Evolutionary Music

Al Biles
Rochester Institute of Technology
www.it.rit.edu/~jab

Overview
- Define music and musical tasks
- Survey of EC musical systems
- In-depth example: GenJam
- Key issues for EC in musical domains

Music
- What is music?
  - Lots of opinions, styles, genres, religions...
  - Music vs. noise
    - “I may not know music, but I know what I like”
    - Usually means, “I like what I know…”
- Two defining characteristics:
  - Music is aural (heard)
  - Music is temporal (happens in real time)
- Music is temporally organized sound

Aspects of Music
- Pitch (not necessarily tonality)
  - Melody: Horizontal (temporal) arrangements
  - Harmony: Vertical (simultaneous) arrangements
- Rhythm (timing, not necessarily a pulse)
  - Temporal sequences, relationships of events
  - Repetition, meter, tempo
- Timbre (any sounds are fair game)
  - Traditional instrument sounds, ambient sounds
  - Computer-generated sounds (anything possible)
- Form (maybe emergent, even random)
  - Structure, organization, conception
  - Hierarchy (multiple levels)
Musical Tasks
- Composition: Create score (abstraction)
- Performance: Realize score in sound
- Synthesis: Generate sounds electronically
- Listening: Derive abstraction from sounds
- Improvisation: Everything simultaneously

EC in Music
- Dates back to 1991
  - Horner and Goldberg: Thematic bridging
  - Gibson and Byrne: NEUROGEN
- Activity increasing rapidly
  - Reviewed over 120 articles for this tutorial
  - EC music class projects appearing on the www

Generative Systems
- Certainly evolutionary, certainly relevant
  - Cellular Automata (music apps since 1980’s)
  - Swarms (emergent behavior, colonies)
  - Artificial Life
  - Sonification of data, DNA (Genetic music)
  - Fractals, chaotic systems (music since 1970’s)
- Not my primary focus, due to time

Survey of EC Applied to Music
- Organized around musical tasks
  - Task analysis of the musical domain
  - Choose subtasks where EC used
- Some representative examples
  - See my Web site for references and links
    - www.it.rit.edu/~jab
- Goals
  - Recruit some new blood
  - Motivate discussion of fundamental EC issues

© 2004 John A. Biles
**EC in Composition**
- First application area (1991)
- Largest application area
- Agenda
  - Describe subtasks of composition
  - Cite some examples
  - Summarize themes and variations

**Composition Subtasks**
- Generate melodies (motives)
  - Generate melodic line (sequence of pitches)
  - Generate rhythm (sequence of durations)
- Develop (extend, enhance) melodies
  - Generate variations
  - Combine motives to create longer lines
  - Generate countermelodies

**Composition Subtasks**
- Harmonization
  - Generate harmony parts (hymns, chorales)
  - Generate harmonic foundation (chord changes)
- Arranging
  - Rhythm section accompaniment
  - Counterpoint
- Structure
  - Generate or adhere to form
  - Generate sections, higher level units

**A Few Examples**
- Horner and Goldberg (1991)
  - Thematic bridging (melody morphing)
  - Bred sequence of operations to transform one motive into another
  - Fitness - hit target, if so check bridge length
- NEUROGEN (Gibson and Byrne, 1991)
  - Rhythm - GA with NN fitness function
  - Add pitch - GA, 2 NN (interval, structure)
  - Harmony - Simple rule base
**variations (Bruce Jacob, 1995)**

- Three components, all GAs
  - Composer - builds phrases from user-supplied motives
  - Ear - Judges the composer’s output (fitness)
  - Arranger - Orders phrases into composition, fitness by user
- Starts at motive level (above notes)
- Co-evolution of Composers and Ears
- Sample: Hegemon-Fibre, 1st movement

**GP-Music (Johanson & Poli, 97)**

- GP melody generator (short, monophonic)
  - Terminals - pitches or rest
  - Functions - musical development
- No real rhythm (all notes same length)
- Fitness
  - Interactive (1-100 rating, pair-wise comparison)
  - Neural nets trained on ratings from interactive runs (1-100 version worked less badly)
- Even toy domains are tricky

**GenDash (Rodney Waschka II)**

- New music composer, not a techie
- GenDash - GA tool he tweaks for each piece (since mid-1990’s)
  - Initial population: 26 measures of music
  - Random selection, crossover at note level
  - All children of each generation heard
  - Around five generations per aria
- Highly collaborative, artistic

**Harmonization - SATB**

- Soprano Alto Tenor Bass (classic four-part)
  - Voicing individual chords and voice leading
  - Standard rule sets exist => automatic fitness
- Basically a scheduling problem (optimize)
  - Represent chord sequence or voice sequences
  - Fitness usually number of constraints violated
- Mixed success
  - Easy if chords specified (more constrained)
  - Harder if chords evolved too (more creative)
Harmonization Examples

  - Melody and chord symbols -> 4-part harmony
  - Broke problem into 2 parts
    - Enumerate all possible voicings for each chord
    - GA to find best sequence of voicings (voice leading)
  - Evolved chords themselves as well
  - More creative, less tractable
  - Rule-based system worked better
- EC probably not the best approach

Rhythm Examples

- Horowitz (1994)
  - Representation - params to generating function
  - One-measure drum textures presented visually
  - Mentor listens, selects favorites to survive/breed
- CONGA (Tokui and Iba, 2000)
  - 4 to 16 measure patterns (user specifies)
  - GA evolves half or one-measure patterns (grid)
  - GP arranges patterns into phrases (hierarchy)
  - Levels evolved separately (mentor switches)
  - Neural net to thin the GA population

Rhythm - Drum Machine

- Generate single-measure or longer patterns
- 2D grid (standard drum machine interface)
  - Time on X axis
  - Instrument on Y axis
  - MIDI velocity in the cells (0-127)
- Build textures
  - Loop one measure
  - Build longer phrases from multiple patterns

SBEAT (Tatsuo Unemi, 2002)

- Currently in third version
- Representation (individuals are measures)
  - 16 events (fixed time grid) X
  - 3 chromosomes (pitch, rhythm, velocity) X
  - Up to 23 parts (13 solo, 2 chord, 8 rhythm)
- Collaborative system - User can
  - Select individuals to breed
  - Manipulate underlying chord/scale
  - Enter and protect parts
  - Arrange measures into score (piece)
### Pitch/Duration Representations

- **Pitch**
  - Absolute pitch (scale degree, MIDI note, Hz)
  - Relative interval
    - From previous pitch
    - From beginning of phrase or composition
    - From tonic of key or root of chord
- **Durations**
  - Beat-oriented (multiples/divisions of beat)
  - Absolute (milliseconds)

### Melody Chromosomes

- **Position-based**
  - Time windows on fixed temporal grid (beats/fractions)
  - Enforces beat/measure/phrase structure
  - Tilts toward beat-oriented music
- **Order-based**
  - Pitch/duration pairs (durations can be arbitrary)
  - Measure lines ignored, superimposed, or irrelevant
  - Facilitates non-pulse music
- **Tree-based** (GP)
  - Terminals usually notes (pitch, maybe duration)
  - Functions usually musical operators
  - Facilitates more complex forms (extend hierarchy)

### Melody Fitness

- **Explicit rules and heuristics**
  - From music theory or hip pocket
  - Usually combined via weighted average
- **Interactive** (human mentor, critic, rater)
  - Display individuals; rater selects and rates
  - Perform in musical context (real-time)
- **Learn from examples** (neural networks)
  - Input either features or melodic fragments
  - Examples come from desired style

### Operators - Initialization

- **Random** - Start from scratch
  - White noise generator
  - Fractals
  - Markov chains
- **Sampled**
  - User supplied motive(s) to develop
  - Licks from analyzed corpus
**Operators - Selection**

- Traditional fitness-based
  - Encourages convergence
  - Can be problem if diversity critical
- Musically aware
  - Look for individuals to fill a role
- Random - no fitness
  - Works if individuals all musically meritorious
  - Maximum diversity

**Crossover and Mutation**

- Is purpose to alter or develop?
  - Alter - more random, less guided
  - Develop - more musically aware
- Crossover point(s)
  - At bit vs. musical boundaries (note, measure)
  - Random vs. musically meaningful
- Mutations
  - Flip bits - likely to be unmusical
  - Musically meaningful - may be too “safe”

**EC in Performance**

- Expressive performance of score not trivial
  - Classical: alter note onsets, length, envelopes
  - Jazz: also alter notes (add, delete, change)
- Annotate jazz performance (Grachten)
  - GA to minimize cost of edit-distance operations to transform score to performance
  - Use training sets of “correct” performances

**Audience Mediated Performance**

- GenJam Populi (more later)
- Sound Gallery (Woolf and Thompson)
  - Artistic installation piece
  - Speakers in corners of room (four islands)
- Each driven by evolving hardware distorting a source sound
  - Fitness: location of patrons (closer is better)
  - Migration to keep people moving
Performance (kind of)

- GA to enhance public speaking voice (Sato)
  - Three “genes” - pitch, volume, speed
  - Fitness - from mentors
  - Not real-time yet...
- HPDJ (Hewlett Packard Disc Jockey)
  - Select tunes, sequence them, do crossfades
  - Fitness: crowd animation level

EC in Synthesis

- Control synthesis algorithms/techniques
- Goal: Higher level (more musical) interface
  - Huge, chaotic parameter spaces
  - Provide guided search through synthesis space
- Two different subtasks
  - Match a target sound
  - Generate new (hopefully interesting) sounds

Matching a Target Sound

- Basically an optimization problem
- Fitness - [perceptual] spectral matching
- GA to evolve parameter settings (Horner)
  - Unit generator (UG) parameters (FM, modular)
  - Additive synthesis envelope breakpoints
  - Wavetable, physical modeling parameters
- CSound Recipes (Horner and Ayres, 2002)
- GP to evolve UG topologies (Garcia, 2001)
- Reverb params - match room (Mrozek, 96)

Search for New Sounds

- Explore a synthesis technique’s sound space
- Fitness - mentor preference
- Goal often collaborative tool for sound designers and composers
- Example - Timbre trees (Takala, 1993)
  - Evolve topology of unit generator patches (GP)
  - Sounds synchronized to animated motion
### Granular Synthesis
- Sound objects made up of 1-100 ms grains
  - Each grain has waveform, pitch, envelope, ...
  - Sound object (cloud) has density, shape, ...
- GA to evolve parameters (Johnson, 99)
  - FOF (formant wave-function) synthesis
  - Evolves parameters for CSound function call

### Emergent Granular Synthesis
- Chaosynth (Miranda, 1995-)
  - CA to control grain parameters
  - As CA self-organizes, sound emerges
- Swarm Granulator (Blackwell, 2003)
  - Swarmer - Swarm is the granular cloud
  - Interpreter - Interprets swarm for granulator
  - Granulator - Sound engine (Max/MSP)
  - Real-time interactive performance

### Synthesizer Control
- Commercial Synthesizers hard to control
- Muta-Synth (Palle Dahlstedt, 2001)
  - Customizable S/W controller for Nord synth
  - Extended to real-time interactive performance
- Genophone (Mandelis, 2002)
  - Evolves sounds and gesture mappings
  - Data glove interface
  - Sends SysEx messages to Korg Prophecy

### Breed Actual Waveforms
- Thesis (Cristyn Magnus, SDSU, 2003)
- Representation
  - Waveform (sample array)
  - Genes: segments bounded by zero crossings
- Operators
  - Crossover and mutations at gene level only
  - Eliminates clicks and pops
- Fitness: Match waveform or amp. envelope
- Piece is evolution of initial to target sounds
**EC in Listening**

- NEXTPITCH (Francine Federman, 2000)
  - LCS to predict next pitch in melody
  - Nursery tunes and chorales (simple melodies)
- Accidental evolution of a radio (Layzell, 02)
  - Trying to evolve a hardware oscillator
  - Got a radio that received oscillations from a nearby computer

**EC Listeners in Composers**

- The EAR in Bruce Jacob’s *variations* system
  - IGA to breed set of “data filters” for harmonies
  - Each filter passes an acceptable chord
- Co-evolved critics (Todd and Werner, 99)
  - “Male singers” (32-note song)
  - “Female critics” prefer certain intervals
  - Female selects male with best intervals
  - Best means most surprising

**EC in Improvisation**

- Compose and perform concurrently (Jazz)
- Spontaneous, real-time, interactive
- Has to be “right” the first time
- Jazz is an inherently evolutionary domain
  - Jam session environment highly competitive
  - Survival of fittest (cutting sessions)
  - Players “borrow” others’ ideas (licks)
- Can even trace lineage of licks and soloists

**Spector and Alpern (1994-5)**

- Toward general case-based artist generator
- Traded bebop fours using GP (not real-time)
  - Terminal set: four-bar phrase from human
  - Function set: 13 melody transforms
  - Evolved programs to transform human four
- Fitness
  - Five features from jazz theory literature
  - Neural net trained on Bird licks
  - Hybrid combination worked best
**Papadopoulos and Wiggins (98)**
- Generate blues chorus, not real-time
- Chromosome - 12-bar blues of 1/16th notes
- Initialization - Random
- Crossover - single and two-point, note level
- Mutation - musically meaningful
- Fitness - 8 features in fixed weighted sum
- Goal: Eliminate subjectivity (EC-neat)
- Best sounding result was human-edited

---

**Swarm Music**
- Tim Blackwell, 2003
- Swarm-based collective improvisation
- Basically Swarm Granulator operating at note level instead of grain level
- Self-organization
- Stigmergy - interact by modifying environs
- “Follow me” from CD Swarm Music

---

**GenJam: An In-Depth Example**
- *GenJam = Genetic Jam*mer (1994 - present)
- Models a jazz improviser (agent of sorts)
- Real-time interactive performance (MIDI)
- Lets a trumpet player work as a single
- Versions for 4/4, 3/4, 5/4, 7/4, 12/8, 16/8
- About 250 tunes in repertoire
- Swing, bebop, cool, Latin, funk, new age
- Performed for last night’s Reception

---

**Interactive GenJam Architecture**
**Representation of a Phrase (GenJam Normal Form)**

<table>
<thead>
<tr>
<th>Phrase Population</th>
<th>Measure Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>23 -12 57 57 11 38</td>
<td>11 6 9 7 0 5 7 8 7 5</td>
</tr>
<tr>
<td>38 4 7 8 7 15 15 15 0</td>
<td>22 9 7 0 5 7 15 15 0</td>
</tr>
</tbody>
</table>

**Chord Scale Mappings**

<table>
<thead>
<tr>
<th>Chord</th>
<th>Scale</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cmaj7</td>
<td>Major (avoid 4th)</td>
<td>C D E G A B</td>
</tr>
<tr>
<td>C7</td>
<td>Mixolydian (avoid 4th)</td>
<td>C D E G A Bb</td>
</tr>
<tr>
<td>Cm7</td>
<td>Minor (avoid 5th)</td>
<td>C D Eb F G Bb</td>
</tr>
<tr>
<td>Cm7b5</td>
<td>Lociarian (avoid 2nd)</td>
<td>C Eb F Gb Ab Bb</td>
</tr>
<tr>
<td>Cdim</td>
<td>W/H Diminished</td>
<td>C D Eb F Gb G A B</td>
</tr>
<tr>
<td>Cs</td>
<td>Lydian Augmented</td>
<td>C D E F# G A B</td>
</tr>
<tr>
<td>C7+</td>
<td>Whole Tone</td>
<td>C D E F# G Bb</td>
</tr>
<tr>
<td>C7#11</td>
<td>Lydian Dominant</td>
<td>C D E F G A Bb</td>
</tr>
<tr>
<td>C7alt</td>
<td>Altered Scale</td>
<td>C Db D# E Gb G# Bb</td>
</tr>
<tr>
<td>C7#9</td>
<td>Mix. #2 (avoid 4th)</td>
<td>C D# E G F A B</td>
</tr>
<tr>
<td>C7b9</td>
<td>Harm Minor V (no 6th)</td>
<td>C Db E F G Bb</td>
</tr>
<tr>
<td>CmMaj7</td>
<td>Melodic Minor</td>
<td>C D Db F G A B</td>
</tr>
<tr>
<td>Cm6</td>
<td>Dorian (avoid 7th)</td>
<td>C D Db F G A</td>
</tr>
<tr>
<td>Cm7b9</td>
<td>Melodic Minor II</td>
<td>C Db Eb F G A Bb</td>
</tr>
<tr>
<td>Cmgaj7#11</td>
<td>Lydian</td>
<td>C D E F G A B</td>
</tr>
<tr>
<td>C7sus</td>
<td>Mixolydian</td>
<td>C D E F G A Bb</td>
</tr>
<tr>
<td>Cmgaj7sus</td>
<td>Major</td>
<td>C D E F G A B</td>
</tr>
<tr>
<td>C7bl</td>
<td>Blues</td>
<td>C Eb F G G Bb</td>
</tr>
</tbody>
</table>

**Example Measure Crossover**

Random, bit-level crossover point

Parent 1: 9 7 0 5 7 8 7 5
Parent 2: 1001 0111 0000 0101 0111 100 0 0111 0101
Child 1: 7 8 7 15 15 15 0
Child 2: 0111 1000 0111 0111 1111 11 1 1111 0000

**GenJam’s Genetic Algorithm**

- Fairly standard GA process for both populations
  - Random initialization
  - Tournament selection - 4 individuals in a family
  - 2 fittest family members become parents
  - Single-point crossover creates 2 kids
  - Musically meaningful mutation until kids are unique
  - 2 kids replace 2 least fit family members
- Replace 50% of each population in breed mode
- Replace worst 4 measures, 3 phrases in tweak
**Musically Meaningful Mutations on Measures**

Standard melodic development techniques

Musically Meaningful Mutations on Measures

![Musically Meaningful Mutations on Measures](image)

**Musically Meaningful Mutations on Phrases**

Operate at measure-pointer level, not bit level

<table>
<thead>
<tr>
<th>Mutation Operator</th>
<th>Mutated Phrase</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>57 57 11 38</td>
<td>Original Phrase</td>
</tr>
<tr>
<td>Rotate Right Random</td>
<td>57 11 38 57</td>
<td>3 positions in this case</td>
</tr>
<tr>
<td>Reverse</td>
<td>38 11 57 57</td>
<td>Play measures in reverse order</td>
</tr>
<tr>
<td>True Retrograde</td>
<td>38 11 57 57</td>
<td>Play measures backward too</td>
</tr>
<tr>
<td>Sequence Phrase</td>
<td>57 57 38 38</td>
<td>Repeat a measure</td>
</tr>
<tr>
<td>Genetic Repair</td>
<td>57 57 11 22</td>
<td>Replace worst measure</td>
</tr>
<tr>
<td>Super Phrase</td>
<td>55 13 21 34</td>
<td>Winners of fitness tournaments</td>
</tr>
<tr>
<td>Lick Thinner</td>
<td>22 57 11 38</td>
<td>Replace most common measure</td>
</tr>
<tr>
<td>Orphan Phrase</td>
<td>43 37 53 19</td>
<td>Losers of frequency tournaments</td>
</tr>
</tbody>
</table>

**Intelligent Genetic Operators**

- GA’s usually have dumb operators, smart fitness
- Rely on fitness to guide search
- Leads to fitness bottleneck in IGAs, especially temporal
- GenJam currently uses smart operators
  - Intelligent mutation - Already seen
  - Intelligent initialization - Fractals & Markov chains
  - Intelligent crossover - Preserve horizontal intervals
- Good parents tend to have good children
- Reduces volume through the fitness bottleneck

**GenJam Generations Demo**

- Old GenJam version - improvise 4 choruses
- Tune is Tadd Dameron’s *Lady Bird*
- 16-bar form, straight up rhythm
- Each chorus uses a more mature generation
  - 1st - Generation 0, white noise generator
  - 2nd - Gen 1
  - 3rd - Gen 2
  - 4th - Gen 3

© 2004 John A. Biles
**Real-Time Interaction**

- When GenJam trades fours with human
  - Listen to human’s four (Roland GI-10)
  - Map human phrase to GJNF chromosomes
  - Mutate the phrase and 4 measures
  - Play mutated result as its response
- Use mutation as melodic development
- Results in true conversation
- Highly robust and formidable opponent

**Fault Tolerant Pitch Tracking**

- Pitch tracker makes lots of mistakes
  - Wrong pitch
  - Extra note-on events
  - Extra note-off events
- Not a problem
  - Map to GJNF, which is highly robust
  - Errors not mistakes, they’re “development”
  - Will mutate anyway before playing

**Anatomy of a Four**

I played quote from Prince Albert

GenJam “heard” this from pitch tracker

GenJam mutated and played this back

**Collective Improvisation**

- GenJam and human solo simultaneously
- GenJam listens to human while it’s soloing
- Maps to GJNF
- Plays what human did earlier (delay line)
  - Delay of 1 bar, or n events, 4 bars (smart echo)
  - No mutation - Replay as close as possible
- Human can trade 1’s, play harmony, counterpoint
- Challenge for the human!
**Making GenJam Autonomous**

- GenJam more fun when interactive
  - Fitness not necessary or even possible
  - Good human four -> good GenJam four
  - Initialization is very smart
- GenJam’s full-chorus solos not as good
  - Ideas competent but seldom compelling
  - Initialization not smart enough
  - Move to an autonomous GenJam

**Initialize from Stored Licks**

- Licks Databases (several styles)
  - 4-bar licks come from *1001 Jazz Licks*
  - Map to GJNF by hand
- Initialization algorithm
  - Select 16 4-bar licks from database
  - Seed measure pop with those 64 measures
  - First 16 phrases are the 16 original licks
  - Remaining 32 phrases are smart crossovers

**Evolve Soloist Interactively**

- As human solos, map measures to GJNF
- If a human measure is “good enough”
  - Select measure that best matches endpoints
  - Do intelligent crossover with new measure
  - Pick child that best matches endpoints
  - Replace the parent measure with that child
- Evolves soloist toward human’s solo
**What happened to Fitness?**

- Fitness considered necessary for a GA
- View EC as generate-and-test strategy
  - Generate: Initialize, recombine, mutate
  - Test: Fitness
- Usually generators dumb, fitness smart
- GenJam’s generators are smart
  - Intelligence distributed over generators
  - Nothing left for fitness to do, so eliminate it!
- If generators are good, no need to test

**GenJam in Lake Wobegon**

Where the old licks are strong,  
the new licks sound good,  
and all the children are above average!

**Is GenJam Still an [I]GA?**

- If a GA falls in the forest, and there’s nobody there to provide fitness, is it still Evolutionary Computation?

**No, it’s not!**

- No more Mentor (there goes the “I” part)
- No longer any explicit fitness at all
- No generational search
- No real search at all
- It’s just a fancy melodic transducer!
Yes, it is!
- Employs the evolutionary paradigm
- Uses chromosome (string) representations
- Does genotype -> phenotype mapping
- Uses selection, recombination, mutation
- Generates offspring
- Fitness in deciding whether to breed human and soloist measures, which measures
- I got invited to GECCO…

Big Picture Issues
- What to consider in applying EC to music
- How does music domain bend EC
- Advice to those making music with EC
- Summarize with sweeping generalities

Traditional vs. Musical Domains
- Solve a problem vs. Generate content
- Best vs. Better (maybe just different)
- No such thing as “the best” piece
- Fitness absolute vs. relative
- Fitness objective vs. subjective
- Individuals compete vs. Connect
- Convergence vs. Diversity

Optimization vs. Exploration
- Noticed by many (Todd and Werner, 1999)
- Lewis and Clark analogy
  - Searched for (non-existent) northwest passage
  - Ended up exploring the west (more valuable)
- Usually want to explore a musical space, not optimize it
What are you trying to do?

- Study EC vs. make good music
- Scientist/engineer vs. Artist
- Neat vs. Scruffy dimension from AI in 80’s
  - Neats - Model human intelligence
    - Focus on EC purity (don’t cheat)
    - Goal: Show EC can do what people do (be creative)
  - Scruffies - Solve real problems
    - Use EC as one of many tools (hybrid systems)
    - Goal: Make good music

Fitness Issues

- Easy in a few (optimization) domains
- Harder in creative domains
- Hard to code “that sounds good”
- Just because you can compute it doesn’t mean it’s useful as fitness
- Subjective isn’t bad
- If can’t code it, use human fitness function

Revisit Fitness Approaches

- Automatic
  - Rule-based (heuristics)
- Learned
  - Neural Networks
  - Statistical
- Interactive
  - Explicit feedback from one or more mentors
  - Indirect feedback from an audience
- None

Fitness: Heuristic Features

- Dozens of features proposed/used (Towsey 01)
  - Pitch - variety, range
  - Tonal - in key, non-scale, dissonant intervals
  - Melodic contour - direction, stability, interval size
  - Rhythmic - note/rest density, variety, syncopation
  - Patterns - repeated pitch, rhythm patterns
  - Statistical adherence to Zipf’s law
  - Etc.
- Difference polynomials (often brittle)
**Fitness: Rule-Based**
- Knowledge-based (music theory)
- “Theoretically correct” may sound lousy
  - Theory should explain why something sounds good
- Theory should not decide whether something sounds good
- Limit creative options (style enforcement)

**Fitness: Neural Nets**
- Example-based (training set important)
- Input layer
  - Muscular objects themselves
  - Feature vectors derived from objects
- Seldom seems to work
  - Seldom generalizes
  - Features don’t capture the essence
  - Context of objects ignored

**Fitness: Interactive**
- Most common method in creative domains
- If it’s a judgment, let the human judge
- Central problem: Fitness Bottleneck
  - Mentor must experience all individuals
  - Temporal => can’t experience in parallel
  - Must experience in real time
  - Hard to listen that closely, critically
  - Fatigue a big issue
- However, EC can absorb noisy fitness

**Mentor’s Interface**
- Facilitate mentor’s task
- Usability is primary issue (Takaga, this AM)
- Presentation of individuals must be musically valid (in musical context)
- Mentor should be focusing on the music, not the interface
**Representation**

- Only represent what you want to hear
- Don’t represent music you don’t want to hear
- Don’t represent all possible sounds unless you want to hear all possible sounds
- Decide on genre and taylor representation to that genre

**Initialization**

- White noise generators - often too random
- Pink noise
- Fractal/chaos generators
- Markov process
- User-generated objects
- “Greatest hits” from a corpus
- Random ≠ Creative (most of the time)

**Diversity is Essential**

- Convergence can be disastrous
  - “The lick that ate my solo”
  - Can make a good individual sound bad
- Encourage diversity with
  - Operators
  - Co-evolution
  - Speciation, islands
  - No fitness

**Don’t use EC for everything**

- EC as a solution in search of a problem
- Hybrid systems usually better
- Rules, neural nets, heuristics, procedures, user collaboration are all okay
- Only evolve what you have to
**KISS**

- Simple & robust trumps complex & brittle
- Always competent trumps occasionally brilliant
- Start with simple
- Only get complex if you’re out of simple

**Constraints are good!**

- Stylistic constraints can be positive
- Sticking to a genre isn’t an artistic cop-out if you like the genre
- “Freedom” means a bigger search space
- Meeting an audience’s expectations isn’t bad, especially if you want to get gigs...

**Set the bar at the right level**

- Don’t set the bar too low
  - I think we’ve nailed nursery tunes
  - Toy domains are great for class projects, but solutions seldom scale up
- Don’t set the bar too high
  - Don’t try to solve the “western tonal music” problem
  - Pick a doable task to focus on

**Who’s your audience?**

- Audience as users
  - Listeners build mental model of performance
  - Model enables expectations in performance
  - Adhering to rules meets expectations
  - Breaking rules is a surprise
  - Must balance to engage listener
  - Can engage listener with audience-mediated performance
**Listen to the music!**

- Just because it generated notes doesn’t mean it was successful
- Listen to it with fresh ears (or have fresh ears listen to it)
- If you heard it on the radio, would you change the channel?

**Greatest Hits**

- Bentley and Corne, *Creative Evolutionary Systems*, Morgan Kaufmann, 2001
- Burton and Vladimirova, *CMJ*, 23(2), Summer, 1999
- Lots of links: www.it.rit.edu/~jab