Lecture 8

Genetic Algorithms II

Lusi Li
Department of Computer Science
ODU

Review

- **Last Class**
  - Introduction to Genetic Algorithm
- **This Class**
  - More on Genetic Algorithm
- **Next Class**
  - Constraint Satisfaction Problem

Genetic Algorithms

- Start with n random solutions (the initial population)
- Repeatedly do the following:
  - Evaluate each of the solutions by the fitness function
  - Select a subset of the best solutions probabilistically
  - Use these solutions to generate a new population by
    - "crossover" with a rate
    - "mutation" with a rate
  - The new population is used in the next iteration
- Quit when you have a satisfactory solution (or you run out of time)

Components of a GA

A problem definition as input, and

- **Encoding principles (gene, chromosome)**
- **Initialization procedure (creation)**
- **Selection of parents (selection)**
- **Genetic operators (crossover, mutation)**
- **Evaluation function (fitness to environment)**
- **Termination condition**

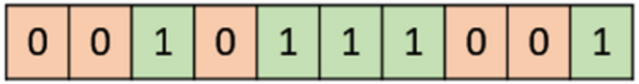
Genetic Algorithms Operators

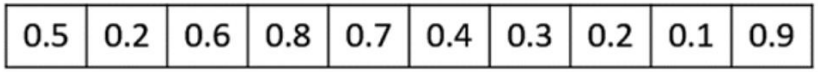
- Methods of representation
- Methods of selection
- Methods of reproduction (crossover and mutation)

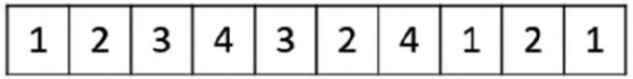
Common Representation Methods

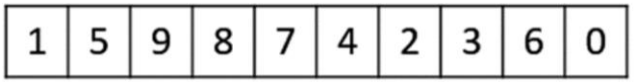
- **Different algorithms may use different data structures to store genetic information.**

- ***Binary Representation***
  - ***{0,1}***
- ***Real Valued Representation***
  - ***e.g., [0,1]***
- ***Integer Representation***
  - ***e.g., {1,2,3,4}***
- ***Permutation Representation***
  - ***an order of the elements***
- ***……***









Methods of Selection

- **There are many different strategies to select the individuals to be copied over into the next generation.**
  - ***Fitness proportionate selection***
    - ***Roulette-wheel selection***
  - ***Rank selection***
  - ***Elitist selection***
  - ***Tournament selection***
  - ***Cutoff selection***
  - ***…***

Roulette Wheel Selection

- **Conceptually, this can be represented as a game of roulette - each individual gets a slice of the wheel, but more fit ones get larger slices than less fit ones.**

No.	String	Fitness	% Of Total	
1	01101	169	14.4	[0, 0.144]
2	11000	576	49.2	(0.144, 0.636]
3	01000	64	5.5	(0.636, 0.691]
4	10011	361	30.9	(0.691, 1]
Total		1170	100.0	

E.g., A randomly generated num. between 0 and 1 = 0.3
=> individual 2 is selected

Other Selection Methods

- ***Rank selection:***

  Select the individuals with higher ranks

- ***Elitist selection*:**

  Select only the most fit individuals to pass to the next generation without crossover and mutation operators.

- **Tournament selection:**

  Select K individuals from the population at random and select the best out of these to become a parent.

- ***Cutoff selection*:**

  Select only those that are above a certain cutoff for the target function.

Methods of Reproduction

- **There are primary methods:**
  - ***Crossover***
  - ***Mutation***

Methods of Reproduction: Crossover

- Two parents produce two offspring
- Two options:

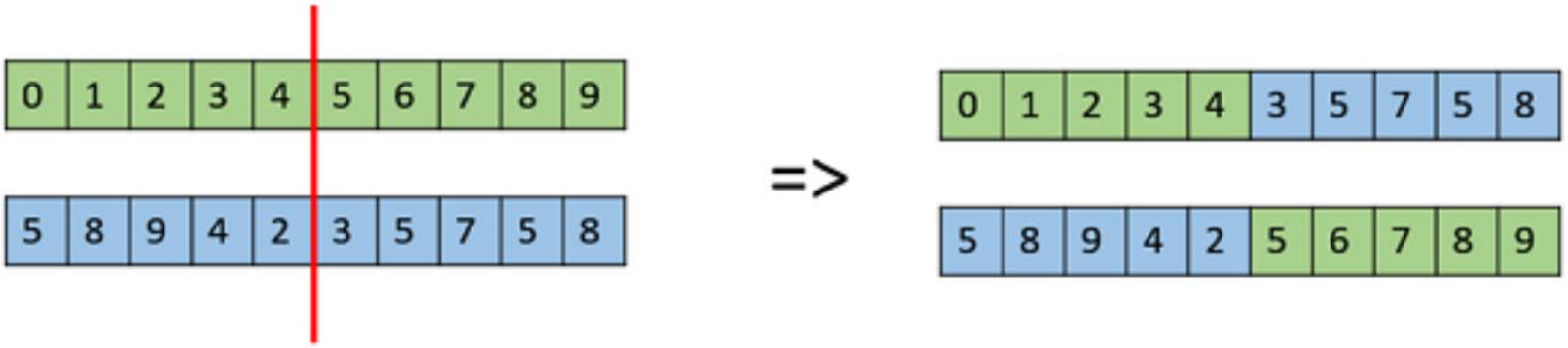
  1. The chromosomes of the two parents are copied to the next generation
  2. The two parents are randomly recombined (crossed-over) to form new offspring

Several possible crossover strategies

- One point crossover
- Multi point crossover
- Uniform crossover
- Order 1 crossover
- …

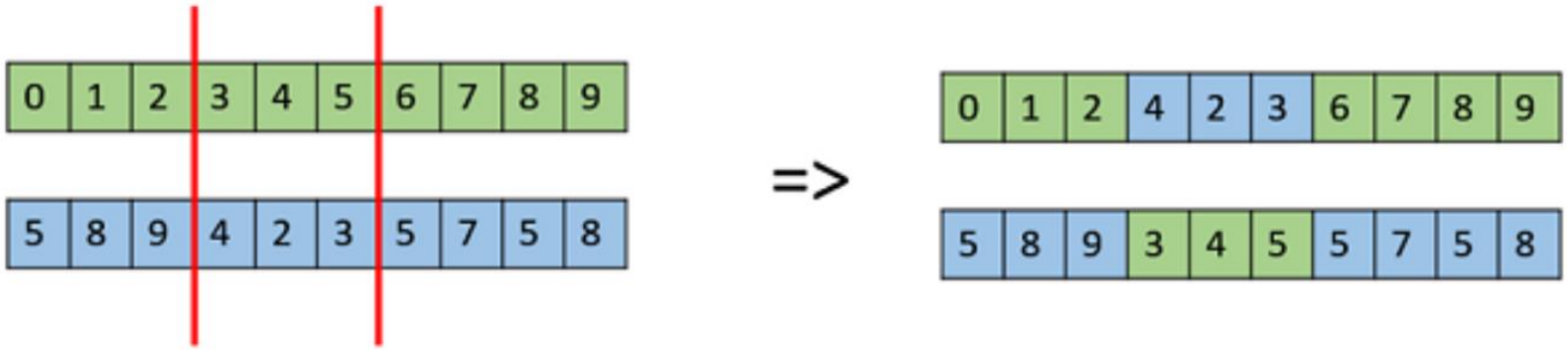
Crossover

- **One-point crossover**



Cross point = 5

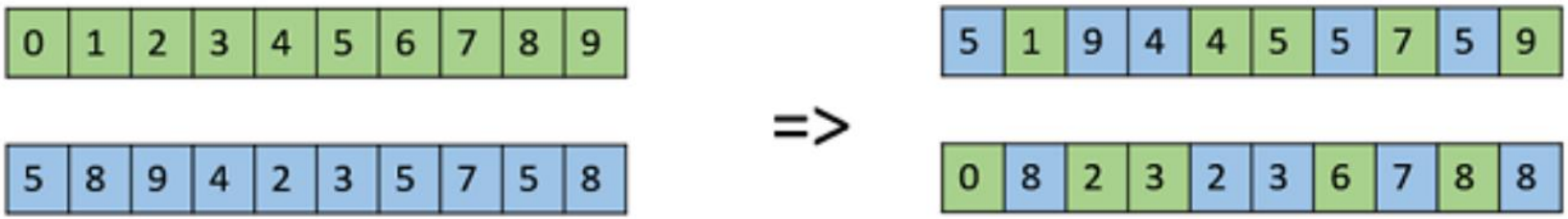
- **Two-point crossover (Multi-point crossover)**



Cross points = 3, 6

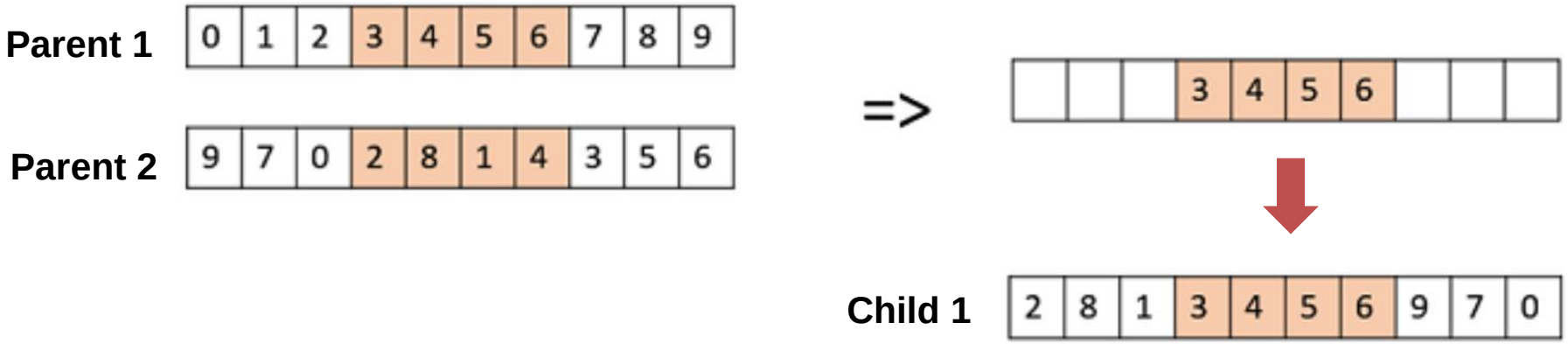
Uniform Crossover

- Treat each gene separately.
- Flip a coin for each chromosome to decide whether or not it'll be included in the off-spring.
- We can also bias the coin to one parent, to have more genetic material in the child from that parent.



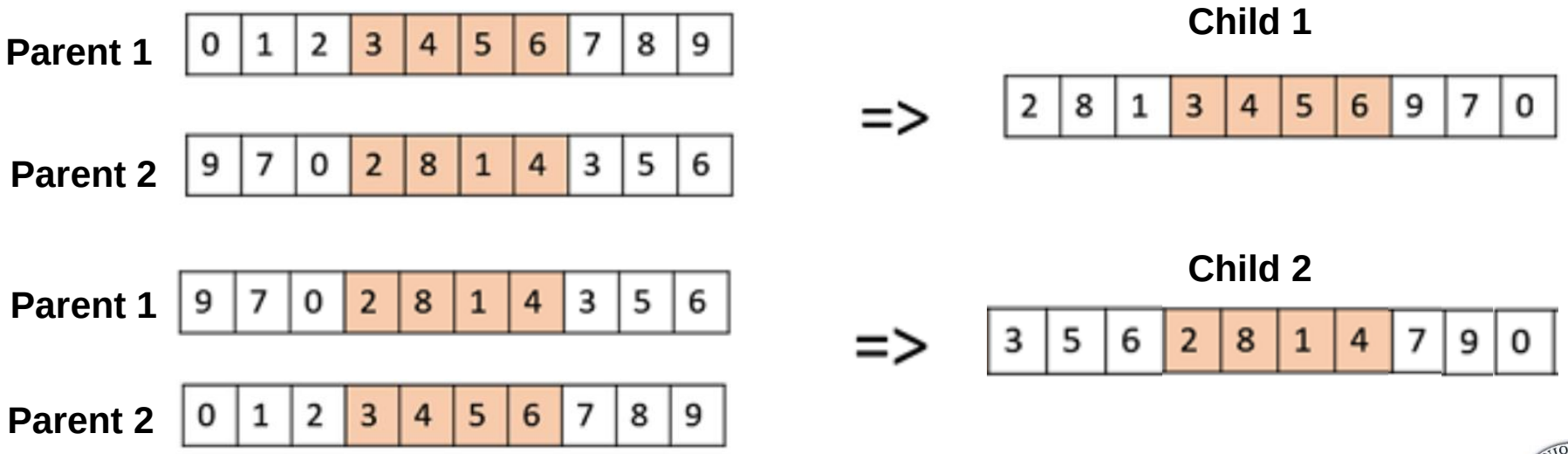
Order 1 Crossover

- **A permutation crossover preserves relative order that elements occur**
  - **Create two random crossover points in the parent and copy the segment between them from the first parent to the first offspring.**
  - **Now, starting from the second crossover point in the second parent, copy the remaining unused numbers from the second parent to the first child, wrapping around the list.**
  - **Repeat for the second child with the parent's role reversed.**
  - **Child 2?**



Order 1 Crossover

- **A permutation crossover preserves relative order that elements occur**
  - **Create two random crossover points in the parent and copy the segment between them from the first parent to the first offspring.**
  - **Now, starting from the second crossover point in the second parent, copy the remaining unused numbers from the second parent to the first child, wrapping around the list.**
  - **Repeat for the second child with the parent's role reversed.**
  - **Child 2?**



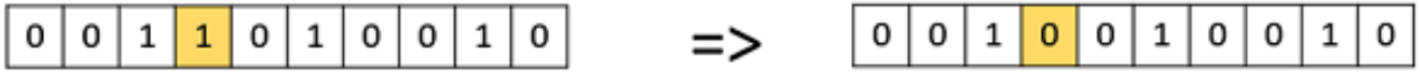
Methods of Reproduction: Mutations

- A small random tweak in a single chromosome, to get a new solution.
  - It is used to maintain and introduce diversity in the genetic population
  - It is usually applied with a low probability

- Common mutations:
  - Bit Flip Mutation
  - Random Resetting
  - Swap Mutation
  - Scramble Mutation
  - Inversion Mutation

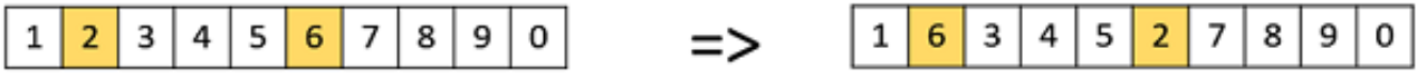
Mutation

- **Bit Flip Mutation for binary encoded GAs**

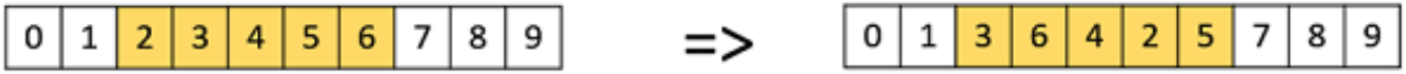


- **Random Resetting**
  - **a random value from the set of permissible values is assigned to a randomly chosen gene**

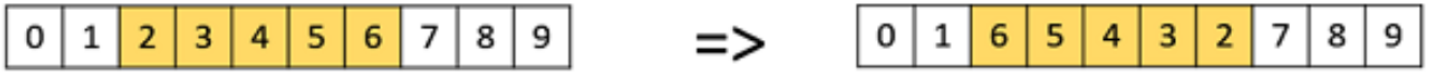
- **Swap Mutation (interchange the values of two random-selected genes)**



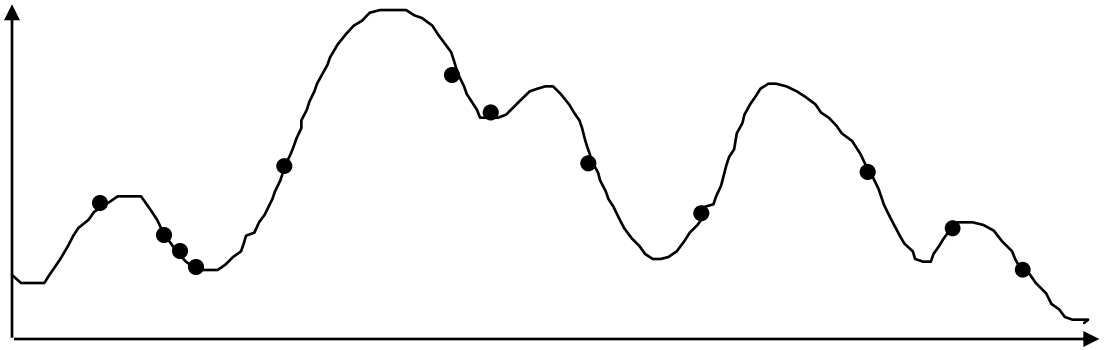
- **Scramble Mutation (scramble a subset of genes)**



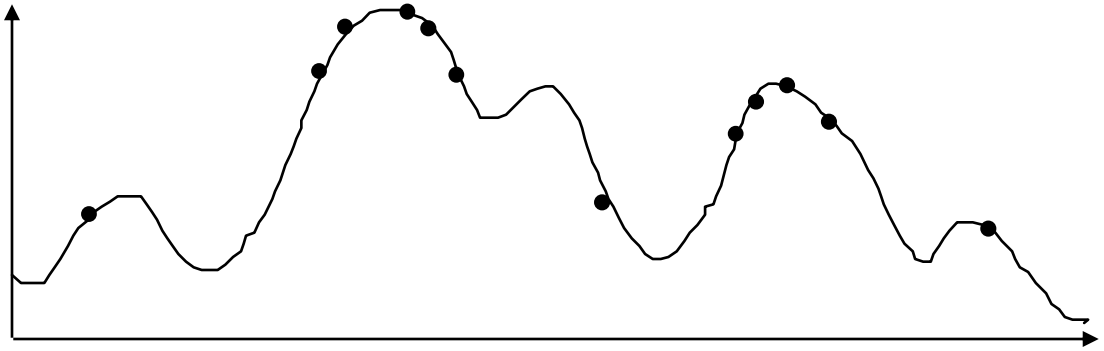
- **Inversion Mutation (invert a subset of genes)**



An Abstract Example

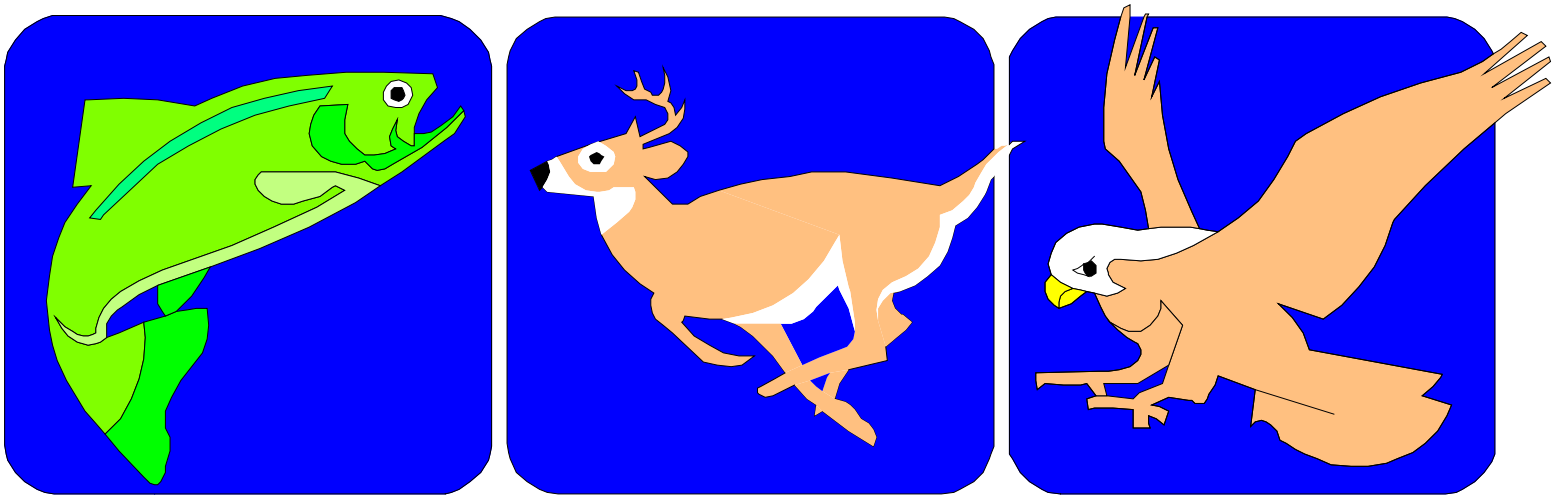


Distribution of Individuals in Generation 0



Distribution of Individuals in Generation N

A Simple Example



"The Gene is by far the most sophisticated program around."

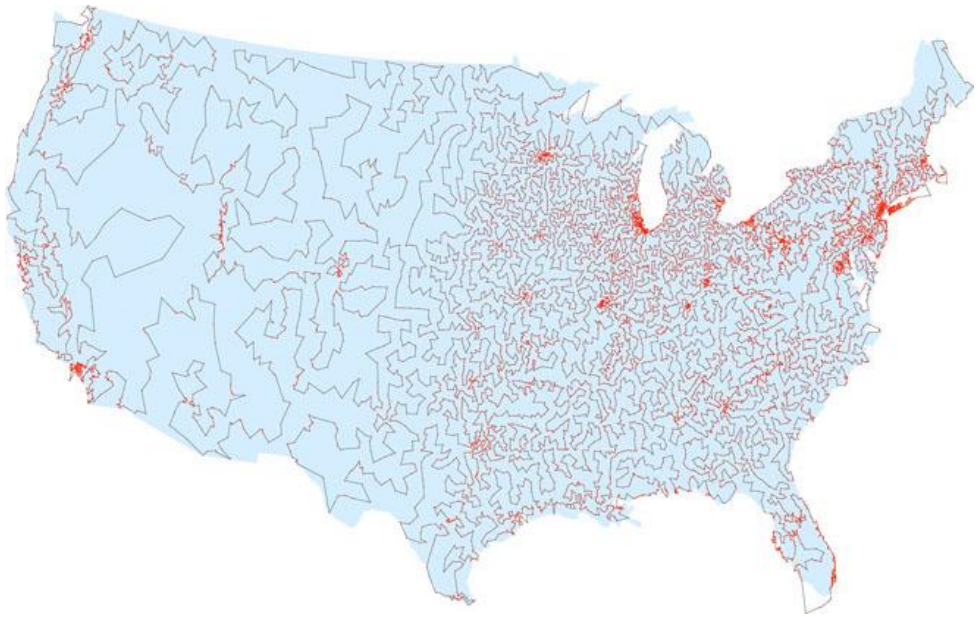
- Bill Gates, *Business Week*, June 27, 1994

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**

- **Finding a solution is NP-hard**



Representation

Representation is an ordered list of city

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

Representation

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

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

CityList1 (3 5 7 2 1 6 4 8)

CityList2 (2 5 7 6 8 1 3 4)

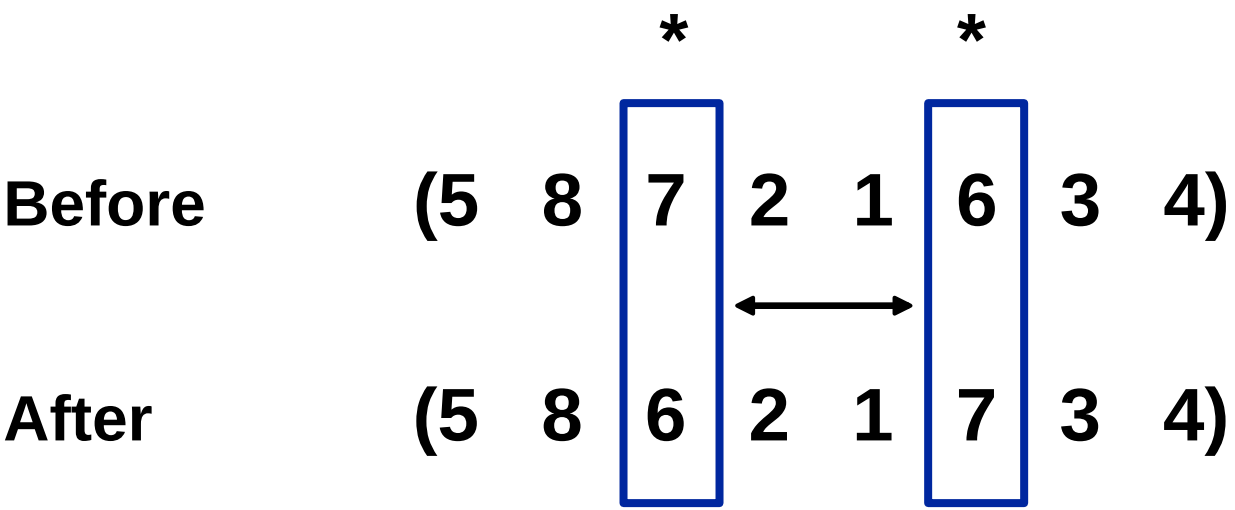
Crossover

Order 1 crossover

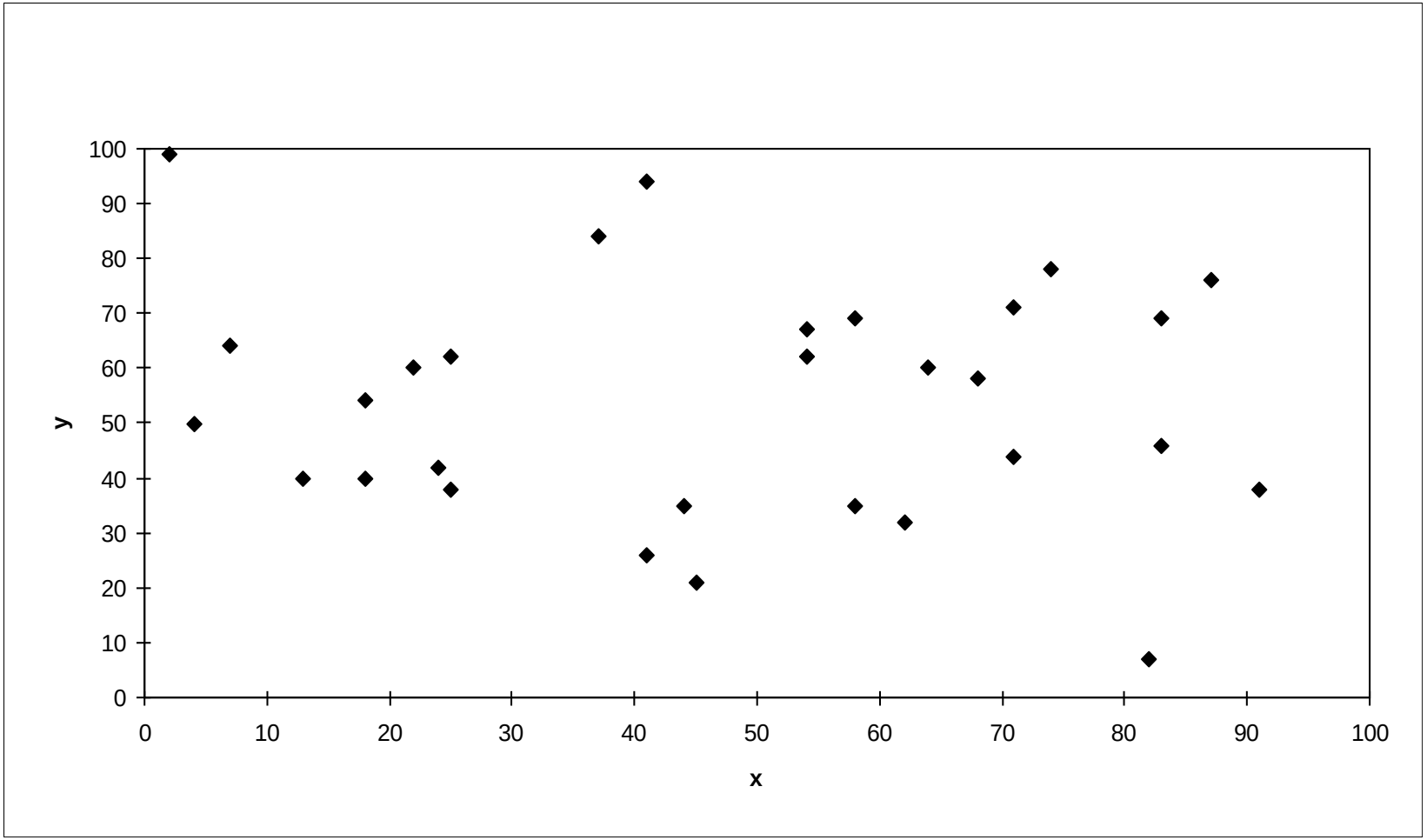
			*			*		
Parent1	(3	5	7	2	1	6	4	8)
Parent2	(2	5	7	6	8	1	3	4)
Child1	(5	8	7	2	1	6	3	4)

Mutation

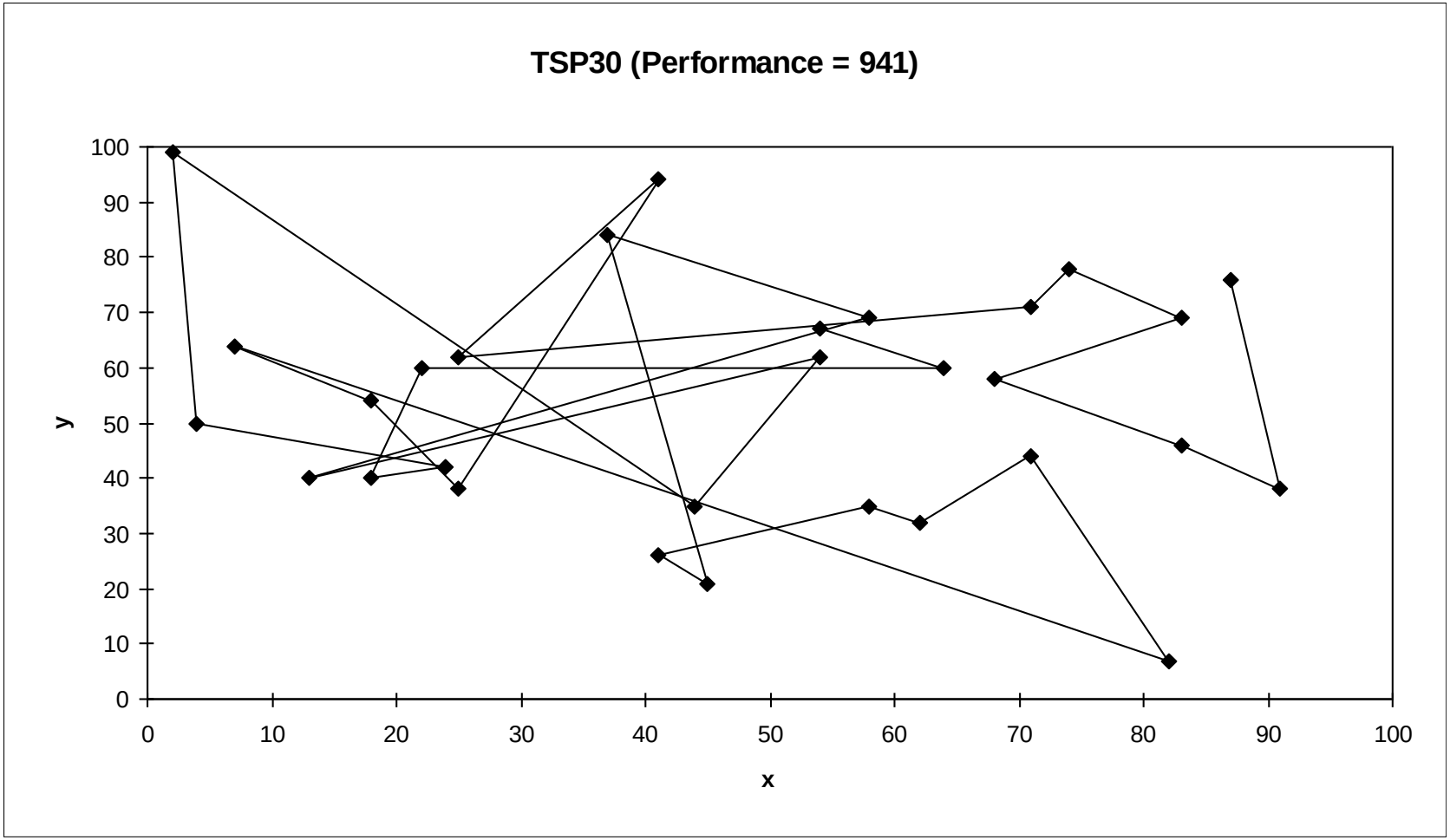
Swap Mutation



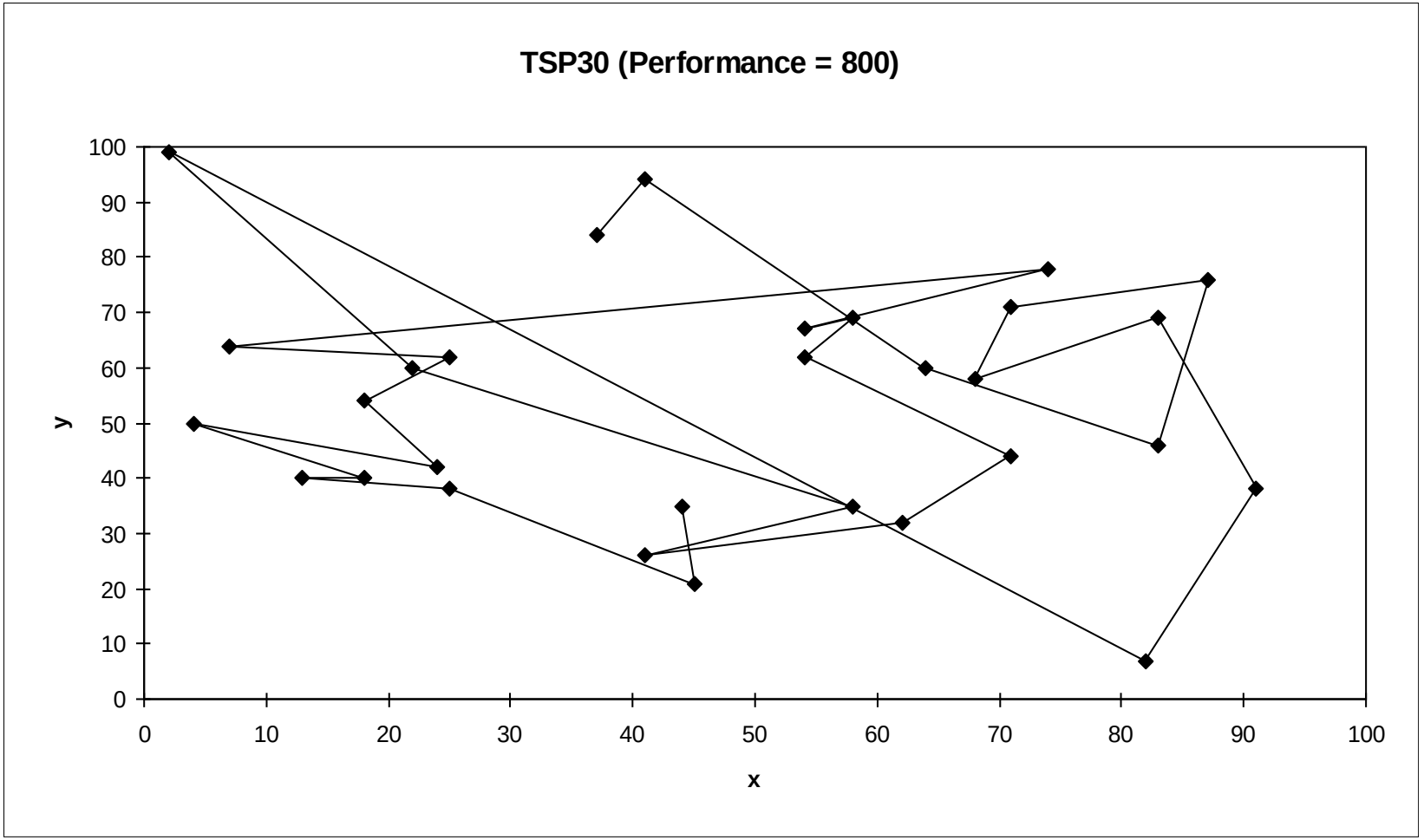
TSP Example: 30 Cities



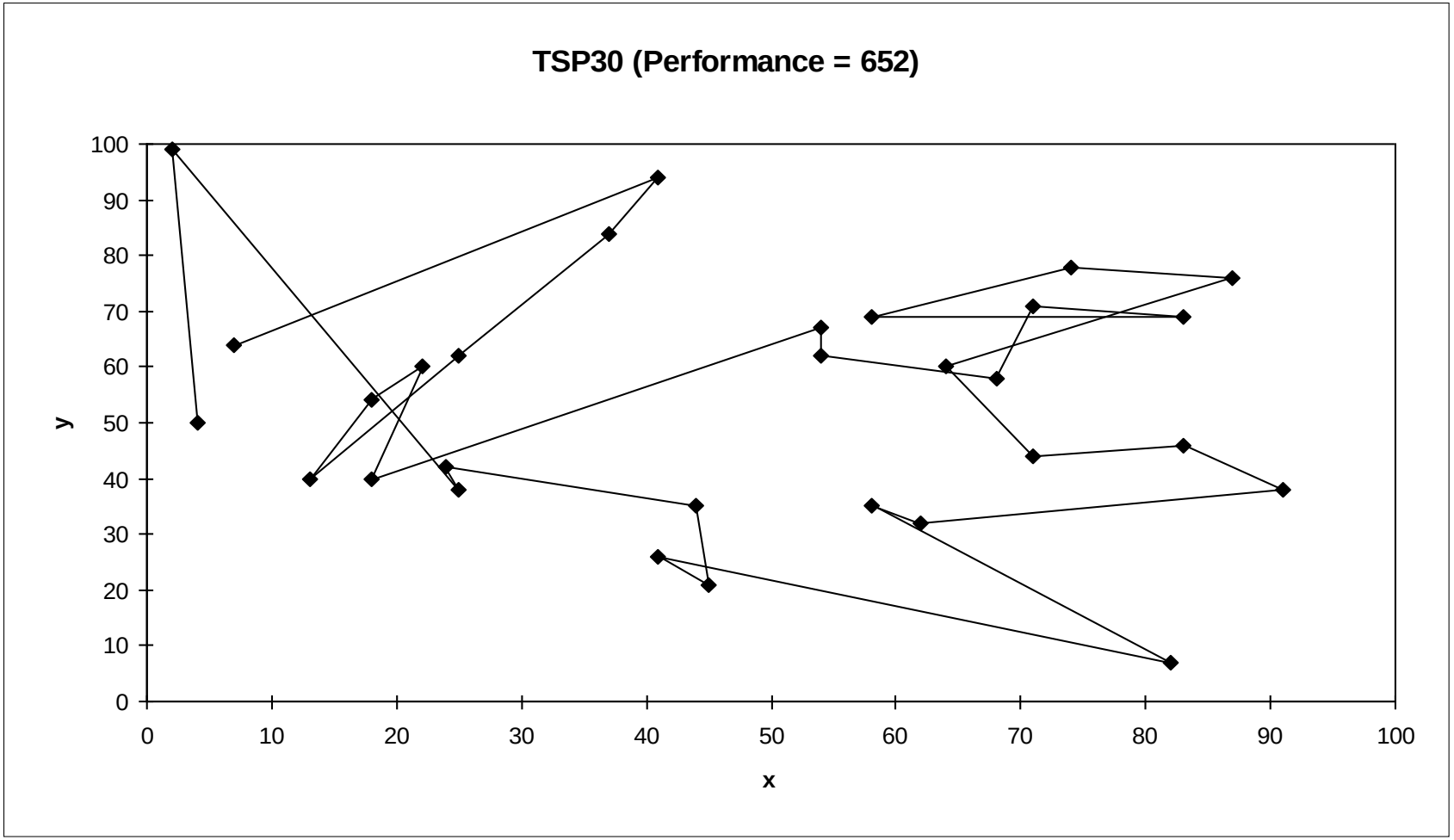
Solution *at 3rd Generation* (Distance=941)



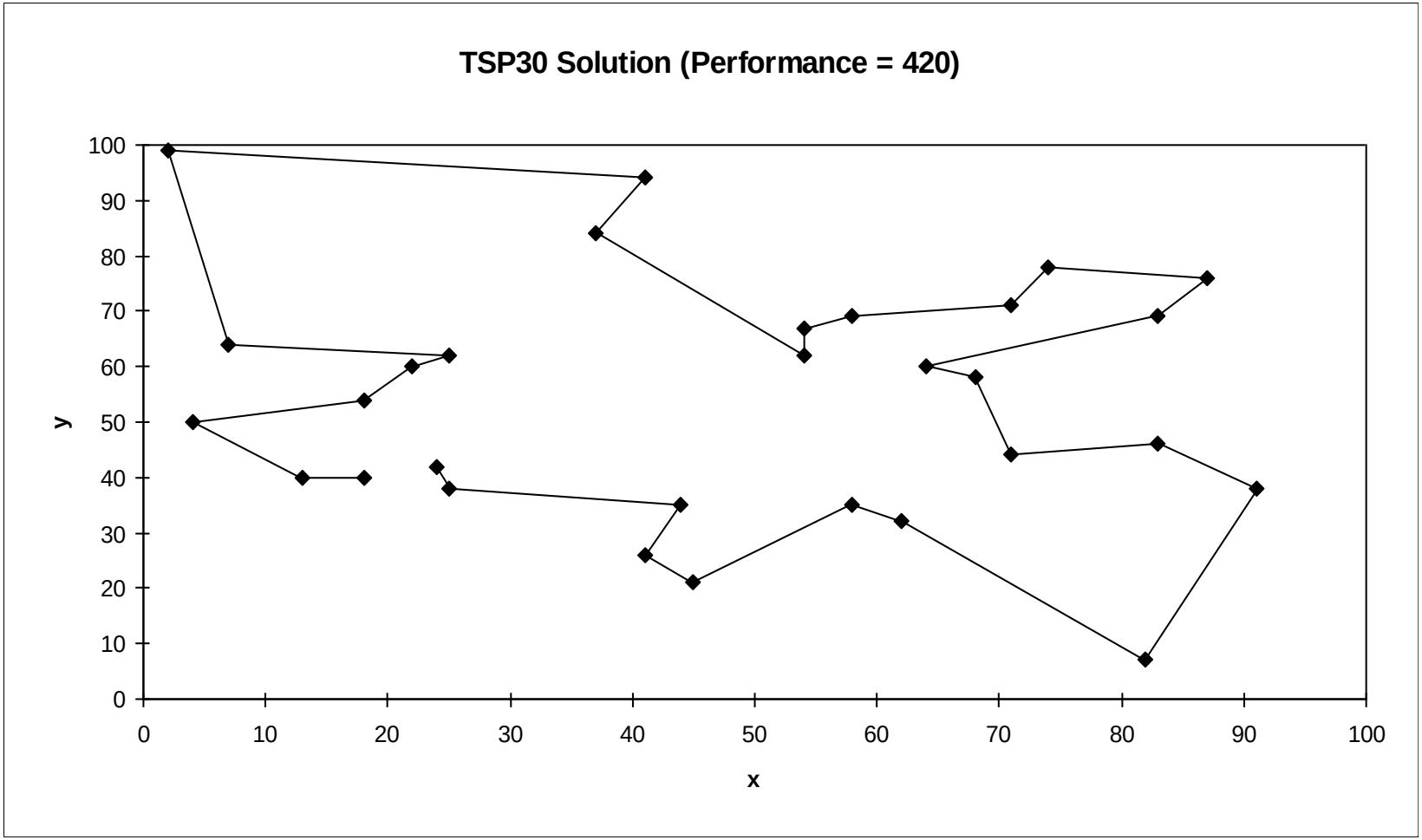
Solution *at 5th Generation* (Distance=800)



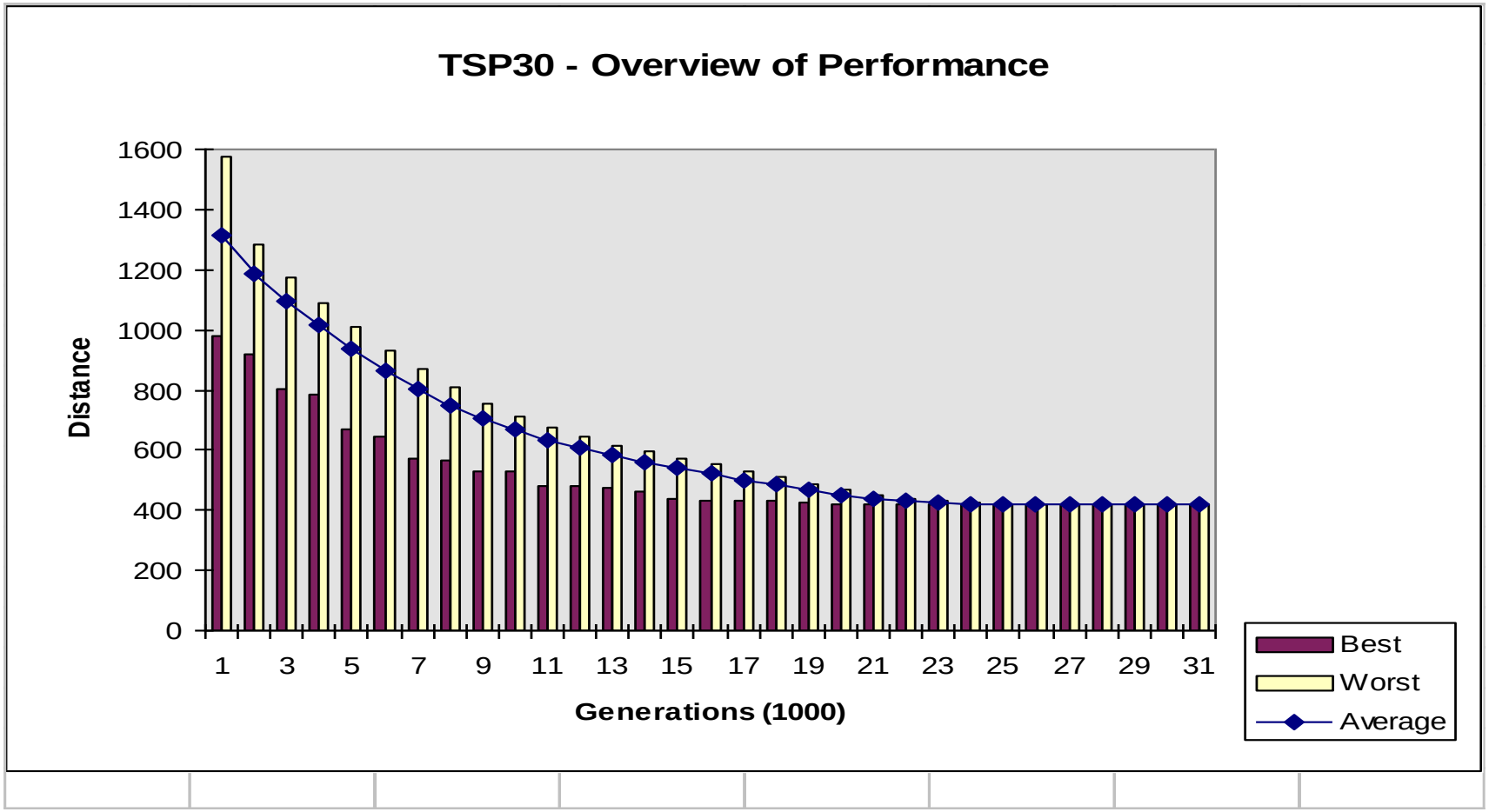
Solution *at 10th Generation* (Distance=652)



Solution at 20th Generation (Distance=420)



Performance Overview



Issues of Genetic Algorithms

- Choosing basic implementation issues:
  - representation
  - population size, mutation/crossover rate, ...
  - selection policies
  - crossover, mutation operators
- Termination criteria
- Performance, scalability
- Solution is only as good as the evaluation function (often hardest part)

Benefits of Genetic Algorithms

- Concept is easy to understand
- Good for “noisy” environments
- Global optimization
- Parallel search; speed up the search
- Handling multiple objectives
- Always an answer; answer gets better with time

Applications of Genetic Algorithms

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's dilemma
Combinatorial Optimization	set covering, travelling salesman, routing, bin packing, graph colouring and partitioning

Summary

- **Genetic Algorithm**
  - **Representation**
  - **Crossover**
  - **Mutation**
  - **Application in Traveling Salesman Problem**

What I want you to do

- **Review course slides**