An Analysis of Dimensionality Reduction Techniques for Visualizing Evolution

Andrea De Lorenzo\textsuperscript{1}, Eric Medvet\textsuperscript{1}, Tea Tušar\textsuperscript{2}, Alberto Bartoli\textsuperscript{1}

\textsuperscript{1}: DIA, University of Trieste, Italy
\textsuperscript{2}: DIS, Jožef Stefan Institute, Ljubljana, Slovenia

VizGEC (@GECCO), 14/7/2019, Prague (Czech Republic)

http://machinelearning.inginf.units.it
Visualization

Visualization:
- enables understanding
- fosters interest
- engages imagination
Landscape

**landscape**

/ˌland(ə)skæp/

**noun**

1. all the visible features of an area of land, often considered in terms of their aesthetic appeal.

"the soft colours of the Northumbrian landscape"
In population-, search-based optimization (and EC):

- where are the solutions? where are they going?
- is the optimization method (or EA) properly driving the population of solutions?
The problem of dimensionality

Problem:
- we can show/visualize data in 2D, at most 3D
- “many” interesting problems have much larger dimensionality

⇒ dimensionality reduction
The problem of dimensionality

Problem:

- we can show/visualize data in 2D, at most 3D
- “many” interesting problems have much larger dimensionality

⇒ dimensionality reduction

- Can we design a general visualization framework for any EA and problem?
- How does dimensionality reduction affect/impact usefulness of visualization?
Table of Contents

1. Scenario

2. Evaluation

3. Wrap-up
General EA/problem

- $S$: search space
  - may be discrete or continuous
- $P_i$: population at the $i$-th generation
  - as a sequence, i.e., solutions sorted according some criterion
- $P_1, \ldots, P_{n_{\text{gen}}}$: an evolutionary run

Goal: **visualizing the evolutionary run** in 2D
**Dimensionality reduction**

\[ m : S^h \rightarrow (\mathbb{R} \times \mathbb{R})^h : \text{a dimensionality reduction} \] function mapping a sequence of solutions \( s_1, \ldots, s_h \) to a sequence of 2D points

- given a point \( s \), its mapping \((x, y)\) depends, in general, on all the solutions in the sequence
- \( S \) can be any space
General framework for the visualization

1. run EA and collect data \((P_1, \ldots, P_{n_{gen}})\)
2. concatenate generations obtaining \(s_1, \ldots, s_h\) (with \(h = \sum_i |P_i|\))
3. map to 2D obtaining \((x_1, y_1), \ldots, (x_h, y_h)\)
4. split sequence of 2D points in subsequences corresponding to generations
   - there is a 2D point for each solution of each generation
5. plot 2D points
Example: 3 solutions evolving for 30 generations

$P_1$, $P_2$, $P_{30}$

$s_1, s_2, s_3, s_4, s_5, s_6, \ldots, \ldots, \ldots, s_{88}, s_{89}, s_{90}$
Example: 3 solutions evolving for 30 generations

\[ P_1 \quad P_2 \quad P_{30} \]

\[ s_1, s_2, s_3, s_4, s_5, s_6, \ldots, \ldots, \ldots, s_{88}, s_{89}, s_{90} \]
Example: 3 solutions evolving for 30 generations

\[ P_1 \quad P_2 \quad P_{30} \]

\[ S_1, S_2, S_3, S_4, S_5, S_6, \ldots, \ldots, \ldots, S_{88}, S_{89}, S_{90} \]

\[ m() \]

\[ (x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, \ldots, \ldots, (x_{88}, y_{88}), (x_{89}, y_{89}), (x_{90}, y_{90}) \]
Example: 3 solutions evolving for 30 generations

\[ P_1, P_2, P_30 \]

\[ s_1, s_2, s_3, s_4, s_5, s_6, \ldots, s_{88}, s_{89}, s_{90} \]

\[ \downarrow \]

\[ m() \]

\[ (x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_{88}, y_{88}), (x_{89}, y_{89}), (x_{90}, y_{90}) \]

mapped \( P_1 \)

mapped \( P_{30} \)
Example: 3 solutions evolving for 30 generations

\[ P_1, P_2, P_{30} \]

\[ s_1, s_2, s_3, s_4, s_5, s_6, \ldots, s_{88}, s_{89}, s_{90} \]

\[ m() \]

\[ (x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_{88}, y_{88}), (x_{89}, y_{89}), (x_{90}, y_{90}) \]

mapped \( P_1 \)

mapped \( P_{30} \)
The general framework:

- any EA
- any* problem
  - only the search space $S$ matters
  - *: provided that a function $m : S^h \to (\mathbb{R} \times \mathbb{R})^h$ is available

In general not trivial, e.g., $S$ is the space of trees...
A special case:

- given a **dissimilarity function** $d : S \times S \rightarrow \mathbb{R}^+$
- consider the subclass of dimensionality reduction functions that
  1. compute the distance matrix $D$ out of $s_1, \ldots, s_h$
  2. obtain $(x_1, y_1), \ldots, (x_h, y_h)$ out of $D$
Scenario

Dimensionality reduction

A special case:

- **given a dissimilarity function** $d : S \times S \rightarrow \mathbb{R}^+$
- consider the subclass of dimensionality reduction functions that
  1. compute the distance matrix $D$ out of $s_1, \ldots, s_h$
  2. obtain $(x_1, y_1), \ldots, (x_h, y_h)$ out of $D$

Requirement “provided that a function $m : S^h \rightarrow (\mathbb{R} \times \mathbb{R})^h$ is available” met with suitable $d$ for $S$

- might be easy to find, closely related to the domain
Table of Contents

1. Scenario
2. Evaluation
3. Wrap-up
Purpose

Does it work?
**Purpose**

Does it work? too broad

Instead:

- which dimensionality reduction works better?
- what/how to plot the 2D points?
Purpose

Does it work? too broad

Instead:
- which dimensionality reduction works better?
  - quantitatively
- what/how to plot the 2D points?
  - qualitatively
Which dimensionality reduction?

Which one works better... in terms of:

RQ1 ability to capture the movements of the population in the search space

RQ2 ability to capture the exploration-exploitation trade-off
Which dimensionality reduction?

Which one works better... in terms of:

**RQ1** ability to capture the movements of the population in the search space

- movement is the essence of progress of the optimization

**RQ2** ability to capture the exploration-exploitation trade-off
Which dimensionality reduction?

Which one works better... in terms of:

**RQ1** ability to capture the movements of the population in the search space
- movement is the essence of progress of the optimization

**RQ2** ability to capture the exploration-exploitation trade-off
- exploration/exploitation: the antagonistic cornerstones of search based optimization
- often explicitly targeted by EA parameters
Experimental setup

Any EA, any problem...
Experimental setup

Any EA, any problem...

Experimented with two (toy) tunable problems with different:

- $S$: either discrete (bit strings of $l$ bits) or continuous ($\mathbb{R}^l$)
- “dimensionality” $l$
  - continuous w/ $l = 2$ is a “baseline”: no actual need of dim. red.
- number of optima $n$

Fitness is the distance (Hamming or Euclidean) to the closest optimum
Experimiental setup

Any EA, any problem…

Experimented with two (toy) tunable problems with different:
- \(S\): either discrete (bit strings of \(l\) bits) or continuous (\(\mathbb{R}^l\))
- “dimensionality” \(l\)
  - continuous \(w/\ l = 2\) is a “baseline”: no actual need of dim. red.
- number of optima \(n\)

Fitness is the distance (Hamming or Euclidean) to the closest optimum

And an EA with variants:
- simple non-overlapping generational model \(w/\) random init, tournament, suitable operators (depend on \(S\))
- same \(w/\) diversity promotion
  - fitness sharing with strength \(n_{NN}\) (0 means no promotion)
  - allows to explore different exploration-exploitation trade-offs

10 runs with \(n_{pop} = 50, n_{gen} = 50\)
Dimensionality reduction techniques

3 + 1 contenders:

- Multidimensional Scaling (MDS)
- t-Distributed Stochastic Neighbor Embedding (t-SNE)
  - devised for visualization, “reduce the tendency to crows points”
- Uniform Manifold Approximation and Projection (UMAP)
  - devised for visualization, fast
- Principal Component Analysis (PCA)
  - the +1: works only if $S = \mathbb{R}^l$

Distances: Hamming for bit strings, Euclidean for $\mathbb{R}^l$
RQ1: movements

Idea: how well the movements of the best individual are captured in 2D?

Procedure:

1. consider the trajectory (sequence of positions) of the best individual across the $n_{gen}$ generations
   - $s_1^*, \ldots, s_{n_{gen}}^*$ in $S$
   - $(x_1^*, y_1^*), \ldots, (x_{n_{gen}}^*, y_{n_{gen}}^*)$ in 2D

2. compute inter-generation best distances from the trajectory
   - $\Delta^S = (\delta^S_1, \ldots, \delta^S_{n_{gen}-1})$ in $S$ using a suitable $d$
   - $\Delta^{2D} = (\delta^{2D}_1, \ldots, \delta^{2D}_{n_{gen}-1})$ in 2D using Euclidean distance

3. compute the Pearson’s correlation between $\Delta^S$ and $\Delta^{2D}$
   - ideally 1; the greater, the better
RQ1: results, overall

\( n = 1, \ n_{\text{NN}} = 0 \)

<table>
<thead>
<tr>
<th>Bit string optimization</th>
<th>Continuous optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td>PCA</td>
<td></td>
</tr>
<tr>
<td>MDS</td>
<td>0.95</td>
</tr>
<tr>
<td>t-SNE</td>
<td>0.59</td>
</tr>
<tr>
<td>UMAP</td>
<td>0.35</td>
</tr>
</tbody>
</table>

⇒ MDS is the best performer
RQ1: results, overall

\[ n = 1, \ n_{NN} = 0 \]

<table>
<thead>
<tr>
<th></th>
<th>Bit string optimization</th>
<th>Continuous optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>PCA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDS</td>
<td>0.95</td>
<td>0.91</td>
</tr>
<tr>
<td>t-SNE</td>
<td>0.59</td>
<td>0.58</td>
</tr>
<tr>
<td>UMAP</td>
<td>0.35</td>
<td>0.52</td>
</tr>
</tbody>
</table>

⇒ MDS is the best performer
RQ1: results, good/bad examples, detail

MDS on $\mathbb{R}^{10}$, $n = 1$, $n_{NN} = 1$

t-SNE on $\mathbb{R}^{5}$, $n = 1$, $n_{NN} = 1$
RQ2: exploration-exploitation trade-off

Idea: how well exploration rate is captured in 2D?

- No single, widely accepted way of measuring if an ongoing search is exploring or exploiting

Borrow existing definition, **Similarity to Closest Neighbour (SCN)**:

- measured at each birth: the distance from the closest solution in the history of the search up to current generation
- based on a dissimilarity measure \(d\) (hence actually a dissimilarity)

\[
SCN^S(s) = \min_{j<i} \min_{s' \in P_j} d(s, s')
\]

- the larger, the more \(s\) is exploring
RQ2: measuring exploration-exploitation

Procedure:

1. for each solution $s$ at each generation, compute:
   - $SCN^S(s)$ in $S$, using a $d$ suitable to $S$
   - $SCN^{2D}(x, y)$ in 2D, using a Euclidean distance

2. compute the medians $\overline{SCN}^S$ and $\overline{SCN}^{2D}$

3. for each $i$-th generation, compute exploration rate $\tau_i$ as the fraction of solutions with $SCN$ larger than median value:
   - $\tau^S$ in $S$
   - $\tau^{2D}$ in 2D

4. measure the root mean squared error (RMSE) between the two $\tau$ signals, in $S$ and in 2D
   - ideally 0; the lower, the better
RQ2: results, overall

$n = 1, n_{NN} = 0$ (top); $n = 4, n_{NN} = 4$ (bottom)

<table>
<thead>
<tr>
<th></th>
<th>Bit string optimization</th>
<th>Continuous optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>PCA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDS</td>
<td>0.43</td>
<td>0.29</td>
</tr>
<tr>
<td>t-SNE</td>
<td>0.49</td>
<td>0.46</td>
</tr>
<tr>
<td>UMAP</td>
<td>0.54</td>
<td>0.53</td>
</tr>
</tbody>
</table>

MDS is again the best performer
RQ2: results, overall

\( n = 1, \ n_{NN} = 0 \) (top); \( n = 4, \ n_{NN} = 4 \) (bottom)

<table>
<thead>
<tr>
<th></th>
<th>Bit string optimization</th>
<th>Continuous optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16 20 24 32</td>
<td>2 5 10 15</td>
</tr>
<tr>
<td>PCA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDS</td>
<td>0.43 0.29 0.23 0.55</td>
<td>0.00 0.45 0.56 0.60</td>
</tr>
<tr>
<td>t-SNE</td>
<td>0.49 0.46 0.47 0.58</td>
<td>0.50 0.56 0.57 0.56</td>
</tr>
<tr>
<td>UMAP</td>
<td>0.54 0.53 0.56 0.64</td>
<td>0.55 0.63 0.66 0.66</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCA</td>
<td></td>
<td>0.00 0.43 0.49 0.54</td>
</tr>
<tr>
<td>MDS</td>
<td>0.37 0.37 0.36 0.54</td>
<td>0.01 0.41 0.49 0.55</td>
</tr>
<tr>
<td>t-SNE</td>
<td>0.48 0.49 0.49 0.57</td>
<td>0.51 0.59 0.61 0.54</td>
</tr>
<tr>
<td>UMAP</td>
<td>0.53 0.56 0.56 0.63</td>
<td>0.56 0.65 0.68 0.65</td>
</tr>
</tbody>
</table>

⇒ MDS is again the best performer
RQ2: results, good/bad examples, detail

MDS on $\mathbb{R}^{15}$, $n = 1$, $n_{NN} = 1$

$\tau^S_i$ $\tau^2D_i$

Generation

0 20 40

0.2 0.4 0.6 0.8 1

t-SNE on $\mathbb{R}^{15}$, $n = 2$, $n_{NN} = 2$

$\tau^S_i$ $\tau^2D_i$

Generation

0 20 40

0.2 0.4 0.6 0.8 1
RQ3: what/how to plot the 2D points?

Many non-primary goals:

- show fitness
- show ancestry
- highlight trajectory of the best solution
- highlight history of the previous populations

Two approaches for showing evolution progress:

- 2D animation: one frame for each generation
- 3D: stacked 2D layers, one for each generation

Interactivity
2D animation and non-primary goals

MDS on $\mathbb{R}^{15}$, $n = 3$, $n_{\text{NN}} = 4$
3D stacked: different runs, dim. reduction functions

$\mathbb{R}^2$, 
$n = 1, 
 n_{NN} = 0$

$\mathbb{R}^{15}$, 
$n = 3, 
 n_{NN} = 4$

$\{0, 1\}^{16}$, 
$n = 3, 
 n_{NN} = 4$
Demo
Table of Contents

1. Scenario
2. Evaluation
3. Wrap-up
General framework for visualizing evolution:

- can be done, requires distance in $S$ and dimensionality reduction
- MDS works best
  - does not “arbitrarily” magnify crowded regions

But...  

- cannot be used online
- computationally expensive
  - minutes with MDS, tens of seconds with UMAP for each run
Thanks!