Generative Art via Grammatical Evolution

Erik Fredericks, Abigail Diller, and Jared Moore
GI2023 ➔ May 20, 2023
Motivation
Background - Generative Art

Visualization of algorithms and/or mathematical functions [1-5]
- Creative coding [11-15]
- Real-world displays [16]

\[ y = 75.0 \times \sin(x) \]

https://p5js.org/examples/math-sine-wave.html
Background - Generative Art

Visualization of mathematical functions in aesthetically-pleasing ways

\[ y = 75.0 \times \sin(x) \]

https://p5js.org/examples/math-sine-wave.html
Included techniques

Stippling
Cellular automata
Circle packing
Flow fields (two implementations)
Drunkard's walk
Dithering
Pixel sorting
– any technique is viable!

https://editor.p5js.org/frederer/sketches/DbWjEErKy
Background - Grammatical Evolution

Subset of evolutionary computation (generally genetic programming-based) [24]
- Grammar-based genome [6-7]
- Uses very similar genetic operators (e.g., mutation, crossover, etc.)

Effective at constraining the solution space
- Target for genetic improvement! [7]

rules = {
    'ordered_pattern': ['#techniques#'],
    'techniques': ['#technique#', '#techniques#', '#technique#'],
    'technique': ['stippledBG', 'flowField'],
    ...
}

Tracery [9] grammar
Background - Grammars

Top-level rule is "flattened" and production rules expanded
- Tracery enables randomness (among other things)

```
rules = {
    'ordered_pattern': ['#techniques#'],
    'techniques': ['#technique#', '#techniques#, #technique#'],
    'technique': ['stippledBG', 'flowField'],
    ...
}
```

Tracery [9] grammar

```
stippledBG(params\textsubscript{a}), stippledBG(params\textsubscript{b}), flowField(params\textsubscript{c}), stippledBG(params\textsubscript{d})
```

Expanded output
Tracery [9]

http://www.crystalcodepalace.com/traceryTut.html
Background - Lexicase Selection [10]

Many-objective selection operator
- Not pareto-based!

Evaluates individuals on an objective-by-objective basis

Each selection:
- Sample of population taken
- Comparison on first objective
  - If one individual better, selected
  - Else, advance to next objective
- If all objectives exhausted
  - Random selection

$\epsilon$-Lexicase selection [30]
- Individuals tied if within $\epsilon$
- Important for real-valued fitness objectives
  (observable output may not change)

$\epsilon = 0.85$
Project

Apply GI techniques to **optimize the grammar** defining the order and parameters of a set of generative art techniques

- I like glitch art, which makes for a lovely (and naive) proof of concept
  - (Pixel sorting below ➔ [https://github.com/satyarth/pixelsort](https://github.com/satyarth/pixelsort))
Approach
(1) Convert Techniques to Grammar

Suite of generative art techniques required as input

def flowField(type, palette, zoom):
    # type: ['edgy', 'curves']
    # palette: list of colors
    # zoom: float between 0.001 and 0.500

'flow-field' : '#flow-field-type#:#palette#:#flow-field-zoom#',
'flow-field-type' : ['edgy', 'curves'],
'flow-field-zoom': [str(x) for x in np.arange(0.001, 0.5, 0.001)],
...
(2) Translate Technique

Each generative art technique must also be translated to framework requirements

E.g., flow field must (minimally) accept a Pillow Image object

def flowField(image, type, palette, zoom):
    # image: PIL image object
    # type: ['edgy', 'curves']
    # palette: list of colors
    # zoom: float between 0.001 and 0.500
(3) Configure Search

Selection operator:
- Many-objective (Lexicase selection)
- Single-objective (Tournament selection)
- Random (No selection)

Standard configurable parameters
- E.g., population size, mutation rate, etc.
(3) Configure Search

Fitness functions

\[ f_{\min(\text{genome})} \] : minimize duplicate genes
\[ f_{\max(\text{techniques})} \] : maximizing diversity of included techniques
\[ f_{\max(\text{RMS|Chebyshev})} \] : maximize pixel differences between images

Many-objective search uses all four fitness functions

Single-objective search only uses \[ f_{\max(\text{RMS})} \]
(4) Execute Search

Search executed according to (3)
- Each genome evaluated on flattened grammar

- *For this work*, image object **not** cleared
  - Subject of future work

Depending on configuration
- Many-objective: Lexicase
- Single-objective: Tournament selection
- Random: No selection
(5) Output Best Images

Final population of image objects stored to disk upon completion
## Experiment Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental replicates</td>
<td>10</td>
</tr>
<tr>
<td>Image size (pixels)</td>
<td>1000 x 1000</td>
</tr>
<tr>
<td>Number of generative techniques</td>
<td>8</td>
</tr>
<tr>
<td>Generations</td>
<td>100</td>
</tr>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.5</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.4</td>
</tr>
<tr>
<td>Number of Lexicase objectives</td>
<td>4</td>
</tr>
<tr>
<td>$\epsilon$ (Lexicase - many-objective)</td>
<td>0.85</td>
</tr>
</tbody>
</table>

**TABLE I: Evolutionary parameter configuration.**
Results - Novelty

Wilcoxon rank-sum test with Bonferroni correction

Lexicase vs. random (p < 0.01)
Single-objective vs. random (p < 0.03)
Lexicase vs. single-objective (p > 0.03)
Results - Gene Uniqueness

Wilcoxon rank-sum test with Bonferroni correction

Lexicase vs. random (p < 0.001)
Single-objective vs. random (p < 0.001)
Lexicase vs. single-objective (p > 0.001)
Discussion / Sample Outputs

Single- and many-objective both tended to converge towards both multiple and common suites of techniques
- On a 'per run' basis
- Resulting from maximizing pixel differences and maximizing the number of techniques

Random search tended towards 'blank space'
- Fewer techniques in genome

Lexicase tended to converge to a smaller set of techniques with common outputs
Related Work

Generative art via artificial intelligence
- Extremely popular right now thanks to large language models!
  - DALL-E, Midjourney, Stable Diffusion, VQGAN+CLIP, etc [15,35,36].
  - All require a massive dataset and massive amount of computing power
- GenerativeGI only requires a suite of techniques and computing necessary for evolutionary search

https://www.reddit.com/r/StableDiffusion/comments/yxtdrh
Related Work

Generative art via search heuristics
- Often used in visualization and creative coding domains
- Visualize 3D models of mathematical formulae [4]
- Creating environments within game worlds [5]
  - GenerativeGI focuses on fine-tuned control over artistic techniques
- Taxonomy of fitness metrics for evolutionary art/music [8]
  - Metrics can be non-trivial to evaluate (e.g., human preference)
  - Target for future work of this paper
Future Work

Additional fitness functions
- Guide towards specific outputs

Human in the loop
- What constitutes a "good" output?
- How do you measure aesthetic preference?

Merging artistic techniques
- How can distinct techniques provide a seamless output?
  - E.g., a flow field into an automaton
Thanks to..

Award 80NSSC20M0124
References


References