Partial Sampling Operator and Tree-Structural Distance for Multi-Objective GP

Makoto OHKI
Tottori University, Japan
1. Introduction

Program Synthesis

Function Generation

Rule Discovery

Target of these applications expressed by a tree structure.

Optimize with EC

Genetic Programming
1. Introduction

Effective Search ⇔ Bloat Control

* Schema Theory for GP [Holland 1992]
* Probabilistic Incremental Program Evolution [Salustowicz 1997]
* Depth Limitation [Langdon 1999]
* Size-Fair Model GP [Langdon 2000]
* Grammar-Guided GP [Ratle 2000]
* FREQT [Asai 2001]
* Subtree Swapping Crossover [Poli 2003]
* TAG3P [Hoai 2004]
* Tree Size Limitation [Ryan 2006]
* Stochastic Grammar-based GP [Ratle 2006]
* Semantic Building Blocks [McPhee 2008]

. . . . . . .
1. Introduction

**Effective Search ⇔ Bloat Control**

* Schema Theory for GP [Holland 1992]
* Probabilistic Incremental Program Evolution [Salustowicz 1997]
* Depth Limitation [Langdon 1999]
* Size-Fair Model GP [Langdon 2000]
* Grammar-Guided GP [Ratle 2000]
* FREQT [Asai 2002]
* Subtree Swapping Crossover [Poli 2003]
* TAG3P [Hoai 2004]
* Tree Size Limitation [Ryan 2006]
* Stochastic Grammar-based GP [Ratle 2006]
* Semantic Building Blocks [McPhee 2008]
1. Introduction

Effective Search ⇔ Bloat Control

* Schema Theory for GP [Holland 1992]
* Probabilistic Incremental Program Evolution [Salustowicz 1997]
* Depth Limitation [Langdon 1999]
* Size-Fair Model GP [Langdon 2000]
* Grammar-Guided GP [Ratle 2000]
* FREQT [Asai 2001]
* Subtree Swapping Crossover [Poli 2003]
* TAG3P [Hoai 2004]
* Tree Size Limitation [Ryan 2006]
* Stochastic Grammar-based GP [Ratle 2006]
* Semantic Building Blocks [McPhee 2008]
1. Introduction

Effective Search ⇔ Bloat Control

* Schema Theory for GP [Holland 1992]
* Probabilistic Incremental Program Evolution [Salustowicz 1997]
* Depth Limitation [Langdon 1999]
* Size-Fair Model GP [Langdon 2000]
* Grammar-Guided GP [Ratle 2000]
* FREQT [Asai 2001]
* Subtree Swapping Crossover [Poli 2003]
* TAG3P [Hoai 2004]
* Tree Size Limitation [Ryan 2006]
* Stochastic Grammar-based GP [Ratle 2006]
* Semantic Building Blocks [McPhee 2008]

1. Introduction
2. Partial Sampling
3. MOGP with SD
4. Verification
5. Conclusion

10th International Joint Conf. on Computational Intelligence, Seville, Spain, September 18-20, 2018
1. Introduction

In this paper,

- Partial Sampling (PS) operator instead of Crossover and Mutation

- A technique of Multi-Objective GP by applying NSGA-II
  - index of goodness of the tree
  - the Size of the tree
  - tree position in the population by Tree-Structural Distance (TSD)

- Apply TSD instead of Crowding Distance (CD) of NSGA-II

- Double Spiral Problem for verification
2. Partial Sampling Operator for Mating

- Partial Sampling Operator for Mating

1. Introduction
2. Partial Sampling
3. MOGP with SD
4. Verification
5. Conclusion
2. Partial Sampling Operator for Mating

Proliferation Termination Probability $p_t$

\[
\begin{align*}
    p_t^0 &= \frac{1}{\text{AverageSize } R^g}, \\
    p_t^{g+1} &= \frac{1}{\text{Succeed } P^g - p_t^0} - p_t^0 \\
    &= \frac{1}{\text{Succeed } R^g - p_t^0} - p_t^0 \\
    &= p_t^g - p_t^0 + p_t^0,
\end{align*}
\]

$R^g$ : population at $g$-th generation
$P^g$ : parent set at $g$-th generation
AverageSize $\bullet$ : average size of tree structure
Succeed $\bullet$ : average size of partial tree structure of set succeeded from previous generation
2. Partial Sampling Operator for Mating

- 2 kinds of metastasis

![Diagram showing partial sampling operator for mating]

- Initial proliferation
- Random metastasis
- Upper node depend metastasis
3. Multi-Objective GP with Tree-Structural Distance

3 Objective Functions

① objective function according to Goodness of the tree structure

\[ h_1(\text{indiv}_i) = \text{performance(root}_i) \]

② objective function according to the size of the tree structure

\[ h_2(\text{indiv}_i) = \frac{1}{\text{size(root}_i)} \]

③ objective function according to average of TSD in the population

\[ h_3(\text{indiv}_i) = \frac{1}{N_{\text{pop}}} \sum_{k=1}^{N_{\text{pop}}} \text{TSD indiv}_i,\text{indiv}_k \]
3. Multi-Objective GP with Tree-Structural Distance

- **Tree-Structural Distance (TSD)**

![Diagram showing tree-structural distance with nodes A, B, C, D, E, F, G, H, I, J, and K, and corresponding probabilities for each node.]

- **Nodes and Probabilities:**
  - A: 1
  - B: 1/4
  - C: 1/4
  - D: 1/24
  - E: 1/24
  - F: 1/24
  - G: 1/8
  - H: 1/96
  - I: 1/96
  - J: 1/32
  - K: 1/32

1. Introduction
2. Partial Sampling
3. MOGP with SD
4. Verification
5. Conclusion
3. Multi-Objective GP with Tree-Structural Distance

Tree-Structural Distance (TSD)

\[ TSD(root_i, root_k) = \frac{1}{24} + \frac{1}{24} + \frac{1}{8} = \frac{5}{24} \]
3. Multi-Objective GP with Tree-Structural Distance

- NSGA-II (conventional)

- non-dominated sorting

- population

- rank 1
- rank 2
- rank 3
- rank 4
- rank 5
- others

- parents

- rank 1
- rank 2
- rank 3

- CD

- mating

- children

- Q_g

- next generation

- R_{g+1}

- P_g

- R^g: population
- P^g: parents
3. Multi-Objective GP with Tree-Structural Distance

- NSGA-II with TSD instead of CD

- Population: $R_g$
  - Rank 1
  - Rank 2
  - Rank 3
  - Rank 4
  - Rank 5
  - Others

- Parents: $P_g$
  - Rank 1
  - Rank 2
  - Rank 3

- Non-dominated sorting

- Children: $Q_g$

- Mating

- Next generation: $R_{g+1}$

$R^g$: population
$P^g$: parents
4. Verification by **Double Spiral Problem**

\[ \begin{align*}
  f(x, y) > 0 & \iff (x, y) \in D_1 \\
  f(x, y) < 0 & \iff (x, y) \in D_2 \\
  f(x, y) = 0 & \iff \text{FALSE}
\end{align*} \]

difficult even by the neural network.

- non-terminal node $\in \{ +, -, *, \div, \sin, \cos, \tan, \text{ifltz} \}$
- terminal node $\in \{ x, y, \text{constant} \}$

\[
\text{ifltz}(a, b, c) \triangleq \begin{cases} 
  b & \text{if } a < 0 \\
  c & \text{otherwise}
\end{cases}
\]
4. Verification by Double Spiral Problem

Objective function $h_1$ according to the goodness of tree

$$h_1(\text{indiv}_i) = \text{performance(root}_i) = \frac{1}{|D_1 \cup D_2|} \sum_{k=1}^{D_1 \cup D_2} g(x_k, y_k)$$

$$g(x, y) = \begin{cases} 
1 & f(x, y) > 0 \land (x, y) \in D_1, \\
0 & f(x, y) > 0 \land (x, y) \in D_2, \\
1 & f(x, y) < 0 \land (x, y) \in D_2, \\
0 & f(x, y) < 0 \land (x, y) \in D_1, \\
0 & f(x, y) = 0 
\end{cases}$$
4. Verification by Double Spiral Problem

Final Solution Distribution

- CO+MU & CD
- CO+MU & SD
- PSM & CD
- PSM & SD

1. Introduction
2. Partial Sampling
3. MOGP with SD
4. Verification
5. Conclusion
4. Verification by Double Spiral Problem

- Comparison among 3-Objective, 2-Objective, 1-Objective GPs

![Graph showing comparison among objectives with SD and CD](image)

1. Introduction
2. Partial Sampling
3. MOGP with SD
4. Verification
5. Conclusion
4. Verification by Double Spiral Problem

Comparison of results on MS-Norm plane

- CO+MU & CD
- CO+MU & SD
- PS0.25 & CD
- PS0.25 & SD
- PS0.50 & CD
- PS0.50 & SD
- PS0.75 & CD
- PS0.75 & SD
5. Conclusion

In this paper,

- **Multi-Objective GP**
- In addition to goodness of the tree, 2 objective functions
  - tree size
  - Tree-Structural Distance (TSD)
- **Partial Sampling (PS)** for mating
- Double Spiral Problem for verification.
- The proposed technique (PS + TSD $\rightarrow$ NSGA-II) is effective.
5. Conclusion

In the future,

- Enhance the capability of numerical optimization
- Ranking Selection technique harmonizing CD and TSD
- Mechanism to forcibly exit from PS
Thank you very much!

Ask me simply, even if you have.
MS and Norm

degree of spread of $FFS$

$$MS = \sqrt{\sum_{i=1}^{m} \left(\frac{|FFS|}{\max_j f_i(x_j)} - \frac{|FFS|}{\min_j f_i(x_j)}\right)^2}$$

degree of convergence to $POS$

$$Norm = \frac{1}{|FFS|} \sum_{j=1}^{m} \sqrt{\sum_{i=1}^{m} f_i(x_j)^2}$$
1. Introduction

Program Synthesis [David2017]

```ml
type list = Nil | Cons of int * list
let rec even x =
  match x with
  | Nil -> Nil
  | Cons(u, Nil) -> Cons(u, Nil)
  | Cons(u, Cons(_, us))
    -> Cons(u, even us)
let rec sum x =
  match x with
  | Nil -> 0
  | Cons(u, us) -> u + sum us
let rec sum_even = ??
let main x = assert (sum (even x) = sum_even x)
```

Solution

```ml
let rec sum_even x =
  match x with
  | Nil -> 0
  | Cons (u, Nil) -> u
  | Cons (u, Cons(_, us)) -> u + sum_even us
```
1. Introduction

Applications of Genetic Programming (GP)

- Program Synthesis
- Function Generation
- Rule Set Discovery

・・・
1. Introduction

- Function Generation [Jamali2017]

\[ U(x_p) = -Ce^{-\frac{(x_p - x_g)^2}{r^2}} \]

Potential Function
1. Introduction

Rule Set Discovery [Ohmoto2013]

Observed Data

$x$ is $A'$

Knowledge Base

if $x$ is $A_1$ then $y$ is $B_1$
if $x$ is $A_2$ then $y$ is $B_2$
・・・・・・
if $x$ is $A_n$ then $y$ is $B_n$

$y$ is $B'$

$A_1, A_1, ..., A_n, B_1, B_2, ..., B_n$: Fuzzy Sets
1. Introduction

- They can be expressed by a tree structure data.

Optimization

Genetic Programming: GP
1. Introduction

Intron

\[
\begin{align*}
3 & + \text{ zero} = 3 \\
\text{if} \quad \text{FALSE} & \quad B \quad C = B
\end{align*}
\]