Evolutionary Programming Using a Mixed Mutation Strategy and an Online Demo

Jun He

School of Computer Science
University of Birmingham
Global Optimization Problem

Global optimization problem:

\[
\begin{align*}
\text{minimize} & \quad f(x), \quad x = (x_1, \cdots, x_n) \in \mathbb{R}^n, \\
\text{subject to} & \quad g_k(x) \leq 0, \quad k = 1, \cdots, m.
\end{align*}
\]

where \( f \): the objective function and \( g_k \): \( m \) constrains.
Global Optimization Problem

Global optimization problem:

\[
\begin{align*}
\text{minimize } & \quad f(x), \quad x = (x_1, \cdots, x_n) \in \mathcal{R}^n, \\
\text{subject to } & \quad g_k(x) \leq 0, \quad k = 1, \cdots, m.
\end{align*}
\]

where \( f \): the objective function and \( g_k \): \( m \) constrains.

A very popular problem in science, engineering and economics etc.
Global Optimization Problem

- Global optimization problem:

\[
\begin{align*}
\text{minimize} & \quad f(x), \quad x = (x_1, \ldots, x_n) \in \mathbb{R}^n, \\
\text{subject to} & \quad g_k(x) \leq 0, \quad k = 1, \ldots, m.
\end{align*}
\]

where \( f \): the objective function and \( g_k \): \( m \) constrains.

- A very popular problem in science, engineering and economics etc.

- Many existing algorithms for it.
Global Optimization Problem

- Global optimization problem:

\[
\begin{align*}
\text{minimize } & \quad f(\mathbf{x}), \quad \mathbf{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n, \\
\text{subject to } & \quad g_k(\mathbf{x}) \leq 0, \quad k = 1, \cdots, m.
\end{align*}
\]

where \( f \): the objective function and \( g_k \): \( m \) constrains.

- A very popular problem in science, engineering and economics etc.

- Many existing algorithms for it.

- Q: How many algorithms do you know?
EAs for Function Optimization

Q: Why apply EAs to solve global optimization problem?
EAs for Function Optimization

Q: Why apply EAs to solve global optimization problem?

Possible reasons
EAs for Function Optimization

Q: Why apply EAs to solve global optimization problem?

Possible reasons

- some problems are too difficult and complex to existing algorithms
EAs for Function Optimization

Q: Why apply EAs to solve global optimization problem?

Possible reasons

- some problems are too difficult and complex to existing algorithms
- EAs are easy to design and implement, and don’t need much mathematical knowledge
Evolutionary Programming (EP) is equivalent to Evolutionary Strategy (ES), but proposed by L. Fogel.
Evolutionary Programming (EP)

- Equivalent to Evolutionary Strategy (ES), but proposed by L. Fogel
- Encoding: real-vector. A point \((x_1, \cdots, x_n)\) and its search step \((\sigma_1, \cdots, \sigma_n)\).
Evolutionary Programming (EP)

- Equivalent to Evolutionary Strategy (ES), but proposed by L. Fogel
- Encoding: real-vector. A point \((x_1, \cdots, x_n)\) and its search step \((\sigma_1, \cdots, \sigma_n)\).
- Mutation:

\[
\begin{cases}
\sigma^{(t+1)} = \sigma^{(t)} \times \text{RandomNumber1}, \\
x^{(t)} = x^{(t)} + \sigma^{(t)} \times \text{RandomNumber2},
\end{cases}
\]
Evolutionary Programming (EP)

- Equivalent to Evolutionary Strategy (ES), but proposed by L. Fogel
- Encoding: real-vector. A point \((x_1, \cdots, x_n)\) and its search step \((\sigma_1, \cdots, \sigma_n)\).
- Mutation:

\[
\begin{align*}
\sigma^{(t+1)} &= \sigma^{(t)} \times \text{RandomNumber1}, \\
x^{(t)} &= x^{(t)} + \sigma^{(t)} \times \text{RandomNumber2},
\end{align*}
\]

- Selection: e.g. \(k\)-tournament section. (choose the best among \(k\) candidates; and repeat this procedure until a population is generated)
Evolutionary Programming (EP)

- Equivalent to Evolutionary Strategy (ES), but proposed by L. Fogel
- Encoding: real-vector. A point \((x_1, \cdots, x_n)\) and its search step \((\sigma_1, \cdots, \sigma_n)\).
- Mutation:

\[
\begin{align*}
\sigma^{(t+1)} &= \sigma^{(t)} \times \text{RandomNumber}_1, \\
x^{(t)} &= x^{(t)} + \sigma^{(t)} \times \text{RandomNumber}_2,
\end{align*}
\]

- Selection: e.g. \(k\)-tournament section. (choose the best among \(k\) candidates; and repeat this procedure until a population is generated)
- Q: How to calculate random numbers?
Mutations

Q: How many mutation operators do you know?
Mutations

Q: How many mutation operators do you know?
Researchers have developed many types of mutations, e.g.
- Gaussian mutation using Gaussian distribution
- Cauchy mutation using Cauchy distribution
- Lévy mutation using Lévy distribution
Mutations

Q: How many mutation operators do you know?

Researchers have developed many types of mutations, e.g.

- Gaussian mutation using Gaussian distribution
- Cauchy mutation using Cauchy distribution
- Lévy mutation using Lévy distribution

Q: Can you develop a new one?
Gaussian Mutation

Gaussian Mutation (1 dimension space):

\[ x^{(t+1)} = x^{(t)} + \sigma^{(t+1)} N(0, 1), \]

where \( N(0, 1) \) is a Gaussian random variable.
Gaussian Mutation

- Gaussian Mutation (1 dimension space):
  \[ x^{(t+1)} = x^{(t)} + \sigma^{(t+1)} N(0, 1), \]
  where \( N(0, 1) \) is a Gaussian random variable.

- Standardized Gaussian distribution:
  \[ \frac{1}{\sqrt{2\pi}} \exp \left( -x^2 \right) \]
Cauchy Mutation

Cauchy Mutation (1 dimension space)

\[ x^{(t+1)} = x(t) + \sigma^{(t+1)} C, \]

where \( C \) is a Cauchy random variable.
Cauchy Mutation

- Cauchy Mutation (1 dimension space)

\[ x^{(t+1)} = x(t) + \sigma^{(t+1)} C, \]

where \( C \) is a Cauchy random variable.

- Cauchy distribution:

\[ \frac{1}{\pi} \frac{t}{t^2 + x^2} \]

where \( t > 0 \) scale parameter.
Gaussian vs Cauchy distributions

Q: What is the difference between them?
Gaussian vs Cauchy distributions

Q: What is the difference between them?
A: Cauchy distribution has a longer tail, so the search step is much bigger
Best Mutation?

Q: Which is better between Gaussian and Cauchy Mutations?
Best Mutation?

Q: Which is better between Gaussian and Cauchy Mutations?

A: do several experiments.

- for functions with many local optimal points, Cauchy is better. Why?
- For functions with only a few local optimal points, Gaussian is better. Why?
Best Mutation?

Q: Which is better between Gaussian and Cauchy Mutations?

A: do several experiments.
   - for functions with many local optimal points, Cauchy is better. Why?
   - For functions with only a few local optimal points, Gaussian is better. Why?

An explanation.
Mixed Mutation Strategy

Q: why using a mixed strategy?
Mixed Mutation Strategy

Q: why using a mixed strategy?

Reasons

- You have several mutations in hand, but you don’t know which is the best for a problem
- A mixed strategy will possibly integrate the advantages of different mutations
Design of a mixed strategy

Q: How to design a mixed strategy to mix different strategies
Design of a mixed strategy

Q: How to design a mixed strategy to mix different strategies

A simple solution, called “adaptive mutation”

- an individual generates two individual by Gaussian and Cauchy mutation.
- select the best.
Design of a mixed strategy

Q: How to design a mixed strategy to mix different strategies

A simple solution, called “adaptive mutation”

- an individual generates two individuals by Gaussian and Cauchy mutation.
- select the best.

Q: What is the meaning of “adaptive” mutation?
Q: How to design a mixed strategy to mix different strategies

A simple solution, called “adaptive mutation”
- an individual generates two individual by Gaussian and Cauchy mutation.
- select the best.

Q: What is the meaning of “adaptive” mutation?
Q: Can you develop other approaches?
A Game Theory Approach

- Player: individuals.
A Game Theory Approach

- Player: individuals.
- Strategy: mutation operators.
A Game Theory Approach

- Player: individuals.
- Strategy: mutation operators.
- Output: offspring generated by applying a strategy.
A Game Theory Approach

- Player: individuals.
- Strategy: mutation operators.
- Output: offspring generated by applying a strategy.
- Payoff: a performance measure of a strategy against other strategy.
A Game Theory Approach

- Player: individuals.
- Strategy: mutation operators.
- Output: offspring generated by applying a strategy.
- Payoff: a performance measure of a strategy against other strategy.
- A mixed strategy: the probability distribution to select over its strategy set.
Probability of choosing a strategy

Assume player 1 uses strategy $s_1$ and player 2 uses strategy $s_2$, 
Probability of choosing a strategy

- Assume player 1 uses strategy $s_1$ and player 2 uses strategy $s_2$,
- next time, choose strategy 1 with a probability

$$p(s_1) = \frac{f(s_1)}{f(s_1) + f(s_2)},$$

where $f(s_1)$ is the result of player 1 using strategy 1, and $f(x_2)$ for player 2 using strategy 2.
Probability of choosing a strategy

- Assume player 1 uses strategy $s_1$ and player 2 uses strategy $s_2$,
- next time, choose strategy 1 with a probability
  \[ p(s_1) = \frac{f(s_1)}{f(s_1) + f(s_2)}, \]
  where $f(s_1)$ is the result of player 1 using strategy 1, and $f(x_2)$ for player 2 using strategy 2.
- choose strategy 2 with a probability
  \[ p(s_2) = \frac{f(s_2)}{f(s_1) + f(s_2)}. \]
An Online Demo

- Design a software demo application, which includes different mutation strategies
An Online Demo

- Design a software demo application, which includes different mutation strategies
- User can take it as benchmark software and compare own algorithms with classical algorithms
An Online Demo

- Design a software demo application, which includes different mutation strategies
- User can take it as benchmark software and compare own algorithms with classical algorithms
- Provide a set of test functions
An Online Demo

- Design a software demo application, which includes different mutation strategies
- User can take it as benchmark software and compare own algorithms with classical algorithms
- Provide a set of test functions
- User can input their own functions
An Online Demo

- Design a software demo application, which includes different mutation strategies
- User can take it as benchmark software and compare own algorithms with classical algorithms
- Provide a set of test functions
- User can input their own functions
- User can add more mutations inside
Server

Server side: an expression parser and an EP solver. It is used to implement the task of computation and return result to the client.
Server

- Server side: an expression parser and an EP solver. It is used to implement the task of computation and return result to the client.
- Functions
Server

- Server side: an expression parser and an EP solver. It is used to implement the task of computation and return result to the client.

- Functions
  - Mathematic expression parser: It checks the validity of the function provided by users.
Server

- Server side: an expression parser and an EP solver. It is used to implement the task of computation and return result to the client.

- Functions
  - Mathematic expression parser: It checks the validity of the function provided by users.
  - EA solver: To apply an EP to solve the function optimization problem. EA part receives the parsed mathematic expression. The solver is a Java implementation of the EP using mixed mutation strategy, described in the previous section.
User interface

- Client side: a web page interface
User interface

- Client side: a web page interface
- Interface application. It accepts the input from users and sends parameters to the server for calling EP solver.
User interface

- Client side: a web page interface
- Interface application. It accepts the input from users and sends parameters to the server for calling EP solver.
- Parameters initialization
User interface

- Client side: a web page interface
- Interface application. It accepts the input from users and sends parameters to the server for calling EP solver.
  - Parameters initialization
  - Mathematic function initialization
User interface

- Client side: a web page interface
- Interface application. It accepts the input from users and sends parameters to the server for calling EP solver.
  - Parameters initialization
  - Mathematic function initialization
  - Predefined test functions
Running results

Q: When each strategy should be applied?
Running results

Q: When each strategy should be applied?

Experiment observation:

Figure 2: Cauchy is better first and then Gaussian dominate
### Which is the best

<table>
<thead>
<tr>
<th></th>
<th>Mixed best</th>
<th>Adaptive mean best</th>
<th>Cauchy mean best</th>
<th>Gaussian mean best</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>9.151e-06</td>
<td>4.16e-5</td>
<td>5.72e-4</td>
<td>1.91e-4</td>
</tr>
<tr>
<td>$f_2$</td>
<td>1.269e-03</td>
<td>2.44e-2</td>
<td>7.60e-2</td>
<td>2.29e-2</td>
</tr>
<tr>
<td>$f_3$</td>
<td>6.590e-04</td>
<td>4.83e-3</td>
<td>1.76e-2</td>
<td>8.79</td>
</tr>
<tr>
<td>$f_4$</td>
<td>1.706e-02</td>
<td>4.54e-2</td>
<td>2.49e-2</td>
<td>8.13e-2</td>
</tr>
<tr>
<td>$f_5$</td>
<td>-8.774e+00</td>
<td>-6.46</td>
<td>-5.50</td>
<td>-6.43</td>
</tr>
<tr>
<td>$f_6$</td>
<td>-9.735e+00</td>
<td>-7.10</td>
<td>5.73</td>
<td>7.62</td>
</tr>
<tr>
<td>$f_7$</td>
<td>-9.841e+00</td>
<td>-7.80</td>
<td>6.41</td>
<td>8.86</td>
</tr>
</tbody>
</table>
Summary

Expect you have learned that

- the procedure of EP for solving global optimization problem
Summary

Expect you have learned that

- the procedure of EP for solving global optimization problem
- two basic mutations: Gaussian and Cauchy mutations
Summary

Expect you have learned that

- the procedure of EP for solving global optimization problem
- two basic mutations: Gaussian and Cauchy mutations
- the idea of mixed strategy
Summary

Expect you have learned that

- the procedure of EP for solving global optimization problem
- two basic mutations: Gaussian and Cauchy mutations
- the idea of mixed strategy
- develop online demo application
Open Questions

How to design a better mixed mutation strategy?
Open Questions

- How to design a better mixed mutation strategy?
- How to mix different crossover or selection?
Open Questions

- How to design a better mixed mutation strategy?
- How to mix different crossover or selection?
- How to develop an open source software program from the demo?
Further Readings


