Computational Complexity Analysis of Multi-Objective Genetic Programming

Frank Neumann

School of Computer Science

The University of Adelaide

Introduction

There are many

- successful applications
- experimental studies

of Genetic Programming.

We want to

- argue in a rigorous way about GP algorithms and
- contribute to their theoretical understanding

This is also important for the acceptance of our algorithms outside our community.

This Talk

Definition of two general problems for the runtime analysis of genetic programming:

- Weighted ORDER
- Weighted MAJORITY

Rigorous runtime analysis of two mechanisms for dealing with the bloat problem:

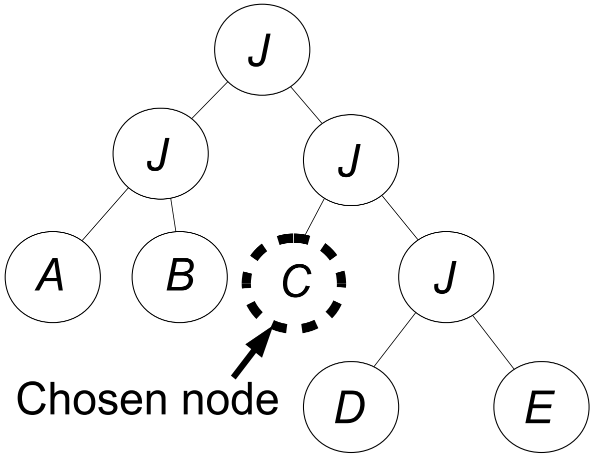
- Parsimony approach
- Multi-objective approach

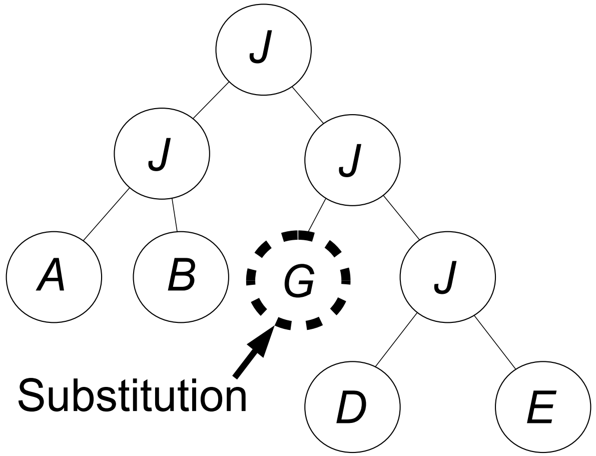
Lot of open questions.

Runtime Analysis of GP

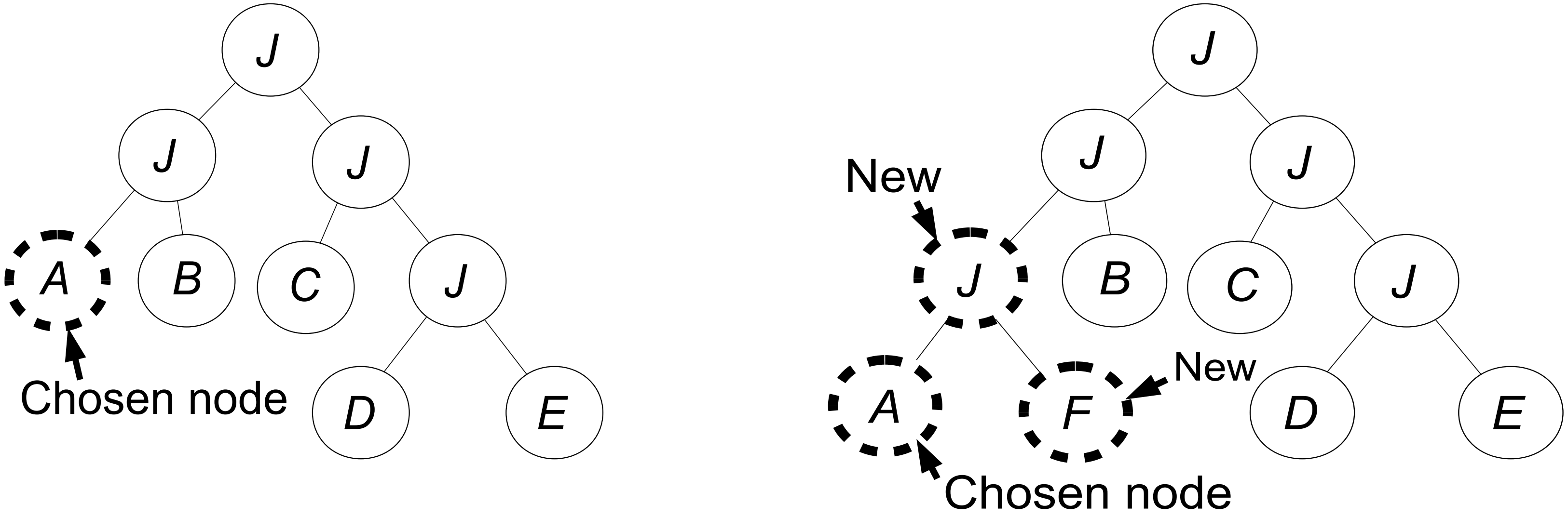
- Rigorous runtime analysis of genetic programming is relatively new.
- We want to understand in a rigorous way how genetic programming works.
- We consider simple mutation-based algorithms.
- Studies should enable analysis of more complex algorithms in the future.

Substitution

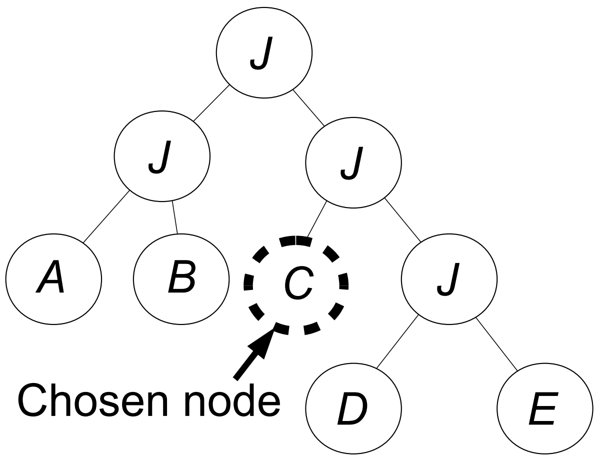


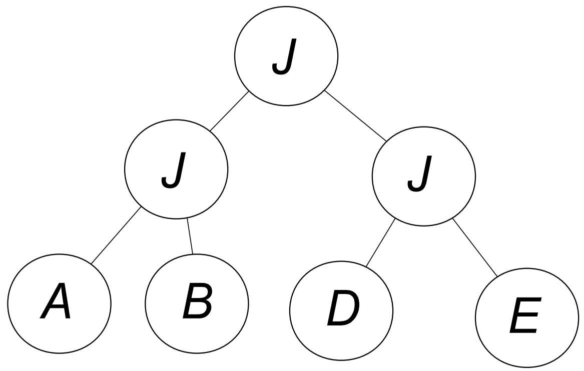


Insertion



Deletion





Baseline Algorithm

Algorithm 1 ((1+1) GP)**.**

1. *Choose an initial solution X.*
2. *Repeat*
   - *Set $Y := X$.*
   - *Apply mutation to Y.*
   - *If selection favors Y over X then $X := Y$.*

(1+1) GP-single: select one operation of {insert, delete, substitute} uniformly at random and apply it to Y.

Expected optimization time := Expected number of iterations to obtain an optimal solution.

Weighted ORDER and MAJORITY

- $F := \{J\}$, J has arity 2.
- $L := \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$

Each variable x_i has a corresponding weight

$$w_i \in \mathbb{R},\ 1 \leq i \leq n$$

Without loss of generality, we assume

$$w_1 \geq w_2 \geq \ldots \geq w_n > 0$$

Weighted ORDER and MAJORITY

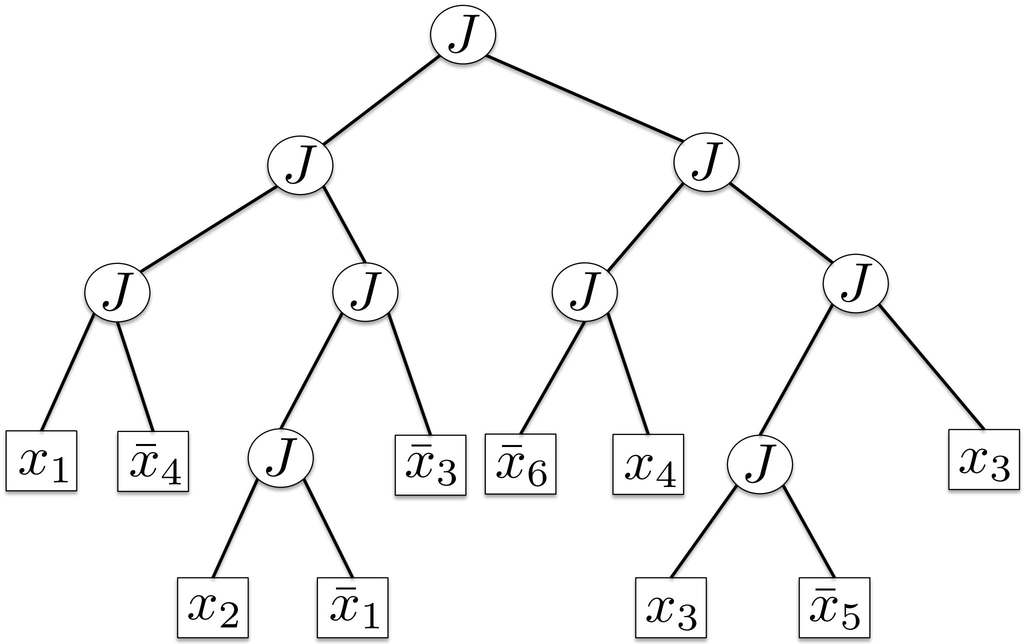
Weighted ORDER: w_i contributes to fitness if x_i is expressed, i. e. x_i is seen before $\bar{x}_i$ in an inorder parse.

Weighted MAJORITY: w_i contributes to fitness if x_i is expressed, i. e. x_i is present and there are at least as many x_i as $\bar{x}_i$.

Special case: $w_i = 1, 1 \leq i \leq n$

ORDER and MAJORITY

Let $n = 6$ and $w_1 = 13,\ w_2 = 11,\ w_3 = 8,\ w_4 = 7,\ w_5 = 5,\ w_6 = 3$



We get: $l = (x_1, \bar{x}_4, x_2, \bar{x}_1, \bar{x}_3, \bar{x}_6, x_4, x_3, \bar{x}_5, x_3)$

$$\mathrm{WORDER}(X) = w_1 + w_2 = 13 + 11 = 24$$

$$\mathrm{WMAJORITY}(X) = w_1 + w_2 + w_3 + w_4 = 13 + 11 + 8 + 7 = 39$$

Parsimony Approach

Attach to each solution X a complexity value C(X)

- C(X): number of nodes in X.

$$\text{MO-WORDER }(X) = (\text{WORDER }(X), C(X))$$

$$\text{MO-WMAJORITY }(X) = (\text{WMAJORITY }(X), C(X))$$

A solution X is called non-redundant if no variable can be deleted without decreasing fitness.

Otherwise, we call X a redundant solution.

Parsimony Approach

Parsimony approach:

- If two solutions have same quality according to F, select a solution of lowest complexity.
- Selection:

(1+1) GP-single on MO-F: Favor Y over X iff

$$(F(Y) > F(X)) \vee ((F(Y) = F(X)) \wedge (C(Y) \leq C(X))).$$

Denote by T_{init} the complexity of the initial solution.

Theorem:

The expected optimization of (1+1) GP-single on Weighted ORDER and Weighted MAJORITY is $O(T_{init} + n \log n)$.

Proof ideas:

- Using single operations the difference between complexity and the number of expressed variables can not increase.
- After $O(T_{init} + n \log n)$ steps the current solution is non-redundant, i. e. no deletion is possible without decreasing fitness.
- Missing variables can be inserted at any position.
- Additional phase of O(n log n) steps gives optimal solution (coupon collector).

Multi-Objective GP

Consider the two objectives F and C as equally important:

1. A solution Y *weakly dominates* a solution X (denoted by $Y \succeq X$) iff $(F(Y) \geq F(X) \wedge C(Y) \leq C(X))$.

2. A solution Y *dominates* a solution X (denoted by $Y \succ X$) iff $(Y \succeq X) \wedge (F(Y) > F(X) \vee C(Y) < C(X))$.

Algorithm 5. *SMO-GP*

1. *Choose an initial solution* X.
2. *Set* $P := \{X\}$.
3. *Repeat*
   - *Choose* $X \in P$ *uniformly at random.*
   - *Set* $Y := X$.
   - *Apply mutation to* Y.
   - *If* $\{Z \in P \mid Z \succ Y\} = \emptyset$,
     set $P := (P \setminus \{Z \in P \mid Y \succeq Z\}) \cup \{Y\}$.

Multi-Operations: Choose the number of operations in each mutation steps according to 1+Pois(1).

Expected optimization time := Expected number of fitness evaluations to obtain the whole Pareto front.

Theorem:

The expected optimization time of SMO-GP-single and SMO-GP-multi on MO-ORDER and MO-MAJORITY is $O(nT_{init} + n^2 \log n)$.

Proof ideas:

- Population size is upper bounded by n+1.
- Empty solution is included in the population after $O(nT_{init})$ steps.
- Other Pareto optimal solutions are obtained in time $O(n^2 \log n)$ by introducing one of the missing variables at any position (coupon collector slowed by population of size n).

Theorem:

Starting with a non-redundant initial solution, the expected optimization time of SMO-GP-single on MO-WORDER and MO-WMAJORITY is $O(n^3)$.

Proof ideas:

- No redundant solutions are accepted during the run of the algorithm.
- Population size is upper bounded by n+1.
- Empty solution is included in the population after $O(n^2)$ steps.
- Afterwards remaining Pareto optimal solutions are produced in time $O(n^3)$ (fitness-based partitions for linear pseudo-Boolean functions slowed down by population of size n).

Summary

Summary:

- Weighted ORDER and MAJORITY are generalizations of ORDER and MAJORITY (similar to OneMax to linear functions for binary strings)
- Parsimony approach provably reaches the optimal solution quickly.
- Multi-objective approach provably computes the Pareto front in expected polynomial time when starting with non-redundant solution.

Open Problems

- Determine the expected optimization time of (1+1) GP-multi which chooses k according to 1+Pois(1) on MO-WORDER and WORDER, MO-WMAJORITY, and WMAJORITY.

- Determine the expected optimization time of SMO-GP-multi on MO-WORDER and MO-WMAJORITY.