Towards a Geometric Theory of Interactive Domains

Anthony Bucci
DEMO Lab, Brandeis University, Waltham MA USA

http://www.cs.brandeis.edu/~abucci/
http://demo.cs.brandeis.edu/
Coevolution and Interaction

- Coevolution research is concerned with finding good solutions in problems lacking objective measures of goodness.
- Typically, there is an interaction available: solutions can interact with something and we can see what happens.
- The question is how to use that information to perform search.
Interactive Domains

Formally, an interactive domain consists of:

\[ S \text{ set of candidate solutions} \]
\[ T \text{ set of tests} \]
\[ R \text{ ordered set of outcomes} \]
\[ p : S \times T \rightarrow R \]

function encoding the interactions
Example Interactive Domains

Chess

\[ S = T = \{ \text{chess–playing strategies} \} \]
\[ R = \{ \text{lose} < \text{draw} < \text{win} \} \]

\( p \) outputs how player 1 does against player 2.

Multiobjective optimization

\[ S = \{ \text{candidate solutions} \} \]
\[ T = \{ f_1, f_2, \ldots, f_n \} \]
\[ R = \mathbb{R} \]

\( p(s,f_i) = f_i(s) \)
Ideally, a coevolutionary algorithm searching an interactive domain will enter an *arms race dynamics*. In this, all individuals are getting better at the task. Each time one individual improves, the others improve too.
The Arms Race
The Arms Race
The Arms Race
**Pitfalls**

**Intransitivity (Cycling; Mediocre Stable States)**
Instead of continually improving, individuals cycle through suboptimal solutions.

**Overspecialization (Focusing)**
Individuals improve on one *dimension of performance* at the expense of others.
What Is a Dimension of Performance?

In multiobjective optimization, it is clear: the objectives are the dimensions which matter.

But in a general interactive domain, it is not clear what the dimensions are.
Motivating a Geometric Viewpoint

The multiobjective example suggests thinking about all interactive domains in a geometric way. Once we see how to do this precisely, we can reinterpret our ideal and pitfalls as precise definitions rather than vague intuitions. The definitions relate closely to the geometry of the domain.
**Ideals and Pitfalls from Geometric Viewpoint**

**Arms race:** when an algorithm progresses on all dimensions of performance

**Overspecialization:** when an algorithm progresses on only some of the dimensions of performance

**Cycling:** when an algorithm switches between making progress on two dimensions, regressing on one while progressing on the other
Every interactive domain has a natural partial ordering of the individuals, inspired by Pareto dominance in multiobjective optimization (Bucci and Pollack, 2003). For example:

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_2$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$s_4$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Pareto Dominance, II

Putting all the candidates in order this way yields the partial order on the right:

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_2$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$s_4$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Method 1: Poset Decomposition Theorem

Theorem  Any (finite) partial order can be decomposed into the (finite) intersection of linear orders over the same base set.

Example

is realized by the two linear orders $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4$ and $s_1 \rightarrow s_3 \rightarrow s_2 \rightarrow s_4$
Spatial Embedding from Poset Decomposition

The linear orders $L_1 = s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4$ and $L_2 = s_1 \rightarrow s_3 \rightarrow s_2 \rightarrow s_4$ yield a coordinate system:

<table>
<thead>
<tr>
<th></th>
<th>$L_1$</th>
<th>$L_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$s_2$</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$s_3$</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$s_4$</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

![Diagram showing the spatial embedding of elements $s_1$ to $s_4$.]
The dimensions of this embedding are the linear realizers of the ordering of the candidates. The decomposition theorem guarantees the candidate order is preserved. Unfortunately, it costs exponential time in the size of $S$. In practice we apply this to a population much smaller than $S$. Perhaps we can use smaller populations with the information the embedding gives; or, pay the high cost once and design an online algorithm.
Method 2: Intrinsic Coordinate Systems

This method applies when the outputs of $p$ are binary. From $p$ we can define a DAG of the tests:

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_2$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$s_4$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
We remove the "redundant" tests, which show up in the graph as having two in−arrows, and add formal tests at "infinity":

\[
\begin{align*}
\text{Intrinsic Coordinate Systems, II} \\
\text{We remove the "redundant" tests, which show up in the graph as having two in−arrows, and add formal tests at "infinity":}
\end{align*}
\]
**Theorem**  Minus an "origin" test (like $t_1$), this graph will consist of unbranching chains of tests.

Each of these chains is a dimension. It has the property illustrated below:

$$p(s,t)$$

$$u \rightarrow u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow u_4 \rightarrow u \rightarrow t$$

$u_3$ is the coordinate of $s$. 
We can thus use these dimensions to embed the candidates into a space:

<table>
<thead>
<tr>
<th></th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_2$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$s_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$s_4$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Theorem**  *The ordering of the candidates in this embedding is the same as the original ordering.*
Intrinsic Coordinate Systems

Highlights

- Finding minimal-sized coordinate systems might be expensive, but polynomial-time algorithms exist to find suboptimal ones (Bucci, Pollack and de Jong, in review)
- Might be possible to use smaller populations or design an online algorithm to update existing coordinate systems
- Applies to domains with binary outcomes, but we can convert any domain into several new domains of this form.
Regarding Coevolutionary Algorithms

- Both methods suggest we separately select tests for being "good" on dimensions. This is not the same as pressuring candidates to be good at the task.

- Might pressure tests to further discriminate candidates, giving a more "zoomed" coordinate system.

- Losing a dimension can lead to both cycling and overspecialization and should be avoided.
Summary

- Argued that a geometric viewpoint offers a precise, uniform treatment of common ideals and pitfalls in coevolution.
- The viewpoint offers intuitions about interactive domains, coming from dimensions, which might be absent otherwise.
- Gave two concrete ways of embedding an interactive domain into a space.
- Gave suggestions for improving coevolutionary algorithms.