Lecture 8: Differential Evolution
What is differential evolution?

• Differential Evolution (DE) is an EA for continuous function optimization proposed by Kenneth Price and Rainer Storn in 1994 (see http://www1.icsi.berkeley.edu/~storn/code.html).

• DE is also a population based algorithm, in which each individual is called an agent, and is often represented as multi-dimensional real vector.

• The basic idea is to use vector differences for perturbing (mutating) the agents, instead of conventional crossover and mutation used in GA.

Basic steps of DE

- Step 1: Initialize P.
- Step 2: For each agent \( x \) in \( P \),
  - Perturb \( x \) to \( y \);
  - Change \( x \) to \( y \) if \( \text{fitness}(y) > \text{fitness}(x) \).
- Step 3: Stop if terminating condition satisfied; return to Step 2 otherwise.
Method for perturbation

• An agent \( x \) is perturbed in two steps.
• The first step is to find a perturbation vector \( u \).
  – Select at random 3 agents \( r_1, r_2, \) and \( r_3 \), which are mutually different, and also different from \( x \).
  – The **perturbation vector** is found as follows:

\[
    u_i = r_{3i} + F(r_{1i} - r_{2i}), \quad i = 1, 2, \ldots, D
\]

where \( D \) is the dimensionality of the agents, and \( F \) in \([0, 2]\) is called the differential weight (usually, \( F \) is selected in \([0.4, 1]\)).
Method for perturbation

• The second step is to find the perturbed vector $y$ through crossover of $x$ and $u$:

\[
j = \text{rand}[1, D]; \quad // \text{Find an index at random}\n\text{for } i = 1 \text{ to } D
\text{if}(\text{rand}[0, 1] < \text{CR} \text{ or } i == j) \quad y_i = u_i; \\
\text{else} \quad y_i = x_i; \\
\text{end}
\]

where $\text{CR}$ is the crossover rate.

$u$ is also called a donor because it is produced only for donating its parts to the new agent.
Performance of DE

• Storn and Price showed in 1997 that DE was more efficient than SA and GA.
• Ali and Torn found in 2004 that DE was both more accurate and more efficient than controlled random search and another GA.
• In 2004, Lampinen and Storn demonstrated that DE was more accurate than several other optimization methods including four GAs, SA, and EP.
• The question is, why using the differences between existing agents is better than random perturbation? It is necessary to provide a theoretic foundation for DE.
Why DF?

• The process for finding the donor corresponds to an “adaptive” mutation (diversification).
  – The mutation direction and size are defined automatically based on existing solutions.

• The process for finding the new solution corresponds to a “purposeful” crossover (intensification).
  – Perturb a solution partially based on “estimated” direction and size (vs random crossover in GA).

• Direction: determined based on two (or more) solutions found so far.
• Size: determined (randomly) based on the distances between existing solutions.
DE for interactive evolution

Start

Initialization

Get an agent

Perturbation

Selection

Terminating condition?

yes

End

no

Which one is your preferable flower?
DE for synthesizing natural images

- Image morphing is a technique for producing intermediate images from some given images, and has been widely used in movie production.
- In our research, we have tried to generate many different but natural images using DE.
- The naturalness is “subjective”, and is evaluated based on human feeling.
Feature points are used for morphing
Process for image morphing

• Step 1: Find the FPS of the morphed image:
  \[ F_m = \alpha F_s + (1 - \alpha) F_t \]

• Step 2: Warp the source image and the target image
  \[ I_s^w = W(I_s, F_s, F_m) \]
  \[ I_t^w = W(I_t, F_t, F_m). \]

• Step 3: Find the morphed image
  \[ I_m = \alpha I_s^w + (1 - \alpha) I_t^w \]
Example of image morphing
Definition of the agents

• Each agent is a feature point set (FPS) containing $2N$ feature points of 2 images.
• Since each feature point has two coordinates, the dimensionality of each agent is $D=4N$.
• If we try to evolve the agents directly, the search space will be very large.
• In our research, we define an initial FPS roughly, and then fine-tune the FPS using DE.
Feature points used for morphing

Search is performed within a square region 2bx2b for each feature point.
Genotype and phenotype

Genotype: String of real numbers in $[0,1]$

$$g = [g_1, g_2, \ldots, g_{4n}]$$

**Direct approach**

- $x_{sj} = g_j \times w$, $j = 1, 2, \ldots, n$
- $y_{sj} = g_{j+n} \times h$, $j = 1, 2, \ldots, n$
- $x_{rj} = g_{j+2n} \times w$, $j = 1, 2, \ldots, n$
- $y_{rj} = g_{j+3n} \times h$, $j = 1, 2, \ldots, n$

**Fine tuning approach**

- $x_{sj} = x_{sj}^0 + g_j \times a - b$, $j = 1, 2, \ldots, n$
- $y_{sj} = y_{sj}^0 + g_{j+n} \times a - b$, $j = 1, 2, \ldots, n$
- $x_{rj} = x_{rj}^0 + g_{j+2n} \times a - b$, $j = 1, 2, \ldots, n$
- $y_{rj} = y_{rj}^0 + g_{j+3n} \times a - b$, $j = 1, 2, \ldots, n$
DEMO:
Find morphing parameters to generate natural faces
Discussion

• Using IE, we can modify one agent a time, and keep it if the morphing image is natural enough. We can then get many different natural images through evolution.

• Using IE, we can also learn other human feelings, such as beautifulness, kindness, happiness, sadness, peacefulness, etc.

• The system may be useful for producing exercise problems in an e-learning environment (need further improvement).
Problems to solve

• One of the main problems in IE is that, human user’s subjective evaluation may change during evolution.
• That is, the evaluation standard may shift with time, so that the same object can be evaluated differently, and different objects may have the same fitness.
• Learning the “feeling-shift” is important to obtain objects (e.g. synthesized image or music) the user really likes. This problem can be solved by using machine learning technology.

Matlab program

- You can download a set of Matlab programs at


- The programs were written by Markus Buehren, and can be used for optimizing some complex functions.
- There are several demo programs, and you can run and see what happen.
Optimizing the Shekel's Foxholes

3D View

All evaluated members

All evaluation values over parameter 1

All evaluation values over parameter 2
Homework

- Download the Matlab program for DE, and try another demo program.
- There is an obsolete function “wavread”. You may just delete the related lines in the program, or replace it using a new function “audioread”. You can get more detailed explanation about the new function using on-line documents of Matlab.
- Summarize the results, print it in one page, and submit next week.