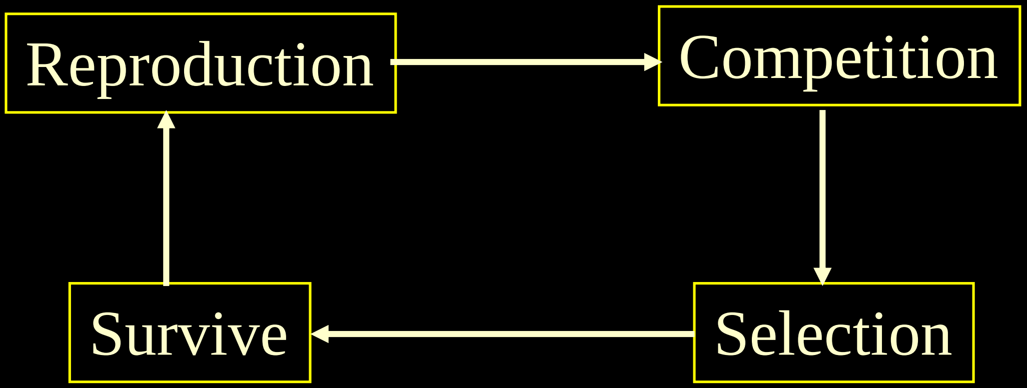
CSCE790 Topics in Information Technology

Department of Computer Science & Engineering
University of South Carolina
Spring, 2002

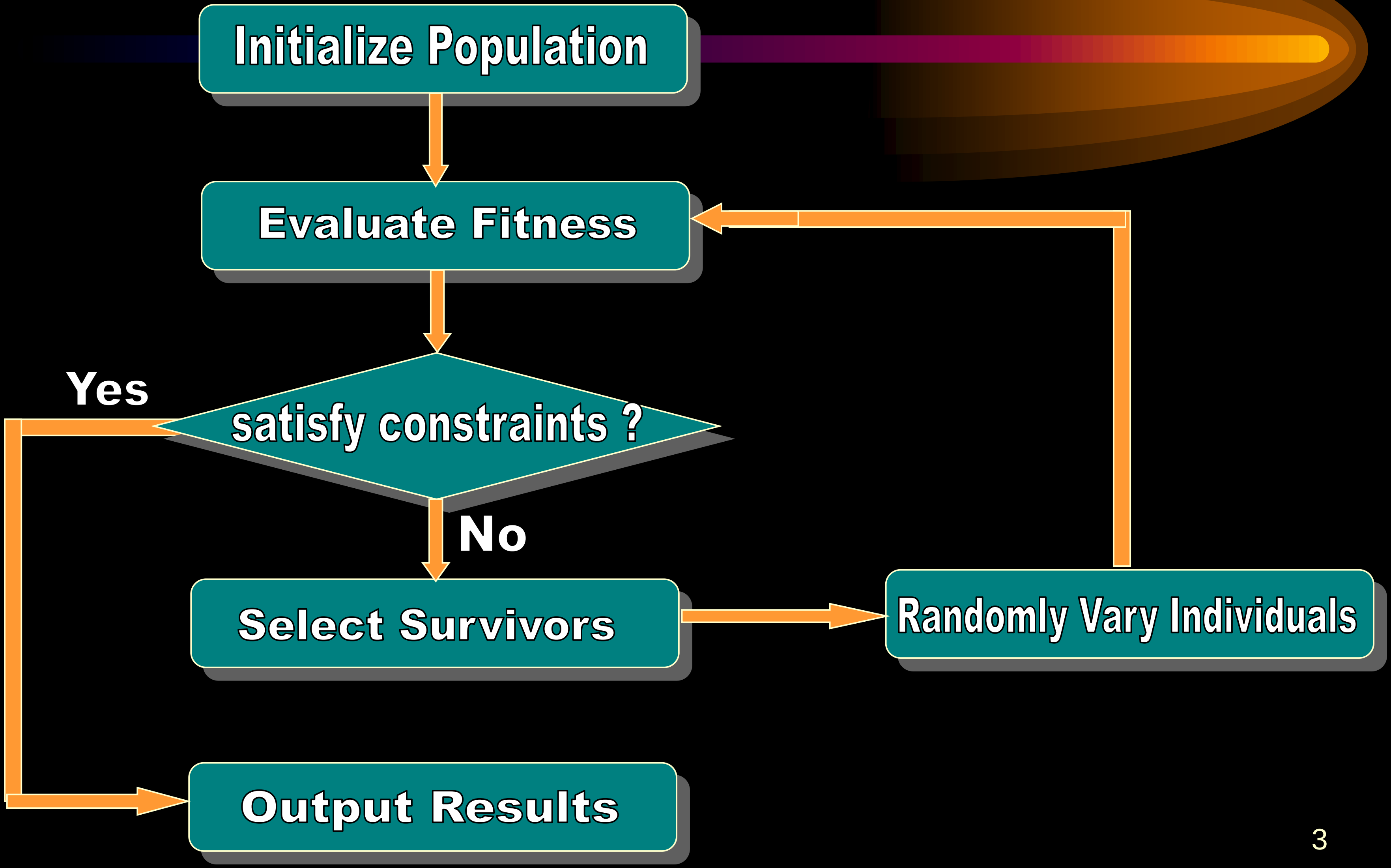
Genetic Algorithm

- Based on Darwinian Paradigm



- Intrinsically a robust search and optimization mechanism

Conceptual Algorithm



A Simple Example

The Traveling Salesman Problem:

Find a tour of a given set of cities so that

- each city is visited only once
- the total distance traveled is minimized

Representation

Representation is an ordered list of city numbers known as an *order-based* GA.

1) London
2) Venice
3) Dunedin
4) Singapore
5) Beijing
6) Phoenix
7) Tokyo
8) Victoria

CityList1 (3 5 7 2 1 6 4 8)

CityList2 (2 5 7 6 8 1 3 4)

Crossover

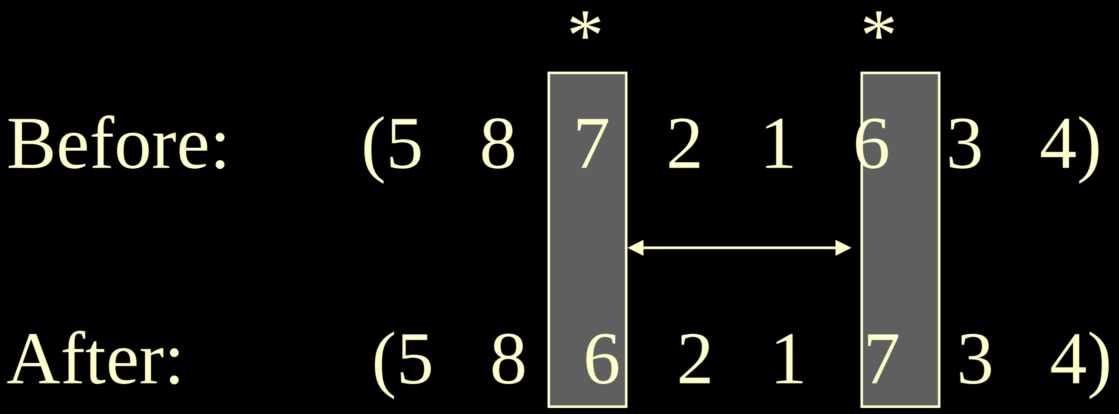
Crossover combines inversion and recombination:

```
                 *           *
Parent1   (3  5    7  2  1  6    4  8)
Parent2   (2  5    7  6  8  1    3  4)
          -----------------------------
Child      (5  8   7  2  1  6    3  4)
```

This operator is called the *Order1* crossover.

Mutation

Mutation involves reordering of the list:

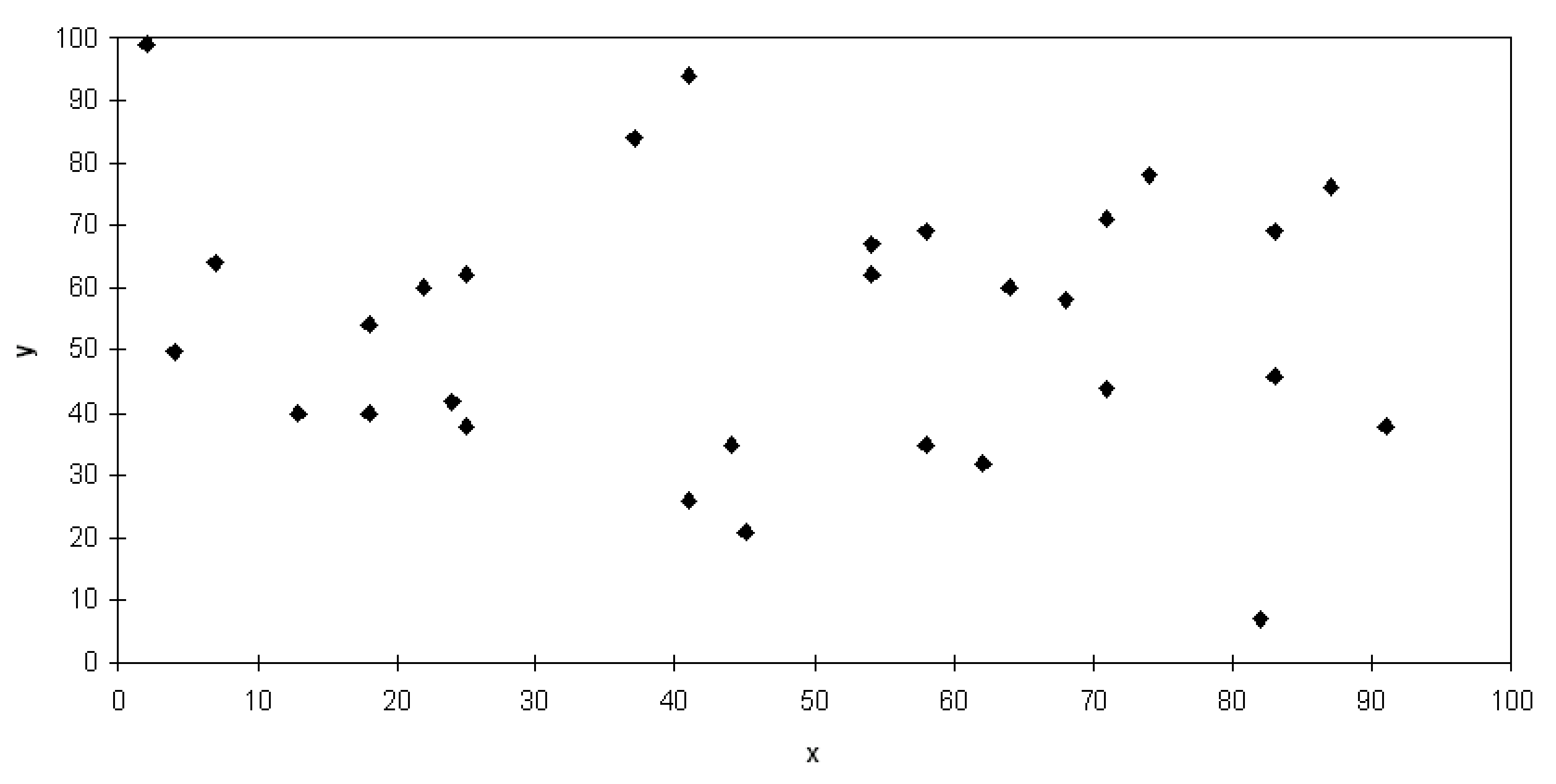


* *

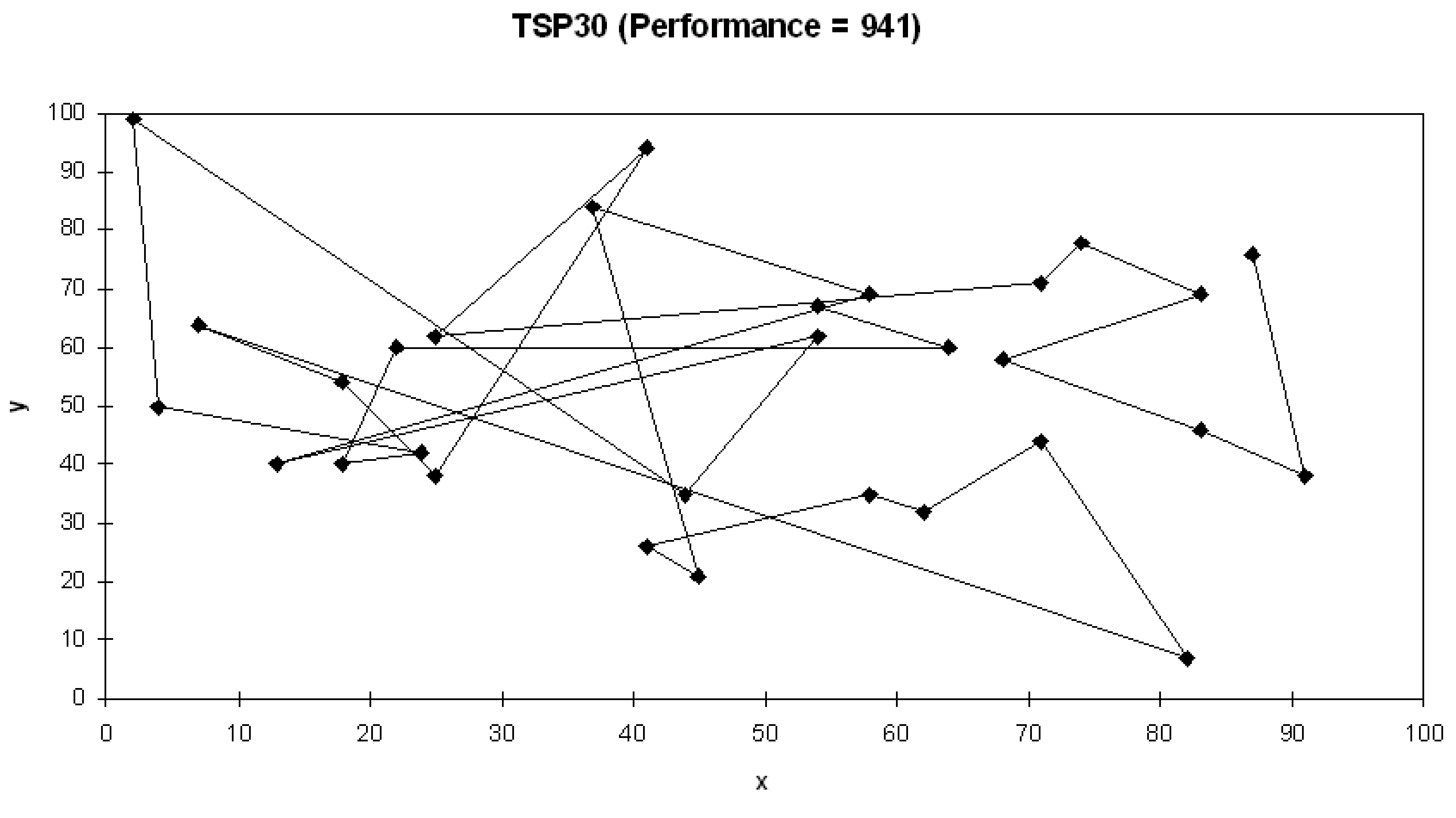
Before: (5 8 7 2 1 6 3 4)

After: (5 8 6 2 1 7 3 4)

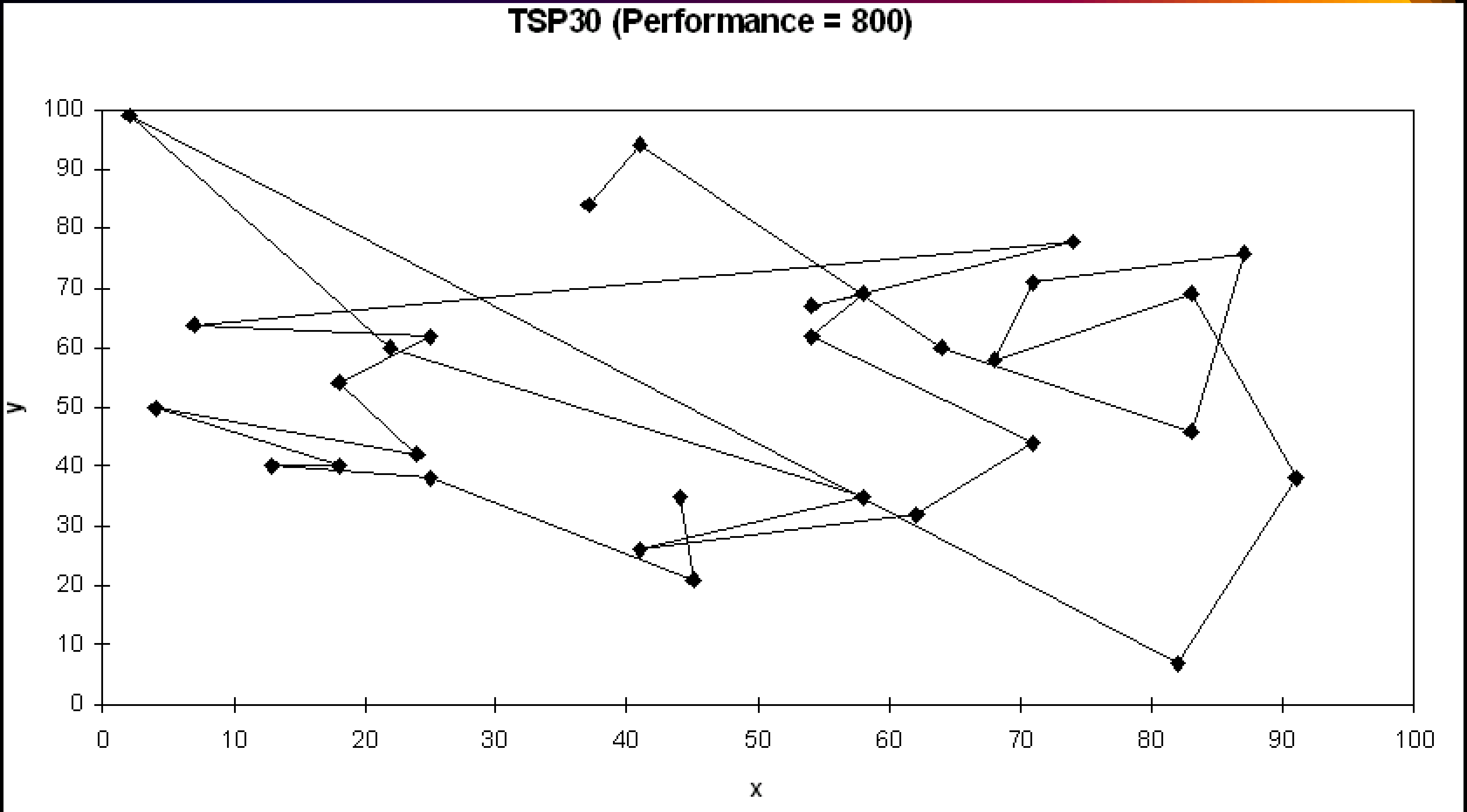
TSP Example: 30 Cities



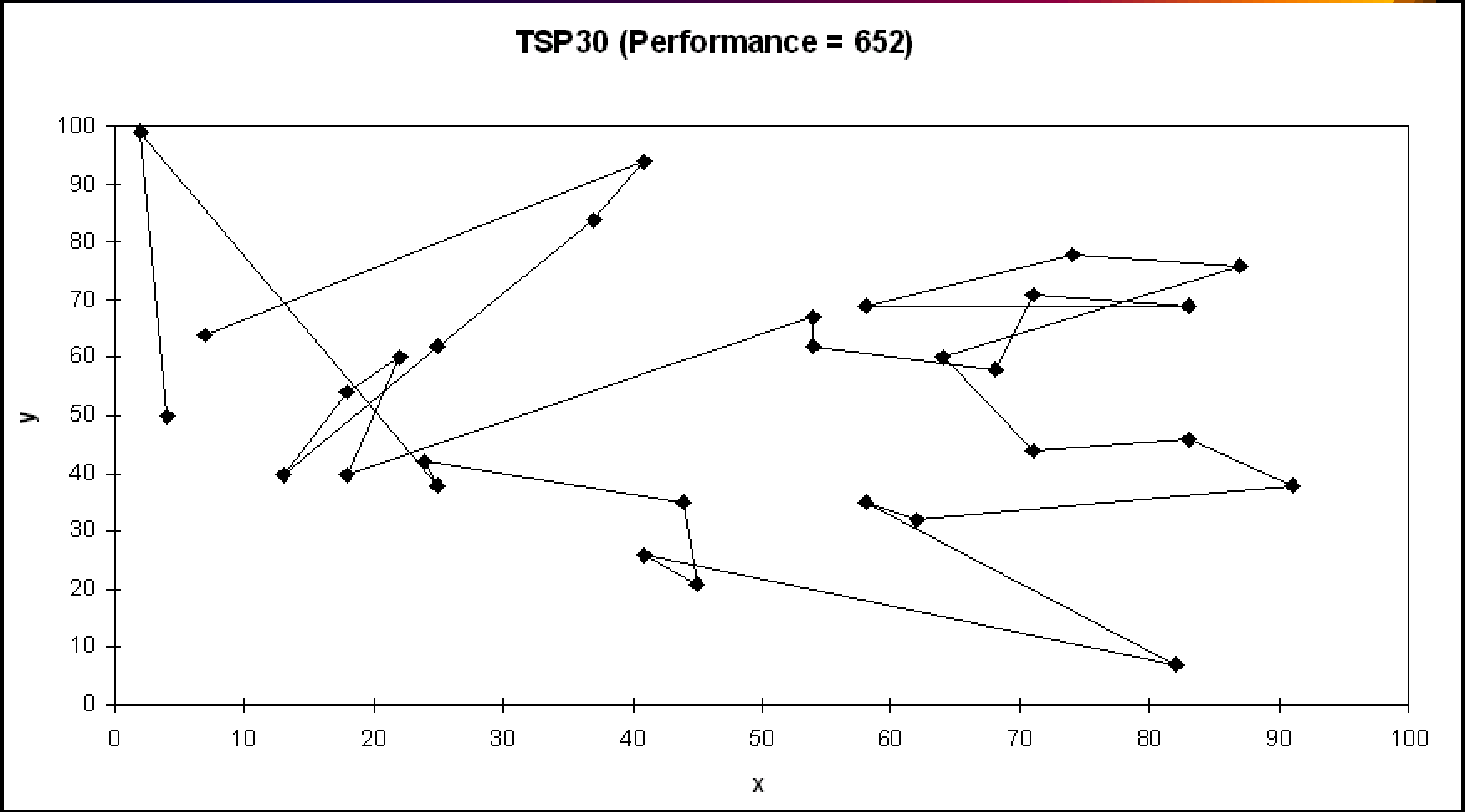
Solution $_i$ (Distance = 941)



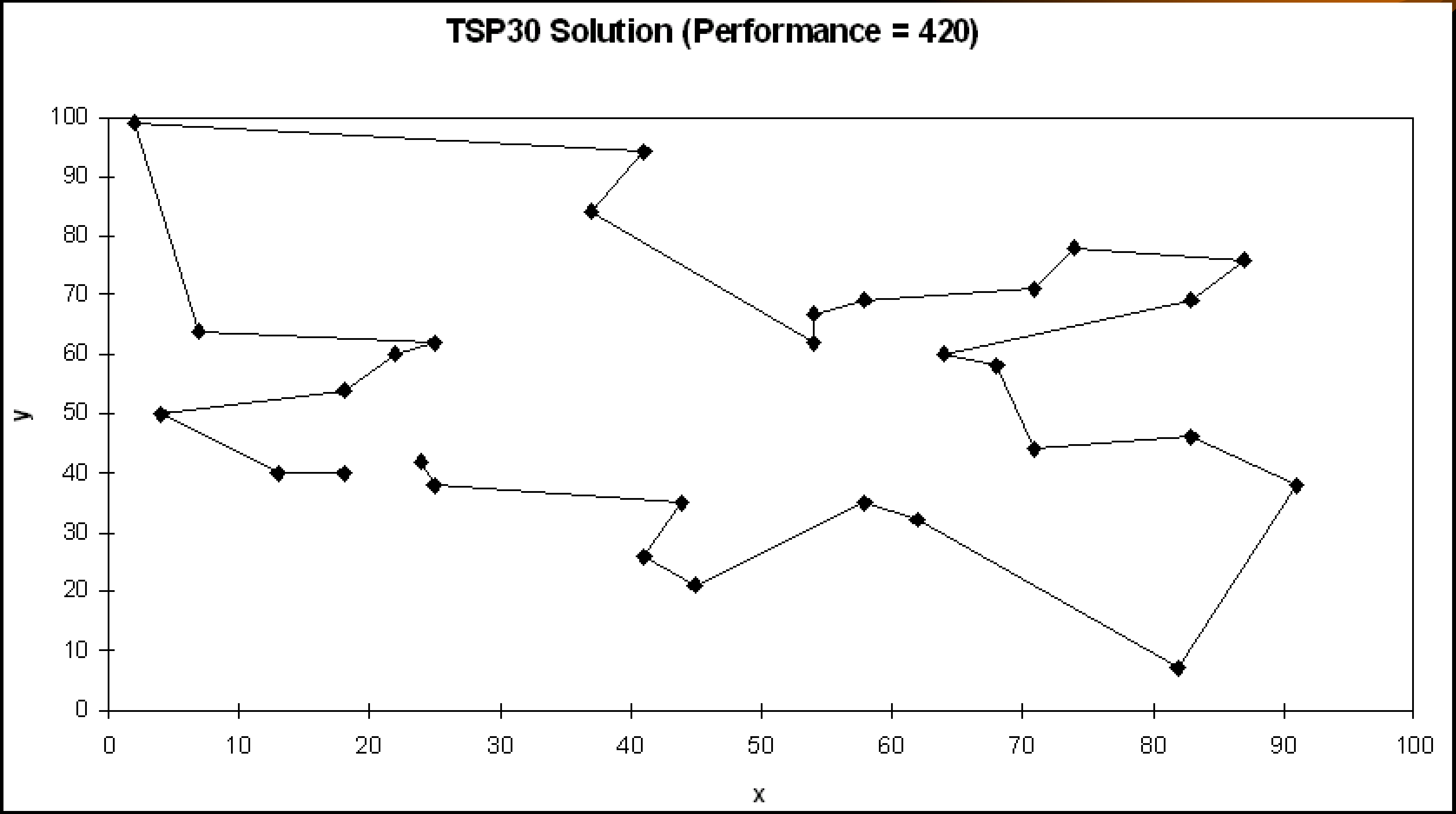
Solution $_j$(Distance = 800)



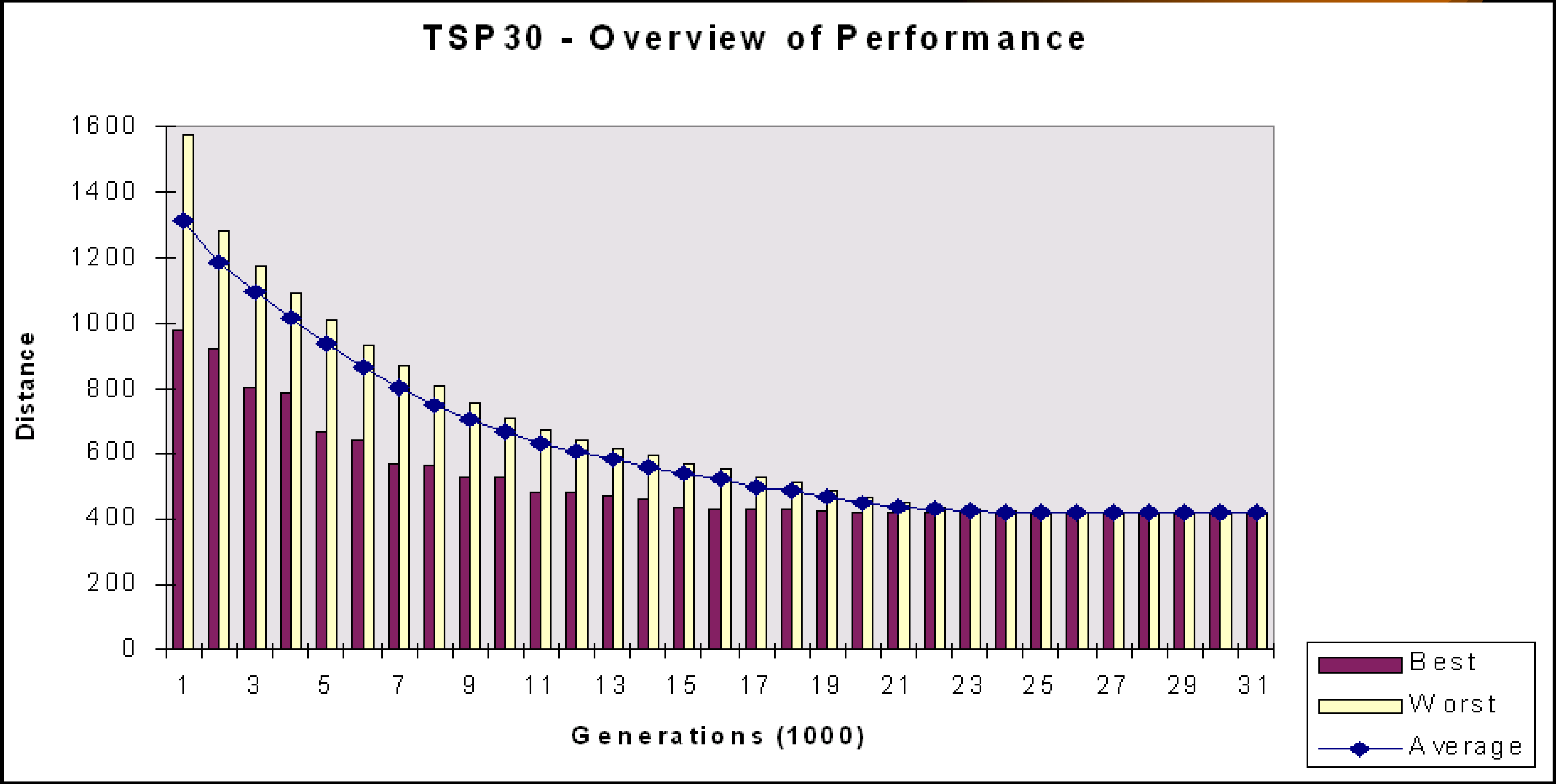
Solution $_k$(Distance = 652)



Best Solution (Distance = 420)



Overview of Performance



Genetic Algorithm

- Encoding
- Fitness Evaluation
- Reproduction
- Survivor Selection

Encoding

- Example:
- *f(**x**) =a+bx+cx^2   0<=x<=3*
- *x = (1011101111100110)* $_2$ *=2.93506*

Encoding

- Example:
- *Travling salesman problem*
- *x = acdebf*
- *Fitnes(x) = total distance of route x.*

Encoding

- individual (chromosome)
- gene
- objectives → fitness

Encoding

Population

Chromosomes could be:

– Bit strings	(0101 ... 1100)
– Real numbers	(43.2 -33.1 ... 0.0 89.2)
– Permutations of element	(E11 E3 E7 ... E1 E15)
– Lists of rules	(R1 R2 R3 ... R22 R23)
– ... any data structure ..	

Fitness Evaluation

- A key component in GA

Reproduction

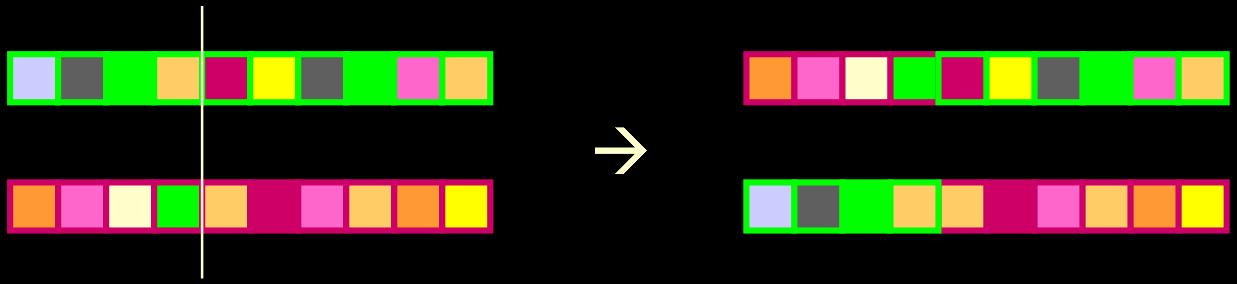
- Reproduction operators
  - Crossover
  - Mutation

Reproduction Operators

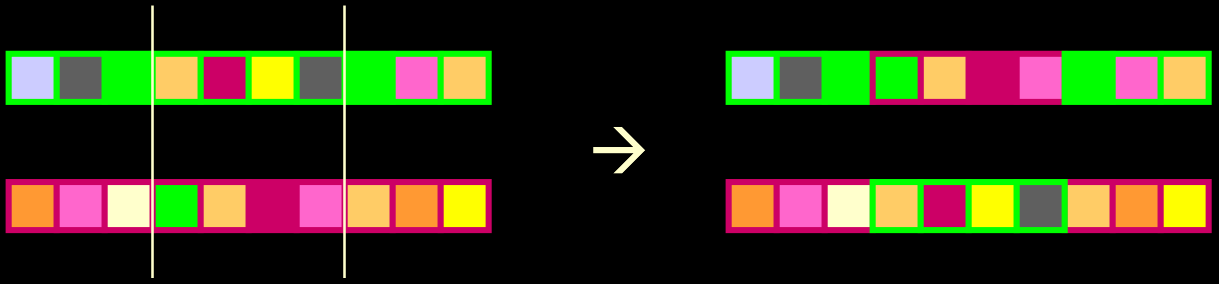
- Crossover
  - Generating offspring from two selected parents
    - Single point crossover
    - Two point crossover (Multi point crossover)
    - Uniform crossover

Reproduction Operators

- Single point crossover

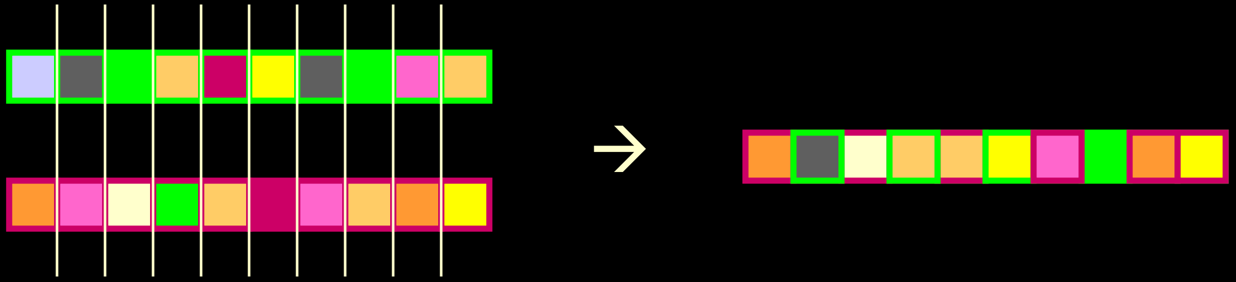


- Two point crossover (Multi point crossover)



Reproduction Operators

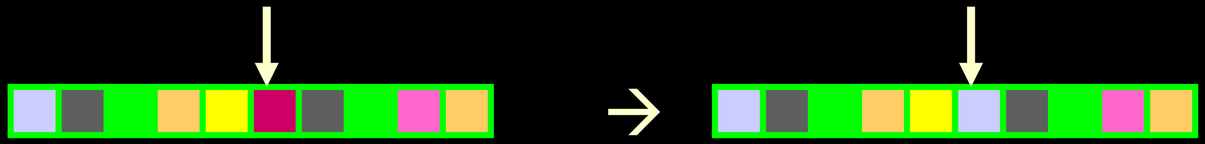
- Uniform crossover



- Is uniform crossover better than single crossover point?
  - Trade off between
    - Exploration: introduction of new combination of features
    - Exploitation: keep the good features in the existing solution

Reproduction Operators

- Mutation
  - Generating new offspring from single parent



  - Maintaining the diversity of the individuals
    - Crossover can only explore the combinations of the current gene pool
    - Mutation can “generate” new genes

Mutation: Local Modification

Before: (1 0 1 1 0 1 1 0)

After: (0 1 1 0 0 1 1 0)

Before: (1.38 -69.4 326.44 0.1)

After: (1.38 -67.5 326.44 0.1)

Reproduction Operators

- Control parameters: **population size, crossover/mutation probability**
  - Problem specific
  - Increase population size
    - Increase diversity and computation time for each generation
  - Increase crossover probability
    - Increase the opportunity for recombination but also disruption of good combination
  - Increase mutation probability
    - Closer to randomly search
    - Help to introduce new gene or reintroduce the lost gene
- Varies the population
  - Usually using crossover operators to recombine the genes to generate the new population, then using mutation operators on the new population

Parent/Survivor Selection

- Strategies
  - Parent selection
    - Uniform randomly selection
    - Probability selection : proportional to their fitness
    - Etc.

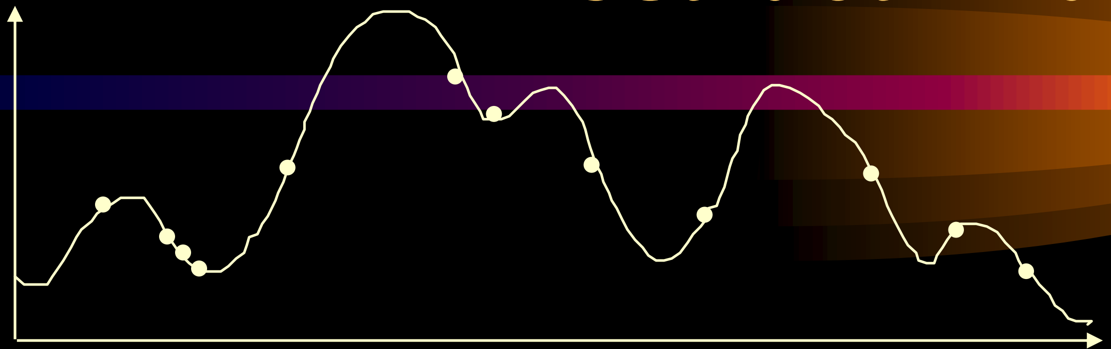
Parent/Survivor Selection

- Strategies
  - Survivor selection
    - Always keep the best one
    - Elitist: deletion of the K worst
    - Etc.

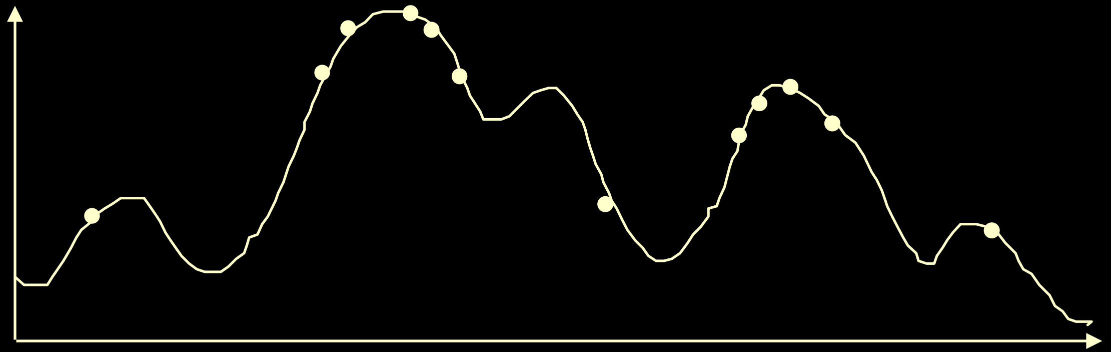
Parent/Survivor Selection

- Too strong fitness selection can lead to sub-optimal solution
- Too little fitness selection results in unfocused and meandering search

An Abstract Example



Distribution of Individuals in Generation 0



Distribution of Individuals in Generation N

Some GA Application Types

Domain	Application Types
Control	gas pipeline, pole balancing, missile evasion, pursuit
Design	semiconductor layout, aircraft design, keyboard configuration, communication networks
Scheduling	manufacturing, facility scheduling, resource allocation
Robotics	trajectory planning
Machine Learning	designing neural networks, improving classification algorithms, classifier systems
Signal Processing	filter design
Game Playing	poker, checkers, prisoner抯dilemma
Combinatorial Optimization	set covering, travelling salesman, routing, bin packing, graph colouring and partitioning