Naïve Crossover Biases in GP

On the Evolution of Parsimonious Solutions

Case for Parsimonious Solutions

- Avg(program size increase)
  - $\propto O(\text{generation}^2)$
- Evaluation Overhead Increases
- Solution Transparency Decreases
- Does boat $\rightarrow$ more effective search?
  - What is an effective search?
    - Does bloat $\rightarrow$ better population diversity or more duplication?
- Restrict our discussion to Tree structured GP

Generic Approaches to Date

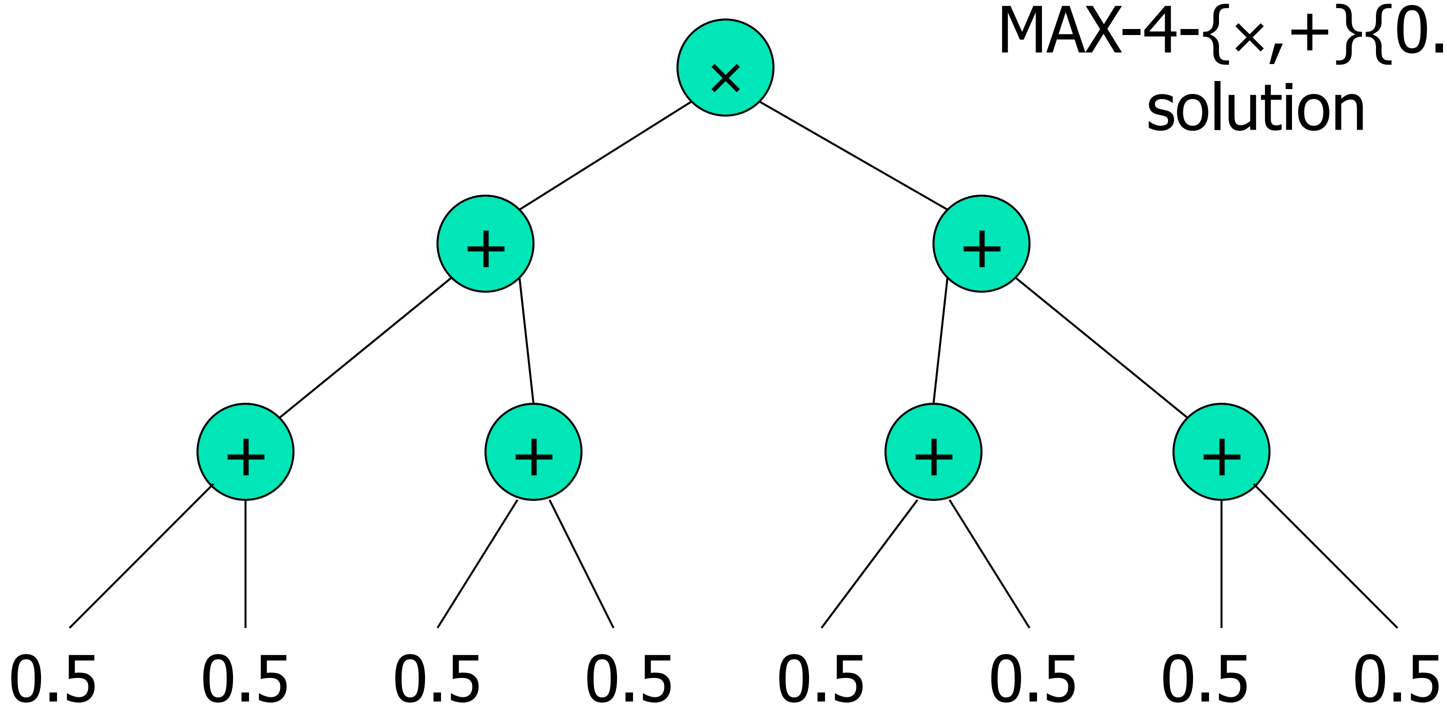
- Size and Depth Limits
  - [1]
- Parsimony Pressure
  - [2], [3], [4]
- Non-destructive crossover
  - [5]
- Biased crossover operator
  - [6], [7]

Size and Depth [1]

- Explicit,
  - Max node count/ Depth limit;
  - Unwanted interaction with crossover operator [8].

Max Problem [8]

MAX-4-{×,+}{0.5}
solution



Visualizing a tree (1)

J. M. Daida et al GECCO'03

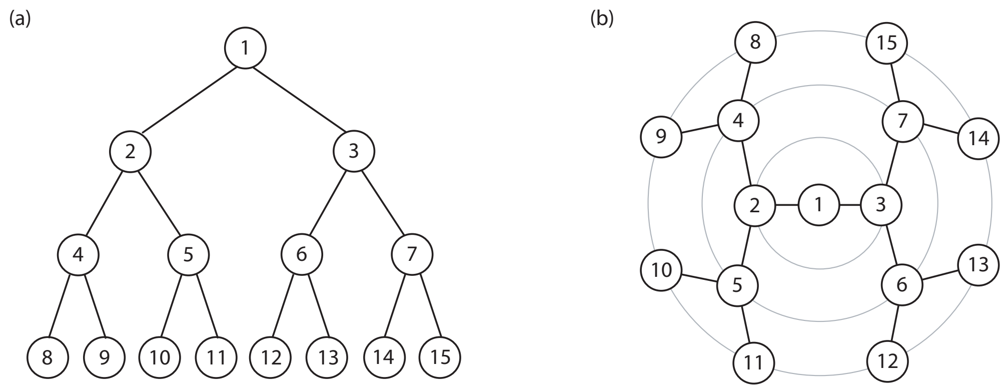


Fig. 1. Mapping a Full binary tree to a circular grid. (a) Full binary tree of depth 3. (b) Corresponding circular grid

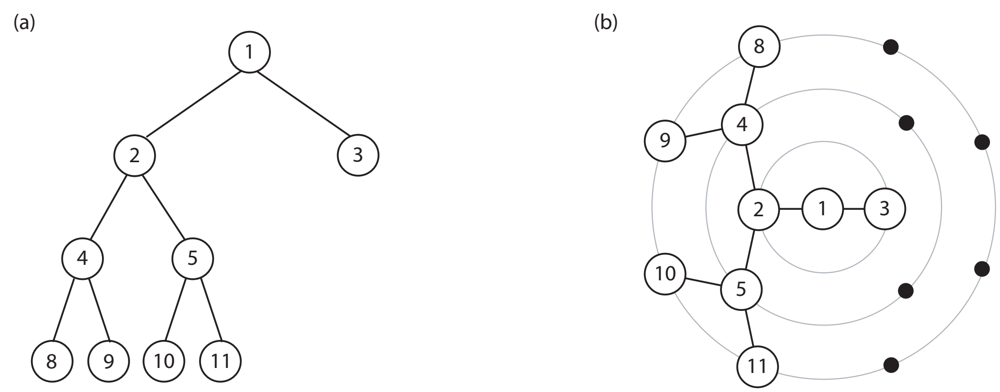


Fig. 2. Mapping an arbitrary plane binary tree to a circular grid. (a) Arbitrary plane binary tree of depth 3. (b) Corresponding circular grid

Visualizing a tree (2)

J. M. Daida et al GECCO'03

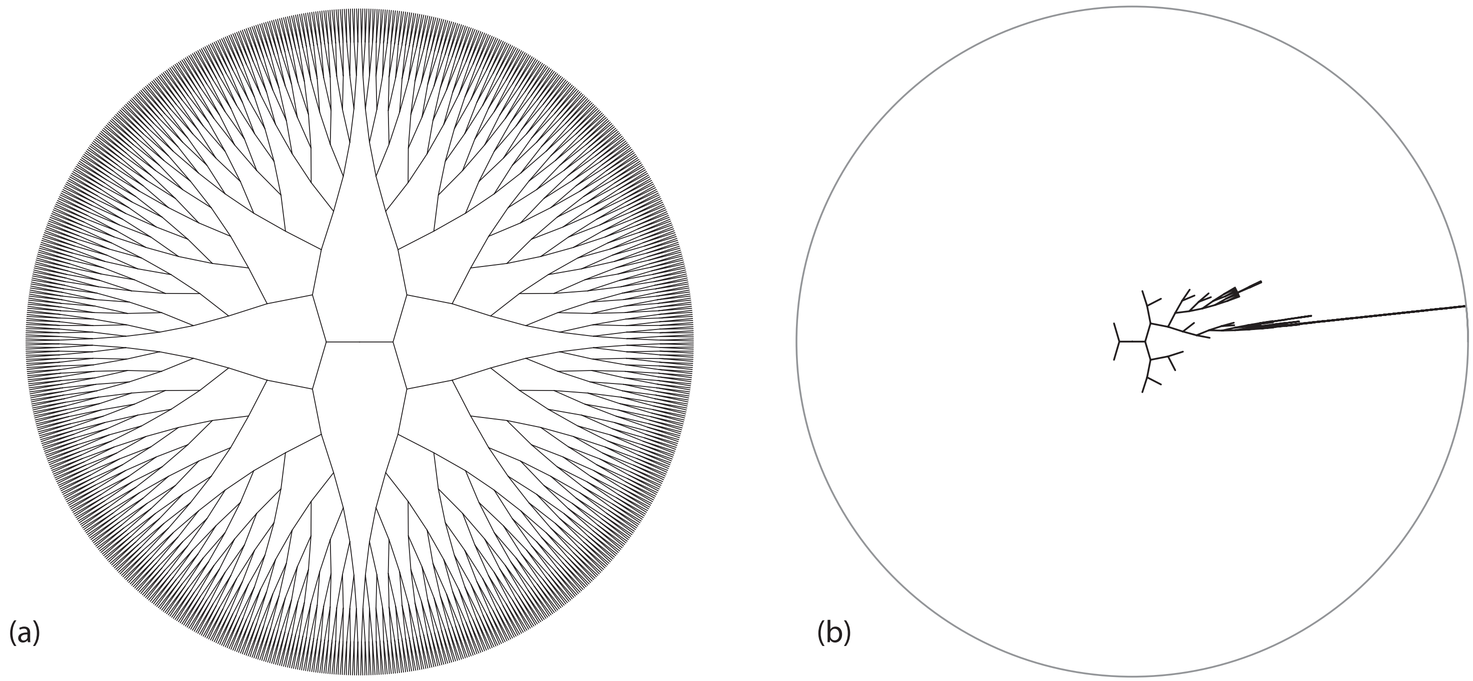
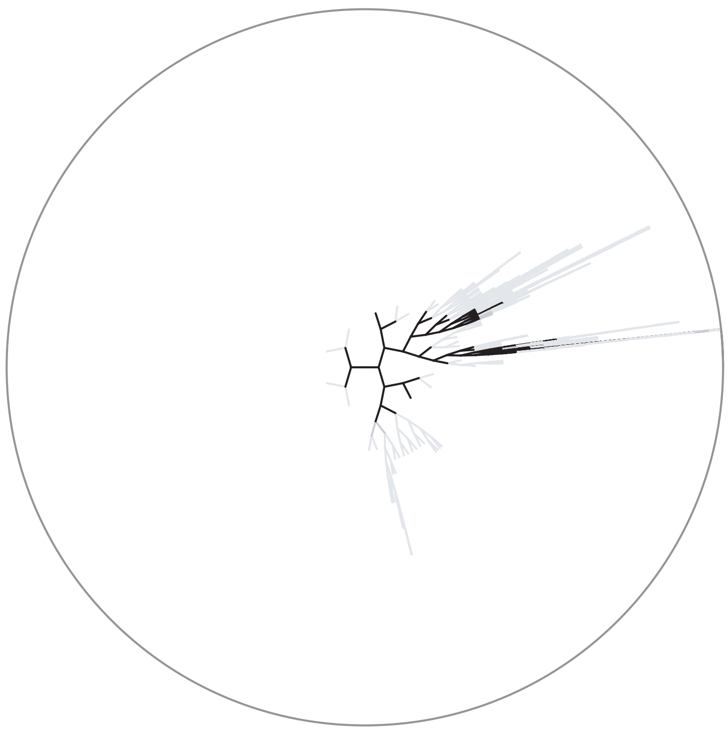


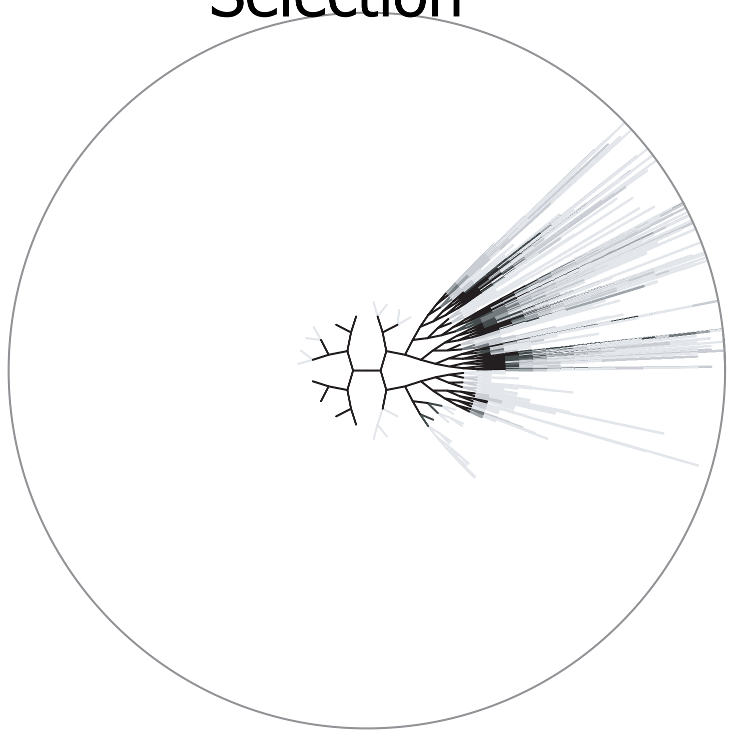
Fig. 3. Two examples of plane binary trees. (a) Full binary tree, depth 10. (b) Structure of a GP-generated solution (using arity-2 functions), depth 26. The gray circle around (b) represents depth 26 and is shown for reference

Visualizing a tree (3)
J. M. Daida et al GECCO'03

Tournament Selection



Fitness Proportional Selection



Parsimony Pressure

- Fitness = ‘error’ + α ‘complexity’ [2]
  - α is problem dependent;
  - Evolution of α ?
- Pareto Optimal [3]
  - Multi-objective fitness function
- Lexicographic [4]
  - Only use ‘complexity’ as a tie breaker;
  - Limited to countable problems?

Non-destructive Crossover [5]

- IF
  - child performance ≠ parent performance
- THEN
  - accept child
- (not to be confused with genetic 'hill-climbing')

Biased Crossover Operator

- Size Fair or Homologous Crossovers [6]
  - Emphasize routed tree approach to selection of crossover.
  - Second crossover point constrained to path defined by the first.
- Fitness direction
  - Estimate fitness with respect to each node of the individual

Fitness Direction – Motivation

- How significant are naïve biases
  - Each node has a fitness;
- Crossover only produces 1 child
  - Spend more time investigating same shape without recourse to 1-point crossover;
  - Mutation provides further investigation of reproduced ‘child’.

Selection of Crossover Point

- Random
  - Any link chosen with uniform probability.
- Deterministic
  - Select branch with respect to distance metric.
- Both
  - Provide for stochastic selection of either at some ratio.

Directed Crossover Heuristics

- Fitness
  - Estimate fitness at each node as tree evaluated
- Fitness Difference
  - Difference in node fitness
- Roulette-Fitness
  - Nodes allocated ‘roulette-wheel’ area $\propto$ node fitness

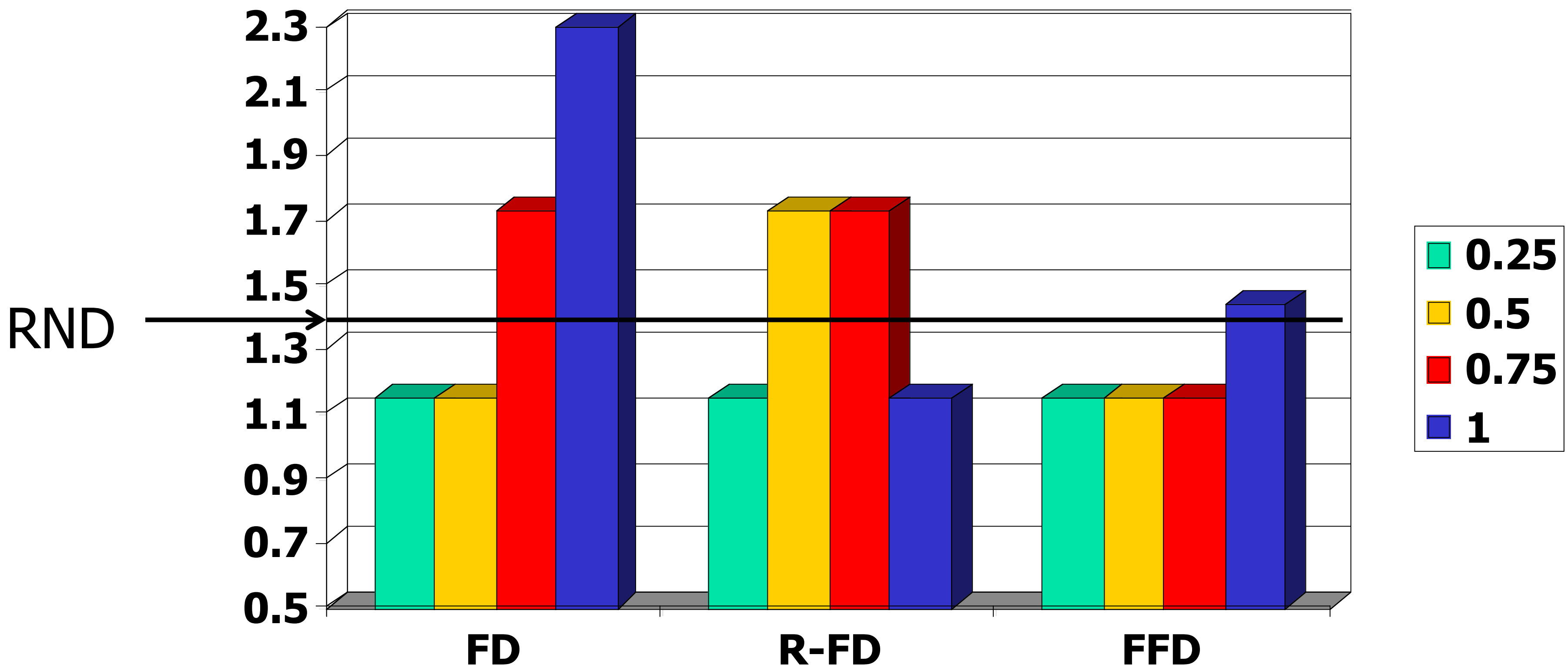
Methodology

- Investigate increasing levels of determinism
  - {0% (random only); 25%, 50%, 75% 100% (directed only)}
- Algorithm
  - IF (apply crossover == TRUE)
  - THEN IF (Directed Xover == TRUE)
    - THEN (apply Directed Xover)
    - ELSE (apply Random Xover)

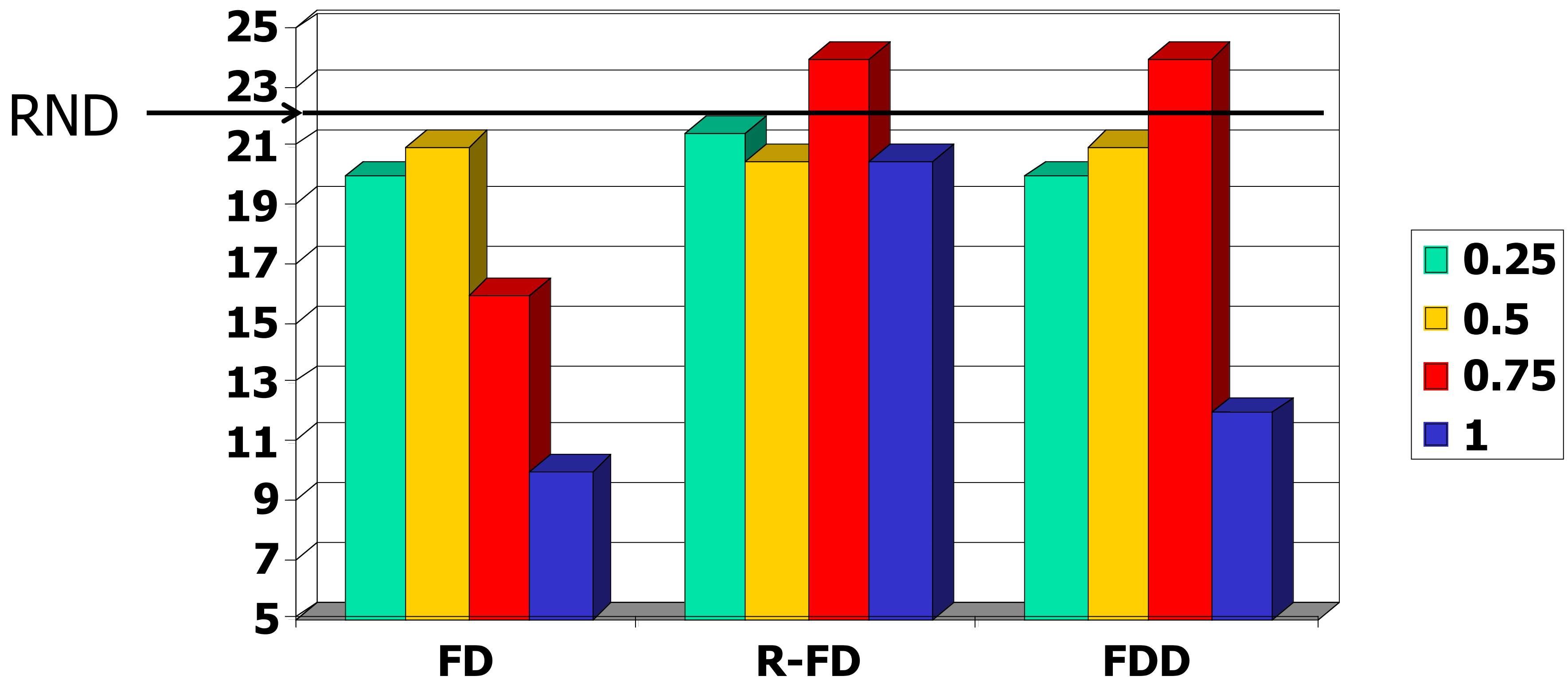
Classification Problems

Data set	Breast	Liver
Train	524	259
Test	175	86
Terminal	x(0),…,x(8)	x(0),…,x(5)
Function	+, -, *, %, sin, cos, sqrt	
Fitness	# correct classifications	
Pop. Size	500	
Search Op.	90% Xover; 50% mutation	
Termination	15,000 tournaments	

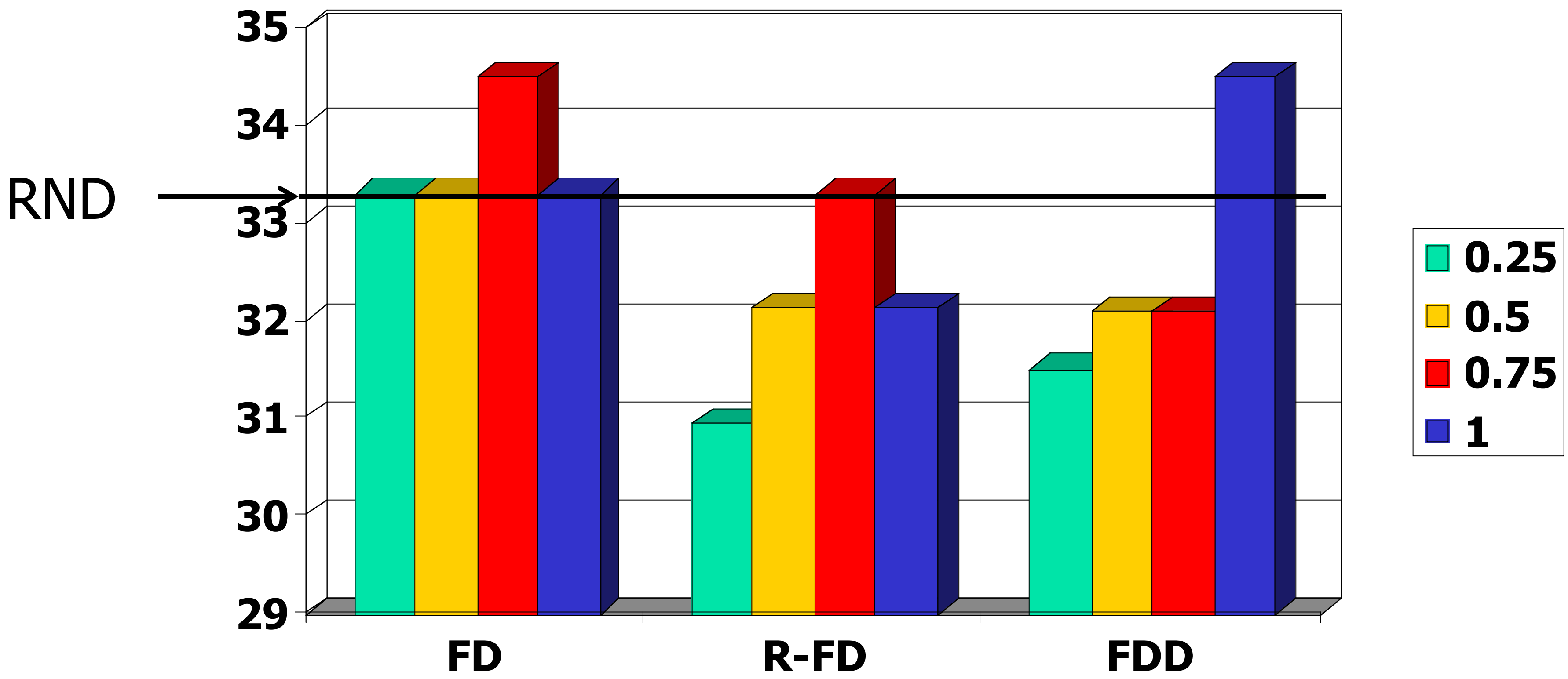
Breast Cancer – Median Test Error



Breast Cancer – Median Nodes per Solution

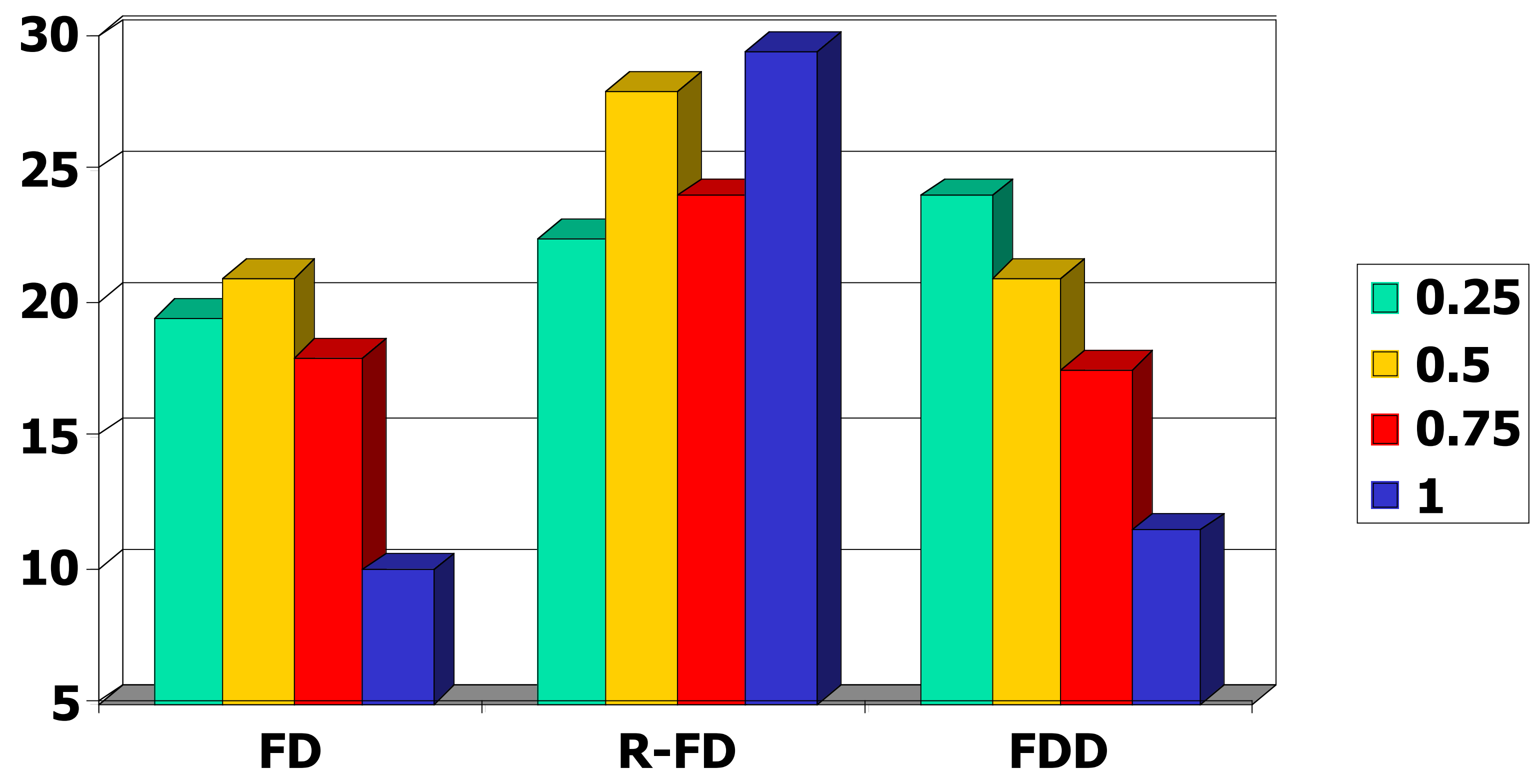


Liver Disease – Median Test Error

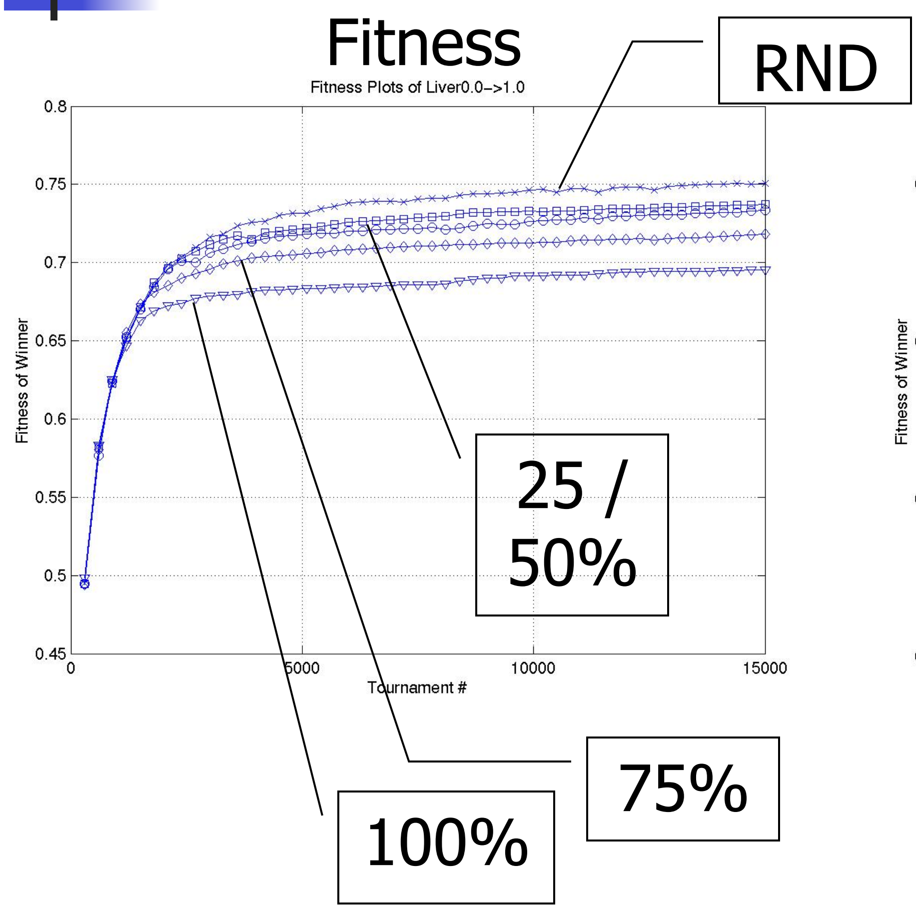


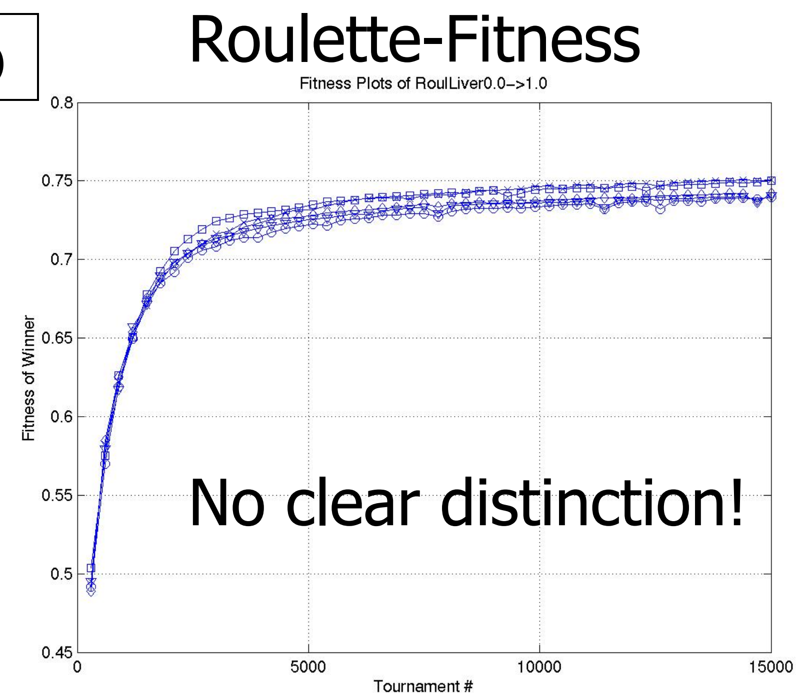
Liver Disease – Median Nodes per Solution

RND
34.5



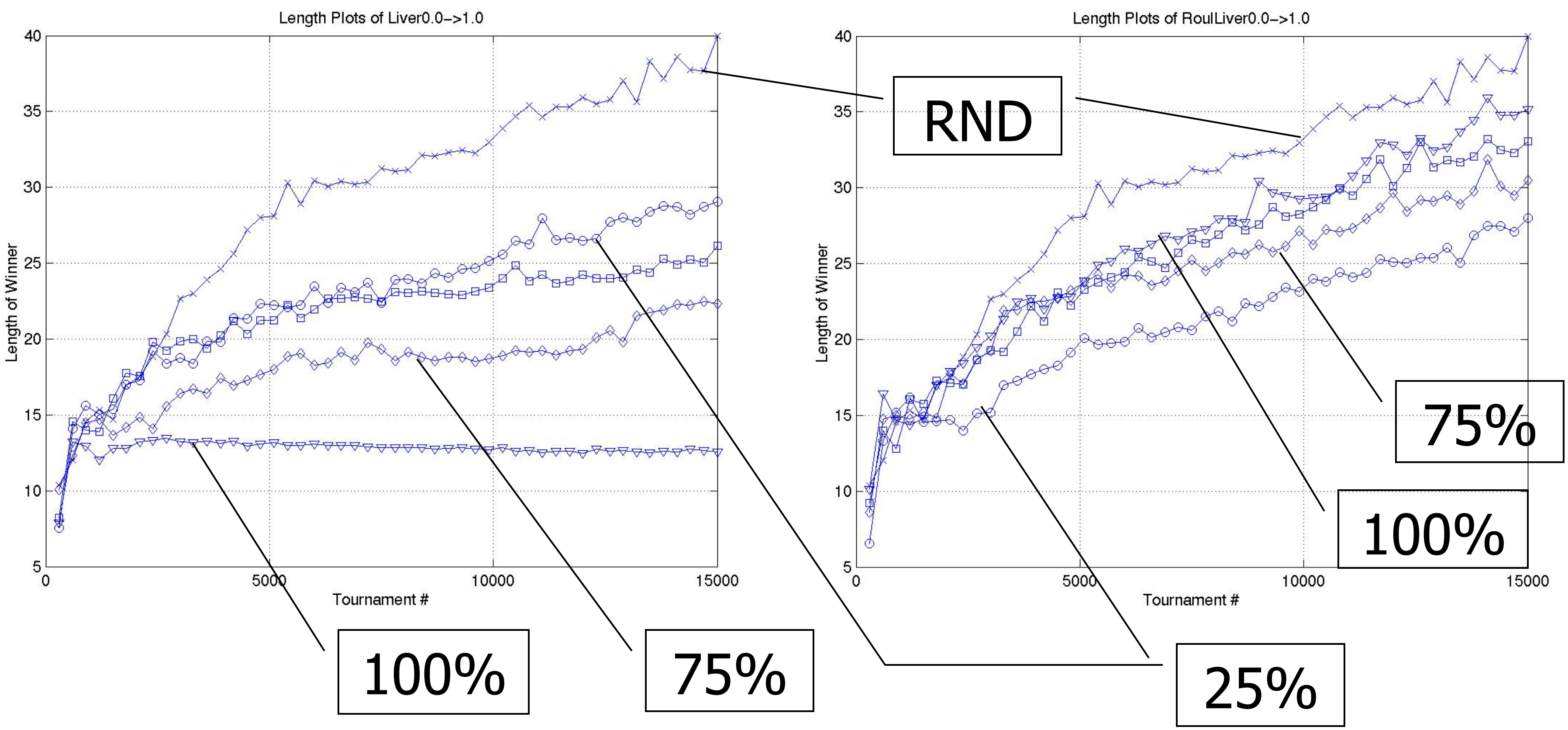
Evolution of Fitness: Liver – *Average* Classification Accuracy





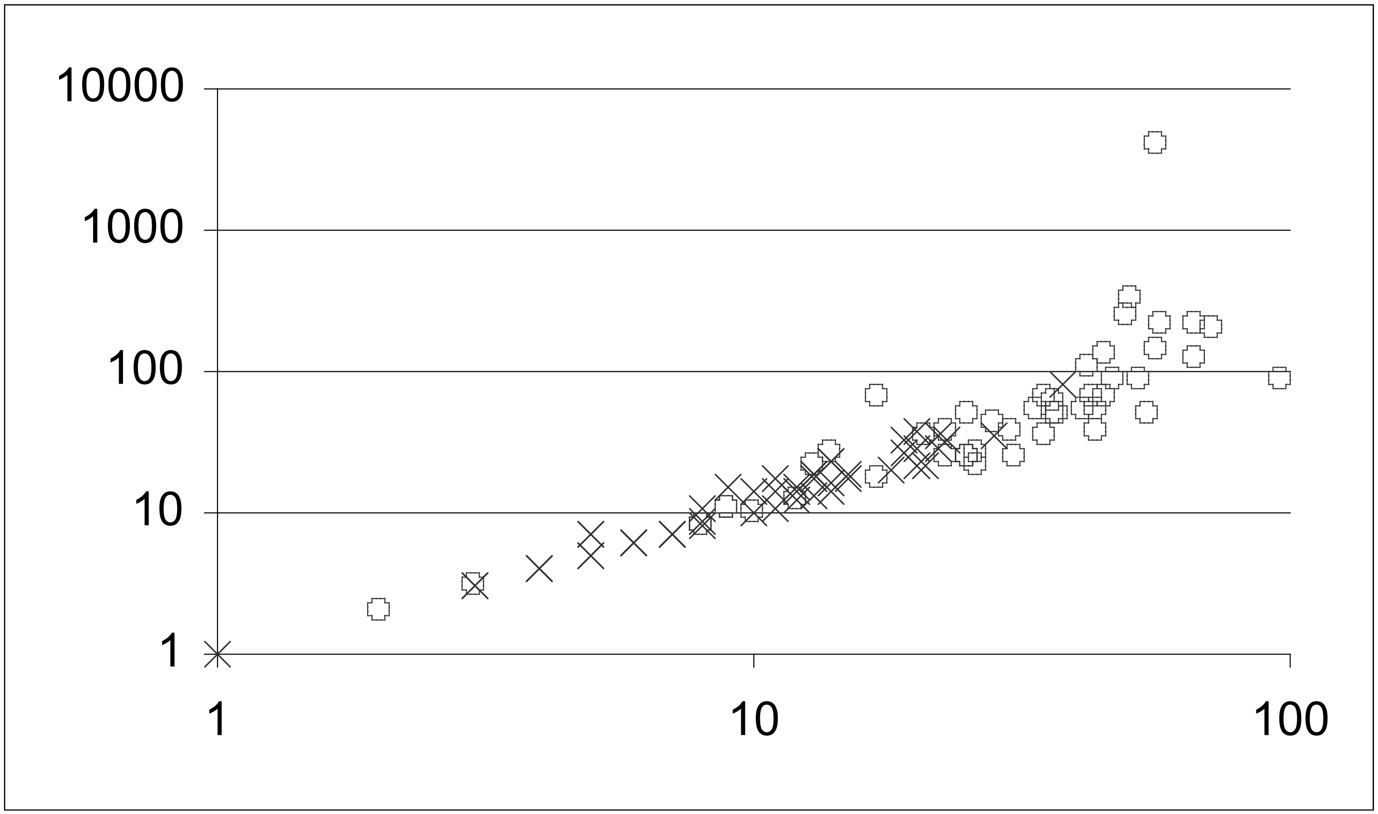
Evolution of Fitness: Liver – *Average* Node Count

Fitness | Roulette-Fitness



MAPLE™ 'simplification'

nodes After Simplification



×- FD
O - RND

nodes Before Simplification

Discussion

- Hands Off – v – Interactive
  - Removing Introns after solution located – v – Inclusion in the evolutionary cycle.
- Complexity – v – Accuracy
  - Can simple solutions still be accurate?
  - Are complex solutions useful?
- Solution bias?
  - Are biases application / difficulty specific?

References

1. Koza J.R., Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, 1992.
2. Soule T., Foster J.A., Effects of Code Growth and Parsimony Pressure on Populations in Genetic Programming. Evolutionary Computation 6(4) 293-309, 1998.
3. Ekart A., Nemeth S.Z., Selection based on the Pareto non-domination criterion for controlling code growth in GP. Genetic Programming and Evolvable Machines. 2(1) 61-73, 2001.
4. Luke S., Panait L., Lexicographic Parsimony Pressure. GECCO' 2002. Morgan Kaufmann, 829-836.

References (conti.)

5. Soule T. Foster J.A., Removal Bias, Proceedings of IEEE Int. Congress on Evolutionary Computation. 781-786, 1998.
6. Langdon W.B., Size Fair and Homologous Tree Crossovers for Tree GP. Genetic Programming and Evolvable Machines. 1(1/2) 95-120, 2000.
7. Terrio D., Heywood M.I., On Naïve Crossover Biases with Reproduction for Simple Solutions to Classification Problems. GECCO, 2004.
8. Gathercole C., Ross P., An adverse interaction between crossover operator and a restriction on tree depth. Proceedings of the First Annual Conference on GP. 291-296, 1996.
9. Daida J.M. et al. Visualizing tree structures in Genetic Programming. GECCO’03 pp 1652-1664