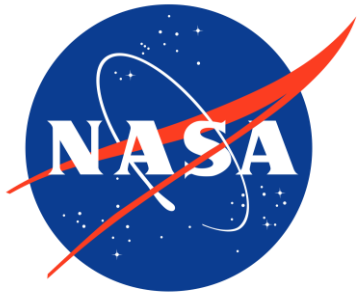


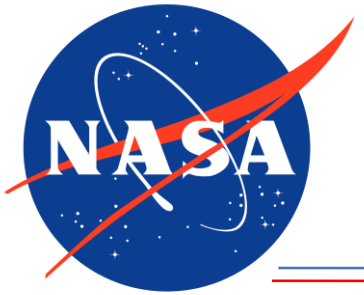
Benchmarks in Bingo

Diana Vera, University of Illinois at Chicago

Mentor – Geoff Bomarito

NASA Langley Research Center

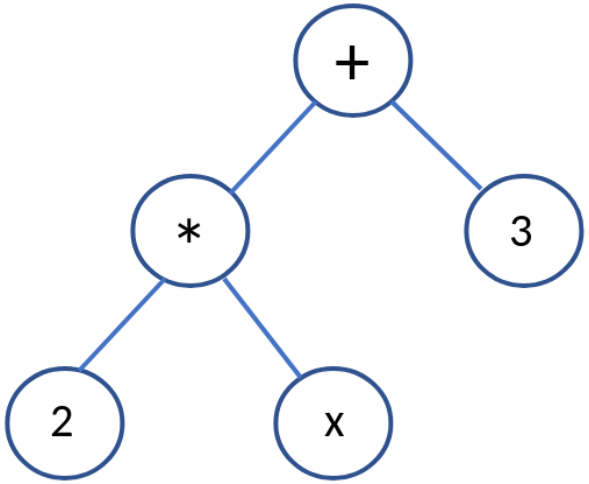
Summer 2019



Contents

Terms to Know

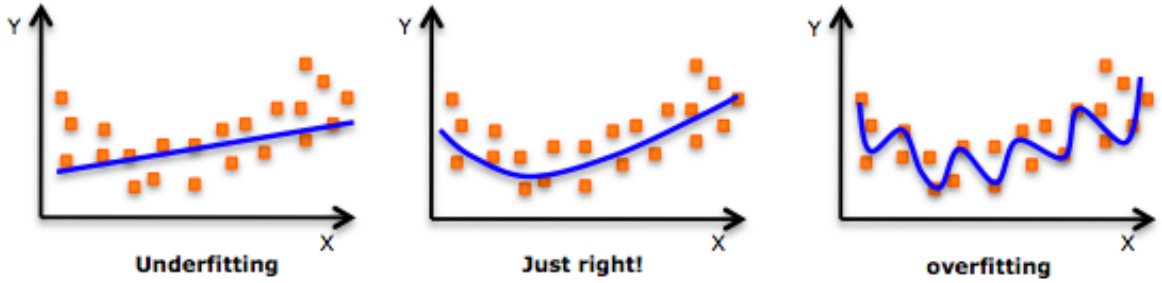
- AGraph – Acyclic Graphs that represent equations in Bingo
- Genetic Programming (GP) – A method of breeding a population of individuals to find a solution to a problem (in this case, individuals are AGraphs)
- Benchmark – Tests the performance of an operation. Useful for comparing relative performance of different models



$= 2x + 3$

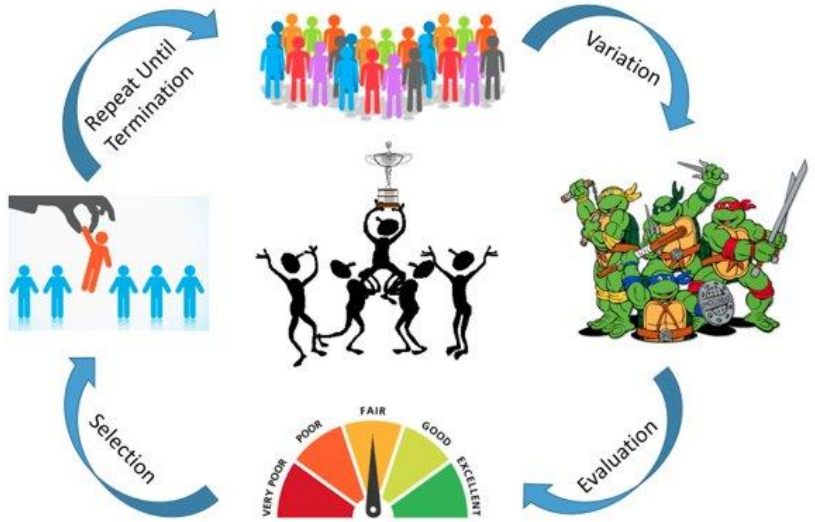
Background: Symbolic Regression

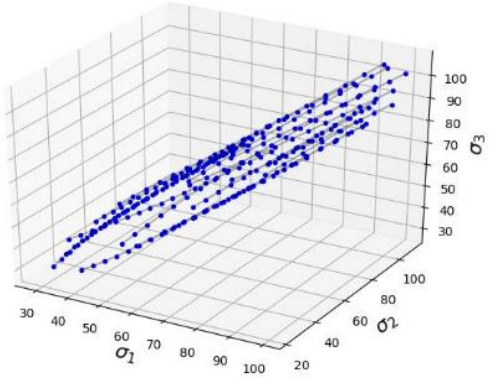
- Symbolic Regression
  - Finding a model to best fit a given dataset
  - "Best fit" = Accuracy + Simplicity
  - Challenges: Overfitting can yield a solution that accurately describes a particular set of data but is too specific to reliably account for new data
    - "Memorizing" rather than "learning"
  - Optimal solutions find the trend in a dataset but are general enough to apply to new data



Background: Bingo

- Bingo
  - Open source package for performing symbolic regression
  - Can be used as a general-purpose evolutionary optimization package
  - NASA application to help learn and understand material models

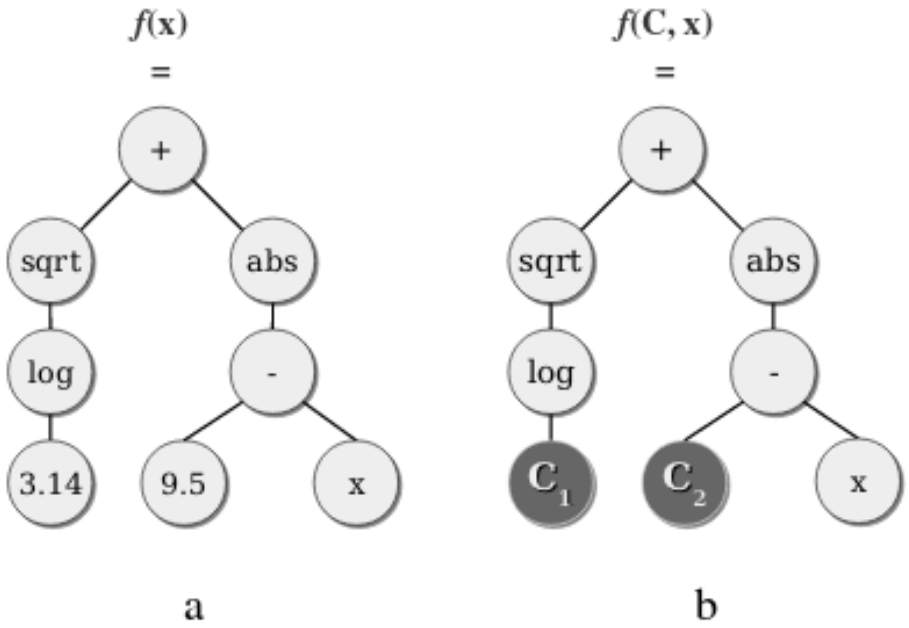




$$\Phi = f(\sigma)$$

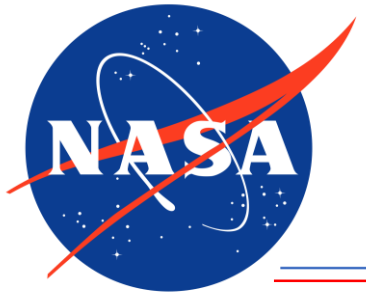
Background: Local Optimization

- Individuals are marked to indicate whether they need to undergo optimization.
- A predefined fitness function is used alongside an optimization algorithm that optimizes numeric constants in an individual using least-squares minimization
- We used the Levenberg–Marquardt algorithm but there are many to choose from
- The default is Nelder-Mead



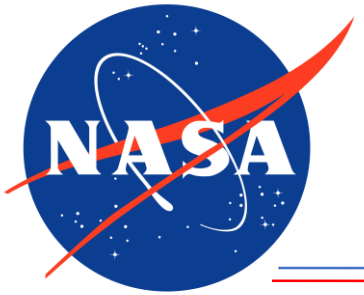
Objective

- Create a benchmark suite for Bingo
  - Benchmarks will provide a metric to compare performance
  - We can measure the benefit of local optimization
  - Can also measure the differences in performance of different evolutionary algorithms



Benchmarks in Bingo

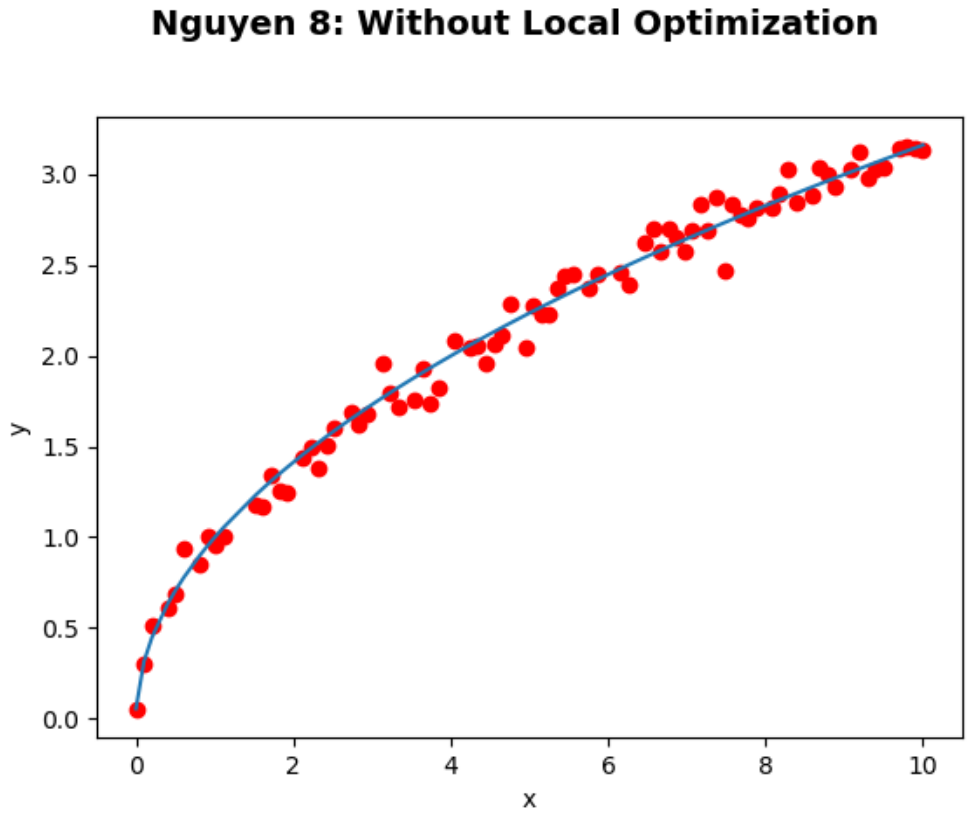
- Equations pulled from popular symbolic regression benchmarks
- Benchmark Suite
  - By default, includes 9 Benchmarks
  - User can specify which to include or exclude
- Benchmark
  - Name of benchmark
  - Target function
  - Training set
  - Testing set

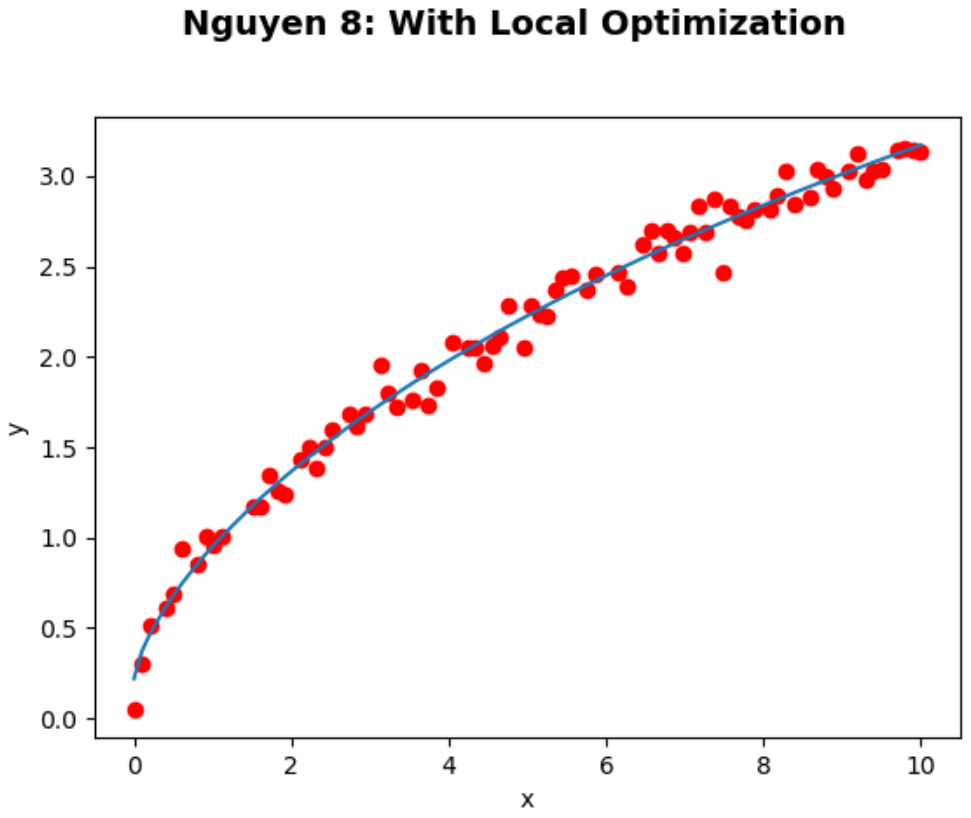


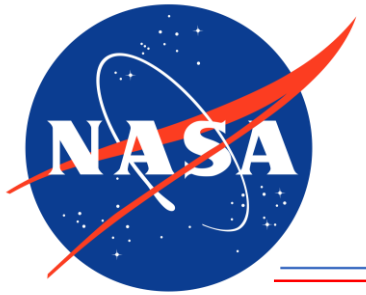
Benchmarks in Bingo

Name	Function
Koza-1	$x^4 + x^3 + x^2 + x$
Koza-2	$x^5 - 2x^3 + x$
Koza-3	$x^6 - 2x^4 + x^2$
Nguyen-1	$x^3 + x^2 + x$
Nguyen-3	$x^5 + x^4 + x^3 + x^2 + x$
Nguyen-4	$x^6 + x^5 + x^4 + x^3 + x^2 + x$
Nguyen-5	$\sin(x^2)\cos(x) - 1$
Nguyen-6	$\sin(x) + \sin(x + x^2)$
Nguyen-8	$\sqrt{x}$

Benchmarks in Bingo: Nguyen-8







Benchmarks in Bingo

Without local opt:

WITHOUT CONTINUOUS LOCAL OPTIMIZATION

:::: REGRESSION BENCHMARKS ::::

NAME	MEAN +- STD	MIN	MAX
koza_1	62.1947 +- 0.1485	62.0720	62.4037
koza_2	61.5925 +- 0.0322	61.5668	61.6380
koza_3	62.3496 +- 0.3052	61.9761	62.7237
nguyen_1	61.8624 +- 0.1673	61.6549	62.0646
nguyen_3	62.2615 +- 0.3234	61.9342	62.7018
nguyen_4	62.3746 +- 0.2445	62.1972	62.7203
nguyen_5	62.7328 +- 0.2187	62.5148	63.0318
nguyen_6	62.3097 +- 0.1309	62.1679	62.4836
nguyen_8	62.4529 +- 0.5627	61.6747	62.9863

With local opt:

WITH CONTINUOUS LOCAL OPTIMIZATION

:::: REGRESSION BENCHMARKS ::::

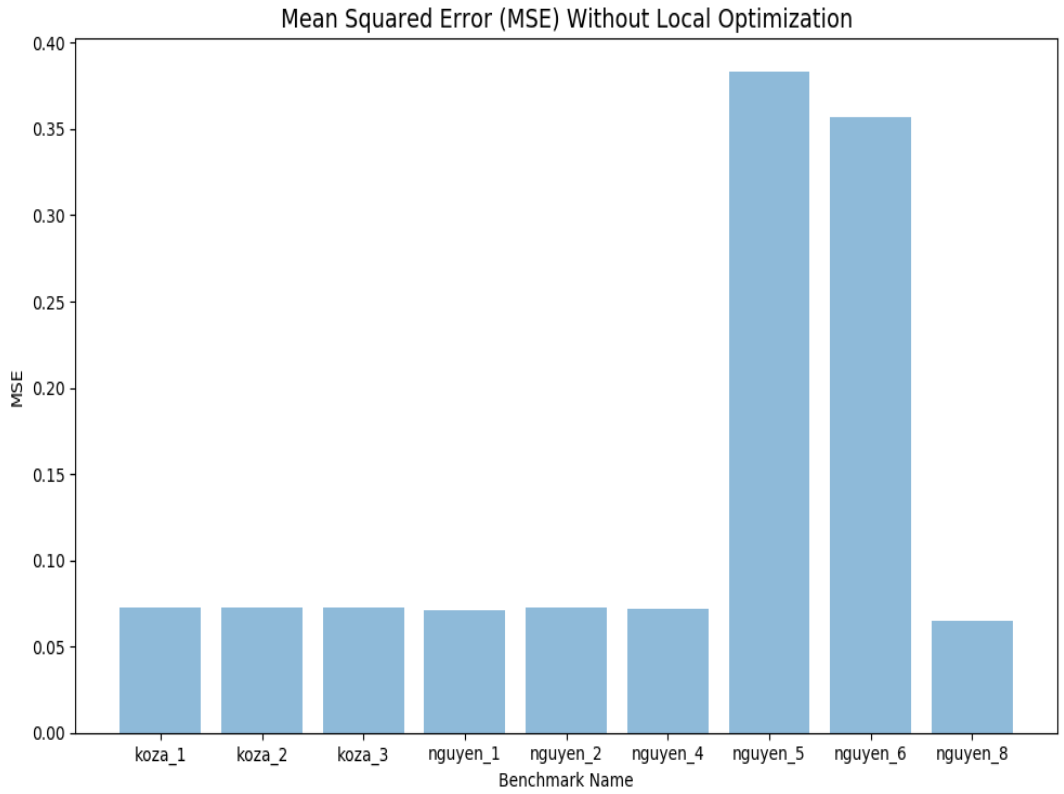
NAME	MEAN +- STD	MIN	MAX
koza_1	49.3313 +- 0.9735	48.2622	50.6171
koza_2	49.7498 +- 0.9792	48.3650	50.4461
koza_3	46.2997 +- 0.1841	46.0519	46.4928
nguyen_1	48.3795 +- 0.6077	47.6669	49.1518
nguyen_3	48.2649 +- 0.0463	48.2128	48.3253
nguyen_4	46.7042 +- 0.4394	46.1929	47.2658
nguyen_5	51.0687 +- 0.1565	50.8588	51.2343
nguyen_6	47.0395 +- 0.7966	46.0917	48.0407
nguyen_8	50.3101 +- 0.7088	49.3240	50.9589

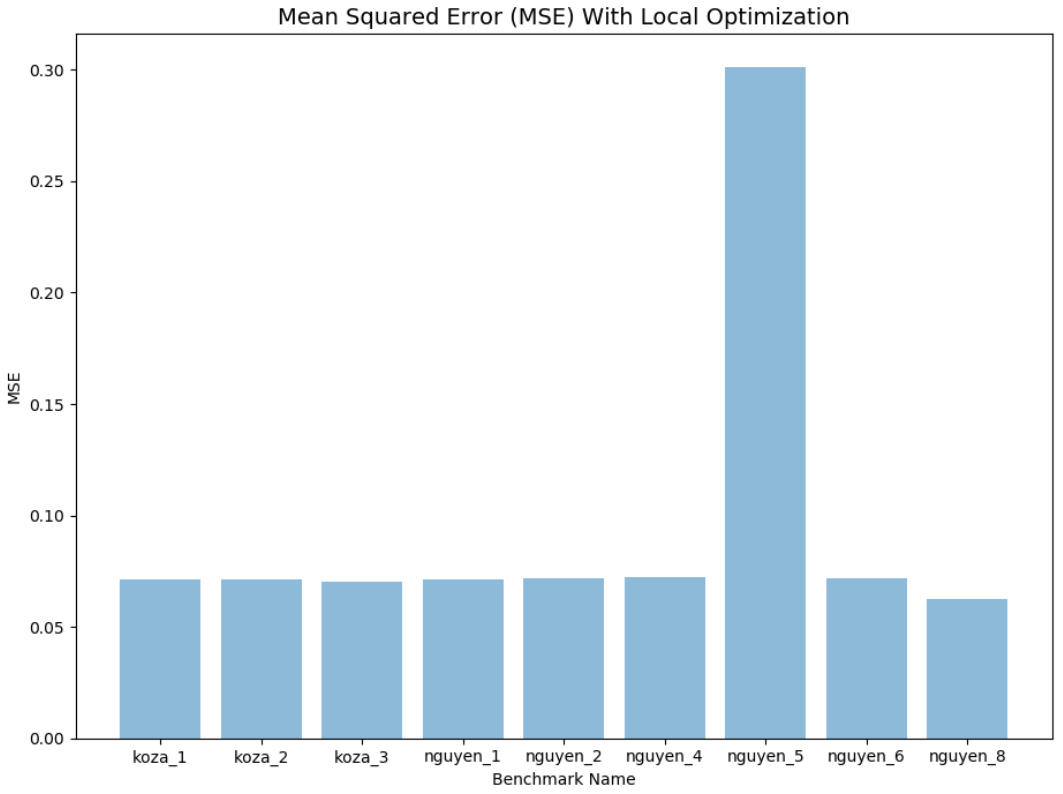
Benchmarks in Bingo

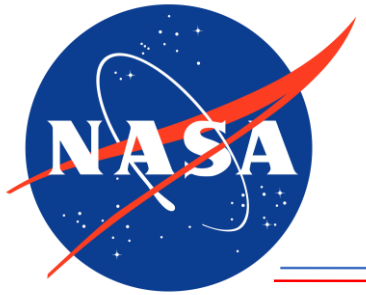
Name	% improvement in performance
Koza-1	20.68%
Koza-2	19.23%
Koza-3	25.74%
Nguyen-1	21.79%
Nguyen-3	22.48%
Nguyen-4	25.12%
Nguyen-5	18.59%
Nguyen-6	24.51%
Nguyen-8	19.44%

Average improvement in performance: 21.95%

Benchmarks in Bingo: Error

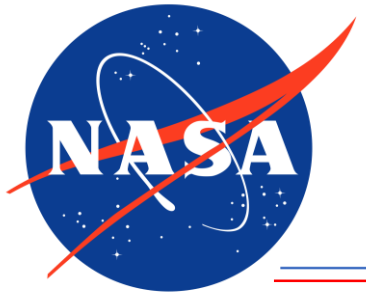






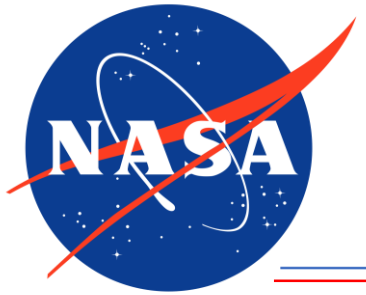
Value Added

- To NASA:
  - Bingo now has Benchmark and BenchmarkSuite classes that allow the user to measure performance on a standard set of equations
  - BenchmarkSuite class allows the user to add custom Benchmarks if desired
  - In the Spring 2019 term, my work included implementation of a few different evolutionary algorithms such as DeterministicCrowding that allow the user to swap out models that best suit their needs
  - The addition of a BenchmarkSuite provides a standardized way for users to assess the performance of those algorithms



Value Added

- To me:
  - Learned to use Test-Driven Development when writing code
    - Extremely useful in a professional environment but not taught in my school curriculum
  - Gained experience with Genetic Programming, not only by contributing to Bingo itself but also by reading about the challenges currently facing the GP field
  - Not related to my work but I got to see an airplane get dropped from the gantry. Coolest summer camp ever.



Conclusion

- Benchmarks add a useful metric of performance to Bingo
- Using these Benchmarks, we were able to show that local optimization helps find solutions more quickly, with comparable accuracy
- The Benchmark Suite can also be used to measure the performance of different evolutionary algorithms to see which one best fits the problem at hand