Don't Evaluate, Inherit

Kumara Sastry
David E. Goldberg
Martin Pelikan

Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801
http://www-illigal.ge.uiuc.edu

Genetic and Evolutionary Computation Conference
(GECCO-2001)
July 7-11, 2001
San Francisco, CA

Foreword

- Fitness inheritance is simple
- Inheritance eliminates evaluation
- Can be easily modeled & predicted

Overview

- Background & motivation
- Fitness inheritance
- Objective
- Facetwise models
  - Constant solution quality
  - Fixed population size
- Conclusions

Background

- Design of *competent* GAs: A key challenge
  - Solve hard problems quickly, reliably, and accurately
  - Significant progress made (Goldberg, 1999)
  - Require subquadratic function evaluations
- High for large-scale problems
- *Efficiency enhancement techniques* (EETs)
  - Parallelism, Hybridization, Time Continuation, Evaluation Relaxation

Motivation

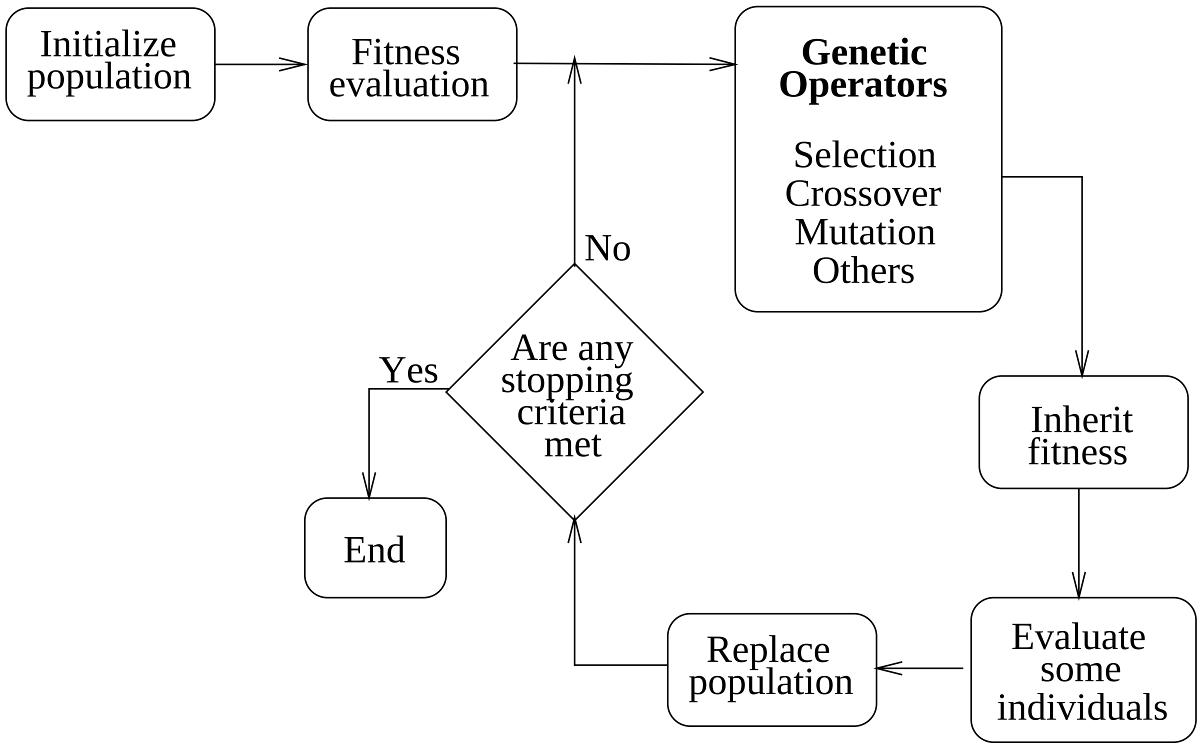
- Fitness inheritance is an EET
  - Offspring inherits fitness from parents
  - Thus eliminates fitness evaluation
- Smith, Dike & Stegmann (1995)
  - Offsprings inherit average parent fitness
  - Reported very high speed-up
- Analytical investigation required

Fitness Inheritance

- Offsprings assigned average fitness of parents
- Computed during crossover
- Inheritance is a very simple operation
- Fitness evaluation eliminated
- Introduces noise

Parents	Fitness	Offsprings	Inh Fit	True Fit
110000	111111	2	3	6
001111	000000	4	3	0

GA With Fitness Inheritance



Objective

- Model fitness inheritance in GAs
  - Convergence time model
  - Population-sizing model
- Optimize inheritance for greatest speedup
- Verify theory with empirical results
- Compare results with those of Smith et al

Assumptions

- Fixed population size
- Non-overlapping population
- Generational GAs
- Binary encoding and fixed string length
- Stationary fitness functions
- models for OneMax domain

Fitness Inheritance Model

- Actual fitness distribution: Gaussian

$$f \sim \mathcal{N}(\mu_{f,t}, \sigma^2_{f,t})$$

- Inherited fitness is average BB fitness
- Inherited fitness distribution: Gaussian

$$f' \sim \mathcal{N}(\mu_{f,t}, (1-p_i)\sigma^2_{f,t})$$

- p_i is inheritance proportion: (n_i/n)
  - n_i: No. of indivs. with inherited fitness
  - n: Population size

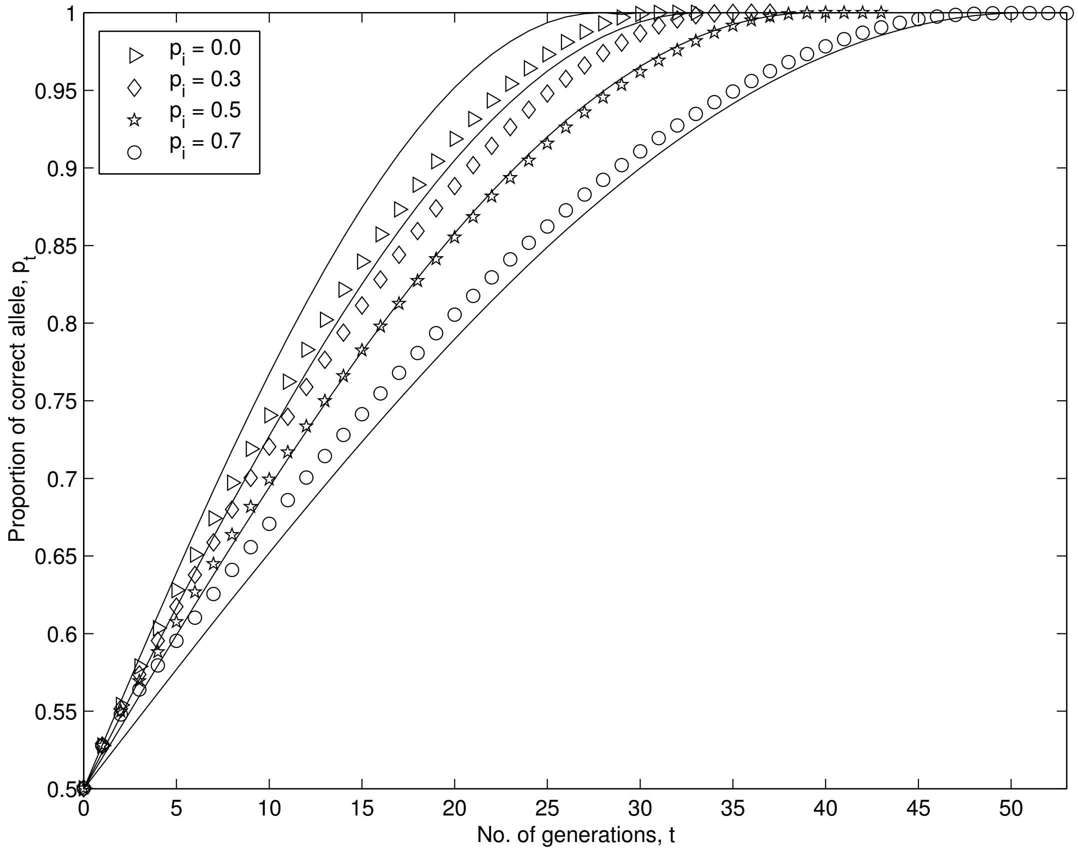
Convergence Time Model

- Mühlenbein & Schlierkamp-Voosen (1993); Thierens & Goldberg (1993); Bäck (1994); Miller & Goldberg (1996)
  - Use Selection Intensity (Bulmer,1980)
- Prop. of correct BBs & convergence time

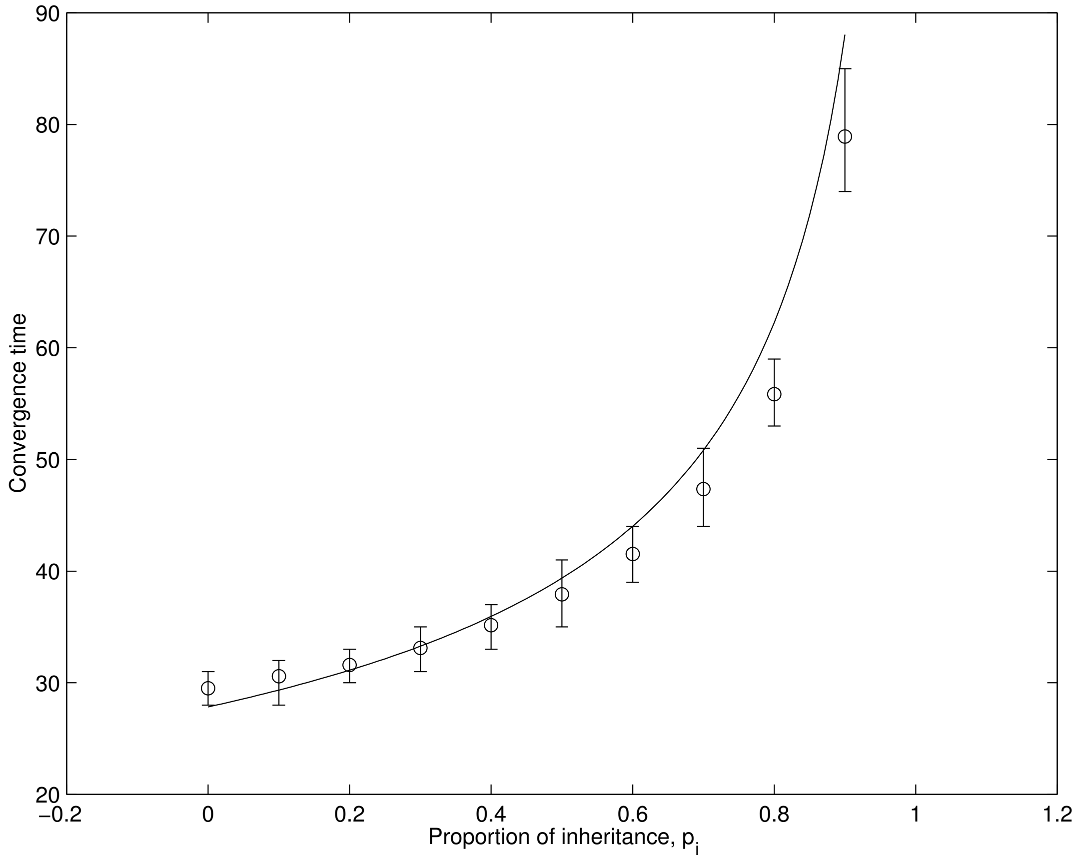
$$p_t = \frac{1}{2}\left[1 + \sin\left(It\sqrt{\frac{1-p_i}{\ell}}\right)\right]$$

$$t_{\text{conv}} = \frac{\pi}{2I}\sqrt{\frac{\ell}{1-p_i}}$$

Proportion of Correct BBs



Convergence Time

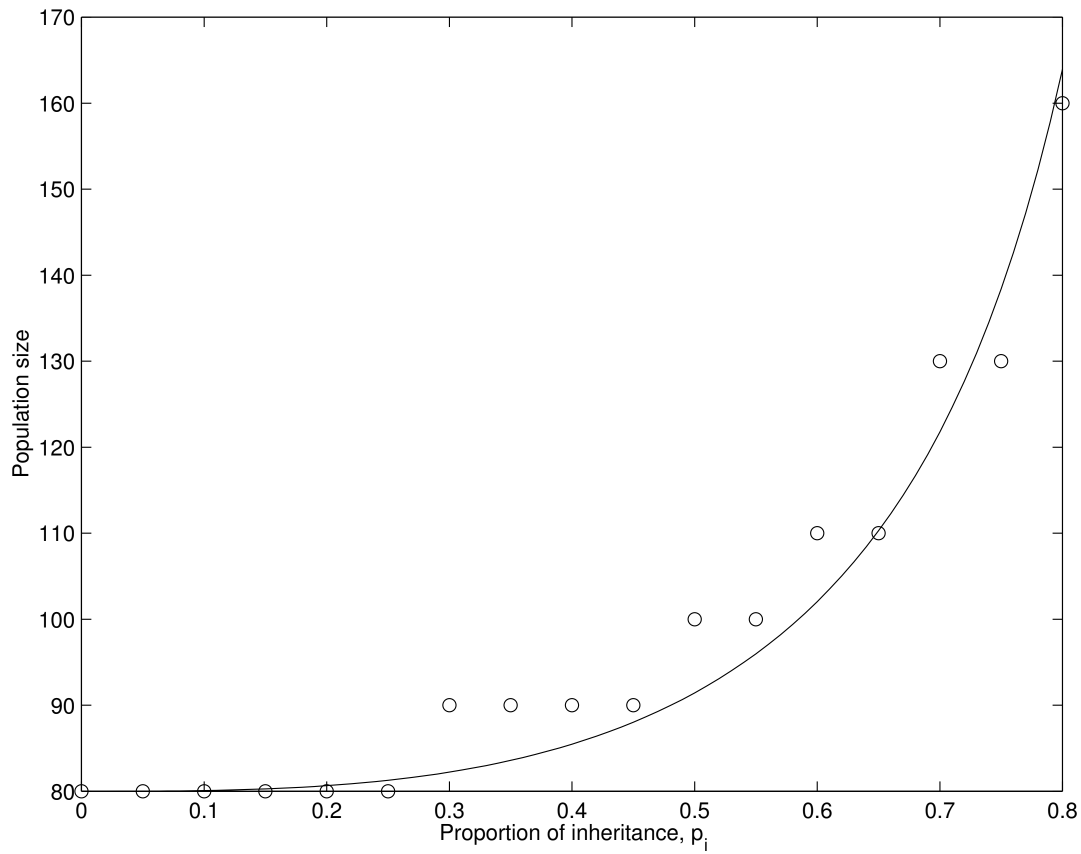


Population-Sizing Model

- Gambler's ruin population-sizing model (Harik, Cantú-Paz, Goldberg & Miller,1997)
  - Combines BB supply & decision making model
- Population sizing for noisy environments (Miller,1997)

$$n = \frac{-2^{k-1}\log\alpha\sqrt{\pi\sigma_{f,t}^2}}{(1-p_i^3)}$$

Population Size



Optimal Inheritance Proportion

- $t_{\text{conv}} \propto (1 - p_i)^{-\frac{1}{2}}$, $n \propto (1 - p_i^3)^{-1}$
- Low p_i does not yield high speed-up
- High p_i
  - Requires large population size
  - Longer convergence time
- Optimal p_i yields greatest speed-up

Optimal Inheritance Proportion

- Neglect inheritance cost
- Minimize number of function evaluations

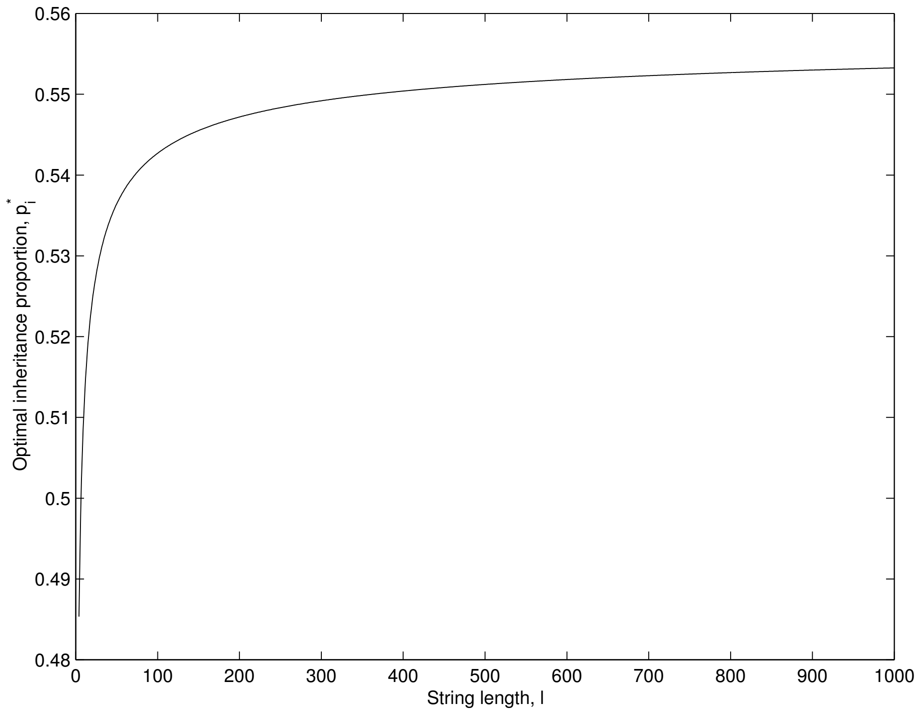
$$N_{fe} = n\left[t_{\text{conv}}(1-p_i) + p_i\right]$$

- Solve $\frac{\partial N_{fe}}{\partial p_i} = 0$

$$3p_i^2\left[\frac{\pi\sqrt{\ell}}{2I}(1-p_i) + p_i\sqrt{1-p_i}\right] + (1-p_i^3)\left[\sqrt{1-p_i} - \frac{\pi\sqrt{\ell}}{4I}\right] = 0$$

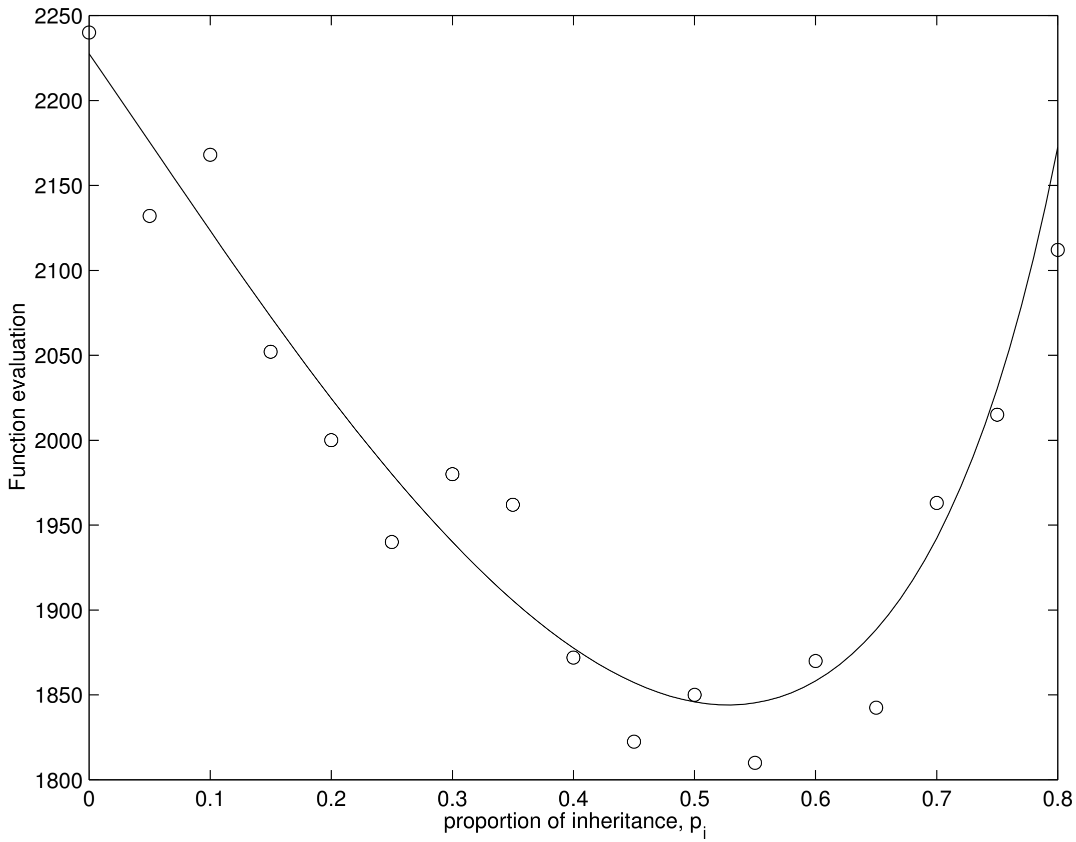
- Solve numerically or Use approximations

Optimal Inheritance Proportion



- Optimal Inheritance: $0.54 \leq p_i^* \leq 0.558$

Results: 100-Bit OneMax



Optimal Speed-Up

- $N_{fe}(p_i^*)$: No. of func. evals. with optimal inheritance
- $N_{fe}(p_i = 0)$: No. of func. evals without inheritance
- Speed-Up, $\eta_s = N_{fe}(p_i = 0)/N_{fe}(p_i^*)$
- Greatest speed-up: $1.2422 \leq \eta_s \leq 1.2428$

Result Comparison

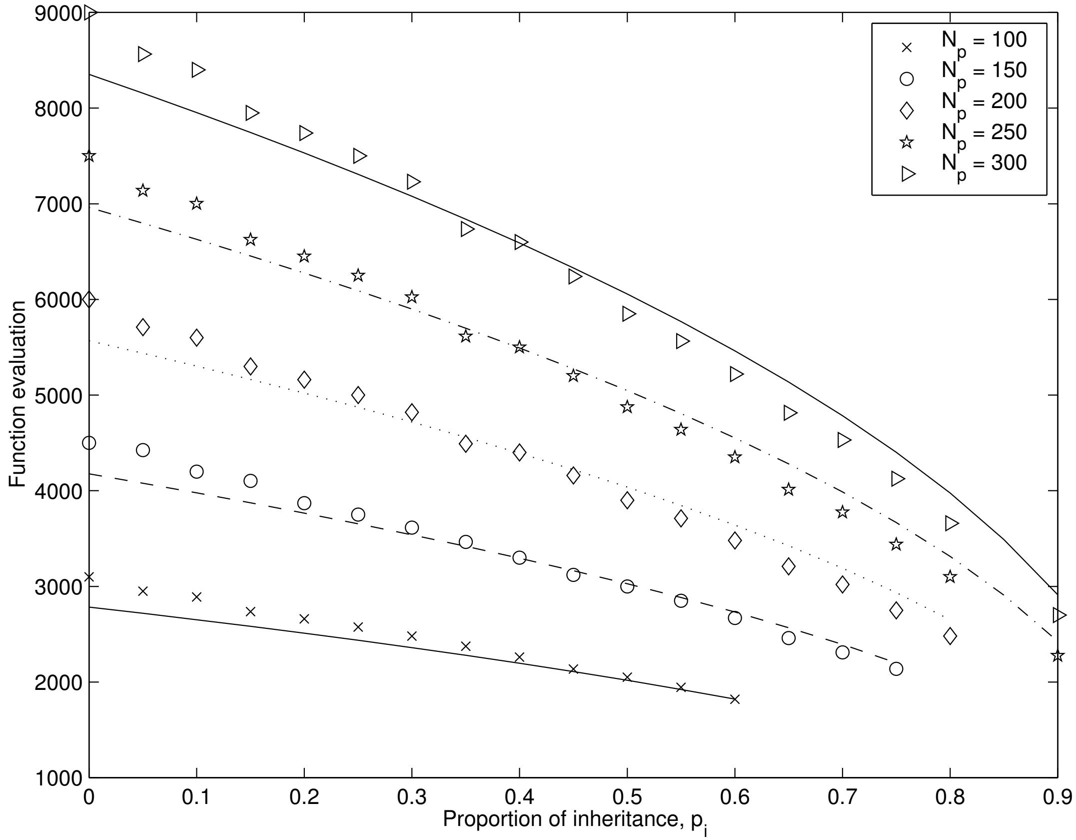
- Smith, Dike & Stegmann (1995) reported higher speed-up
- Our Study: $\eta_s \approx 1.24$
- Is something wrong?
- Fixed population size assumption
- Adjust population size for constant solution quality

Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801. USA.
http://www-illigal.ge.uiuc.edu

Apparent Speed-Up

- Assume fixed population size
  - Irrespective of p_i value
- Speed-up obtained is *apparent* speed-up
- Results agree with Smith, Dike & Stegmann (1995)

Apparent Speed-Up

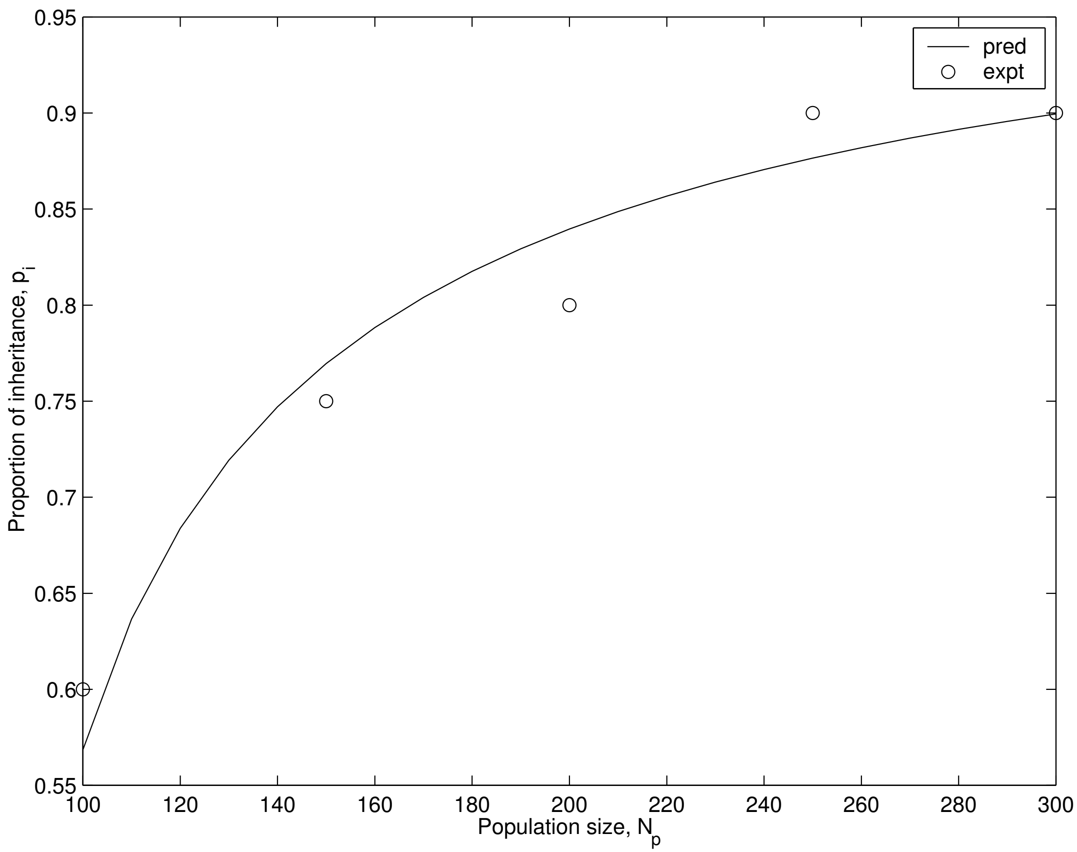


Optimal Apparent Speed-Up

$$p^*_{i,\text{app}} = \left(1 - \frac{\kappa}{n}\right)^{\frac{1}{3}}$$

- κ a constant, depends on
  - Building block size
  - Failure probability
- $n < \kappa$: Premature convergence
- $n > \kappa$: Higher speed-up

Optimal Apparent Speed-Up



Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801. USA.
http://www-illigal.ge.uiuc.edu

Summary

- Modeled fitness inheritance
  - Convergence time model
  - Semi-empirical population-sizing model
- Optimal inheritance for greatest speed-up
- Fitness inheritance: 20% reduction
- A loose upper bound
- Fixed population size—higher speed-up

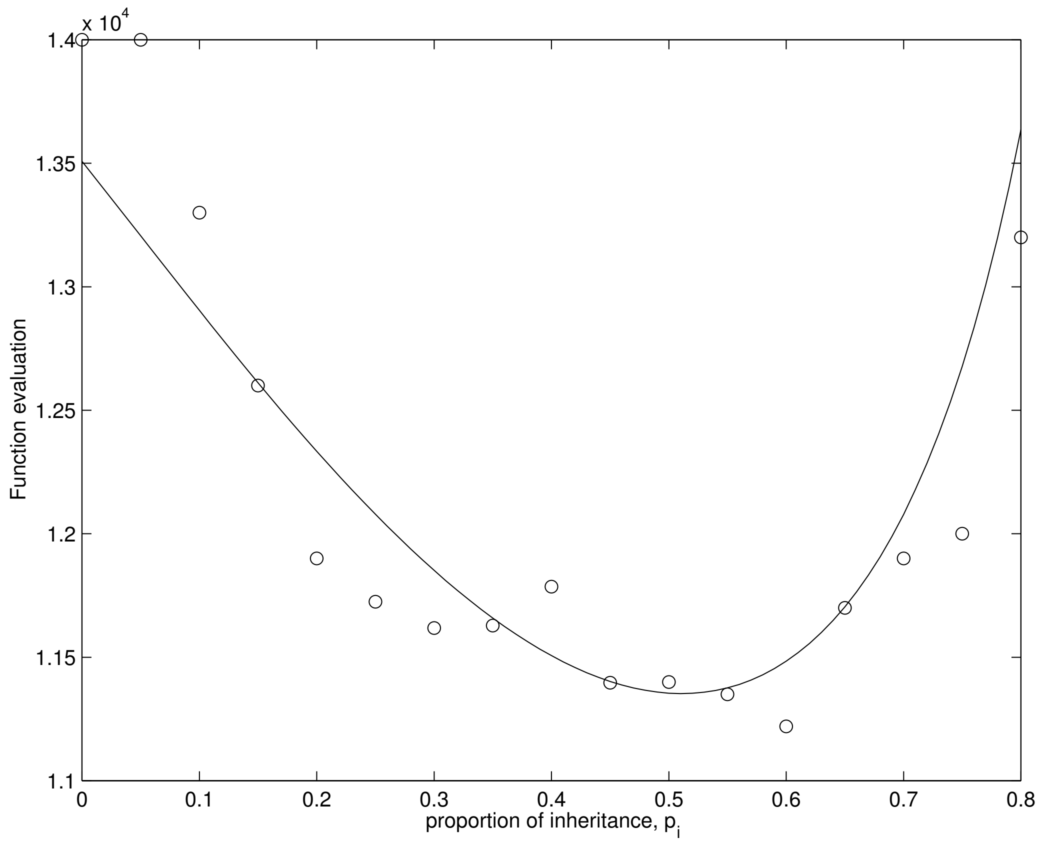
Future Work

- Extend analysis to other problem domain
- Analytical population-sizing model
- Model other inheritance techniques
- Apply to complex, real-world problems

Acknowledgments

- Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-1-0163.
- National Science Foundation under grant DMI-9908252.
- U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0003.

Results: 40-Bit Trap, $p_c = 0.9$



Results: 40-Bit Trap, $s = 8$