Finding a Sorting Algorithm Using Genetic Programming

May 22, 2003
Hyunkoo Jee
Defining a Virtual Machine and the Syntax of the Programming Language (1/4)

- Resources that can be used by the VM
  - Array of registers: A[0], ..., A[N−1]
  - Index registers: I1, I2 (initial value: 0)

- Machine Instructions and the Syntax
  - Syntax
    - One instruction (and its operands) per one line
    - An instruction can have at most two operands
    - Line number is assigned as 0, 1, 2, ...
Defining a Virtual Machine and the Syntax of the Programming Language (2/4)

- **Example**

  0: PUT I2 ← I1
  1: INC I2
  2: EXCH
  3: BR (I2 >= N-1) 5
  4: BR (I1 < I2) 1
  5: INC I1
  6: BR (I1 < I2) 0
Defining a Virtual Machine and the Syntax of the Programming Language (3/4)

Instructions

- **EXCH** (Exchange): First, it compares \(A[I_1]\) and \(A[I_2]\). And, if there is an inversion, it exchanges the values of \(A[I_1]\) and \(A[I_2]\). If \(0 \leq I_1 \leq N-1\) and \(0 \leq I_2 \leq N-1\) is not satisfied, this instruction doesn't do anything.

- **INC operand1** (Increase): It increases the value of the register specified in operand1. Operand1 can be \(I_1\) or \(I_2\).

- **DEC operand1** (Decrease): It decreases the value of the register specified in operand1. Operand1 can be \(I_1\) or \(I_2\).

- **PUT operand1**: It copies the value of a register into another register. Operand1 can be \(I_2 \leftarrow I_1\) or \(I_1 \leftarrow I_2\).

- **BR operand1 operand2** (Conditional branch): If the condition specified in operand1 is true, jump to the line specified in operand2. Operand1 can be...
  - \(I_1 \leq 0\)
  - \(I_2 \leq 0\)
  - \(I_1 \geq N-1\)
  - \(I_2 \geq N-1\)
  - \(I_1 < I_2\)
  - \(I_1 > I_2\)
  - \(I_1 = I_2\)

Operand2 is the line number to jump to.
A priori constraints that the programs must satisfy

- Use the instruction “EXCH” exactly one time
Defining the Cost Function (or Fitness Function)

- Weighted sum of
  - Inversion
    - \( \#( \{(i,j) \mid i < j \text{ and } A[i] > A[j], \ 0 \leq i, j \leq N-1\} ) \)
  - Runtime
    - (how many instructions executed)
  - Length
    - (how many lines)
Evaluation

- Implement the virtual machine, or an emulating program
- Execute the individual candidates (=programs) on this VM, and compute the cost function of each candidates.
- Here we generate T arrays, each array has N elements. Let each individual sort these T arrays. And we compute the average of the T cost values of each individual.
- If “runtime” exceed some threshold, stop the VM (unless it would run an infinite loop).
Defining the Evolutionary Operations

4 Mutation Methods
- Insert a new line
  - When inserting a line including “BR”, the “operand2” should be in an adequate range.
  - Don’t insert “EXCH” line.
- Delete a line
  - Don’t delete “EXCH” line.
- Swap two lines
- Mutate a line
  - Do not touch the instruction. Mutate the operands only.

To mutate an individual candidate, randomly apply