Parameterized versus Generative Representations in Structural Design

An Empirical Comparison

Rafal Kicinger
Tomasz Arciszewski
Kenneth De Jong
Outline

- Motivation
- Research Questions
- Research Methodology
- Experimental Results
- Conclusions
- Further Work
Motivation

- Design representations - one of the key aspects of any computational design activity
  - Describe their form, elements, etc.
  - Incorporate domain-specific knowledge
  - Determine the space in which solutions are sought

- Traditionally – simple direct encoding of structural members in a linear genome (*parameterized* representations)

```
600000 5112210011 010001110011
| bracings   | beams     | supports |
```
Motivation

Generative representations
Motivation

We still not sure what type of representation to use in a given design situation:

- Parameterized – successfully applied to many design optimization problems, but ...
  - Problems with scalability
  - Generally not sufficient when novel design concepts are sought

- Generative – improved scalability, but...
  - not well understood
  - very few design applications
Research Questions

- For what types of structural design problems should, or should not, generative representations be used?
- What are qualitative and quantitative differences between generative and parameterized representations?
Two classes of complex structural design problems investigated:

- Conceptual design of a **wind bracing system** in a tall building (Problem I)
- Conceptual design of an **entire steel structural system** in a tall building (Problem II)
Research Methodology: Design Problems

Three subclasses of Problem I were considered:

- Design of a wind bracing system composed of simple X bracing and no bracing – empty cell (Problem Ia)
- Design of a wind bracing system composed of K bracing and no bracing (Problem Ib)
- Design of a wind bracing system composed of 7 types of bracings (Problem Ic)
Research Methodology: Representations

Two types of design representations investigated:

- **Parameterized** – direct mapping between structural elements and genes
  
  ![Parameterized example](image)

- **Generative** – designs are generated, or grown from initial configurations (design embryos) using 1D cellular automata (design rules) (Kicinger 2004, Kicinger et al. 2004a, 2004b)
  
  ![Generative example](image)
Research Methodology: Representations

Example: Generative representation for Problems Ia and Ib

- Design embryo
- Design rule

Process of generating a wind bracing system from its representation

standard 1D CA rule

or totalistic 1D CA rule
Research Methodology: Representations

Generative representation → Growth process → Design concept → Application of loads → Structural analysis and sizing optimization → Detailed design

Assign fitness
Assignment of loads
Total weight

GECCO'05, Washington, DC - Copyright Rafal Kicinger
Experimental Results: Mutation Rates

Lower mutation rates produced better results for **parameterized** representations:

**Problem Ia**
(Similar results obtained for all problems)
Experimental Results: Mutation Rates

Pattern obtained for parameterized representations

Problem Ia

(Similar results obtained for all problems)
Experimental Results: Mutation Rates

Dramatically different pattern obtained for **generative** representations (standard and totalistic CAs)

**Problem Ia**

(Similar results obtained for all problems)
Experimental Results: Average Performance

Generative representations (standard CAs and totalistic CAs) outperformed parameterized representations and produced high-performance design much faster.

**Problem II**
(Similar results obtained for problems Ib and Ic)
Experimental Results: Differences

- **Quantitative Differences:**
  - Generative representations based on *standard CAs* produced *6-12% better* results than parameterized representations for problems *Ib, Ic*, and *II*
  - Generative representations based on *totalistic CAs* produced *12-20% better* results than parameterized representations for problems *Ib, Ic*, and *II*

- **Qualitative Differences:**
  - Evolutionary processes with generative representations identified high-performance solutions *much faster* than with parameterized representations
    - totalistic CAs were the fastest in generating high-performance solutions for problems *Ib, Ic*, and *II*
  - Generated design concepts showed *qualitatively different shaping patterns* than those produced by parameterized representations
Experimental Results: Shaping Patterns

Design concepts (Problem Ic) produced using generative and parameterized representations

Generative (standard and totalistic CAs)  Parameterized
Experimental Results: Problem Types

Types of structural design problems for which generative representations based on cellular automata may be used:

- Composed of a relatively large number of identical, simple, and locally interacting elements (e.g. wind bracings)
- High-performance solutions exhibit some regularity (patterns)
- Design evaluation process is computationally expensive
Experimental Results: Problem Types

Limitation:

High-performance solutions exhibit too complex patterns than cannot be attained by iteration of simple standard and totalistic CAs

Problem Ia

Parameterized
(520,349 lbs.)

CA
(547,428 lbs.)
Conclusions

Generative representations based on cellular automata:

- Scale well with the geometry of a design problem but NOT with the number of cell values
  - Solution: use totalistic CAs
- Good for problem domains where some regularity/patterns are expected, or desired
  - BUT...
  - may not generate solutions exhibiting too complex patterns
  - Solution: Use more sophisticated types of cellular automata representations (future work)
Conclusions

Identified differences between parameterized and generative representations:

- CAs produce quantitatively better solutions (6-20% average performance improvement)
- CAs find these high-performance solutions faster
- CAs produce qualitatively different shaping patterns, including majority of patterns found in existing structural systems
Future Work

- More sophisticated types of generative encodings based on cellular automata:
  - CAs with a self-adaptation mechanism
  - Non-uniform CAs
  - ...

- Other structural design problems:
  - Discrete problems
    - space structures
    - bridges
    - ...
  - Continuum problems (finite elements)
    - plates
    - shells
Backup Slides
Elementary Cellular Automaton

design rule (1D CA rule)

a)

b)

design embryo

t=0

t=1

t=2

References


Design Experiments

Design Problem Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of stories</td>
<td>30</td>
</tr>
<tr>
<td>Number of bays</td>
<td>5</td>
</tr>
<tr>
<td>Bay width</td>
<td>20 feet (6.01 m)</td>
</tr>
<tr>
<td>Story height</td>
<td>14 feet (4.27 m)</td>
</tr>
<tr>
<td>Structural analysis method</td>
<td>1\textsuperscript{st} order</td>
</tr>
<tr>
<td>Beams</td>
<td>pinned, fixed (Problem II)</td>
</tr>
<tr>
<td>Column</td>
<td>fixed</td>
</tr>
<tr>
<td>Supports</td>
<td>pinned, fixed (Problem II)</td>
</tr>
<tr>
<td>Wind bracings</td>
<td>no, diagonal (/), diagonal (), K, V, simple X, and X</td>
</tr>
</tbody>
</table>
Design Experiments

Representational Parameters

<table>
<thead>
<tr>
<th>Parameterized:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- number of gene values</td>
<td>2, or 7</td>
</tr>
<tr>
<td>- length</td>
<td>150(Ia, Ib, Ic), or 306(II)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generative:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- type of CAs</td>
<td>standard, or totalistic</td>
</tr>
<tr>
<td>- radius of the local</td>
<td>1, or 2</td>
</tr>
<tr>
<td>neighborhood</td>
<td></td>
</tr>
<tr>
<td>- CA boundaries</td>
<td>periodic</td>
</tr>
<tr>
<td>- number of cell values</td>
<td>2, or 7</td>
</tr>
<tr>
<td>- length</td>
<td></td>
</tr>
<tr>
<td>standard CAs, radius 1</td>
<td>13(Ia, Ib), 348(Ic), or 367(II)</td>
</tr>
<tr>
<td>standard CAs, radius 2</td>
<td>37(Ia, Ib), 16812(Ic), or 16855(II)</td>
</tr>
<tr>
<td>totalistic CAs, radius 1</td>
<td>9(Ia, Ib), 24(Ic), or 39(II)</td>
</tr>
<tr>
<td>totalistic CAs, radius 2</td>
<td>11(Ia, Ib), 36(Ic), or 53(II)</td>
</tr>
</tbody>
</table>
## Design Experiments

### EC Parameters:

<table>
<thead>
<tr>
<th>EA</th>
<th>ES, GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pop. sizes (parent, offspring)</td>
<td>(1,5), (1,25), (5,25), (5,125) or (50,250) for ES (5,25), or (50,50) for GA</td>
</tr>
<tr>
<td>Generational model</td>
<td>Overlapping for ES(μ+λ), Nonoverlapping for ES(μ,λ) and GA</td>
</tr>
<tr>
<td>Selection (parent, survival)</td>
<td>(uniform stoch., truncation) for ES, (fitness prop., uniform stoch.) for GA</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.025, 0.1, 0.3, or 0.5</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0, 0.2, or 0.5</td>
</tr>
<tr>
<td>Fitness</td>
<td>Total weight of the structural system (determined by the 1st-order analysis)</td>
</tr>
<tr>
<td>Initial. method</td>
<td>random</td>
</tr>
<tr>
<td>Constraint handling method</td>
<td>death penalty (infeasible designs assigned 0 fitness)</td>
</tr>
<tr>
<td>Termination criterion</td>
<td>1000 evaluations (short-term), or 10,000 evaluations (long-term)</td>
</tr>
<tr>
<td>Number of runs</td>
<td>5 (in each experiment)</td>
</tr>
</tbody>
</table>
Experimental Results: RQ3

Generational Model:

- No differences in performance between ES($\mu + \lambda$) and ES($\mu, \lambda$)
Experimental Results: RQ3

Population sizes:
- Tiny and small population sizes produced comparable results, large population sizes produced inferior results for parameterized representations (Problem II)
Experimental Results: RQ3

Population sizes:
- Small population sizes produced best results, tiny and large population sizes produced inferior results for generative representations (Problem II)
Experimental Results: RQ3

Evolutionary Algorithm:
- ES produced better results than GAs (Problem Ia)
Design Embryo

A design embryo is understood as an ordered set of cell values representing an initial configuration (here one-dimensional) of structural members (e.g. types of wind bracing, beams, columns) from which a design concept is developed.
Design Rule

- A *design rule* is a formal description of a transformation that changes the current configuration of structural members into a new configuration.
- This transformation defines a unit time step.

![Diagram](image.png)
### Generative Representation

<table>
<thead>
<tr>
<th>Wind bracing subsystem</th>
<th>Beam subsystem</th>
<th>Supports</th>
</tr>
</thead>
<tbody>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
<tr>
<td>design rule</td>
<td>design rule</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>design embryo (bracings)</th>
<th>design rule (bracings)</th>
<th>design embryo (beams)</th>
<th>design rule (beams)</th>
<th>support configuration</th>
</tr>
</thead>
</table>

**Note:** The diagram illustrates the relationship between wind bracing subsystems and beam subsystems, indicating how design rules and support configurations are interconnected.
Conclusions

- EC parameters that have had the largest impact on the success of evolutionary design processes:
  - Type of EA
    - ES produced significantly better results than GAs
  - Parent population size
    - Moderate sizes (e.g. 5) produced best results
  - Mutation rate
    - Low for parameterized (e.g. 0.025)
    - High for generative (e.g. 0.3)