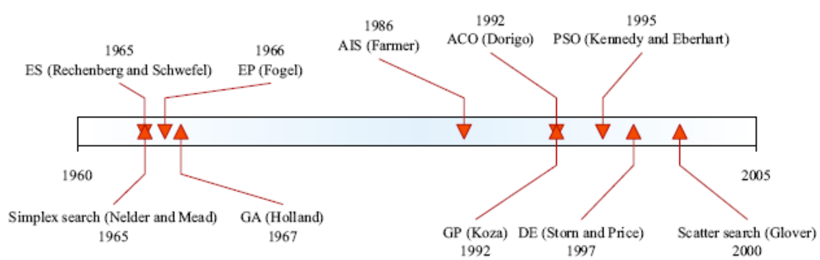
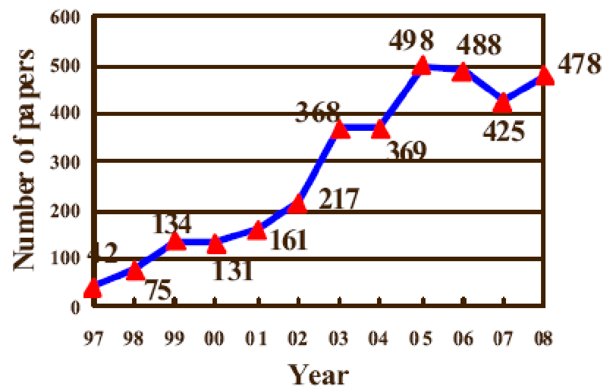
Computational Intelligence

Unit # 2

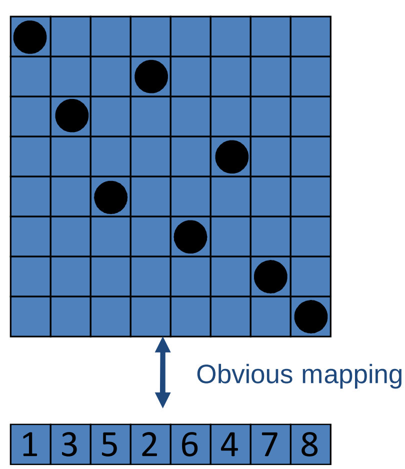
Recap: History of Evolutionary Computation



Number of EC Papers Published in ISI Indexed Journals



The 8-Queen Problem: Representation



The 8-Queen Problem: Fitness Evaluation

- Penalty of one queen:
  the number of queens she can check.
- Penalty of a configuration:
  the sum of the penalties of all queens.
- Note: penalty is to be minimized
- Fitness of a configuration:
  inverse penalty to be maximized

The 8-Queen Problem: Mutation

- Small variation in one permutation, e.g.:
  - swapping values of two randomly chosen positions

1	3	5	2	6	4	7	8

→

1	3	7	2	6	4	5	8

The 8-Queens Problem: Crossover

- Combining two permutations into two new permutations:
  - choose random crossover point
  - copy first parts into children
  - create second part by inserting values from other parent:
    - in the order they appear there
    - beginning after crossover point
    - skipping values already in child

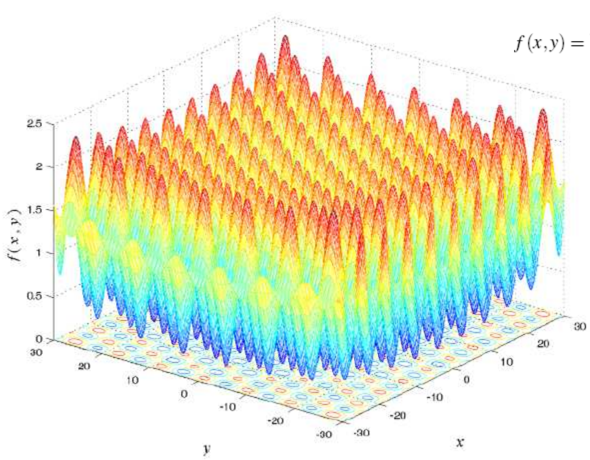
1	3	5	2	6	4	7	8
8	7	6	5	4	3	2	1

→

1	3	5	4	2	8	7	6
8	7	6	2	4	1	3	5

Function for HW # 1

$$f(x,y) = \frac{x^2+y^2}{4000} - \cos(x)\cos\left(\frac{y}{\sqrt{2}}\right) + 1$$



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.

General Pattern

- A population of constant size μ is evolved over time.
- The current population is used as a source of parents to produce λ offspring.
- The expanded population is reduced from $\mu+\lambda$ to μ individuals.

Selection

- Selection is one of the main operators in EAs, and relates directly to the Darwinian concept of survival of the fittest.
  - **Selection of the new population:** A new population of candidate solutions is selected at the end of each generation to serve as the population of the next generation. The selection operator should ensure that good individuals do survive to next generations.
  - **Reproduction:** Offspring are created through the application of crossover and/or mutation operators.

Selective Pressure

- Selection operators are characterized by their selective pressure, also referred to as the takeover time, which relates to the time it requires to produce a uniform population.
- It is defined as the speed at which the best solution will occupy the entire population by repeated application of the selection operator alone.
- An operator with a high selective pressure decreases diversity in the population more rapidly than operators with a low selective pressure, which may lead to premature convergence to suboptimal solutions.
- A high selective pressure limits the exploration abilities of the population.

Selection Procedure

- Selection in evolutionary algorithms is the process of choosing which individuals reproduce offspring and which individuals survive to the next generation.
- When selection is used to choose which individuals reproduce, the process is referred to as ***pre-selection*** (parent(s) selection).
- When it is used to select the individuals that survive to the next generation it is called ***post-selection*** (survival selection).

Selection Procedure (Cont'd)

- ***Deterministic selection*** tends to behave more like greedy hill-climbing algorithms and exploits the nearest areas with promising solutions.
- ***Probabilistic selection*** schemes are more exploratory and search the landscape.
- Schemes based on exploration are said to have a low selection pressure, while schemes based on exploitation are said to have greater selection pressure.
- In other words, selection pressure is a vague measure of how often more fit individuals are selected to reproduce and/or live to the next generation.

Selection Procedure (Cont'd)

- Selection schemes can be further categorized into generational or steady-state schemes.
- A selection scheme is ***generational*** when the entire current population is replaced by its offspring to create the next generation
- A scheme is referred to as ***steady-state*** when a selected few offspring replace a few members of the current generation to form the next generation.

Selection Schemes

- Fitness Proportional
- Rank Selection
- Tournament Selection
- Truncation
- Elitist
- Uniform Stochastic

Parent Selection Mechanism

- Assigns variable probabilities of individuals acting as parents depending on their fitness
- Usually probabilistic
  - high quality solutions more likely to become parents than low quality
  - but not guaranteed
  - even worst in current population usually has non-zero probability of becoming a parent
- This *stochastic* nature can aid escape from local optima

Fitness Proportional Selection

- Proportional selection, proposed by Holland, biases selection towards the most fit individuals.
- A probability distribution proportional to the fitness is created, and individuals are selected by sampling the distribution.
- Because selection is directly proportional to fitness, it is possible that strong individuals may dominate in producing offspring, thereby limiting the diversity of the new population. This is known as **premature convergence**.
- In other words, proportional selection has a high selective pressure.

Fitness Proportional Selection (Cont'd)

- When fitness values are all very close together, there is almost no selection pressure. Therefore, later in a run, when some convergence has taken place and the worst individuals are gone, the performance only increases very slowly.
- Also known as Roulette Wheel Selection.

Rank Based Selection

- Attempts to remove problems of FPS by basing selection probabilities on *relative* rather than *absolute* fitness
- Rank population according to fitness and then base selection probabilities on rank where fittest has rank μ and worst rank 1
- This imposes a sorting overhead on the algorithm, but this is usually negligible compared to the fitness evaluation time.

Rank Based Selection (Cont'd)

- Selection is independent of actual fitness values, with the advantage that the best individual will not dominate in the selection process.
- It preserves a constant selection pressure by sorting the population on the basis of fitness, and then allocating selection probabilities to individuals according to their rank, rather than according to their actual fitness values.

Tournament Selection

- FP and R selection methods and the algorithms used to sample from their probability distribution relied on a knowledge of the entire population.
- In certain situations, if the population is very large, or if the population is distributed in some way (perhaps on a parallel system), obtaining this knowledge is either highly time consuming or at worst impossible.
- In other cases, there might not be a universal fitness definition at all. For instance, think of an application evolving game playing strategies. In this case we might not be able to quantify the strength of a given strategy but we can compare any two of them by simulating a game played by these strategies as opponents.

Tournament Selection (Cont'd)

- Informal Procedure:
  - Pick n_{ts} members at random then select the best of these
  - Repeat to select more individuals
- Inherits the advantage of rank selection
- Does not require global reordering
- For crossover with two parents, tournament selection is done twice, once for the selection of each parent.

Tournament Selection (Cont'd)

- Provided that the tournament size, n_{ts}, is not too large, tournament selection prevents the best individual from dominating, thus having a lower selection pressure.
- On the other hand, if n_{ts} is too small, the chances that bad individuals are selected increase. Thus, the the selective pressure is directly related to n_{ts}.
- *If $n_{ts} = n_s$, the best individual will always* be selected, resulting in a very high selective pressure.
- On the other hand, if $n_{ts} = 1$, random selection is obtained.

Tournament Selection (Cont'd)

- There also exists a non-deterministic variant of this selection where this is not necessarily the case. Therefore, a probability p is defined. The best individual in the tournament is selected with probability p, the second best with probability p(1 − p), the third best with probability p(1 − p)2 and so on.
- Tournament selection is perhaps the most widely used selection operators in the modern applications of EAs, due to its extreme simplicity and the fact that the selection pressure is easy to control by varying the tournament size.

Random Selection

- Random selection is the simplest selection operator, where each individual has the same probability *to be selected.*
- *No fitness* information is used, which means that the best and the worst individuals have exactly the same probability of surviving to the next generation.
- Random selection has the lowest selective pressure.

Random Selection

- Random selection returns elements by chance. A possible preceding fitness assignment process as well as the objective values of the individuals play no role at all.
- This hinders the optimization algorithm to follow any gradient in the fitness landscape – it is effectively turned into a random walk.
- Random selection is thus not applied exclusively, but can serve as mating selection scheme in conjunction with a separate environmental selection.
- It maximally preserves the diversity and can be a good choice if used to pick elements from an optimal set.

Survivor Selection

- a.k.a. ***replacement***
- Most EAs use fixed population size so need a way of going from (parents + offspring) to next generation
- Often deterministic
  - Fitness based : e.g., rank parents+offspring and take best
  - Age based: make as many offspring as parents and delete all parents
- Sometimes do combination (elitism)

Age-Based and Fitness-Based Replacement

- The basis of these schemes is that the fitness of individuals is not taken into account during the selection of which individuals to replace in the population, rather they are designed so that each individual exists in the population for the same number of EA iterations.
- A wide number of strategies have been proposed for choosing which μ of the $\mu + \lambda$ parents and offspring should go forward to the next EA iteration.

Replace Worst/Truncation

- In this scheme the worst lambda members of the population are selected for replacement.
- Although this can lead to very rapid improvements in the mean population fitness, it can also lead to premature convergence as the population tends to rapidly focus on the fittest member currently present.

Elitism

- Elitism refers to the process of ensuring that the best individuals of the current population survive to the next generation.
- The best individuals are copied to the new population without being mutated.
- The more individuals that survive to the next generation, the less the diversity of the new population.

Elitist Selection

- In elitism at least one copy of the best individual in the population is always passed onto the next generation
  - The main advantage is that convergence is guaranteed (i.e., if the global maximum is discovered, the EA converges to the maximum).
  - By the same token, however, there is a risk of being trapped in a local maximum.

Hall of Fame

- The hall of fame is a selection scheme similar to the list of best players of an arcade game.
- For each generation, the best individual is selected to be inserted into the hall of fame.
- The hall of fame will therefore contain an archive of the best individuals found from the first generation.
- The hall of fame can be used as a parent pool for the crossover operator, or, at the last generation, the best individual is selected as the best one in the hall of fame.

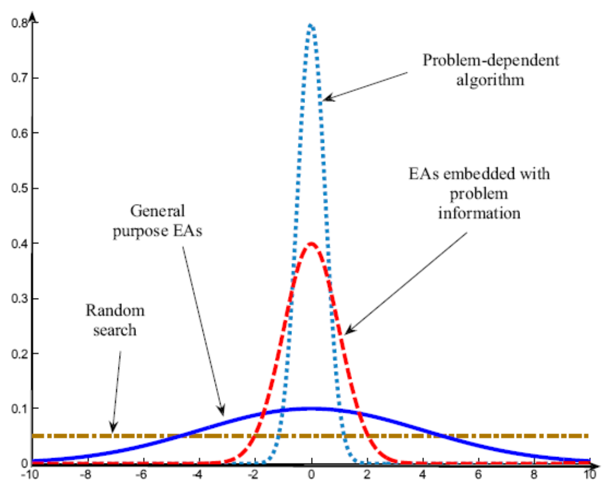
Elitism

- This scheme is commonly used in conjunction with age-based and stochastic fitness-based replacement schemes, in an attempt to prevent the loss of the current fittest member of the population.
- In essence a trace is kept of the current fittest member, and it is always kept in the population.
- The main advantage is that convergence is guaranteed (i.e., if the global maximum is discovered, the EA converges to the maximum). By the same token, however, there is a risk of being trapped in a local maximum.

No Free Lunch Theorem

- Wolpert and Macready published a paper with a very strong title: "No Free Lunch Theorems for Optimization". The key contents of the paper can be quoted as follows:
  - For both static and time dependent optimization problems, the average performance of any pair of algorithms across all possible problems is identical.

One Possible Interpretation of NFL Theorem



NFL Theorem (Cont'd)

- The more we understand the problem, the more specific technique we could design for solving it, and the better performance it will have, but the less robust it will be for other problems.
- We need to demonstrate that our algorithms are better than random search on the problem we face.
- General purpose EAs are reliable methods when you are doing a blind or near blind search in most cases.
- If problem information could be embedded into the encoding and decoding process and into operators, together with a problem-dependent local search method, the performance of the algorithm would be improved at the expense of lower adaptability for other problems.

NFL Theorem (Cont'd)

- To sum up, the No Free Lunch theorem is the sword of Damocles when you want to prove that your algorithm is an ideal one.

Performance Indicator: Efficacy

- We want to evaluate the quality of the results the algorithm provides and do not care about the speed.
- The **mean best fitness** (MBF) is defined as the average of the best fitness in the last population over all runs.
- Apart from the best fitness in the last population, the **best fitness values thus far** could be used as a more absolute evaluation for efficacy.

Performance Indicator: Reliability

- We want to know the extent to which the algorithm can provide acceptable results.
- *Success rate (SR) is defined as the percentage of runs terminated* with success.
- We define a successful run as the difference between the best fitness value in the last generation f_{best} *and a predefined value f* under a predefined* threshold ε.
- SR, combined with MBF, could be used in situations with time requirements, so we want to get good results with limited runs.

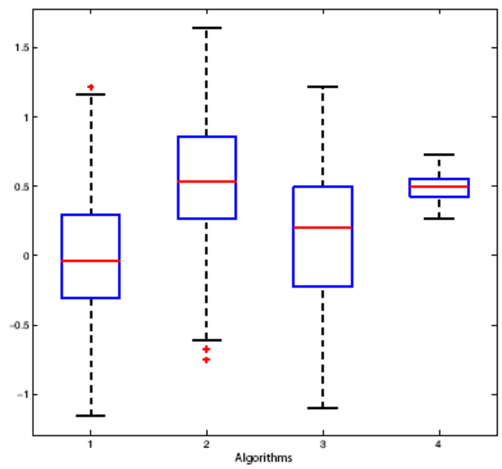
Performance Indicator: Efficiency

- In general, we want to find the global optimal solution as soon as possible.
- So the average number of evaluations to a solution (AES) is defined as the number of evaluations it takes on average for the successful runs to find an optimum,
- If an algorithm has no successful runs, its AES is undefined.

Performance Indicator (Cont'd)

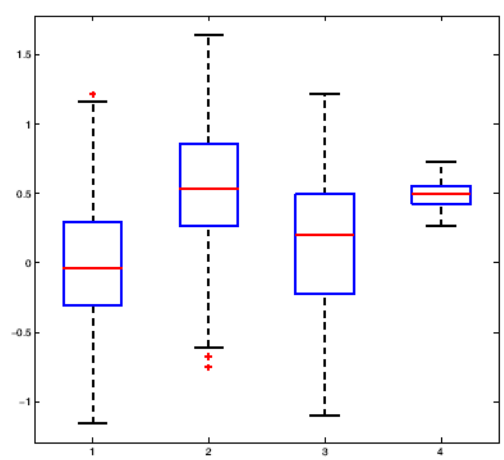
- **Best-so-far (BSF) -** We record the best solution found by the algorithm thus far for each generation in every run.
- **Average-of-current-population (ACP) -** We record the average solution in each generation in every run.
- **Worst-of-current-population (WCP) -** We record the worst solution in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

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?

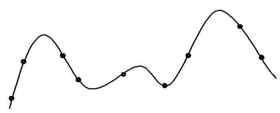
Example (Cont'd)

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**	

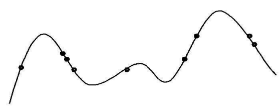
- Standard deviations supply additional info
- T-test (and alike) indicate the chance that the values came from the same underlying distribution (difference is due to random effects)

Typical behaviour of an EA

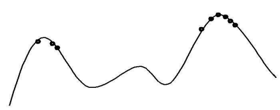
Phases in optimising on a 1-dimensional fitness landscape



Early phase:
quasi-random population distribution

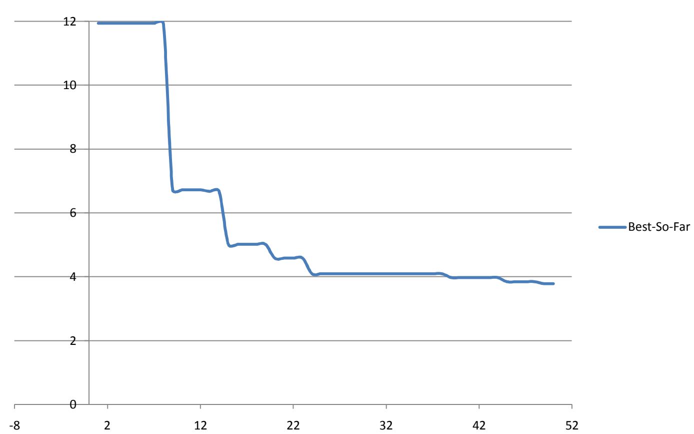


Mid-phase:
population arranged around/on hills

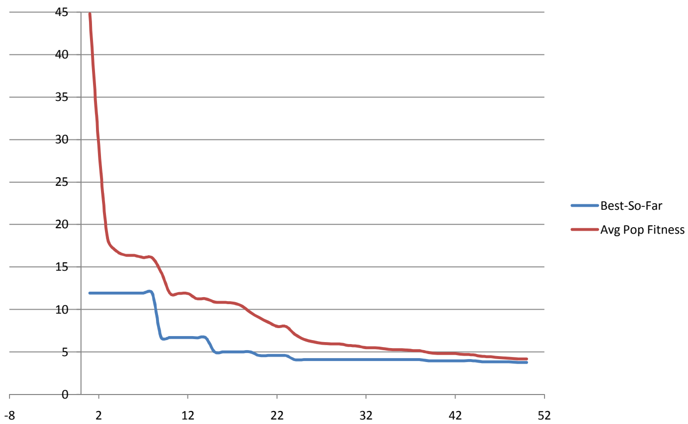


Late phase:
population concentrated on high hills

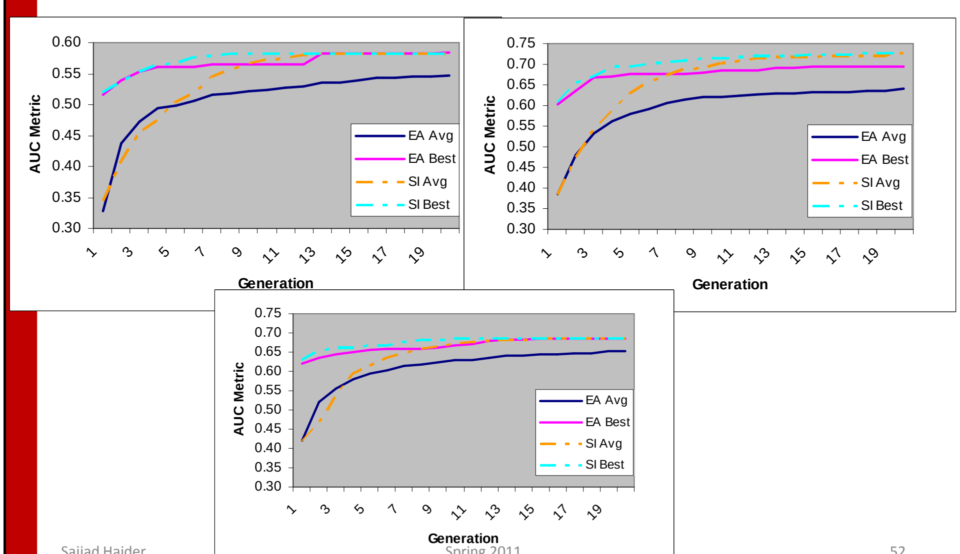
Evolution of Bipedal Walk for RoboCup



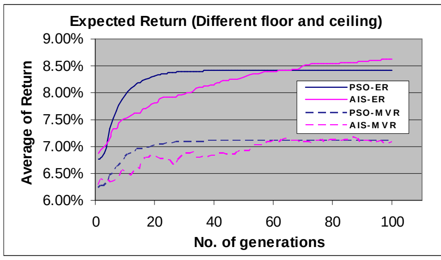
Evolution of Bipedal Walk for RoboCup (Cont'd)

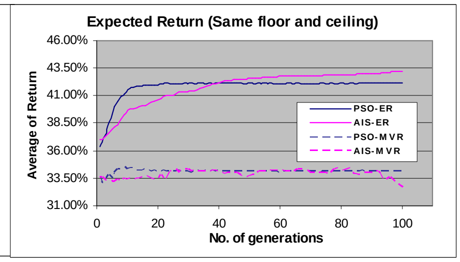


PSO vs. EA in Strategy Optimization

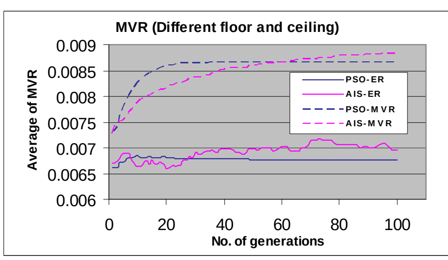


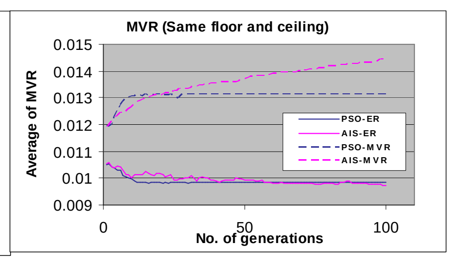
PSO vs. AIS in Portfolio Optimization



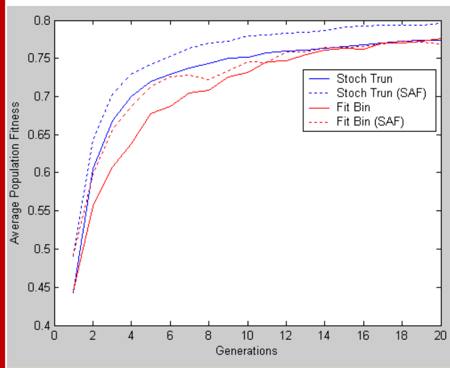


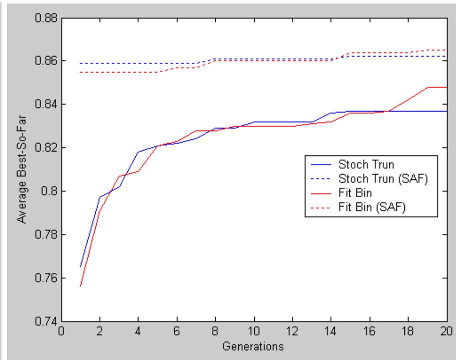
PSO vs. AIS in Portfolio Optimization (Cont'd)





Random vs. Heuristic based Initialization





Random vs. Heuristic based Initialization (Cont'd)