5.2 Evolutionary Programming

李长河

中国地质大学（武汉）自动化学院

lichanghe@cug.edu.cn

李长河

自动化学院 710

lichanghe@cug.edu.cn

Contents

Evolutionary Programming

Evolutionary programming (EP) was firstly proposed to simulate the evolution process and generate artificial intelligence and then applied in optimization domains.

There is a consistent pursuing for our human to obtain the artificial intelligence.

About sixty years ago

Researchers attempted to build the model of natural evolution to realize the automatic programming, sequence prediction and so on. For example, Friedberg aimed to design the algorithm to find the program with certain inputs and outputs.

Friedberg

The pioneer of automatic programming

The performance of the method

Unfortunately, the mechanism proved worse even than a pure random search, resulted mainly by:

- The absence of effective selection pressure
- The extreme disassociation(Small changes in program syntax usually cause large changes in the input-output behaviour of the program)

In the 1960s

Having the knowledge of Friedberg' s disappointing results, Bremermann focused on the work of relatively simple optimization problems, especially for the linear programming and convex programming.

The performance of the method

It is too limited for real optimization applications.

Evolutionary Programming-The Emerging of Evolutionary Programming

In 1964

L. J.Fogel formally proposed a kind of evolutionary algorithms, called evolutionary programming (EP).

The mechanism of EP

The individual, on behalf of a transition table of finite-state-machine (FSM), mutates to reproduce new FSMs, meanwhile, whether the individual was able to mutate in the next generation depended on the performance in the evaluation testing.

The performance of the method

Compared with Friedberg′ s and Bremermann′ s algorithms:

- Applying EP algorithm to more sophisticated optimization problems
- Withith appropriate selective pressure provided

In 1970s

Nevertheless, the refined work did not receive the remarkable attention in the field, until the genetic algorithm and evolution strategies were fully accepted in 1970s.

The limitation in Friedberg and Bremermann's experiments caused the ignorance of Fogel's works in almost thirty years.

The characteristic of the initial EP

- Fixed chromosome structure
- Changing numerical parameters evolving along with decision variables

The specific mechanism was brought by many other EAs and obtained a formal term, self-adaptation.

Evolutionary Programming-The Classical Evolutionary Programming

The categories of initial EP algorithms

- Standard EP, which has no self-adaptation specialities.
- Continuous standard EP, different from the generation-based algorithms, in which, the individual is evaluated and added into the population.
- Meta-EP, into which, the variance of mutation step size is cooperated.
- Continuous meta-EP, in which, the individual is evaluated and added into the population due to the variance of the mutation operator.
- Rmeta-EP, which cooperates covariances and standard deviations for self-adaptation.

The key elements in the process of EP algorithms

- Representation of solutions
- Mutation operators
- Selection mechanisms

The optimization domain $\boldsymbol{x}$, the space dimension D, the variances $\boldsymbol{\nu}$, the domain of real numbers $\mathbb{R}$.

Representation

Individuals of standard EP can be described as:

$$\mathbf{p} = (\boldsymbol{x}, \boldsymbol{\nu}) \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^D$, $\boldsymbol{\nu} = \mathbf{0}$.
Slightly different from standard EP, individuals in meta-EP represent can be described as:

$$\mathbf{p} = (\boldsymbol{x}, \boldsymbol{\nu}) \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^D, \boldsymbol{\nu} \in {\mathbb{R}_+}^D$.

In standard EP, Gaussian mutation operator, with the form of $\mathbf{m} = (\{\beta_1, \ldots, \beta_D\}, \{\gamma_1, \ldots, \gamma_D\})$, is applied, where proportionality constant vector $\boldsymbol{\beta}$ and offset $\boldsymbol{\gamma}$ are parameters that must be tuned for a particular task. Often, however, constants in $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are set to one and zero respectively.

Mutation (standard EP)

The operator works with a standard deviation that is determined by the square root of a linear transformation of the fitness value $f(\boldsymbol{x})$ for each element in variable vector. Mutating with Gaussian mutation operator, the element x_i in variable vector $\boldsymbol{x}$ updated by:

$$x_i' \leftarrow x_i + \sqrt{\beta_i \cdot f(\boldsymbol{x}) + \gamma_i} \cdot \mathcal{N}_i(0,1) \tag{3}$$

where $\beta_i = 1, \gamma_i = 0$ are commonly used, therefore the mutation operator can be simplified:

$$x_i' \leftarrow x_i + \sqrt{f(\boldsymbol{x})} \cdot \mathcal{N}_i(0,1) \tag{4}$$

The difficulties of the mutation operator used in standard EP

- When the fitness value is large, the step size in the search will be large, which results in almost like random search.
- It needs lots of efforts to tune the extra 2D parameters.
- If the fitness value of global optimum is not zero, the approaching to the global optimum is not possible.

Mutation (meta-EP)

Compared with standard EP, meta-EP has main specialty of adapting variances to overcome the tuning difficulties. Different mutation operators combined with the variances vector $\boldsymbol{\nu}$ are applied:

$$x'_i \leftarrow x_i + \sqrt{\nu_i} \cdot \mathcal{N}_i(0,1) \tag{5}$$

$$\nu_i \leftarrow \nu_i + \sqrt{\zeta \cdot \nu_i} \cdot \mathcal{N}_i(0,1) \tag{6}$$

where the parameter ζ is introduced to assure $\boldsymbol{\nu}$ to be positive all the time.

Selection

Carrying out the mutation policy, all individuals produce its offspring, resulting in a larger population with the size of $2\mathtt{N}$. To reduce the size from $2\mathtt{N}$ back to $\mathtt{N}$, selection mechanism is requisite. For example, the tournament selection is a popular method to weed out $\mathtt{N}$ individuals based on fitness value in the pairwise comparison.

1. 5.2 Evolutionary Programming
    - 5.2.1 The Emerging of Evolutionary Programming
    - 5.2.2 The Classical Evolutionary Programming
    - 5.2.3 Framework and Parameter Settings

Algorithm 1: meta-EP algorithm

Input: Maximum Generation $G_{\max}$
Initialization: $t = 0$, $\mathbb{P}(t) = \{\mathbf{p}_i(t)\}, \mathbf{p}_i = (\boldsymbol{x}_i, \boldsymbol{\nu}_i, f(\boldsymbol{x}_i))), i = 1, \cdots, N$;
Evaluate all individuals;
while $t < G_{\max}$ **do**
  Mutate: $\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \sqrt{\boldsymbol{\nu}_i(t)} \cdot \boldsymbol{\mathcal{N}}(0, 1)$;
  Evaluate: $f(\boldsymbol{x}(t+1))$;
  Select: $\mathbb{P}(t+1) = \mathbb{P}(t) \bigcup New\ Generated$;
  $t := t + 1$;
end

The algorithm starts with a set of solutions with randomly generated positions and mutation step sizes. Then the population undergoes mutation, evaluation, and selection procedures until the stop criterion is satisfied.

Parameter settings for standard EP and meta-EP

The default parameter settings of standard EP and meta-EP can be seen in Table 1.

表: Parameter settings for standard EP and meta-EP

Paramter	EP	meta-EP	Default
Domain range l_i, u_i	$\times$	$\times$	$l_i = -50, u_i = 50$
Upper bound c of σ_i		$\times$	$c = 25$
Proportionality constants β_i	$\times$		$\beta_i = 1$
Offset constants γ_i	$\times$		$\gamma_i = 0$
Meta-parameter ζ for adaptation		$\times$	$\zeta = 6$
Tournament size q	$\times$	$\times$	$q = 10$
Population size N	$\times$	$\times$	$N = 200$