Differential Evolution for Self-adaptive Triangular Brushstrokes

Uroš Mlakar, Janez Brest, Aleš Zamuda

University of Maribor

{*uros.mlakar,janez.brest,ales.zamuda*}*@um.si*

Student Workshop on Bioinspired Optimization Methods and their Applications (BIOMA 2014)

13th International Conference on Parallel Problem Solving from Nature (PPSN 2014)

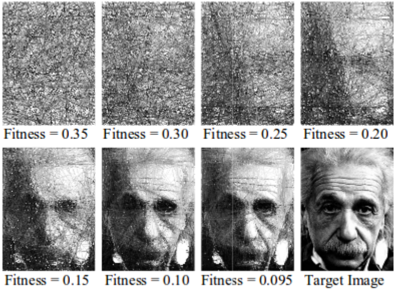
Ljubljana, Slovenia, September 13, 2014

Overview

Motivation: Line Strokes

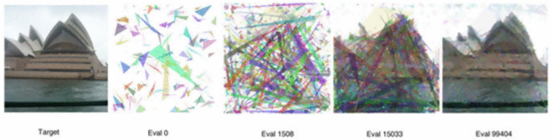
- Riley et al. (WCCI Barcelona, July 2010) compared 2 representations
  - variable-length classic genetic algorithm and
  - tree-based genetic algorithm.

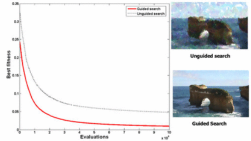
- Line strokes to generate evolved images
  - best fitness was $\sim 9.4\%$ of the original image,
  - using a tree-based algorithm.



Motivation: Triangular Brushstrokes

- Izadi et al. (AJCAI, Adelaide, December 2010) used GP for the creation of non-photorealistic animations
  - unguided and guided searches: the guided yields better results,
  - filled and empty brushstrokes,
  - reported results are:
    - unguided: $\sim 5\%$
    - guided: $\sim 2\%$ - requires the source image in the phenotype rendering.





Differential Evolution (DE)

- A floating-point encoding EA for global optimization over continuous spaces,
  - through **generations**,
    the evolution process improves **population of vectors**,
  - iteratively by combining a parent individual and several other individuals of the same population.
- We choose the **strategy** *jDE/rand/1/bin*
  - **mutation**: $\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F \times (\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G})$,
  - **crossover**: $u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & \text{if } \mathrm{rand}(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{i,j,G} & \text{otherwise} \end{cases}$,
  - **selection**: $\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G+1} & \text{if } f(\mathbf{u}_{i,G+1}) < f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G} & \text{otherwise} \end{cases}$,
  - includes mechanism of F and CR control parameters self-adaptation.

The Proposed Method (Encoding)

- An individual encoded image is stored into a DE vector: $\mathbf{x} = (x_1, x_2, ..., x_{8T^{max}}, F, CR, T^L, T^U)$, size is $D + 4$, $D = 8T^{max}$,
- the scaling factor F and crossover rate CR as used by the jDE,
- then T^L and T^U follow.
- The parameters T^L and T^U define the number of triangles T_i:
  - T_i rendered in the evolved image,
  - T^L and T^U updated similarly as the F control parameter.

The Proposed Method (Genotype $\rightarrow$ Phenotype Rendering) 1/3

- DE vector $\mathbf{x}_i, \forall i \in \{1, ...NP\}$ constituting a genotype rendered into a phenotype image $\mathbf{z}_i$ (to be compared against $\mathbf{z}^*$).
- Each brushstroke is represented as $(c_x, c_y, r, \alpha_1, \alpha_2, b^Y, b^{Cb}, b^{Cr})$:

$$c_x \in [0, ..., R_x), \quad c_y \in [0, ..., R_y), \quad r \in \left[0, \frac{R_x}{\sqrt{T_{max}}}\right), \quad \alpha_1 \in [0^\circ, 360^\circ),$$

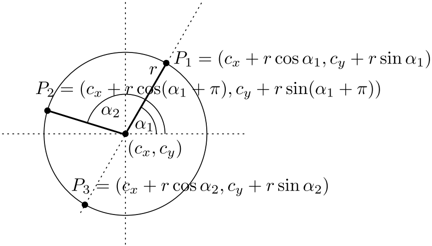
$$\alpha_2 \in [0^\circ, 180^\circ), \quad b^Y \in [16, 236), \quad b^{Cb} \in [16, 241), \quad b^{Cr} \in [16, 241).$$

- c_x and c_y define the center of the triangle to be rendered,
- r defines its circumscribed circle,
- α_1, α_2 define the points of the triangle on the circumscribed triangle,
- b^Y,b^{Cb},b^{Cr} are the color components of its brush.

The Proposed Method (Genotype → Phenotype Rendering) 2/3

- The triangle vertices encoded by $\mathbf{x}_i$ construct T_i triangles,
- each triangle $\mathbf{T}_k = (c_x, c_y, r, \alpha_1, \alpha_2)$ defines vertices as in Figure on the right, Eq. 1.
- For optimization, the **YCbCr** color space is used.
- For rendering, the brush color $\mathbf{b}_k^{YCbCr}$ is transformed to the RGB color space using the Eq. 2.

Figure: Triangle brush definition



Eq. 1

$$P_{1,k} = \lfloor (c_{x,k} + r_k \cos\alpha_{1,k}, c_{y,k} + r_k \sin\alpha_{1,k}) \rfloor$$
$$P_{2,k} = \lfloor (c_{x,k} + r_k \cos(\alpha_{1,k} + \pi), c_{y,k} + r_k \sin\alpha_{1,k} + \pi) \rfloor$$
$$P_{1,k} = \lfloor (c_{x,k} + r_k \cos\alpha_{2,k}, c_{y,k} + r_k \sin\alpha_{2,k}) \rfloor$$

Eq. 2

$$b_k^R = \lfloor 1.164(b_k^Y - 16) + 1.596(b_k^{Cr} - 128) \rfloor$$
$$b_k^G = \lfloor 1.164(b_k^Y - 16) - 0.813(b_k^{Cr} - 128) - 0.391(b_k^{Cb} - 128) \rfloor$$
$$b_k^B = \lfloor 1.164(b_k^Y - 16) + 2.018(b_k^{Cb} - 128) \rfloor$$

The Proposed Method (Genotype $\rightarrow$ Phenotype Rendering) 3/3

- For each triangle T_k, a solid color is rendered,
- over the brush area with a transparency factor $\frac{1}{T_i}$,
- which makes the color of the brush: $\mathbf{b}_k = \lfloor \frac{255}{T_i} \mathbf{b}^k_{RGB} \rfloor$;
- this is analogous to blending each triangle as part-transparent triangle withing the evolved image:
    - $\mathbf{z}^k_{x,y} = \sum_{\mathbf{T}_k over(x,y)} \lfloor \frac{255}{T_i} \mathbf{b}^{RGB}_{k,x,y} \rfloor$.
- Triangles defined over the edges of the image canvas are drawn by clipping away pixels outside of the canvas area.

Fitness Evaluation

- After a phenotype image $\mathbf{z}_i$ is rendered: it is compared to a reference image $\mathbf{z}^*$
  - using the evaluation metric:

$$f(\mathbf{z}) = 100 \times \frac{\sum_{y=0}^{R_y-1} \sum_{x=0}^{R_x-1} |z_{x,y}^{*R} - z_{x,y}^{R}| + |z_{x,y}^{*G} - z_{x,y}^{G}| + |z_{x,y}^{*B} - z_{x,y}^{B}|}{3 \times 255 \times R_x R_y}.$$

- The obtained result is the similarity of the evolved image and the reference image.
- The goal of the evolutionary process is to minimize the function value $f(\mathbf{z})$.

Results (Experimental Setup)

- The parameter sets are:
  - $NP = \{25, 50, 100\}$,
  - $T_{max} = \{10, 20, ..., 150\}$,
  - $RN_i = \{0, 1, ...51\}$,
  - $MAXFES$ = 1e+5.
- A total of 45 parameter settings, 2340 independent runs.
- Rendering: GDI+.
- The experiments conducted on 4 images of size 100×100 pixels.
- Additional experiment: all images evolved up to $MAXFES$ = 1e+6.

Baboon

Liberty

Palace

Vegetables

Results (Experiment)

Best fitness values for all parameter sets for all images, $MAXFES = 1e{+}5$

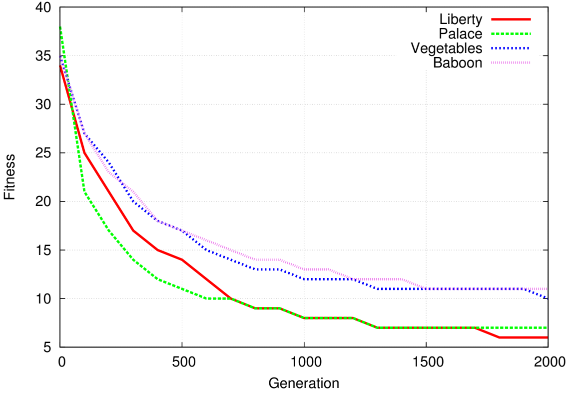
Table 1. Obtained fitness over $T_{\max}$ and NP: test instances Liberty and Palace.

		Liberty				Palace			
NP	$T_{\max}$	Best	Worst	Average	STD	Best	Worst	Average	STD
25	10	8.29	11.99	9.93096	0.8233	8.69	13.69	10.1362	0.9655
25	20	8.03	13.14	10.0935	1.0845	7.83	11.5	9.12173	0.8092
25	30	8.41	13.74	10.0525	1.1712	7.52	11.1	8.97942	0.7992
25	40	8.13	12.81	10.4408	1.1416	7.34	11.36	8.91788	0.8922
25	50	8.49	13.37	10.6767	1.1768	7.65	12.53	8.87442	0.9788
25	60	7.95	14.65	10.9858	1.4284	7.9	11.88	8.99673	0.8761
25	70	8.28	14.21	11.4075	1.3630	7.79	13.17	9.50327	1.0482
25	80	8.72	15.89	11.7554	1.6330	7.97	12.34	9.43558	0.9765
25	90	8.84	16.24	12.1342	1.6608	8.41	13.54	9.82	1.2756
25	100	9.01	16.74	12.4798	1.7521	8.62	12.96	9.83635	0.8869
25	110	8.07	16.78	12.7412	1.7849	9.01	14.42	10.4119	1.2468
25	120	9.67	16.14	12.8467	1.7359	8.93	15.13	10.3858	1.3149
25	130	10.16	17.96	13.2692	1.7193	9.02	14.2	10.2858	1.0292
25	140	9.29	17.99	13.7029	1.7886	8.29	13.51	10.7779	1.0299
25	150	10.82	18.56	14.0373	1.6573	9.89	14.91	11.1206	1.0586
50	10	7.51	9.69	8.45077	0.4198	7.43	11.84	8.68058	0.8825
50	20	6.78	8.99	7.80173	0.4987	**7.1**	11.39	8.79173	0.9592
50	30	6.89	9.17	7.81788	0.5119	7.53	12.58	9.75654	1.1186
50	40	**6.77**	9.87	8.0375	0.6578	8.27	12.24	10.0575	0.9537
50	50	7.08	10.61	8.39923	0.7056	7.97	13.14	10.3338	1.1009
50	60	7.15	10.4	8.67115	0.7472	8.59	12.49	10.7817	1.0754
50	70	7.46	10.9	9.1025	0.8666	7.58	12.8	10.7744	1.1086
50	80	7.6	11.4	9.47981	0.8689	9.15	13.11	11.3802	1.0178
50	90	8.05	12.65	9.67346	0.9115	9.97	13.41	11.5227	0.9315
50	100	8.75	11.75	10.0152	0.7824	8.55	13.62	11.4356	0.9923
50	110	8.93	13.63	10.6356	0.9682	9.32	13.77	12.0712	0.9579
50	120	9.22	13.01	10.7502	0.9840	9.77	14.21	12.429	0.8972
50	130	9.42	12.59	11.0527	0.7707	11.37	14.07	12.7387	0.6134
50	140	9.99	13.39	11.5719	0.7815	9.69	15.5	12.9317	0.9708
50	150	10.2	14.56	12.2633	1.0702	9.58	15.36	12.8092	1.1717
100	10	7.1	9.12	7.98596	0.4241	7.91	13.88	10.9573	1.8019
100	20	6.85	9.77	7.83962	0.5360	8.86	14.59	12.1117	1.2862
100	30	7.15	11.8	8.49077	1.1563	9.59	16.15	12.9098	1.0589
100	40	7.22	13	8.86327	1.1092	9.65	14.97	13.2477	1.1543
100	50	7.41	12.75	9.34846	1.3939	11.01	15.52	13.8606	0.9750
100	60	8.06	12.97	9.77731	1.1539	11.5	16.14	14.1856	1.1234
100	70	8.67	13.28	10.1954	1.3722	10.77	16.32	14.3629	1.1713
100	80	8.73	14.48	11.0929	1.4093	10.98	17.06	14.9348	1.1679
100	90	9.04	14.92	11.3594	1.3483	11.1	16.8	15.104	1.2586
100	100	9.4	16.13	11.6604	1.4952	10.8	17.62	15.36	1.2330
100	110	10.17	15.68	12.3365	1.5685	13.01	17.86	16.0202	0.9744
100	120	10.26	15.45	12.3358	1.5076	11.07	17.99	15.6113	1.6455
100	130	10.22	16.19	13.2212	1.6108	12.33	18.37	16.4085	1.3168
100	140	11.42	16.65	13.7808	1.5502	11.64	18.35	16.1229	1.4990
100	150	11.35	18.68	14.6113	1.9726	10.11	18.34	16.2929	2.0056

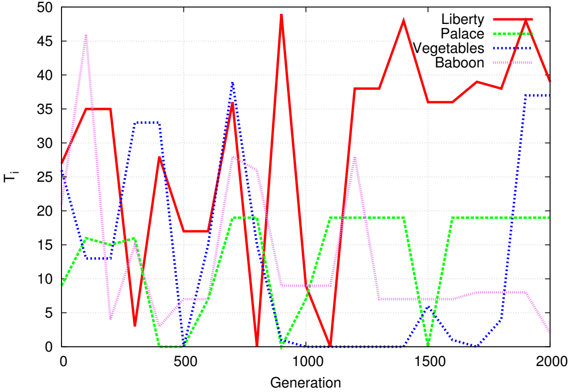
Table 2. Obtained fitness over $T_{\max}$ and NP: test instances Vegetables and Baboon.

		Vegetables				Baboon			
NP	$T_{\max}$	Best	Worst	Average	STD	Best	Worst	Average	STD
25	10	14.13	17.21	15.7269	0.7148	15.02	18.59	16.38	0.7128
25	20	12.56	18.03	14.5658	0.9850	13.44	17.12	15.3815	0.8129
25	30	12.33	15.98	13.9215	0.8475	12.99	19.03	15.0204	1.1150
25	40	11.62	16.21	13.674	1.0436	11.99	16.85	14.4342	1.0135
25	50	12.16	17.08	13.88	1.0726	11.39	17.62	14.4573	1.2299
25	60	11.64	17.88	13.6438	1.2155	11.74	17.51	14.8038	1.2229
25	70	11.29	17.15	13.9056	1.3790	11.88	17.9	14.6267	1.3495
25	80	11.61	16.6	14.0871	1.3881	12.11	17.13	14.3606	1.2815
25	90	11.63	17.96	14.1062	1.4428	11.93	19.41	14.6644	1.5269
25	100	11.34	17	14.4533	1.4694	11.7	18.77	14.7642	1.7438
25	110	11.74	19.66	14.6085	1.7664	12.02	19.11	15.0046	1.7605
25	120	12.26	17.91	14.7737	1.5726	12.2	18.5	15.6467	1.6086
25	130	12.1	19.75	14.6338	1.9283	13.01	19.5	15.4254	1.5505
25	140	11.94	19.01	14.7635	1.6282	12.64	19.37	15.8235	1.8458
25	150	12.82	18.7	14.6487	1.3015	13.13	20.17	15.7952	1.6923
50	10	13.03	15	14.0723	0.4674	13.86	16.52	14.9192	0.5494
50	20	11.66	13.26	12.4644	0.3184	11.8	14.54	13.271	0.5569
50	30	11.12	13.59	12.2425	0.6528	11.59	13.62	12.5506	0.5732
50	40	**10.94**	14.1	12.1848	0.6656	11.1	13.84	12.3137	0.6090
50	50	11.04	13.92	12.2946	0.7609	11.34	14.36	12.4075	0.6304
50	60	11.29	15.86	12.5506	0.9222	11.25	14.1	12.3662	0.6161
50	70	11.18	15.21	12.6104	0.8682	11.54	14.57	12.5437	0.6510
50	80	11.32	15.26	12.8619	0.7658	**11.07**	15.56	12.9473	0.8087
50	90	11.84	15.28	13.0077	0.8038	11.32	16.2	12.857	1.0291
50	100	11.72	15.8	13.5058	0.9565	11.85	15.72	13.2658	0.7972
50	110	12.02	15.92	13.5204	0.8750	11.98	15.56	13.4275	0.7805
50	120	11.9	16.87	13.829	1.1151	12.43	15.66	13.5106	0.7265
50	130	12.51	15.97	14.094	0.8855	12.64	16.32	14.085	0.8259
50	140	12.16	17.07	14.8198	1.2154	12.54	16.31	14.15	0.8865
50	150	13.11	17.98	14.9838	1.2072	13.08	18	14.8765	1.0178
100	10	12.56	16.19	13.9815	0.8083	13.49	16.19	14.5367	0.5672
100	20	11.84	16.45	13.4704	1.0483	12.02	15.87	13.8244	0.8747
100	30	11.83	17.64	13.9133	1.3335	12	15.76	13.7206	0.9727
100	40	12.01	17.95	14.6354	1.3660	11.63	17.01	13.6467	1.3582
100	50	11.87	17.35	14.9156	1.4272	11.99	17.48	14.1658	1.5554
100	60	12.32	18	15.21	1.5119	12.12	17.46	14.5021	1.4517
100	70	12.13	18.05	15.6513	1.2457	12.12	17.16	14.3881	1.3782
100	80	12.9	18.86	16.2008	1.4121	12.13	17.56	14.8656	1.4214
100	90	12.32	20.04	16.3233	1.7789	12.25	18.66	15.2558	1.5144
100	100	12.98	20.55	16.7275	1.7119	13.09	18.42	15.5398	1.5064
100	110	13.76	20.18	17.2896	1.5242	13	19.62	15.84	1.6164
100	120	13.12	20.62	17.626	1.5807	13.34	19.58	16.4725	1.5223
100	130	13.52	20.12	17.9052	1.3516	13.84	19.6	16.9367	1.7362
100	140	14.08	20.52	18.216	1.6975	14.3	21	17.4387	1.7372
100	150	14.97	21.19	19.1221	1.2128	14.75	21.13	17.9488	1.6872

Results (Fitness Convergence Graph)



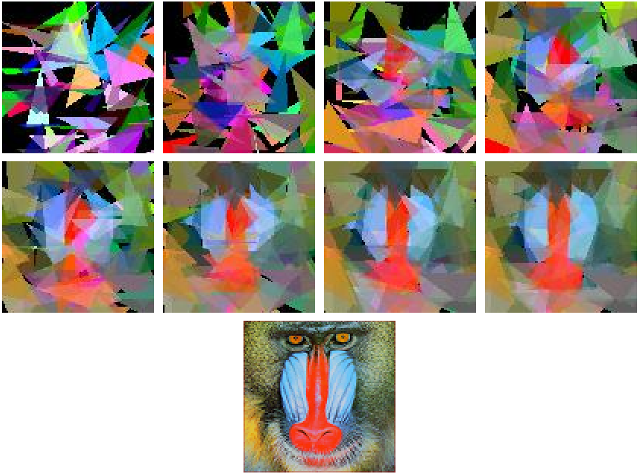
The fitness convergence graph of the best runs for all images.



The dynamics of the number of triangular brushstrokes in the best vectors.

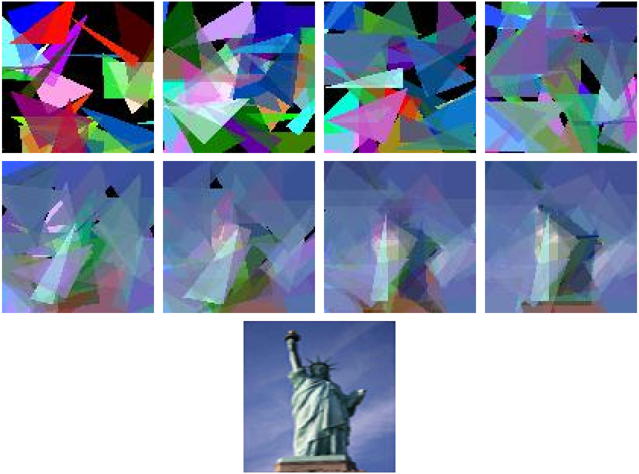
Results (Image: Baboon)

Baboon



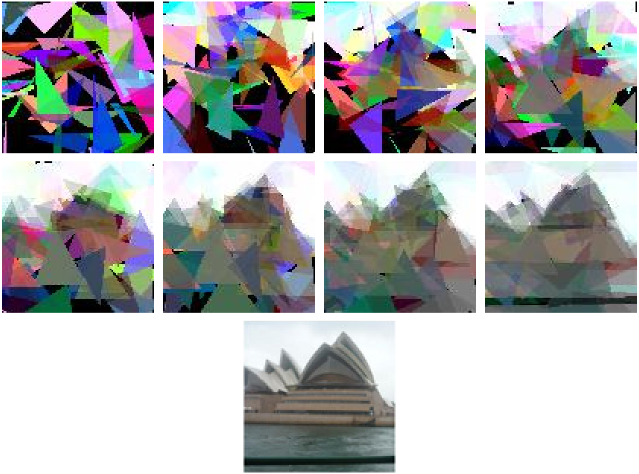
Results (Image: Liberty)

Liberty



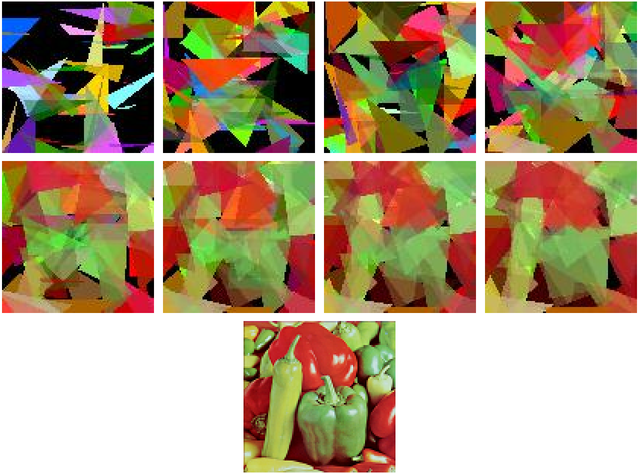
Results (Image: Palace)

Palace



Results (Image: Vegetables)

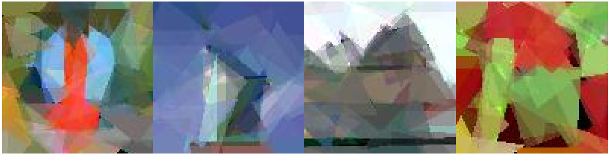
Vegetables



Conclusion

- An evolvable lossy image representation using a jDE algorithm.
- The performance of this encoding: competitive with the GA tree representation.
- Experiments show promising results on sample images.
- In the future we would like to address:
  - different evolutionary operators,
  - change control-parameters updating, and
  - testing on more images with different properties.

Thank you.



Questions?