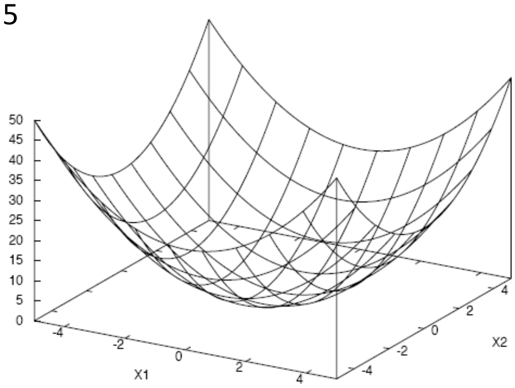
Computational Intelligence

Unit # 4

Assignment # 1 (Function 1)

$f(x, y) = x^2 + y^2$

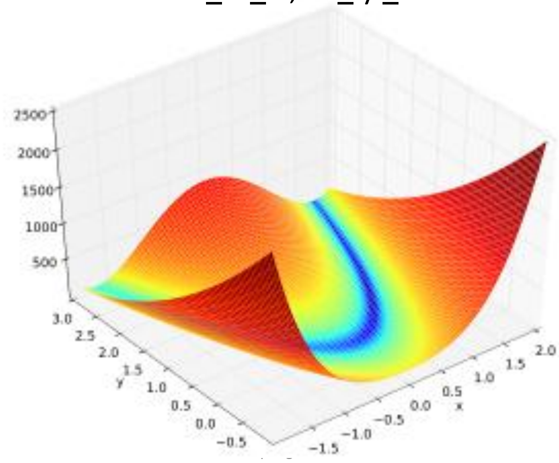
$-5 \leq x, y \leq 5$



Assignment # 1 (Function 2)

$$f(x, y) = 100 * (x^2 - y)^2 + (1 - x)^2$$

$$-2 \le x \le 2, -1 \le y \le 3$$



A Typical Evolutionary Algorithm Cycle

- Step 1: Initialize the population randomly or with potentially good ***solutions***.
- Step 2: Compute the ***fitness*** of each individual in the population.
- Step 3: Select parents using a ***selection procedure***.
- Step 4: Create offspring by ***crossover*** and ***mutation*** operators.
- Step 5: Compute the ***fitness*** of the new offspring.
- Step 6: Select members of population to die using a ***selection procedure***.
- Step 7: Go to Step 2 until termination criteria are met.

Assignment Implementation Details

- Let's pick the population size to 10. So initialize 10 individuals of the form (x, y) randomly.
- Compute their fitness.
- Using one of the parent selection scheme, generate 10 offspring (using crossover and mutation). For mutation, you can use +/- 0.25. Remember that mutation is not applied to each gene. So you need to make it probabilistic as well.
- Compute fitness of the offspring.

Assignment Implementation Details (Cont'd)

- Now you got 10 parents and 10 offspring (20 individuals.
- Using one of the survival selection scheme, pick 10 individuals that survive to the next generation. Discard the remaining individuals.
- Store best survived individual's fitness and average fitness of the whole survived individual.

Assignment Implementation Details (Cont'd)

- Suppose you run this process for 40 generations. You will have recorded the following observations.

Generation #	Best Fitness	Average Fitness
1		
2		
..		
..		
..		
40		

Average Best-So-Far

- You need to repeat this exercise (for a single combination of parent and survival selection schemes) 10 times).
- Each run will start with new randomly initialized population.

Generation #	Run # 1 BSF	Run # 2 BSF	..	..	Run # 10 BSF	Average BSF
1						
2						
..						
..						
..						
40						

Average Average-Fitness

- You need to repeat this exercise (for a single combination of parent and survival selection schemes 10 times) for average fitness values.
- Each run will start with new randomly initialized population.

Generation #	Run # 1 Avg. Fit.	Run # 2 Avg. Fit.	..	..	Run # 10 Avg. Fit.	Average Avg. Fit.
1						
2						
..						
..						
..						
40						

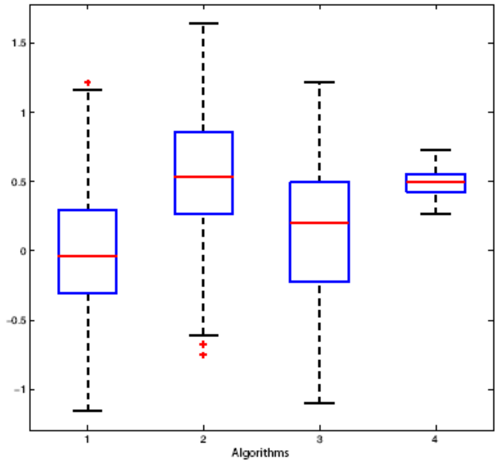
Assignment Implementation Details (Cont'd)

- Once you have gotten (a) average best-so-far values and average average-fitness, you need to plot the values against generation # (separate graphs).
- The scheme described in the previous slides will be repeated for each combination of parent and survival selections, i.e.,
  - FPS and Truncation
  - RBS and Truncation
  - Binary Tournament and Truncation
  - FPS and Binary Tournament
  - RBS and Binary Tournament
  - Binary Tournament and Binary Tournament

Performance Indicator

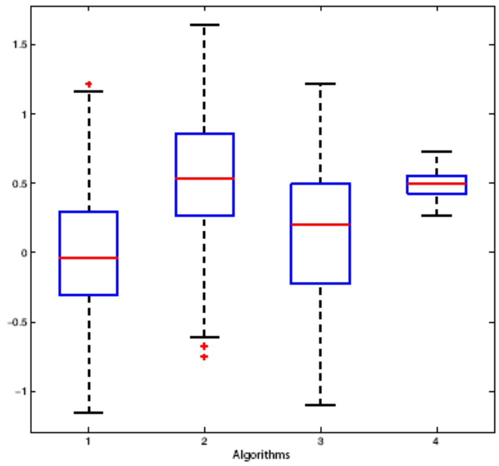
- **Best-so-far (BSF) -** We record the best solution found by the algorithm thus far for each generation in every run.
- **Average-of-current-population (ACP) -** We record the average solution in each generation in every run.
- **Worst-of-current-population (WCP) -** We record the worst solution in each generation in every run.

Performance Graph using Boxplot



Performance Description and Comparison of EA

- *Statistical visualization* uses graphs to describe and compare EAs, which is very illustrative.
- The box plot is the most useful way to graphically illustrate the performance of EAs.



Performance Description and Comparison of EA (Cont'd)

- *Descriptive Statistics* - Graphs are easy to understand, but sometimes the difference between different algorithms is small. Then we need specific numbers to describe and compare the performance.
- The most often used *descriptive statistics are mean and variance (or standard* deviation).

Performance Description and Comparison of EA (Cont'd)

- *Statistical Inference* - Sometimes descriptive statistics is also not strong enough to differentiate between two results, in which case we need *statistical inference.*
- *Statistical inference includes* parameter estimation, hypothesis testing, and many other techniques.
- Here we focus on hypothesis testing to verify whether the difference between two results is statistically significant.

Hypothesis Testing Example

Trial	Old Method	New Method
1	500	657
2	600	543
3	556	654
4	573	565
5	420	654
6	590	712
7	700	456
8	472	564
9	534	675
10	512	643
Average	545.7	612.3

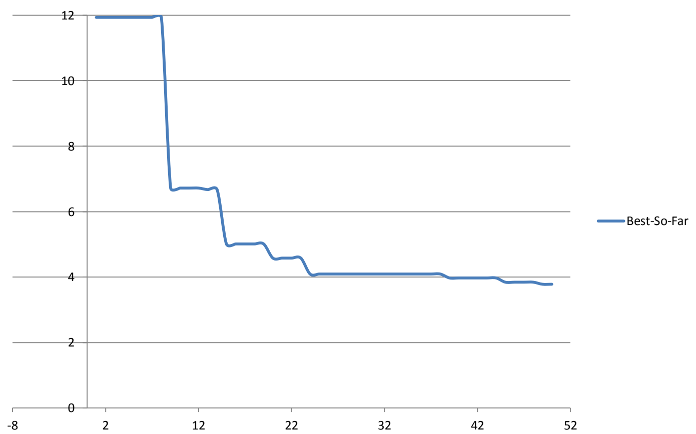
Is the new method better?

Hypothesis Testing Example (Cont'd)

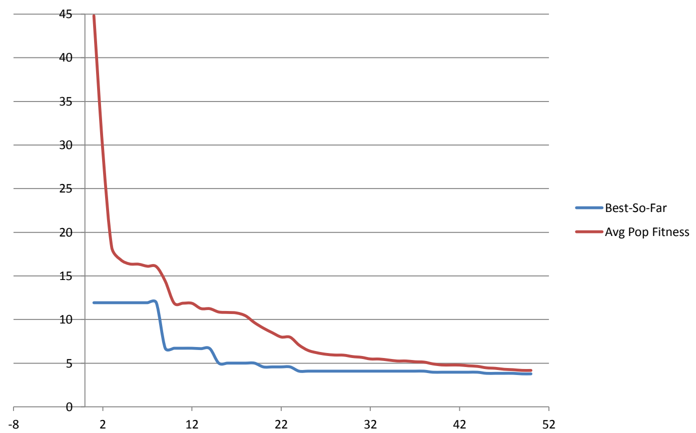
Trial	Old Method	New Method
1	500	657
2	600	543
3	556	654
4	573	565
5	420	654
6	590	712
7	700	456
8	472	564
9	534	675
10	512	643
Average	545.7	612.3
SD	73.5962635	73.5473317
T-test	**0.07080798**	

- Standard deviations supply additional info
- T-test (and alike) indicate the chance that the values came from the same underlying distribution (difference is due to random effects)

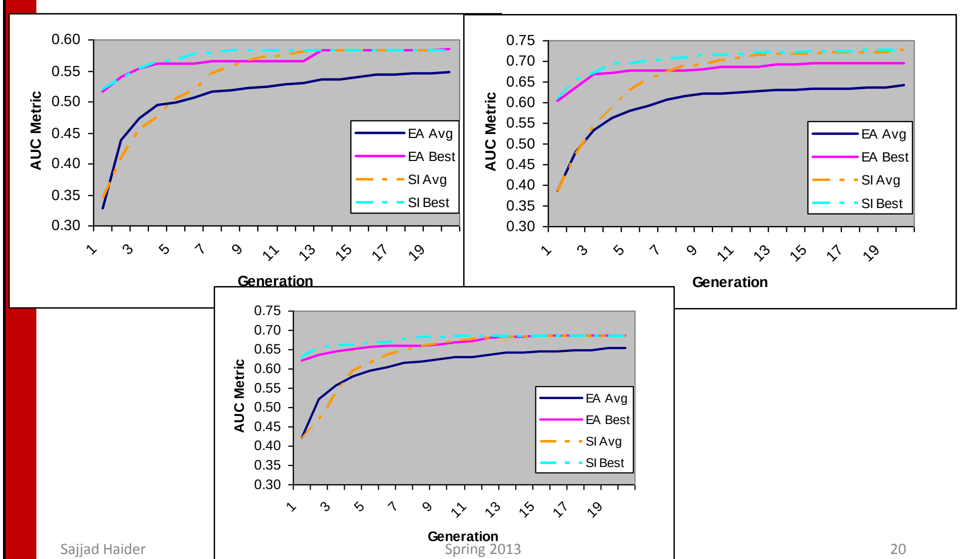
Evolution of Bipedal Walk for RoboCup



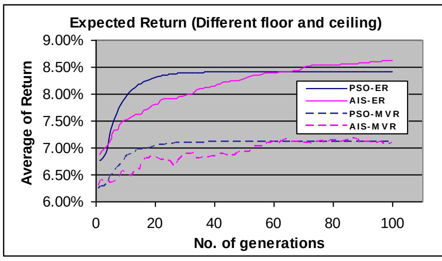
Evolution of Bipedal Walk for RoboCup (Cont'd)

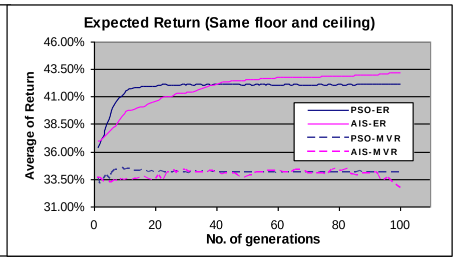


PSO vs. EA in Strategy Optimization

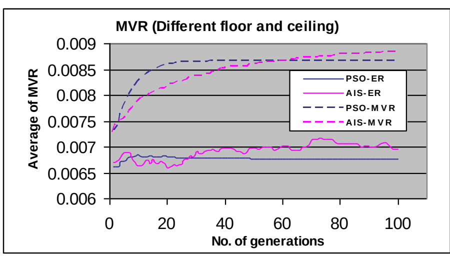


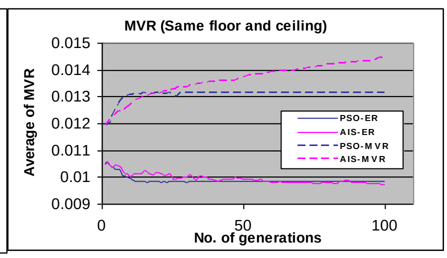
PSO vs. AIS in Portfolio Optimization



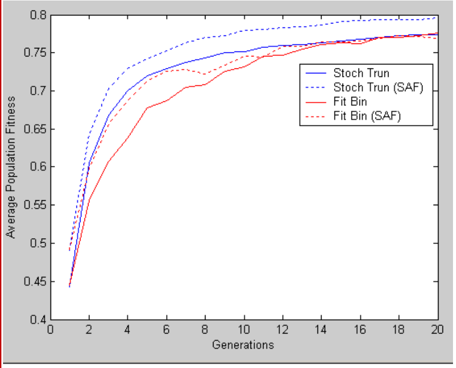


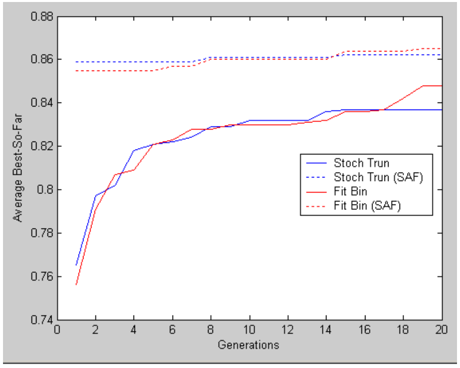
PSO vs. AIS in Portfolio Optimization (Cont'd)





Random vs. Heuristic based Initialization





Random vs. Heuristic based Initialization (Cont'd)