The Rocky Horror Picture Show*.)

*Figure 2. Example encoding of Andrew Lloyd Webber's "Music of the Night," from* Phantom of the Opera *(1986).*

```
[
// Melody by phrases, in the form [start beat within bar
// (allowing for anacrusis or initial rest), followed by an
// array of pitch-duration pairs for each note (pitch defined
// as interval in semitones from C), end beat of bar of phrase,
// and gap until next phrase].
 [
  [0, [4,1, -5,1, 2,1, -5,1], 4, 0],
  [0, [0,0.5, 2,0.5, 4,0.5, 5,0.5, 2,1, 7,1], 4, 0],
  [0, [4,1, -5,1, 2,1, -5,1], 4, 0],
  [0, [0,0.5, 2,0.5, 4,0.5, 5,0.5, 2,1, 7,1], 4, 0],
  [0, [9,0.5, 12,0.5, 12,0.5, 12,0.5, 14,1, 12,0.5, 11,0.5], 4, 0],
  [0, [9,0.5, 12,0.5, 12,0.5, 12,0.5, 14,1, 12,0.5, 11,0.5], 4,0],
  [0, [9,0.5, 12,0.5, 12,0.5, 12,0.5, 14,0.5, 12,0.5, 9,0.5, 5,0.5, 7,2], 2, 1.5],
  [3.5, [4,0.5, 2,0.5, 2,0.5, 2,0.5, 4,0.5, 5,0.5, 7,0.5, 4,0.5, 2, 0.5,0,2], 2, 2]
],

// Chord sequence, as array of pairs, each pair consisting
// of an array of pitches followed by the associated duration.
 [
  [0,4,7],2, [0,4,7]+7,2, [0,4,7],2, [0,4,7]+7,2, [0,4,7],2,
  [0,4,7]+7,2, [0,4,7],2, [0,4,7]+7,2, [0,4,7]+5,2, [0,4,7],2,
  [0,4,7]+5,2, [0,4,7],2, [0,4,7]+5,2, [0,4,7]+10,2, [0,4,7],4,
  [0,4,7]+7,4
 ],

// Medium scale form, interrelationship of phrases, in this
// case ABABCCCD.
 [0,1,0,1,2,2,2,3],
]
```

For a given core song melody, the encoding provides its notes as pitch and rhythm, broken down into phrases, associated chords, and a formal denotation of the melody's internal phrase relationships. The melodic data have a redundancy, in that the start and end position of each phrase within a measure, as well as interphrase intervals are supplied. But these provide a useful check on human error in encoding (the start beat plus the sum of durations within the phrase should lead, modulo time-signature measure length, to the end beat, which, adding the inter-phrase time interval again, should lead to the next start beat). An example is shown in Figure 2, the encoding being itself valid SuperCollider code of nested arrays; the reader can observe the phrase structure with one phrase per nested array (normally falling on one phrase per line of text). All melodies were transposed to a home key of C major or minor, and the standard time signature was 4/4 although other time signatures were permissible. Quarter-note or half-note triplets could be encoded using beat durations of 0.33 or 0.66 (with small variations, so that triplet groups add up to an integer duration). Because representational decisions are key to machine learning, Figure 2 provides insight into the core priorities in musical data for the algorithmic generation system.

Forty-five songs were encoded in this manner; encoding was a relatively intensive process, requiring analytical decisions on phrase boundaries and phrase relationships that may be points of disagreement between analysts, but which were of sufficient quality to form the basis for novel generation of phrases.

The phrase-based encoding allows for statistical analysis of a number of attributes of phrasing in musical theater material. As would be expected from music psychology, phrase durations (assuming an average tempo of 120 bpm) were around 3 seconds in length, corresponding well to the perceptual present and associated memory constraints (London 2012). Chromatic movement was much rarer than diatonic (2,052 diatonic note transitions compared with 213 chromatic note transitions), as might have been anticipated for melody from popular music theater. Note-to-note pitch interval movements were more frequently by step than by leap, in the proportions 44.66 percent adjacent step, 23.26 percent repeated note, 16.68 percent leap upwards, and 15.4 percent leap downwards. Of the 604 leap intervals, 216 were followed by a step, 214 by another leap (65.9 percent of the time in the opposite direction to the previous leap), and 174 were the last interval in a phrase.

Statistics were also extracted for phrase ranges, including mean and median pitches of each phrase. An entire transcribed song extract could provide guide templates for melodic movement. Phrase data from the melodic corpus provided the basis for variable-order Markov models over pitches, melodic intervals, contour classes, durations, and inter-onset interval classes. These models proved useful in generating novel melodies, founded in corpus statistics. Assuming 4/4 time (because the majority of the melodies conformed to this time signature), statistics were also obtained on pitch choices and pitch intervals at each of the eight distinct eighth-note positions in the measure.

## Algorithm for Melody Generation

The melody-generation algorithm creates musical materials at a local level of the phrase, with a medium-scale structure built up by the phrase interrelationships to create song sections, and the final song built up by repetition of sections within a form. The phrases of the melodies in the training corpus are used to train pitch and rhythm models, to construct novel phrases. Novel phrases are specified within a diatonic pitch space, and, in their reuse, these phrase materials are thereby adjusted automatically to work against changing harmonic contexts. The source melodies also provide guidelines for the form over multiple phrases, including the skeleton of pitch height over a melody. The idea of using the mean pitches of a guide melody to constrain new generation bears a relation to the use of an elastic tendency towards the mean pitch of the phrase within previous psychologically inspired treatments (Brown, Gifford, and Davidson 2015).

The central melody-generation routine has quite a number of control arguments, listed in Table 1, giving insight into the flexibility of the calculation. In a number of places, corpus-derived models and statistics naturally inform the underlying routine.

Figure 3 presents two example lead sheets, each restricted to eight measures, to give a flavor of the generation. The parameters are the defaults for the lead-sheet generation algorithm, as per the last column in Table 1. No attempt has been made to "cherry-pick," these being the first two melodies created directly for this example.

## Algorithm for Ostinato Generation

A frequent requirement for musical theater composition with a strong connection to popular music styles is the creation of rhythmic and pitch ostinati, as backings during songs and instrumental filler music. Similar principles to the vocal melody-generation work were applied, but with a separate corpus consisting of some well-known ostinati from popular music and musical theater (e.g., Michael Jackson's "Smooth Criminal," Queen's "Another One Bites the Dust," and "One Day More" from *Les Misérables*).

The backing harmony was either C minor or C major, with no other chord changes; the expected use was that the ostinato could be adjusted to match other chords in a song if needed, but was, in its most basic manifestation, for a groove on a set root. Figure 4 provides a variety of example outputs (again, the first set generated for this article). Note the overly wide-ranging movement in the seventh example, the common initial rhythmic pattern in the first and third, the appearance of dotted and

**Table 1. Control Arguments for the Central Melody-Generation Function**

| *Argument* | *Result* | *Default* |
|---|---|---|
| Key | Set base key for generation | C major |
| Time signature | Set base time signature; no compound signatures, typically 4/4 or 3/4 | 4/4 |
| Range | Set singer's range, permissible compass of notes | 0 to 12 (one octave) |
| Chords | Chord sequence to work to (from chord model, or imposed) | Generated from chord model |
| Eighth-note data | If true, utilize statistics collated separately for each eighth note of the bar, rather than aggregated across all eighth-note positions | 50%/50% true/false |
| On-beat chord probability | Probability of restricting on-beat positions to use only notes of the current chord | 100% |
| Allow sixteenth notes | Allow faster rhythmic units within a melody | 100% |
| Pitch-choice model | Select between two available models for pitch choice: (1) a greedy dynamic programming approach and (2) a variable-order Markov model | Greedy dynamic programming |
| Top jump | Largest leap size in diatonic steps | 8 |
| Patter rhythm probability | Chance of rhythm generation using a "patter rhythm," that is, fast sequence of durations as per Gilbert and Sullivan's "I Am the Very Model of a Modern Major-General" | 0% |
| Use PPM for rhythm | Whether to use a prediction-by-partial-match model for generating rhythmic sequences or a rule-based process | 0% |
| Maximum contiguous syncopation | Maximum number of notes that can be syncopated (falling on an off-beat) in a row | 2 |
| Guide strictness | Whether a template phrase pattern can influence pitch position (the guide consists of the average pitch per phrase) | 100% |
| Impose form | User-specified phrase form rather than derived from a guide melody | False |
| First chord is tonic | Enforces any generated chord sequence to begin on tonic chord of the key | False |

PPM = prediction-by-partial-match.

Scotch snap rhythms in the C-minor patterns, and the syncopation of the sixth ostinato.

## Generation Based on Lyrics

Musical theater composition can proceed led by a musical idea first, or by a lyric. To accommodate a frequent request from the show developers that existing text be accomodated, a front-end process was devised to analyze song lyrics and to be able to set notes to their implicit accent pattern.

Code utilized the Python library NLTK, which provides a function to analyze metrical stress within a word over syllables, as well as a dictionary from the Gutenberg organization (www.gutenberg.org/files/3204) that provided exact syllable breakdowns for common words (e.g., "ac-com-mo-dat-ing", "un-cal-cu-lat-ing"). Text was provided as a block, converted to lowercase ASCII without special characters, and separated into line and words. The prepared text was passed to an external Python program (passing data to and from SuperCollider via auxiliary text files), where the metrical stress analysis came down to a special dictionary lookup (using the `cmudict.dict()` function available with NLTK, which supplies per-word analyses). The Python library gives stresses at three

Figure 3. Two examples of generated eight-bar lead sheets.


levels (see Figure 5 for an example). In Figure 5, the word "authorship" is marked 102, so that "ship" is the highest stress in the whole sentence. (Some readers might disagree with the dictionary here!)

Musically, a reconciliation must be effected between the stress pattern and the metrical frame provided by the time signature; good scansion would normally indicate strong stresses of syllables on strong beats. Syllables (all of which have an associated vowel for singing) might be extended via melisma, but that option was not pursued in the current case. Instead, syllables were allocated measure position based on a default of offbeats for stress level 0, and level 1 stresses falling on the beat; in 4/4, a succession of 0s could fill in across eighth notes, but successive 1s would be spaced by quarter notes.

Figure 6 provides three examples generated using Alex McLean's text. In all three, the split of "end-ing" with "end" on a quarter note shows the lack of flexibility of the software to certain possibilities of patter ("end-ing" could be two eighth notes in line with other parts of that phrase). Note how "ship" always falls on a stressed beat.

The algorithm presented here has trouble with lyrics that have a strongly repeating line-by-line pattern, denoting a common anacrusis, and favors 4/4 over 6/8 interpretations. A facility was added to force a particular pickup structure on the output. It proved practical for generation for this project, but would be open to much future improvement. The natural-language dictionaries themselves were also found to be rather incomplete for song lyrics. In some cases, words had to be provided, already split up, ahead of the syllabification process (the dictionaries might be extended themselves to solve this).

This form of text-to-music generation is in contrast to (but might be expanded through) work based on sentiment analysis, such as the wonderfully named TransProse system (Davis and Mohammad 2014), which creates piano pieces based upon the emotional subtext of novels. There is little prior work generating songs directly from lyrics, except for systems such as MySong/Songsmith (Simon, Morris, and Basu 2008) or the mobile apps Songify and AutoRap developed by Smule, which operate by onset and pitch detection within the audio signal and carry text along with them.

## The Human–Computer Collaborative Design of the Final Music

A fully autonomous program for generating lead sheets was created. It combines the melody-generation and chord-generation modules, coupled with some rules on form. In practice, however, operation of the program was in the domain of computer-assisted composition (Miranda 2009),

Figure 4. Eight generated ostinati (four examples each for C major and C minor).

Figure 5. Example of metrical stress analysis carried out by the Python Natural Language Toolkit. Text originally posted by Alex McLean to Facebook.

Figure 4

```
i got extremely bored of the never ending discussion of authorship around generative art
1 1   0 1   0    1    1 0   1 0   1 0    0 1  0   1   1  0  2   0 1    1 0 0 0    1
```

Figure 5

used to provide material that was then manipulated by human composers. The compromises of working within a high-profile broadcast project with multiple stakeholders necessitated more human intervention before public performance than would have been preferred for pure research; but then, access to a West End venue for evaluation would never have occurred without such oversight.


*Figure 6. Three examples of lead sheets generated from lyrics.*

To maintain some research objectivity concerning the aesthetic choices at the heart of song selection, batches of computer-generated outputs were sent en masse (often 100 songs at a time), without any cherry-picking, to the musical theater specialists. The human composition team essentially selected fragments (somewhat laboriously and without consultation from the research team) from 607 lead sheets and 1,171 ostinati, working with a rehearsal pianist. After particular discovery sessions and in the process of musical development of the final musical theater piece, they sent requests for revisions and novel program output—for example, soliciting a suite of songs in 3/4 instead of 4/4. The musical theater composers' musical preferences and narrative needs had an unavoidable influence on the material making it through to the show, and they frequently freely composed around the skeleton of computer-generated material. The TV production company had mandated an intention to respect the computer-generated material; that the human composers still felt able to range widely from this base is some indication both of limitations in the algorithmic composition and of discomfort in the task of negotiating between algorithm and human vision.

Table 2 lists the 16 songs in the show and their derivation from the computer programs involved in the production. In some cases, the human composition team has only kept a minimal fragment of melody, or in the worst scenario, just a chord sequence (which is a less-than-unique data point, uncopyrightable, and trivially taken unrecognizably far from the original generated material). The production team compiled, together with the human composers, a document detailing the origins of each song in the show (Till et al. 2016) to track the experiment and to assess authorship proportions with respect to publishing rights. Some relevant quotes are reproduced in the table, which uses this source, alongside further analysis of the songs, to attribute the algorithmic components. To complicate matters, the Flow Composer software (Pachet and Roy 2014) was also used to contribute towards a few songs, though it is beyond the scope of the present article to further evaluate that software here (see Colton et al. 2016 for more on the role of Flow Composer).

**Table 2. Songs in the Show and Their Derivation**

| *Song* | *Program* | *Material* | *Computer Contribution* |
|---|---|---|---|
| 1. Green Gate | ALW | Two ostinati, chord sequence, melody, and chords | 50%: Computer-composed eight-bar theme starts the show, and is basis of much further material. |
| 2. We Can Do It Alone | ALW | 16-bar 3/4 central section (chords and melody line) | 20%: As accompaniment material in central section, otherwise human-composed, including all vocal lines. |
| 3. Penetrate the Base | ALW | Chord sequence and two ostinati | 40%: Chord sequence, intact but with interpolated B minor, underlies the verse, although with a human-composed lead vocal. Ostinati are used quite strongly in the composition. The main ostinato is slightly adjusted from the computer original though its derivation is clear and the latter appears later in the song. "I hope the use of this ostinato through this number and at other key dramatic moments of the show will give it the same impact as the ostinato which starts 'Heaven On Their Minds' from *Jesus Christ Superstar* and is later used for the whipping scene. This was one of the references given [to the researchers]. . . I feel the creation of ostinati was a very successful aspect of this process because it also allowed me a great deal of creative freedom when working out what was going on around the ostinato" (Till et al. 2016, p. 11). |
| 4. So Much to Say | ALW | Melody and chords | 20%: The melody of the piece's middle section can be traced to a few bars of program output, but otherwise humans had much more to say. |
| 5. Graceful | ALW | Melody and chords generated to lyrics | 50%: Possibly the most substantially respected computer generation, though there is certainly tweaking of output to best fit lyrics where the automated scansion fails, and additional human-composed material. |
| 6. We Are Greenham | FC | Lead sheet created based on Greenham protest songs | N/A: Lead sheet quite well observed (see Colton et al. 2016). |
| 7. At Our Feet | ALW | Melody and chords | 50%: Much of the material is closely related to the computer source. Core "catchy" elements in verse and chorus are drafted by the computer part, but have been rhythmically tweaked by human hand. |
| 8. Unbreakable | ALW and FC | Melody and chords (both contributed by both programs) | 30%: Shows some connection to the original computer-generated materials, although there is human tweaking, especially in the shifting to a calypso style. |
| 9. How Dare You | ALW | Melody and chords | 50%: A single lead sheet led to all the source materials for the song; some rhythms have been changed, in particular from straight half notes to less symmetrical quarter-and-dotted-half rhythms, but the main verse is a clear derivation from the computer. The chorus is a greater stretch to relate, although it has a basic intervallic cell in common, if shifted in rhythm. Setting to lyrics led to human-composed melodic variations that were more elaborate. |

**Table 2. Continued.**

| *Song* | *Program* | *Material* | *Computer Contribution* |
|---|---|---|---|
| 10. Bouncing Back | ALW | Melody and chords generated to lyrics | 50%: The computer output was substantially adjusted in rhythm because of the demands of the lyrics, and failings in its appreciation of natural scansion. "As a comedy song, the rhythms of the lyrics are so important for the comedy aspect. Break the rhythm that is inherently in the words, and you lose so much of the comedy. As we know already, this system doesn't yet have much of a grasp of stressed syllables versus unstressed ones, let alone meter and form, such as dactyls, iambs, and spondees!" (Till et al. 2016, p. 32) |
| 11. Would It Be So Bad | ALW | Melody and chords | 30%: The computer source is mainly lost here against human composed material, but it is more apparent in the closing ensemble material based on a different lead sheet. |
| 12. Scratch That Itch | FC and ALW | Both programs provided melody and chord material | 10%: Much of the computer material was cut in rehearsals, leaving just some fragments of chord sequences of dubious relation to the original. |
| 13. What's The Point | ALW | Melody and chords | 10%: In the main part of the song only chord sequences from the computer were used, the rest was human composed. The middle eight is claimed to rest on a computer-composed lead sheet (Till et al. 2016, p. 42), though the relationship is too stretched to be apparent. |
| 14. In Our Hearts | ALW | Melody and chords generated to lyrics | 40%: Corrections were made to the rhythm to improve the lyrical setting, but computer material is clearly present in the final version, including the melodic hook of the main chorus. |
| 15. Thank You | ALW | Melody and chords | 30%: The initial trajectory of the song is determined by a 3/4 fragment of computer-generated composition, although the main onrush of the song, with its frantic melodic movement, bears little relation to the computer output. |
| 16. Beyond The Fence / At Our Feet / We Are Greenham/ Green Gate | ALW and FC | | 25%: The first part of this closing number is another "computer-inspired" (Till et al. 2016, p. 56) treatment, taking one program output song as an initial guide. A recap of various parts of the show follows, though the human hand in the composition remains clear. |

ALW = the program developed by the author; FC = flow composer. The Material column enumerates which musical components were algorithmically generated. Each entry in the final column starts with a rough estimate of the amount of the song credited to algorithmic composition, followed by some additional details.

The final column of Table 2 gives an estimated percentage of computer-composed contribution to the final songs for the program presented in this article (listed as "ALW"). The percentage is derived from musical analysis of the final pieces against the original algorithmically composed lead sheets, and from examination of human composer comments on their manipulation of the source song material (Till et al. 2016). This calculation was necessitated by the UK Performing Rights Society registration for the musical, which required a quantitative evaluation. The overall average contribution for the

*Figure 7. Computer-generated original chorus material versus the final version of the song, finished by human hand.*


computer over the 15 songs where ALW was utilized works out as 32 percent, or around one-third of the composition. Although this number cannot be seen as definitive, given the limitations of human self-reflection on creative acts and the working opacity of the machine algorithm, it is suggestive of some algorithmic contribution surviving the process. In cases where two human composers were intimately involved in songs, it points to an equal three-way split between authors (two humans and a computer); in many cases, however, a single human composer worked on a given song, and the contribution percentage is less impressive.

Figure 7 shows the first four bars of the computer-composed chorus material, versus the eventual human-doctored show tune for "At Our Feet"; there is a relation, but there is also a reworking going on that moves rhythms towards more comfortable patterns, streamlines melody, and isn't afraid to reharmonize. The result is a more conventional musical theater composition, and the nature of these adjustments actually has strong potential for showing future revision possibilities for the generating algorithm.

In many cases in the show, a claimed link between computer-composed original and the eventual show score is only vaguely perceptible, or is obfuscated by transformations such as rhythmic value substitutions, new pitches or chord substitutions, and shifting with respect to bar lines to change metrical emphasis (particularly, and perhaps forgivably, used for instances of generation to lyrics). Orchestration in the final production was carried out entirely by human hand, and the live band at the show provided some inherent ambiguity as to the music's origins (the score featured quite a lot of electric guitar in a "power rock" vein).

## Evaluation through Critical Reaction

Few algorithmic composition projects have had the opportunity to receive critical appraisal in a high-pressure, real-world situation with wider exposure than an art music concert of cognoscenti. As detailed in the previous section, although the material had gone through human modification to varying degrees without the involvement of the original researchers, there was a computational presence within the final musical theater piece. On 26 February 2016, a real West End theater show was judged by real theater critics from national media, and the show had a two-week run around this gala performance (see Figure 8).

The theater reported well-engaged audiences, with decent attendance over the two-week run, with many positive Twitter comments and other public feedback. A total of 3,047 people saw the musical, or around 60 percent of the theater's seating capacity during the run (there was virtually no wider marketing budget for the show, and attendance generally followed press that the algorithmic ideology had attracted). As far as it is possible to poll, audiences were mainly drawn from typical West End musical theatergoers, with an unknown proportion of technology-sector workers and academics, who may have attended because of the novelty of the generative component. The press night had a greater proportion of family and friends of cast and creative team. For the final three performances, audiences were polled by Wingspan Productions and asked to rate their enjoyment of the show from 1 (low) to 5 (high). Of 57 respondents, the poll revealed an overwhelmingly high level of enjoyment (the ratings, from lowest to highest, received 1, 1, 6, 10, and 39 votes, respectively).


*Figure 8. The musical at the Arts Theater, London.*

Theater critics are a more volatile group, however. Table 3 quotes some of the most pertinent critical judgments, with a particular emphasis on comments on the music specifically. The more astute critics, such as *The Telegraph*'s Dominic Cavendish, recognized the level of human intervention in the final production:

> *Beyond the Fence* has—if nothing else—considerable curiosity value, even if that value diminishes when you find out about its actual genesis. This experiment to see whether state-of-the-art computing might deliver the next *Sound of Music* has plainly benefited from a lot of human intervention in the six months it has taken to get from its preliminary boot-up to the West End stage. To call it 'computer-generated' is misleading. 'Computer-initiated' and 'computer-assisted,' though less grabby, are more accurate (*The Telegraph*, 27 February 2016).

The broad consensus was that the underlying show was passable but by no means outstanding. In some ways, this is a success for style emulation, although the human cherry-picking from and finessing of the raw computer output provides an additional layer of filtering that tempers confidence in a strong result. That the show was not groundbreaking in its music is unsurprising, given the reliance on databases of musical theater across decades. Statistical analysis aggregated data across time periods, simply selecting hit musicals without any concern for recent trends in musical theater. Unsurprisingly, critics noticed this averaging effect. Design by committee is a lurking issue at the heart of the production.

The project did lead to much media publicity, and can be seen as a landmark in public exposure to computational creativity (Colton et al. 2016). Perhaps the most apt coverage was the article in *New Scientist* (3 March 2016) that quoted from the biography created for the algorithmic composition program: "Other interests include composing music for musical theater, composing musical theater music, music theater composition, and the overthrow of humanity." This review also clearly understood the inchoate technology and its averaging effects, continuing, "for all the algorithmic cleverness behind the technology, a huge amount of its heavy lifting amounts to a kind of fine-grained

**Table 3. Selected Critical Reception in Media Outlets**

| *Source* | *Rating* | *Quote* |
|---|---|---|
| *The Stage* (Paul Vale, 26 February) | 3 | "Little, if any, new ground is broken, either in the structure or the score. . . a varied score." |
| *The Telegraph* (Dominic Cavendish, 27 February) | 3 | "It might have been more satisfying all the same to plump for a scenario of an ostentatiously technological nature, or at least take inspiration from the 'new wave' electronica of the time. . . It looks and sounds analogue, generic, presses no avant-garde buttons. . . A terrific end-of-show number [Thank You] . . . 'Computer Says So-So' then. In a world where flops are the norm, no mean feat." |
| *The Independent* (Holly Williams, 27 February) | 3 | "The result, as you might expect, feels formulaic. The music, piano-led ballads and squealy 80s power-rock, sounds vaguely familiar yet there are no barnstorming, hummable hits. . . I wonder if the computer-generated tag will help or hinder: it's hard to think you'd watch the show without being more interested in the process than the product. And am I being romantic in thinking it's telling that, while the story and songs work fine, the thing that makes it zing is the human-chosen setting? Maybe, but I don't think theater-makers need to start smashing computers any time soon." |
| *The Guardian* (Lyn Gardner, 28 February) | 2 | "A dated, middle-of-the-road show full of pleasant middle-of-the-road songs." |
| *Londonist* (Stuart Black, 29 February) | 3 | "It's quite fun to try and spot stuff the tech has repurposed: a bit of *Chicago* here, a bit of *The Lion King* there—quite a bit of it sounds like Meatloaf at medium throttle." |

All reviews appeared in February 2016. Ratings were on a scale of one to five stars.

market research. . . . The UK's musical theater talents can sleep peacefully at night with little to fear from . . . cybernetic pretenders."

In the course of the research after media coverage, a legal letter was received from a well-known musical composer concerned at the use of a parodic version of his name for the program, and seeking to stop this under trademark law. That letter is quoted here under fair use for the purposes of critique, illuminating as it is to the bias in the old-school entertainment establishment and the backwardness of the law confronting new computational possibilities:

> In addition, our client is concerned about the imputation that is carried by naming the program Android Lloyd Webber. Our client is an innovative composer, yet the name of the program can be understood to imply that our client's musicals have been composed by way of a mechanical process rather than a creative process, which is derogatory (Ashby 2016).

It seems more derogatory that a "mechanical" (computer-programmed) process could not be creative, especially in terms of the creativity of the human author of such a program. It also seems a contradiction to seek to stop a program on commercial grounds from producing output that could be confused with that of a human, and at the same time be so worried as to denigrate the program's capabilities in emulating creativity.

Figure 9 provides a gentle response to criticisms by setting selected comments in a song. This is the first pure output of the program, untouched by further human composition; some motivic reuse is clear, though the melodic line does not stray far. As presented in bare score, there is no human performance mediation; the songs for the musical had the benefit in performance of human

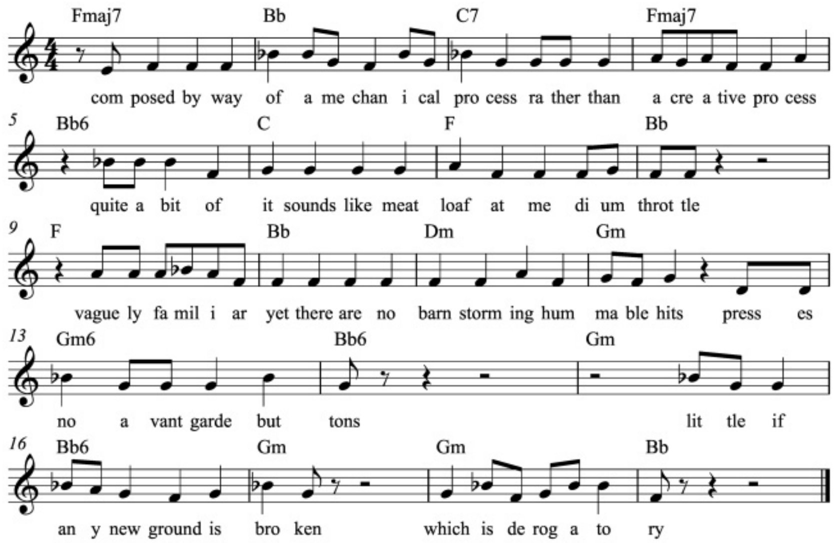


*Figure 9. Some critical reactions algorithmically set to song.*

expression, and human editing and orchestration. These provide a further confutation of experimental control, though again we must offset this problem against the ecological validity of the final product.

Three recommendations are gathered here for future algorithmic composers—that is, those who create algorithmic composition programs, in the position of working with a musical theater team:

1) Expect a push from the musical theater specialists for heavy post-algorithm human editing, and try to stay involved in later stages of the production process.
2) It may be more productive, given the current close links of musical theater composition to popular music, to create an effective pop-song generator with clearly demarcated verses and choruses, and some phrase modulations (i.e., unprepared key shifts) of materials, rather than attempt to work against a corpus of many decades of musical theater shows. For deeper evaluation purposes, a larger historical corpus of musical theater shows should be broken up and subsets assessed to ascertain the effect of different eras on output.
3) Musical theater critics may be disappointed that a computer-generated musical does not engage with computational topics as its essential subject matter. If an algorithmic composer aims to blend in with a mainstream of musical theater composition, success may be taken as blandness of vision!

Despite these challenges, which should not be underestimated as obstructions to pure computer music research, there are great rewards in a real-world project reaching a wider audience beyond specialists. Ultimately, algorithmic composition research must engage with this wider sphere to increase the exposure of such ideas within culture. Because music ultimately stands or falls on general reception, rather than controlled laboratory studies, it is prudent to take opportunities to engage with larger public-facing projects, though methodologies will need careful preparation in future research. The hope is that there are essential aspects of the act of human composition to be discovered through such higher-profile challenges in musical modeling.

## Conclusions

Computational music generation towards a West End show provided a rare chance for very public reaction to algorithmic composition. Despite the clear publicity for "the world's first computer-generated musical," the final piece was highly mediated by human intervention, although much of the musical seed material did originate algorithmically. The demands of an associated television documentary series and human interventions ahead of performance clouded the purity of evaluation, yet it has still been possible to discover new facets of practical generative music based on corpora, and explore text-driven creation of lead sheets. These techniques should also be applicable within various domains in the generation of popular music, in the first instance by switching the source corpus to one of appropriately annotated popular songs. Though methodology necessarily remained pragmatic in negotiation with real-world deadlines and output, the present work should serve as a case study and cautionary tale for future projects that seek to move from academia to fully ecologically valid contexts.

Future work might investigate a number of alternative approaches. Cleaned-up MIDI files may provide a route to a larger corpus of symbolic material. A historical investigation into musical theater composition might benefit from an online repository of late 19th- and early 20th-century works hosted by the Gilbert and Sullivan Archive, with many MIDI files created by Colin M. Johnson (Howarth 2016). A more complicated model of setting text would be crucial to increase the effectiveness of automating song production, allowing for deliberately extended syllables via melisma, and reflecting repeated stress patterns more effectively over lines indicative of a common anacrusis. Musical theater composition itself has not been the prime subject of previous research in algorithmic composition, but it deserves wider future investigation as a locus of practice in popular contemporary composition. Finally, interaction with traditional human composers has much remaining to teach researchers.

## Acknowledgments

Catherine Gale designed and led the overall project, ably coordinating the various research teams involved, and her vision is ultimately to thank for the project reaching the stage. Sky Arts were responsible for the majority of the funding, and Wingspan Productions created the TV programs and led organization of the associated theatrical run and media coverage. Archie Baron, as head of Wingspan, and the musical theater specialists Benjamin Till and Nathan Taylor gave unstintingly of their time and energy to negotiate through the implications of the algorithmic musical material. Archie Baron also supplied the data on theatrical attendance and the audience poll included in the article. Simone Tarsitani and James Tate assisted in encoding musical theater songs for the corpus used as a basis for the algorithm. Bob Sturm provided feedback on an earlier draft, and three anonymous reviewers provided later suggestions ahead of the final published version of this article. One reviewer disliked the main title and suggested a few alternatives, including "AIs and Dolls" and "Random of the Opera," and an editor offered "Oh! Calculata!" For the record, a fuller list of puns created for this project includes "Guys an Droids," "Computer Side Story," "Bits," "Phantom of the Musical," "Boot of Mormon," "Joseph and his Technocoder Dreamcode," "Bytety and the Beast," "Miss Cygon," "Programmer on the Roof," "My Fake Lady," "okcoder!," "Crayzy For You," and "Spamabot."

## References

Ames, C. 1987. "Automated Composition in Retrospect: 1956–1986." *Leonardo* 20(2):169–185.

Ames, C., and M. Domino. 1992. "Cybernetic Composer: An Overview." In M. Balaban, K. Ebcioğlu, and O. Laske, eds. *Understanding Music with AI: Perspectives on Music Cognition*. Menlo Park, CA: AAAI Press, pp. 186–205.

Ashby, K. 2016. Personal communication.

Bird, S., E. Klein, and E. Loper. 2009. *Natural Language Processing with Python*. Sebastopol, California: O'Reilly.

Brown, A., T. Gifford, and R. Davidson. 2015. "Techniques for Generative Melodies Inspired by Music Cognition." *Computer Music Journal* 39(1):11–26.

Collins, N., and A. McLean. 2014. "Algorave: Live Performance of Algorithmic Electronic Dance Music." In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 355–358.

Colton, S., et al. 2016. "The *Beyond the Fence* Musical and 'Computer Says Show' Documentary." In *Proceedings of the International Conference on Computational Creativity*. Available online at www.computationalcreativity.net/iccc2016/wp-content/uploads/2016/01/paper_57.pdf. Accessed August 2016.

Davis, H., and S. M. Mohammad. 2014. "Generating Music from Literature." In *Proceedings of the Workshop on Computational Linguistics for Literature*. Available online at arxiv.org/abs/1403.2124. Accessed August 2016.

De Clercq, T., and D. Temperley. 2011. "A Corpus Analysis of Rock Harmony." *Popular Music* 30(1):47–70.

Eigenfeldt, A., and P. Pasquier. 2013. "Considering Vertical and Horizontal Context in Corpus-based Generative Electronic Dance Music." In *Proceedings of the International Conference on Computational Creativity*, pp. 72–78.

Eno, B. 1996. "Generative Music." *In Motion Magazine*. Available online at www.inmotionmagazine.com/eno1.html. Accessed August 2016.

Howarth, P., ed. 2016. "British Musical Theater." Available online at math.boisestate.edu/gas/british. Accessed February 2016.

London, J. 2012. *Hearing in Time: Psychological Aspects of Musical Meter* (2nd ed.). Oxford, UK: Oxford University Press.

Mauch, M., and S. Dixon. 2010. "Approximate Note Transcription for the Improved Identification of Difficult Chords." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 135–140.

Miranda, E. R. 2009. "Preface: Aesthetic Decisions in Computer-Aided Composition." *Contemporary Music Review* 28(2):129–132.

Pachet, F., and P. Roy. 2014. "Imitative Leadsheet Generation with User Constraints." In *Proceedings of the European Conference on Artificial Intelligence*, pp. 1077–1078.

Pearce, M., D. Meredith, and G. Wiggins. 2002. "Motivations and Methodologies for Automation of the Compositional Process." *Musicae Scientiae* 6(2):119–147.

Pearce, M., and G. Wiggins. 2004. "Improved Methods for Statistical Modelling of Monophonic Music." *Journal of New Music Research* 33(4):367–385.

Pinkerton, R. C. 1956. Information Theory and Melody. *Scientific American* 194(2):77–86.

Psenicka, D. 2009. "Automatic Score Generation with FOMUS." In *Proceedings of the International Computer Music Conference*, pp. 69–72.

Simon, I., D. Morris, and S. Basu. 2008. "MySong: Automatic Accompaniment Generation for Vocal Melodies." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 725–734.

Till, B., et al. 2016. *Beyond the Fence Annotation Notes*. Internal report, Wingspan Productions.