Evolution and Resiliency

Terry Soule
Department of Computer Science
University of Idaho
tsoule@cs.uidaho.edu

This work was made possible by NIH Grant # P20 RR16448 from the COBRE Program of the National Center for Research Resources.

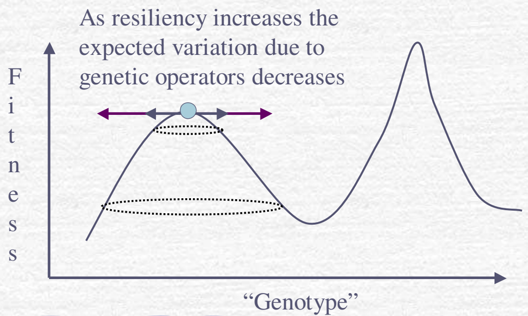
Goals

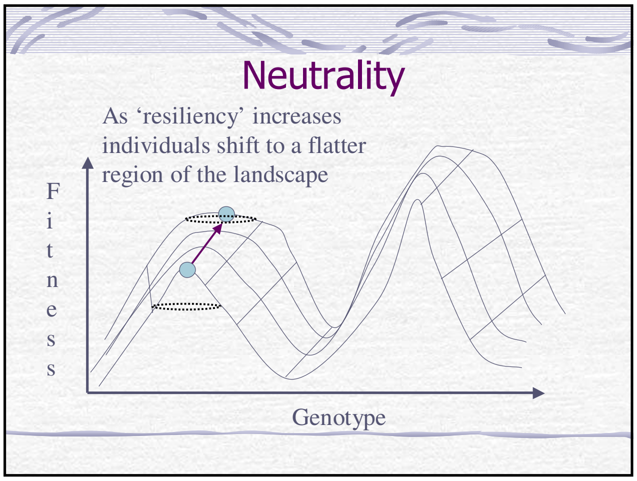
- Introduction to key results and literature on resiliency/genetic robustness (see bibliography)
- Specific illustrative results

Resiliency / (Genetic) Robustness

- “Robustness is the invariance of phenotypes in the face of (heritable) perturbation”[1]
- “Biological systems, from macromolecules to whole organisms, are robust if they continue to function, survive, or reproduce when faced with mutations, environmental change, and internal noise.”[22]

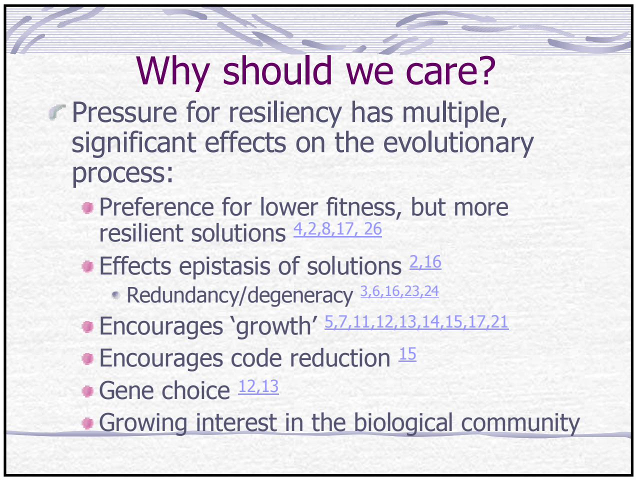
Resiliency/ Genetic Robustness


As resiliency increases the expected variation due to genetic operators decreases



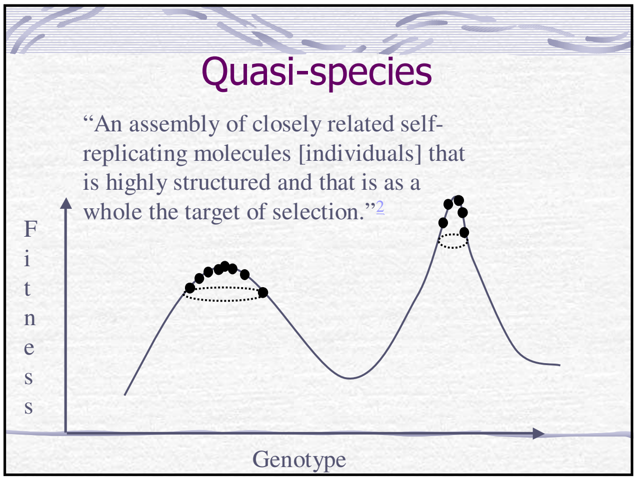
Neutrality

As 'resiliency' increases individuals shift to a flatter region of the landscape



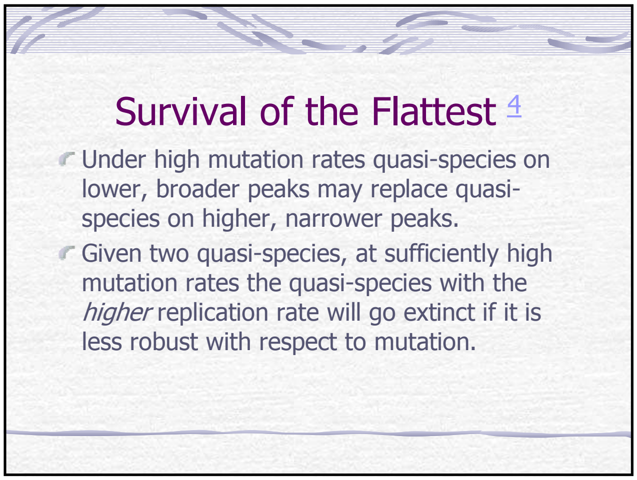
Why should we care?

- Pressure for resiliency has multiple, significant effects on the evolutionary process:
  - Preference for lower fitness, but more resilient solutions [4,2,8,17, 26]
  - Effects epistasis of solutions [2,16]
    - Redundancy/degeneracy [3,6,16,23,24]
  - Encourages 'growth' [5,7,11,12,13,14,15,17,21]
  - Encourages code reduction [15]
  - Gene choice [12,13]
  - Growing interest in the biological community



Quasi-species

"An assembly of closely related self-replicating molecules [individuals] that is highly structured and that is as a whole the target of selection."[2]



Survival of the Flattest [4]

- Under high mutation rates quasi-species on lower, broader peaks may replace quasi-species on higher, narrower peaks.
- Given two quasi-species, at sufficiently high mutation rates the quasi-species with the *higher* replication rate will go extinct if it is less robust with respect to mutation.

Neutral Networks [3]

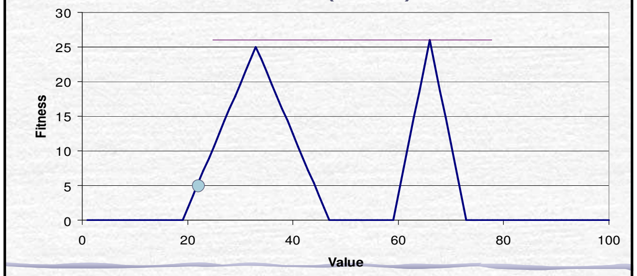
- In competition between quasi-species on neutral networks:
  - With low mutation rates the quasi-species that replicates more slowly goes extinct.
  - With high enough mutation rates the quasi-species on the sparser neutral network will go extinct even if it replicates more rapidly.

Population Size [6]

- Small populations are more likely to contain redundant genes than larger populations where:
  - Redundancy increases robustness
  - Redundancy imposes a ‘cost’ – lowers the replication rate
- In small populations good solutions are easier to lose.

Example: Two Peaks [17]

14140144401 (individual) ⇒ 23 (value=sum) ⇒ 5 (fitness)



Parameters

- Generational
- No mutation
- Crossover rate = 0.9
- Elitism: 2 members
- Tournament selection (3)
- Population size 500
- Crossover: 2 point, variable size

Constant Crossover 15,17

- Crossover:
  - Pick a crossover length for each parent:

    length = 2

    Repeat

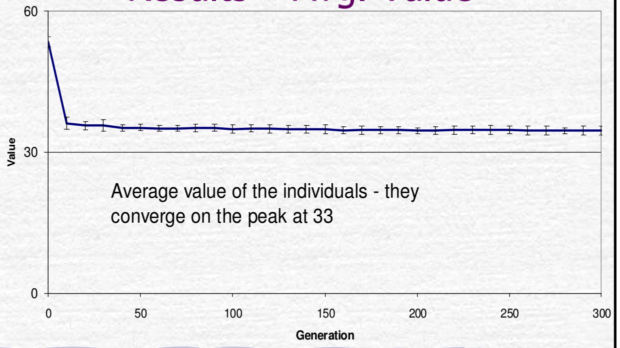
    With probability .5 double length
  - Pick random starting points in each parent
  - Swap regions

0010|4001|0040011

00040401110|14|011

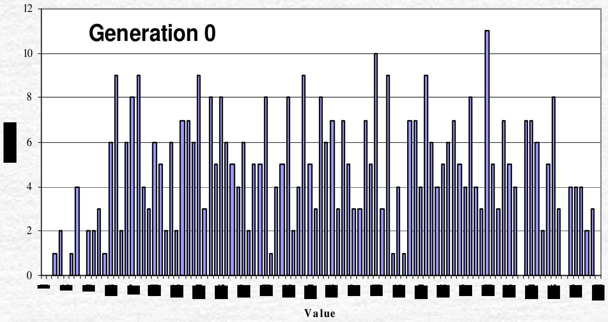
- Exponentially distributed sizes
- Sizes are independent of parent's length

Results – Avg. Value

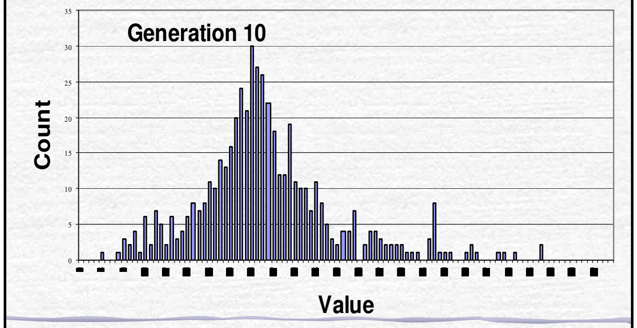


Average value of the individuals - they converge on the peak at 33

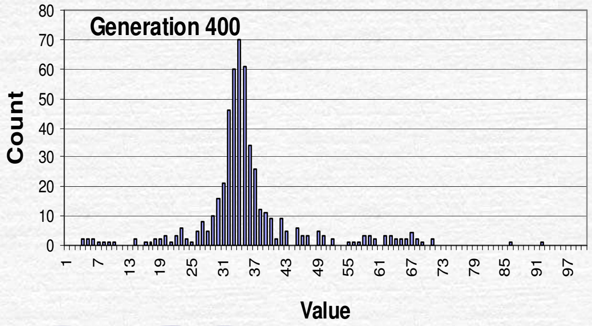
Distribution of Values



Distribution of Values



Distribution of Values



Conclusions I

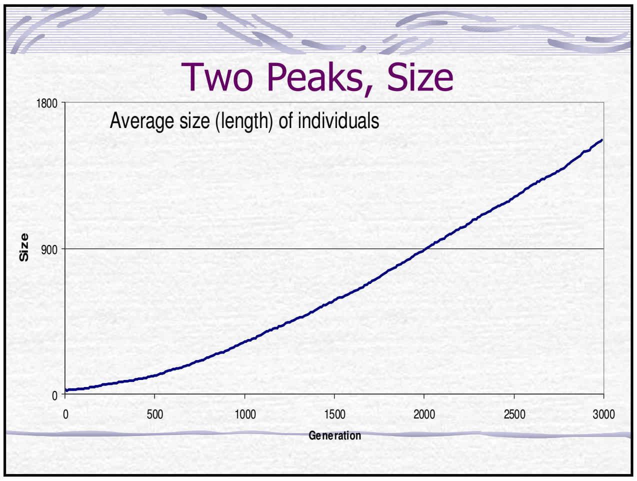
- Robust/Resilient quasi-species may out compete less robust, but more fit quasi-species
- E.g. population converges on lower, broader peak, despite ‘knowledge’ (due to elitism) of higher peak.

Code Growth/Bloat

- Increase in size not correlated to an increase in fitness
  - Growth consists of code having a minimal effect on fitness
- Growth in GP was originally suggested a mechanism to protect against crossover [11,12]
  - Ratio of exons/introns decreases
  - Not necessarily exons and introns [5,12,13]
- E.g. growth increases robustness (w.r.t crossover, a given population, fitness, etc.)

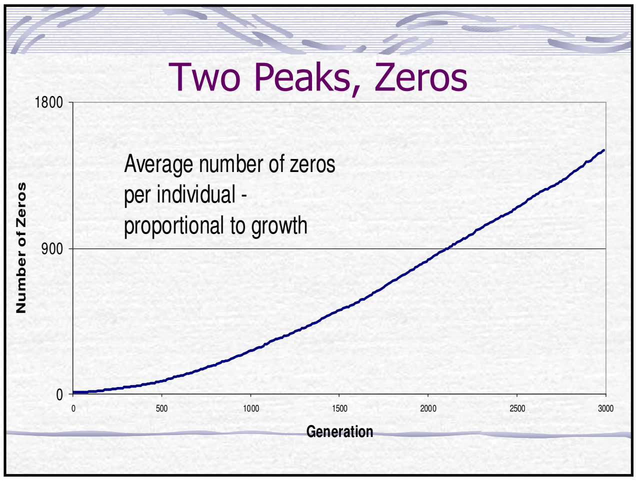
Types of Code

- Introns vs. Exons
- Viable vs. Inviable and Operative vs. Inoperative
- Others [5,11,12]
  1. inviable
  2. inviable for fitness cases
  3. Can be replaced by a no-op
  4. Can be replaced for fitness cases
  5. Continuously defined ‘value’
- Introns not the only source of growth [5,12,13]



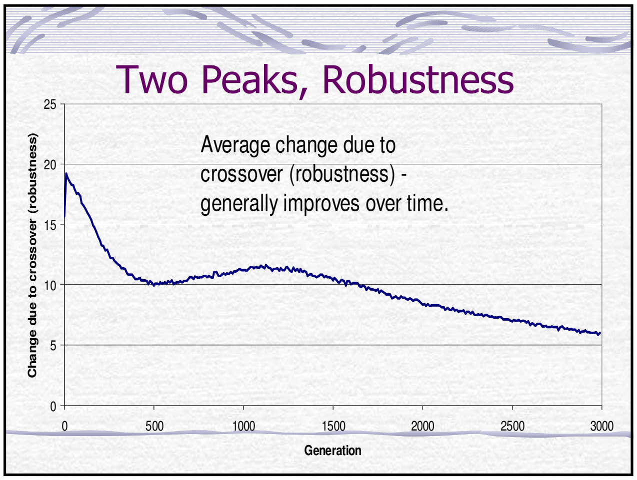
Two Peaks, Size

Average size (length) of individuals



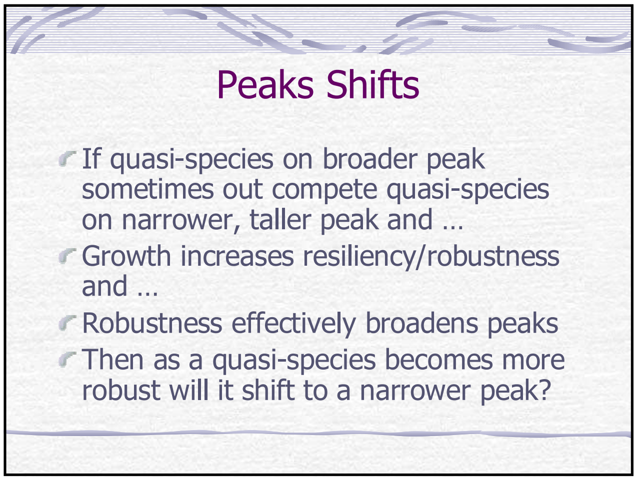
Two Peaks, Zeros

Average number of zeros
per individual -
proportional to growth



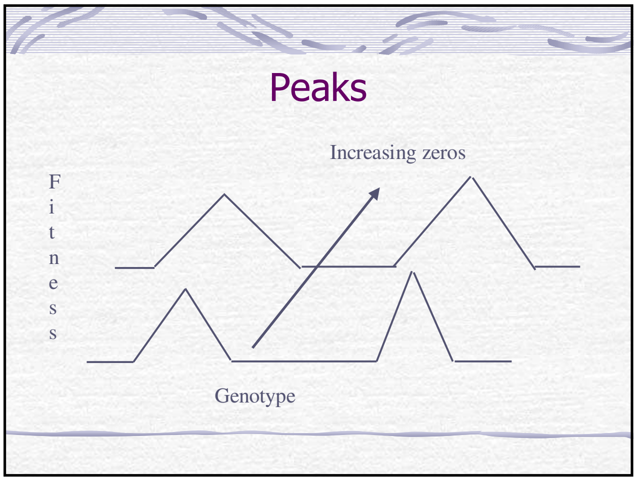
Two Peaks, Robustness

Average change due to
crossover (robustness) -
generally improves over time.

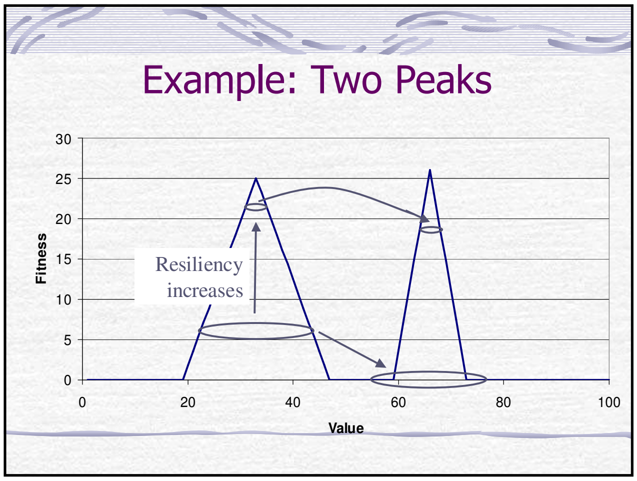


Peaks Shifts

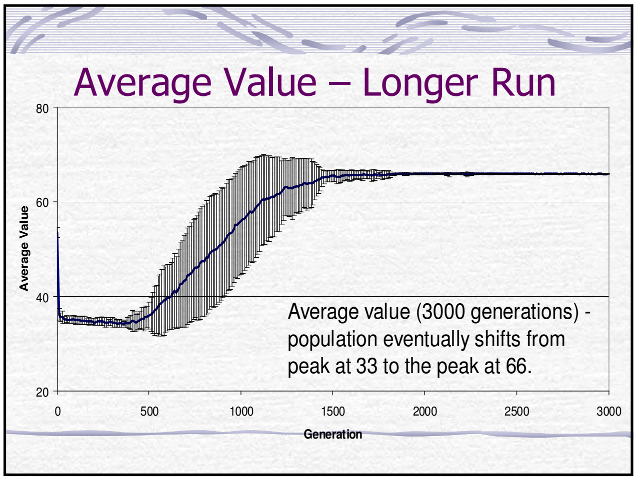
- If quasi-species on broader peak sometimes out compete quasi-species on narrower, taller peak and …
- Growth increases resiliency/robustness and …
- Robustness effectively broadens peaks
- Then as a quasi-species becomes more robust will it shift to a narrower peak?



Peaks

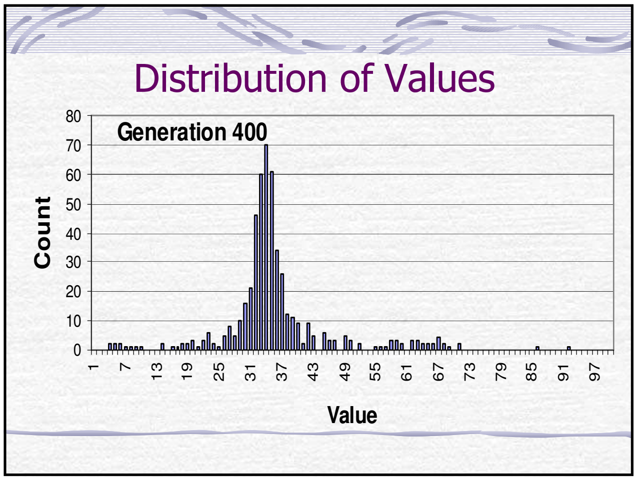


Example: Two Peaks

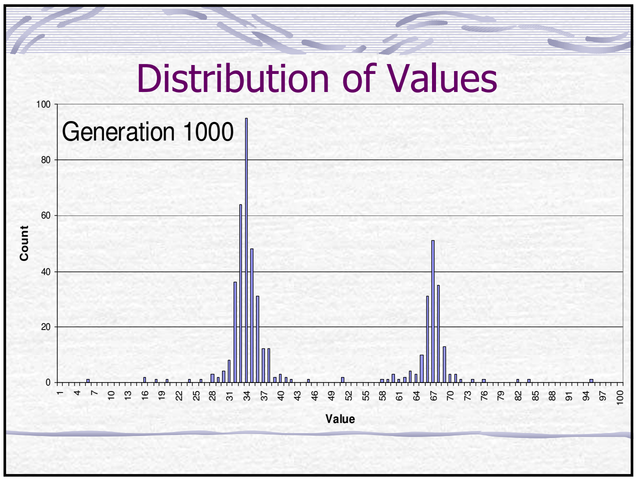


Average Value – Longer Run

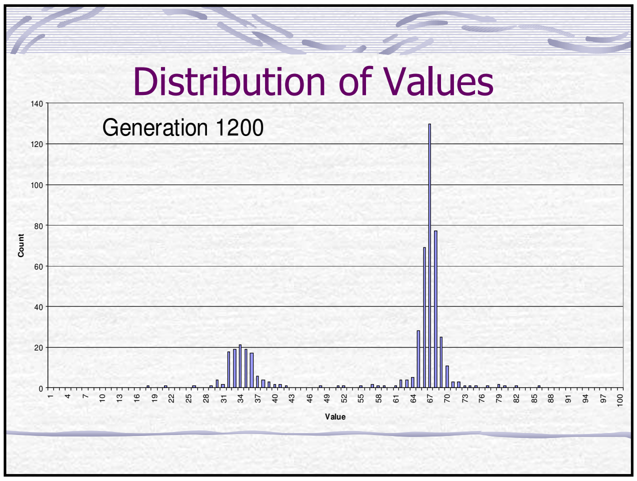
Average value (3000 generations) - population eventually shifts from peak at 33 to the peak at 66.



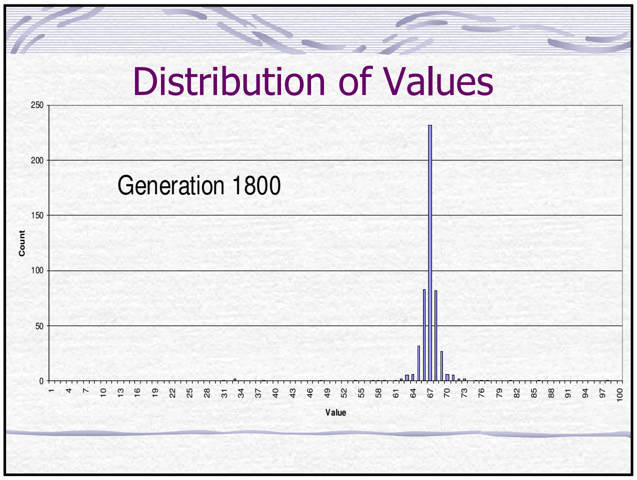
Distribution of Values



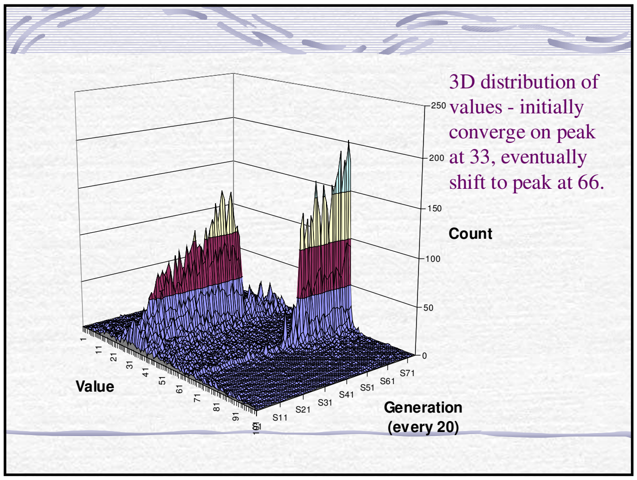
Distribution of Values



Distribution of Values

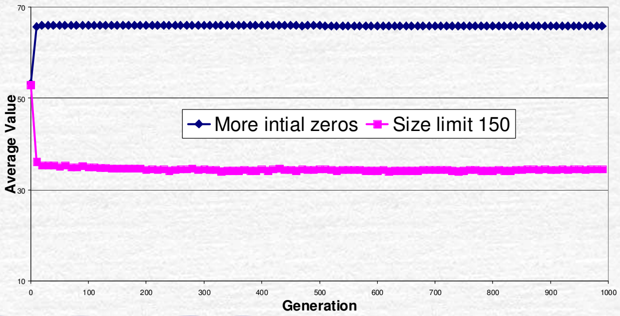


Distribution of Values



3D distribution of values - initially converge on peak at 33, eventually shift to peak at 66.

Controls



Conclusions II

- Broader, but lower, peaks may be favored
- Growth can increase robustness (other robustness strategies exist)
- Increasing robustness allows shifts to narrower peaks
- More robustness (growth) required to shift to narrower peaks (not shown) [17]
- Limiting growth can limit shifts
- Can this dynamic be shown for a more complex problem?[25]

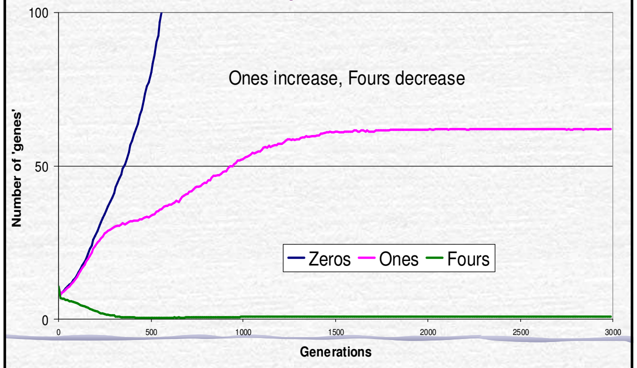
Other Robustness Strategies

- Epistasis [2,16]
- ‘Gene’ choice [12,13]
- Code reduction [15]
- Redundant Genes [6,16]
- Degeneracy [16]
- Gene location?
- Others???

Epistasis [2,16]

- In individuals adapting to a high mutation rate (from a lower one).
  - Increase in the number of neutral mutations
  - Decrease in coupling between genes

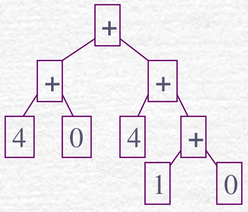
Two Peaks, Gene Choice



GP Experiment

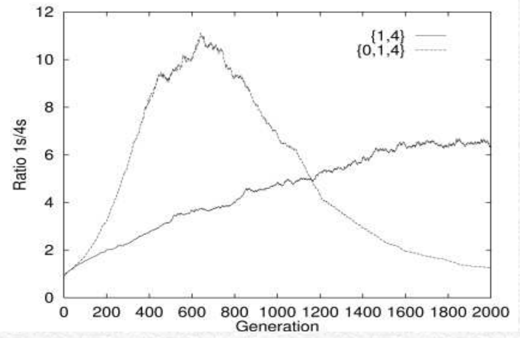
Goal	Expression with value 29
Fitness	|output – 29|
Terminals	0,1,4 or 1, 4
Non-terminal	+
Population size	800
Generations	2000
Selection	3-member tournament
Trials	50
Mutation	0.001/node
Crossover	0.9
Size limit	None
Initial population	Ramped half-and-half

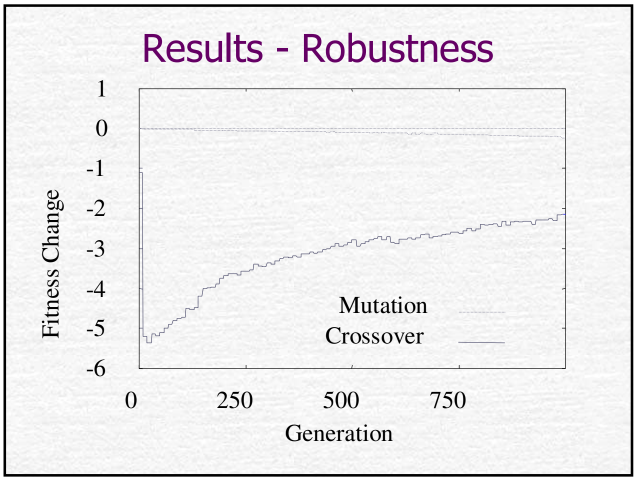
Sample GP Individual



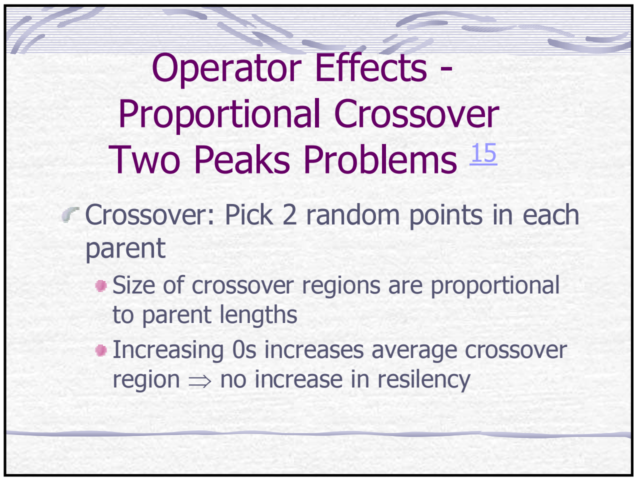
A sample individual with fitness 20.

Results – 1:4 Ratio



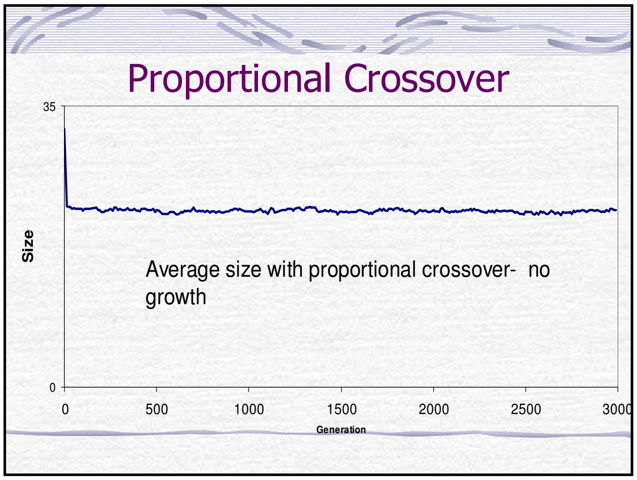


Results - Robustness



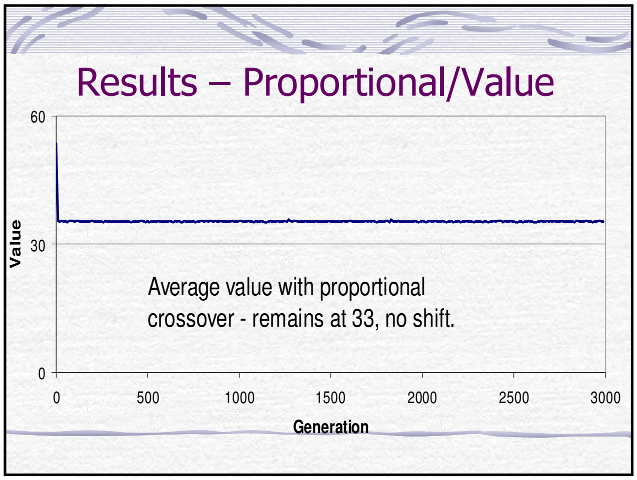
Operator Effects - Proportional Crossover Two Peaks Problems [15]

- Crossover: Pick 2 random points in each parent
  - Size of crossover regions are proportional to parent lengths
  - Increasing 0s increases average crossover region ⇒ no increase in resiliency



Proportional Crossover

Average size with proportional crossover- no growth



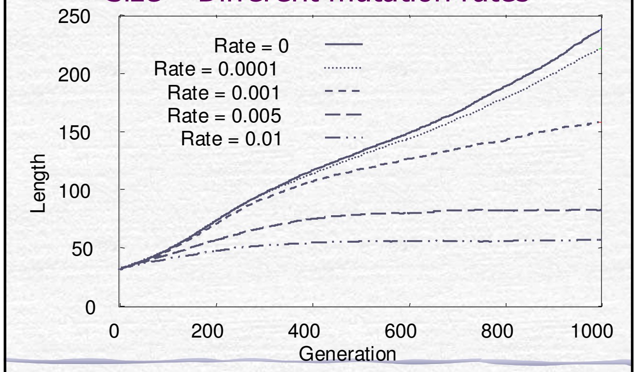
Results – Proportional/Value

Average value with proportional crossover - remains at 33, no shift.

Operator Effects – Mutation I

- Mutation – probability p of mutating a ‘gene’
- Mutation rate per ‘gene’
- More 0s -> greater chance of one of them being mutated
- Single peak experiment

Size – Different mutation rates



Operator Effects - Mutation II

- GP with exactly N mutations per individual
- Mutation rate per individual
- More introns, greater chance of ‘hiding’ a mutation
- With this type of mutation growth increases [14]

Conclusions III

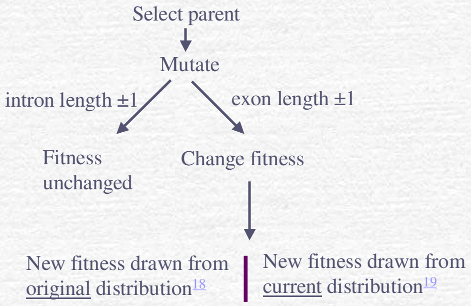
Strategy adopted to increase robustness depends on operators used

- Changes per individual encourage growth.
  - Ex: on average GP crossover effects ~3 nodes per individual.
- Changes per ‘gene’ don’t encourage growth.
  - Ex: on average GP single node mutation effects M percent of the nodes.
- How do operators influence other robustness strategies?

Representationless Model [18,19,20]

- Individuals have no representation
- Individuals have:
  - Fitness (initially assigned arbitrarily from a preset distribution)
  - Non-coding length (l_i)
  - Coding length (l_e)
  - Total length = $l_i + l_e$
- 'Mutation' only
- Mutation effects l_i or l_e, (±1)
- if l_e then fitness changes

Representationless Model


Select parent

Mutate

intron length ±1

exon length ±1

Fitness unchanged

Change fitness

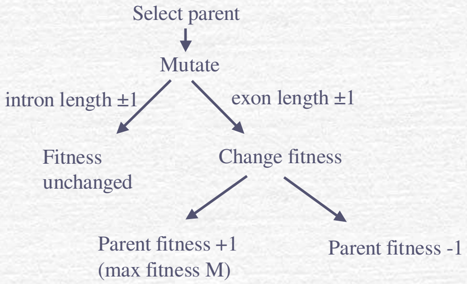
New fitness drawn from original distribution[18]

New fitness drawn from current distribution[19]

Results [18,19]

- Fitness drawn from the initial distribution:
  - (Weak) Growth with selection
  - No growth without selection
- Fitness drawn from the current distribution:
  - Size is a random walk
  - Growth is a result of the boundary at size zero

Representationless Model


Select parent

Mutate

intron length ±1

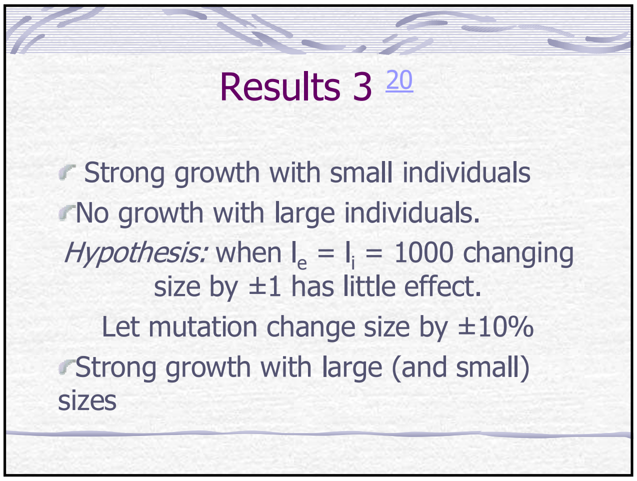
exon length ±1

Fitness unchanged

Change fitness

Parent fitness +1 (max fitness M)

Parent fitness -1



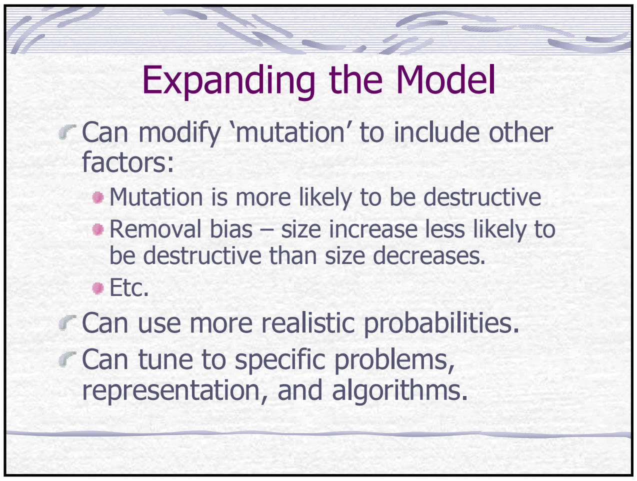
Results 3 20

- Strong growth with small individuals
- No growth with large individuals.

Hypothesis: when $l_e = l_i = 1000$ changing size by ±1 has little effect.

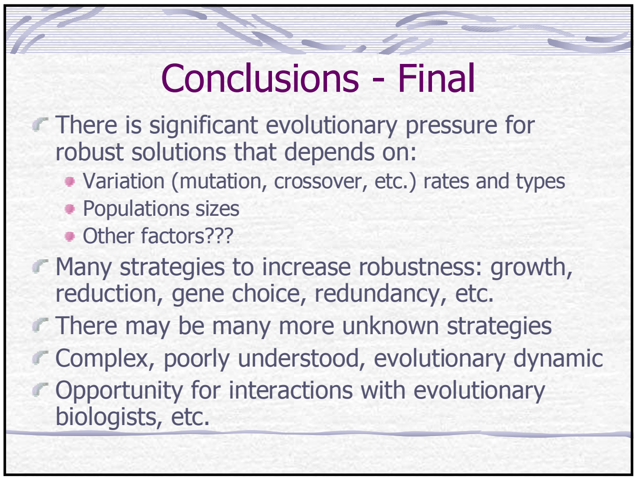
Let mutation change size by ±10%

- Strong growth with large (and small) sizes



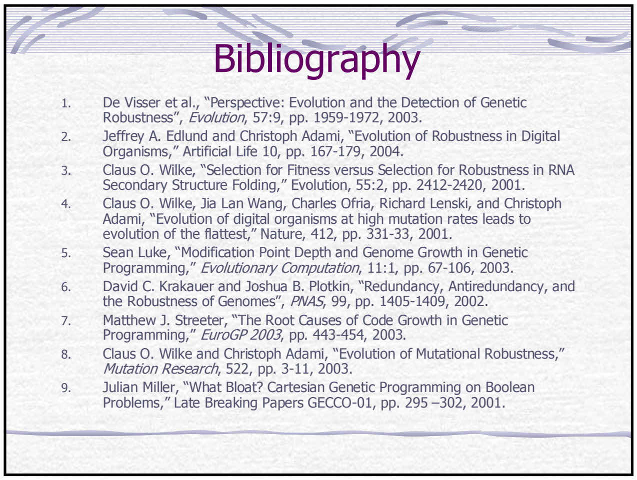
Expanding the Model

- Can modify 'mutation' to include other factors:
  - Mutation is more likely to be destructive
  - Removal bias – size increase less likely to be destructive than size decreases.
  - Etc.
- Can use more realistic probabilities.
- Can tune to specific problems, representation, and algorithms.



Conclusions - Final

- There is significant evolutionary pressure for robust solutions that depends on:
  - Variation (mutation, crossover, etc.) rates and types
  - Populations sizes
  - Other factors???
- Many strategies to increase robustness: growth, reduction, gene choice, redundancy, etc.
- There may be many more unknown strategies
- Complex, poorly understood, evolutionary dynamic
- Opportunity for interactions with evolutionary biologists, etc.



Bibliography

1. De Visser et al., "Perspective: Evolution and the Detection of Genetic Robustness", *Evolution*, 57:9, pp. 1959-1972, 2003.
2. Jeffrey A. Edlund and Christoph Adami, "Evolution of Robustness in Digital Organisms," Artificial Life 10, pp. 167-179, 2004.
3. Claus O. Wilke, "Selection for Fitness versus Selection for Robustness in RNA Secondary Structure Folding," Evolution, 55:2, pp. 2412-2420, 2001.
4. Claus O. Wilke, Jia Lan Wang, Charles Ofria, Richard Lenski, and Christoph Adami, "Evolution of digital organisms at high mutation rates leads to evolution of the flattest," Nature, 412, pp. 331-33, 2001.
5. Sean Luke, "Modification Point Depth and Genome Growth in Genetic Programming," *Evolutionary Computation*, 11:1, pp. 67-106, 2003.
6. David C. Krakauer and Joshua B. Plotkin, "Redundancy, Antiredundancy, and the Robustness of Genomes", *PNAS*, 99, pp. 1405-1409, 2002.
7. Matthew J. Streeter, "The Root Causes of Code Growth in Genetic Programming," *EuroGP 2003*, pp. 443-454, 2003.
8. Claus O. Wilke and Christoph Adami, "Evolution of Mutational Robustness," *Mutation Research*, 522, pp. 3-11, 2003.
9. Julian Miller, "What Bloat? Cartesian Genetic Programming on Boolean Problems," Late Breaking Papers GECCO-01, pp. 295 –302, 2001.

Bibliography

10. Tobias Blickle and Lothar Theile, “Genetic Programming and Redundancy,” *Genetic Algorithms within the Framework of Evolutionary Computation* (Workshop at KI-94, Saarbrucken), pp. 33-38, 1994.
11. Peter Nordin and Wolfgang Banzhaf, “Complexity Compression and Evolution,” *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp. 310-317, 1995.
12. Kim Harries and Peter Smith, “Code Growth, Explicitly Defined Introns and Alternate Selection Schemes,” *Evolutionary Computation*, 6:4, pp. 39-360, 1998.
13. Terence Soule, “Exons and Code Growth in Genetic Programming,” *EuroGP-2002*, pp. 142-151, 2002.
14. Terence Soule and Robert B. Heckendorn, “An Analysis of the Causes of Code Growth in Genetic Programming,” *Genetic Programming and Evolvable Machines*, 3:3, pp. 283-309, 2002.
15. Terence Soule, “Operator Choice and the Evolution of Robust Solutions,” *Genetic Programming Theory and Practice*, Ch. 16, pp. 257-269, 2003.
16. M. Nicolau, C. Ryan and Christopher R. Stephens, “Zero is not a four letter word: Studies in the evolution of language,” *EuroGP-2005*, 2005.
17. Terence Soule, “Resiliency Improves Evolutionary Search,” Journal of Artificial Life, In Press.

Bibliography

18. W. Banzhaf and W. B. Langdon, “Some Considerations on the Reasons for Bloat,” *Genetic Programming and Evolvable Machines*, 3, pp. 81-91, 2002.
19. Justinian Rosca, “A Probablisitic Model of Size Drift,” Genetic *Programming Theory and Practice*, Rick Riolo and Bill Worzel Eds., Kluwer, pp. 119-135, 2003.
20. Xian Liu and Terence Soule, In Preparation, 2005.
21. Nicholas McPhee and Justin Miller, “Accurate Replication in Genetic Programming,” *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp. 303-309, 1995.”
22. Andreas Wagner, “Robustness, evolvability, and neutrality,” *FEBS Letters*, 579, pp. 1172-1778, 2005.
23. Andreas Wagner, “Distributed robustness versus redundancy as causes of mutational robustness,” *BioEssays*, 27, pp. 176=188, 2005.
24. William Langdon and Riccardo Poli, “Repeated Patterns in Tree Genetic Programming,” *EuroGP 2005*, 2005.

Bibliography

25. Alan Piszcz and Terence Soule, “Dynamics of Evolutionary Robustness,” *GECCO 2006*, 2006.
26. Russell Thomason and Terence Soule, “Redundant Genes and the Evolution of Robustness,” *GECCO 2006*, 2006.
27. Xue Zhong and Terence Soule, “Growth of Self-Canceling Code in Evolutionary Systems,” *GECCO 2006*, 2006.