Algorithms in Nature

Genetic algorithms

Some of the popular examples of nature-inspired optimization algorithms include:

- Genetic algorithm,
- Particle swarm optimization,
- Cukcoo search algorithm,
- Ant colony optimization and so on.

History

Applications of Nature-Inspired Algos:

- Digital filter designing
- Image processing
- Machine-learning
- Digital integrator and differentiator designing
- Face-recognition
- Artificial neural networks

History of GAs

- As early as 1962, John Holland's work on adaptive systems laid the foundation for later developments.

- By the 1975, the publication of the book *Adaptation in Natural and Artificial Systems*, by Holland and his students and colleagues.

History of GAs

- early to mid-1980s, genetic algorithms were being applied to a broad range of subjects.

- In 1992 John Koza has used genetic algorithm to evolve programs to perform certain tasks. He called his method "genetic programming" (GP).

What is GA

- A genetic algorithm (or GA) is a search technique used in computing to find true or approximate solutions to optimization and search problems.
- (GA)s are categorized as global search heuristics.
- (GA)s are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).

What is GA

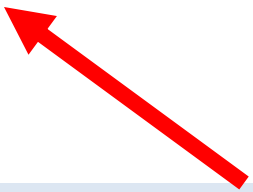
- The evolution usually starts from a population of randomly generated individuals and happens in generations.

- In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected from the current population (based on their fitness), and modified to form a new population.

What is GA

- The new population is used in the next iteration of the algorithm.

- The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.



No convergence rule or guarantee!

Vocabulary

- **Individual** - Any possible solution
- **Population** - Group of all individuals
- **Fitness** – Target function that we are optimizing (each individual has a fitness)
- **Trait** - Possible aspect (features) of an individual
- **Genome** - Collection of all chromosomes (traits) for an individual.

Basic Genetic Algorithm

- Start with a large "population" of randomly generated "attempted solutions" to a problem
- Repeatedly do the following:
  - Evaluate each of the attempted solutions
  - (probabilistically) keep a subset of the best solutions
  - Use these solutions to generate a new population
- Quit when you have a satisfactory solution (or you run out of time)

Example:
the MAXONE problem

Suppose we want to maximize the number of ones in a string of / binary digits

Is it a trivial problem?

It may seem so because we know the answer in advance

However, we can think of it as maximizing the number of correct answers, each encoded by 1, to / yes/no difficult questions`

Example (cont)

- An individual is encoded (naturally) as a string of l binary digits
- The fitness f of a candidate solution to the MAXONE problem is the number of ones in its genetic code
- We start with a population of n random strings. Suppose that l = 10 and n = 6

Example (initialization)

We toss a fair coin 60 times and get the following initial population:

$s_1 = 1111010101 \quad f(s_1) = 7$

$s_2 = 0111000101 \quad f(s_2) = 5$

$s_3 = 1110110101 \quad f(s_3) = 7$

$s_4 = 0100010011 \quad f(s_4) = 4$

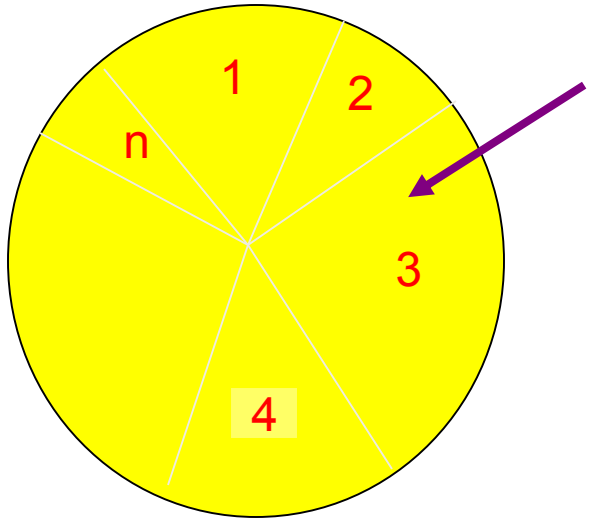
$s_5 = 1110111101 \quad f(s_5) = 8$

$s_6 = 0100110000 \quad f(s_6) = 3$

Step 1: Selection

We randomly (using a biased coin) select a subset of the individuals based on their fitness:

Individual *i* will have a probability to be chosen $\frac{f(i)}{\sum_i f(i)}$



Selected set

Suppose that, after performing selection, we get the following population:

s_1` = 1111010101 (s_1)

s_2` = 1110110101 (s_3)

s_3` = 1110111101 (s_5)

s_4` = 0111000101 (s_2)

s_5` = 0100010011 (s_4)

s_6` = 1110111101 (s_5)

Step 2: crossover

- Next we mate strings for crossover. For each couple we first decide (using some pre-defined probability, for instance 0.6) whether to actually perform the crossover or not
- If we decide to actually perform crossover, we randomly extract the crossover points, for instance 2 and 5

Crossover result

Before crossover:

s_1` = 11110**1**0101 $s_2$` = 1110**1**10101

After crossover:

s_1`` = 11101**1**0101 $s_2$`` = 1111**1**0101

Step 3: mutations

The final step is to apply random mutations: for each bit that we are to copy to the new population we allow a small probability of error (for instance 0.1)

Initial strings	After mutating
s_1`` = 1110110101	s_1``` = 1110100101
s_2`` = 1111010101	s_2``` = 1111110100
s_3`` = 1110111101	s_3``` = 1110101111
s_4`` = 0111000101	s_4``` = 0111000101
s_5`` = 0100011101	s_5``` = 0100011101
s_6`` = 1110110011	s_6``` = 1110110001

And now, iterate …

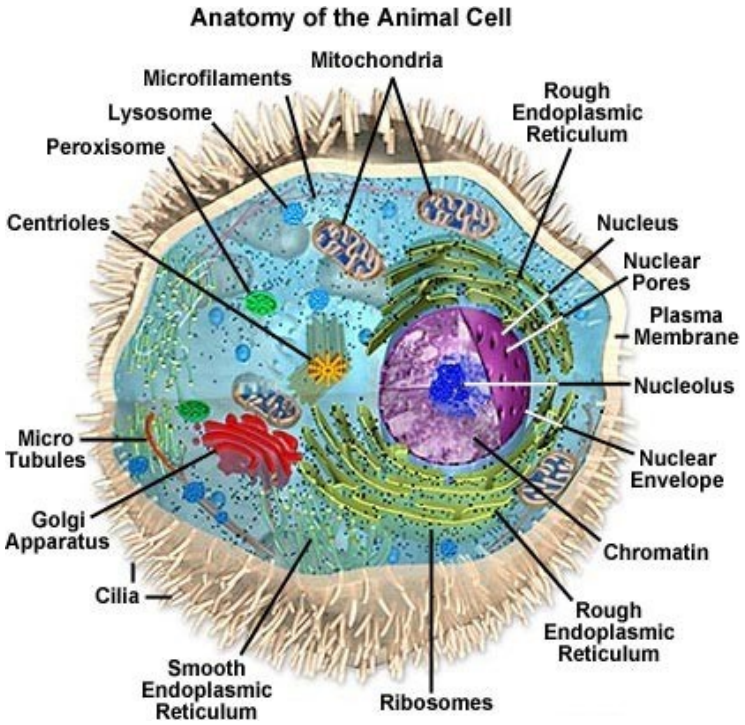
In one generation, the total population fitness changed from 34 to 37, thus improved by ~9%

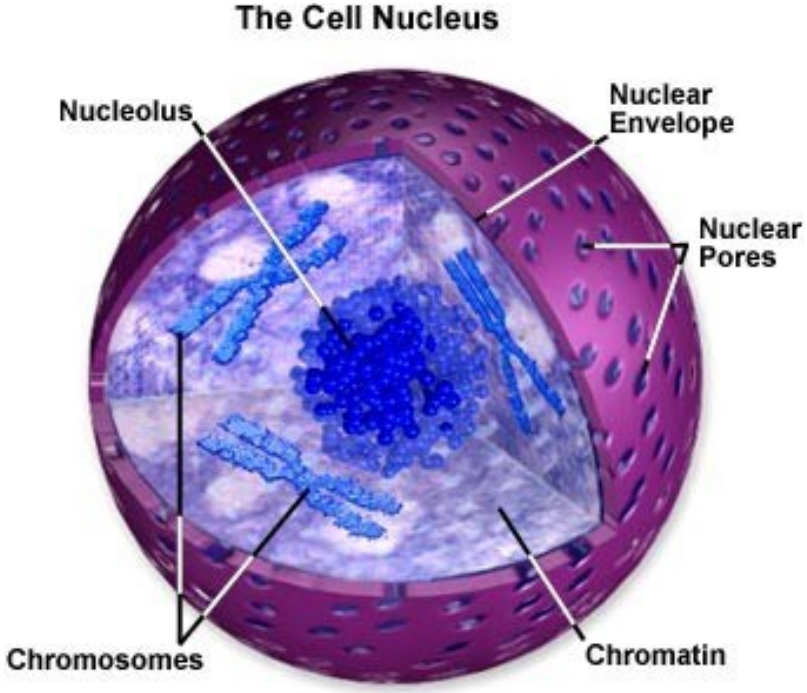
At this point, we go through the same process all over again, until a stopping criterion is met

Biological motivation

Biological Background
"The cell"

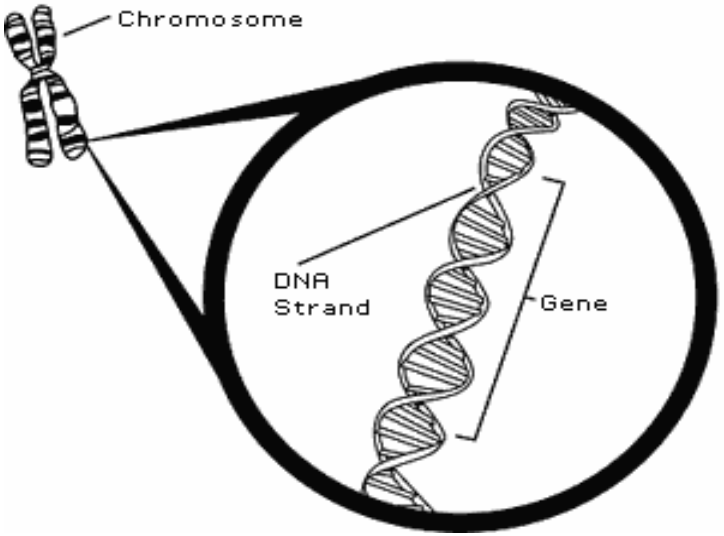
- Every animal cell is a complex of many small "factories" working together.
- The nucleus in the center of the cell.
- The nucleus contains the genetic information





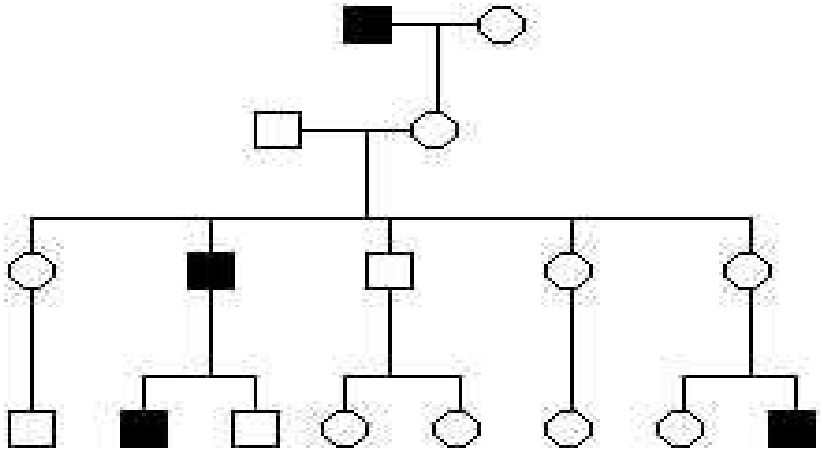
Biological Background
"Chromosomes"

- Genetic information is stored in the chromosomes
- Each chromosome is built of DNA
- Genes are encoded in the chromosomes
- Genes code for proteins
- Every gene has a unique position on the chromosome



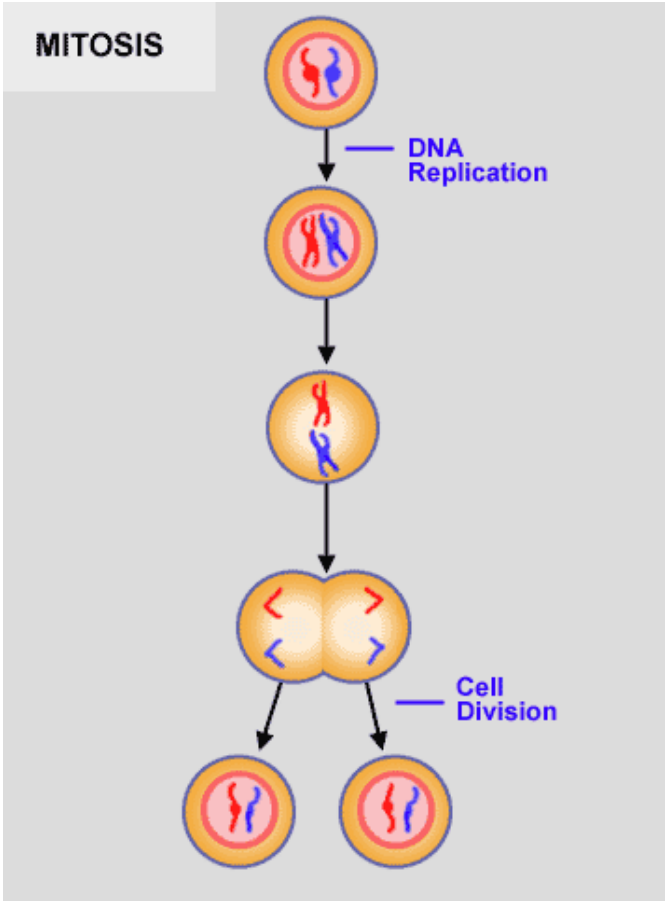
Biological Background: Genotype and phenotype

- The entire combination of genes is called genotype
- A genotype leads to a phenotype (eye color, height, disease predisposition)
- The phenotype is affected by changes to the underlying genetic code



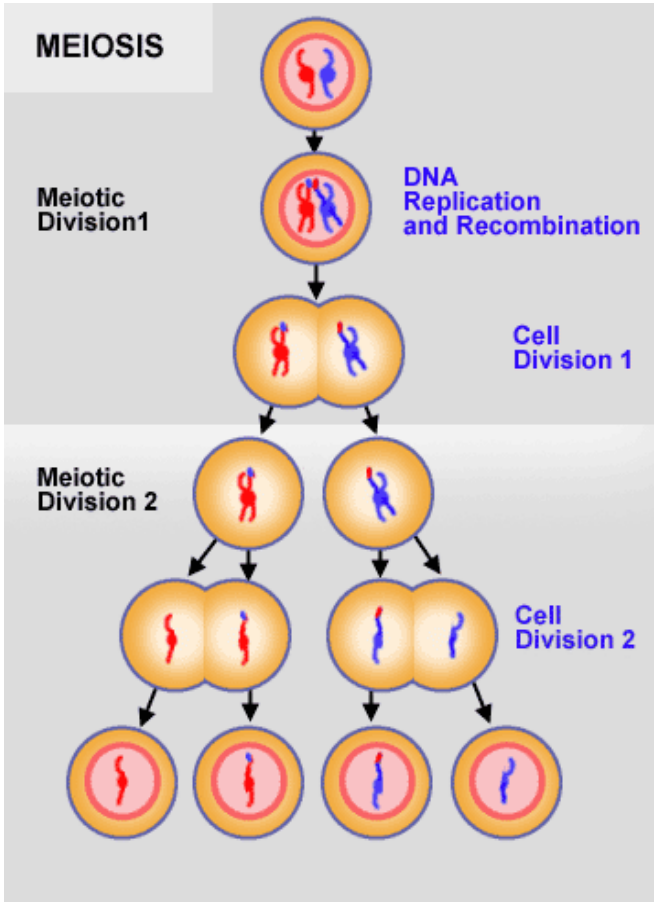
Biological Background
“Reproduction ”

- Reproduction of genetical information
  - ***Mitosis***
  - ***Meiosis***
- Mitosis is copying the same genetic information to new offspring: there is no exchange of information
- Mitosis is the normal way of growing of multicell structures, like organs.



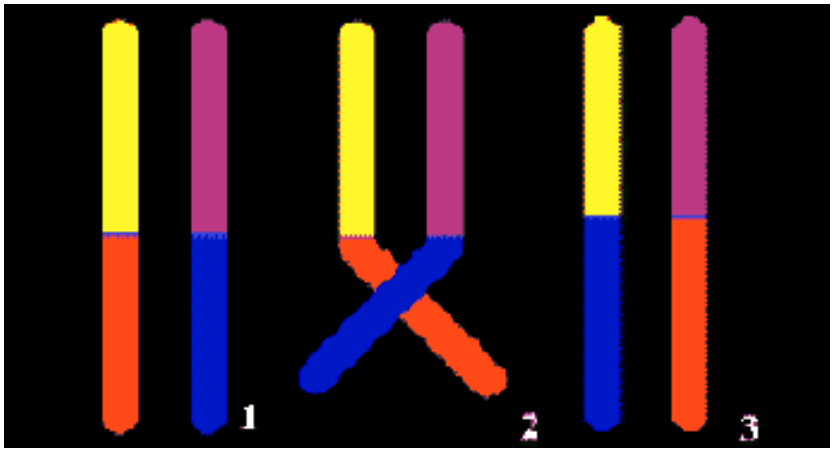
Biological Background
Reproduction

- Meiosis is the basis of sexual reproduction
- After meiotic division 2 gametes appear
- In reproduction two gametes conjugate to a zygote which will become the new individual
- Crossovers leads to new genotype



Mutations

- In any copying process errors can occur, so single (point) mutations ate pretty common.
- Other types of errors, including affecting longer regoins (either deletion, inversions, substitutions etc.) can also occur



“Natural selection”

- The origin of species: “Preservation of favourable variations and rejection of unfavourable variations.”
- There are more individuals born than can survive, so there is a continuous struggle for life.
- Individuals with an advantage have a greater chance for survive: so survival of the fittest.

GA Operators

- **Methods of representation**
- **Methods of selection**
- **Methods of Reproduction**

Common representation methods

- Binary strings.
- Arrays of integers (usually bound)
- Arrays of letters
- ….

Methods of Selection

There are many different strategies to select the individuals to be copied over into the next generation

Methods of Selection

- *Roulette-wheel selection*.
- *Elitist selection*.
- *Fitness-proportionate selection*.
- *Scaling selection*.
- *Rank 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.

Roulette wheel selection

No.	String	Fitness	% Of Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		1170	100.0

Other selection methods

- *Elitist selection*:

  Chose only the most fit members of each generation.
- *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

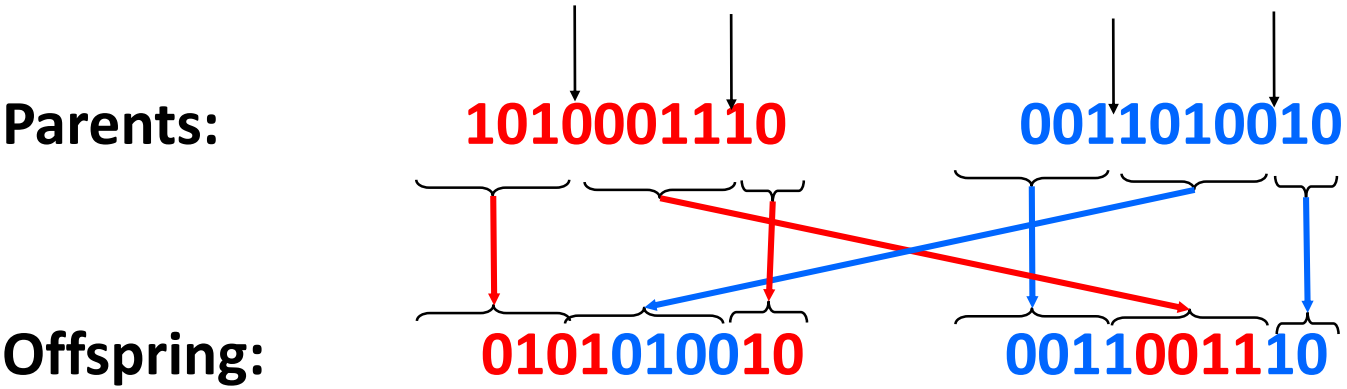
  2. The two parents are randomly recombined (crossed-over) to form new offsprings

Several possible crossover strategies

- Randomly select a single point for a crossover
- Multi point crossover
- Uniform crossover

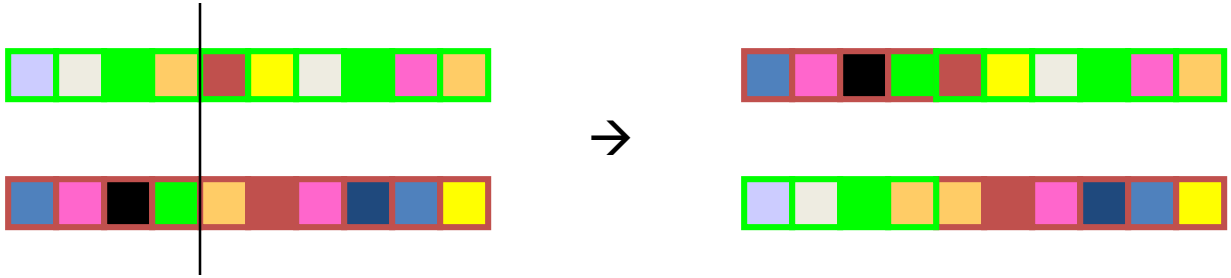
Two-point crossover

- Avoids cases where genes at the beginning and end of a chromosome are always split

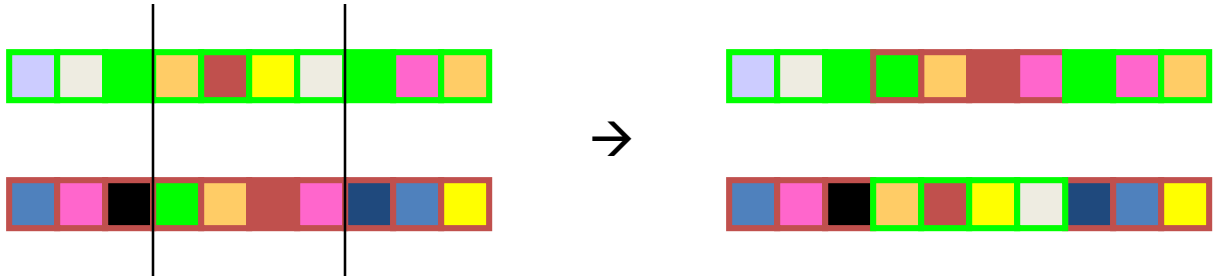


Crossover

- Single point crossover



- Two point crossover (Multi point crossover)



Uniform crossover

- A random subset is chosen
- The subset is taken from parent 1 and the other bits from parent 2.

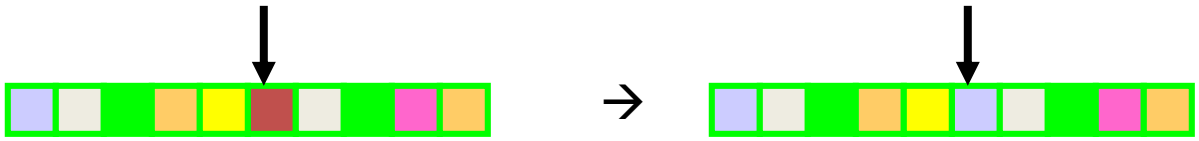
Subset: **BAABBAABBB** **(Randomly generated)**

Parents: **1010001110** **0011010010**

Offspring: **0011001010** **1010010110**

Methods of Reproduction: Mutations

– Generating new offspring from single parent



A (slightly more involved) 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 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

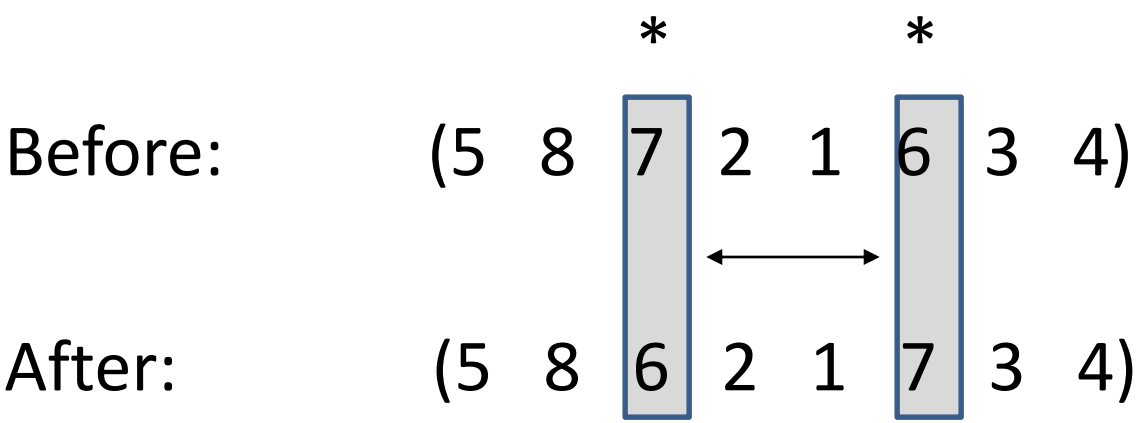
Crossover combines inversion and recombination:

			*			*		
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)

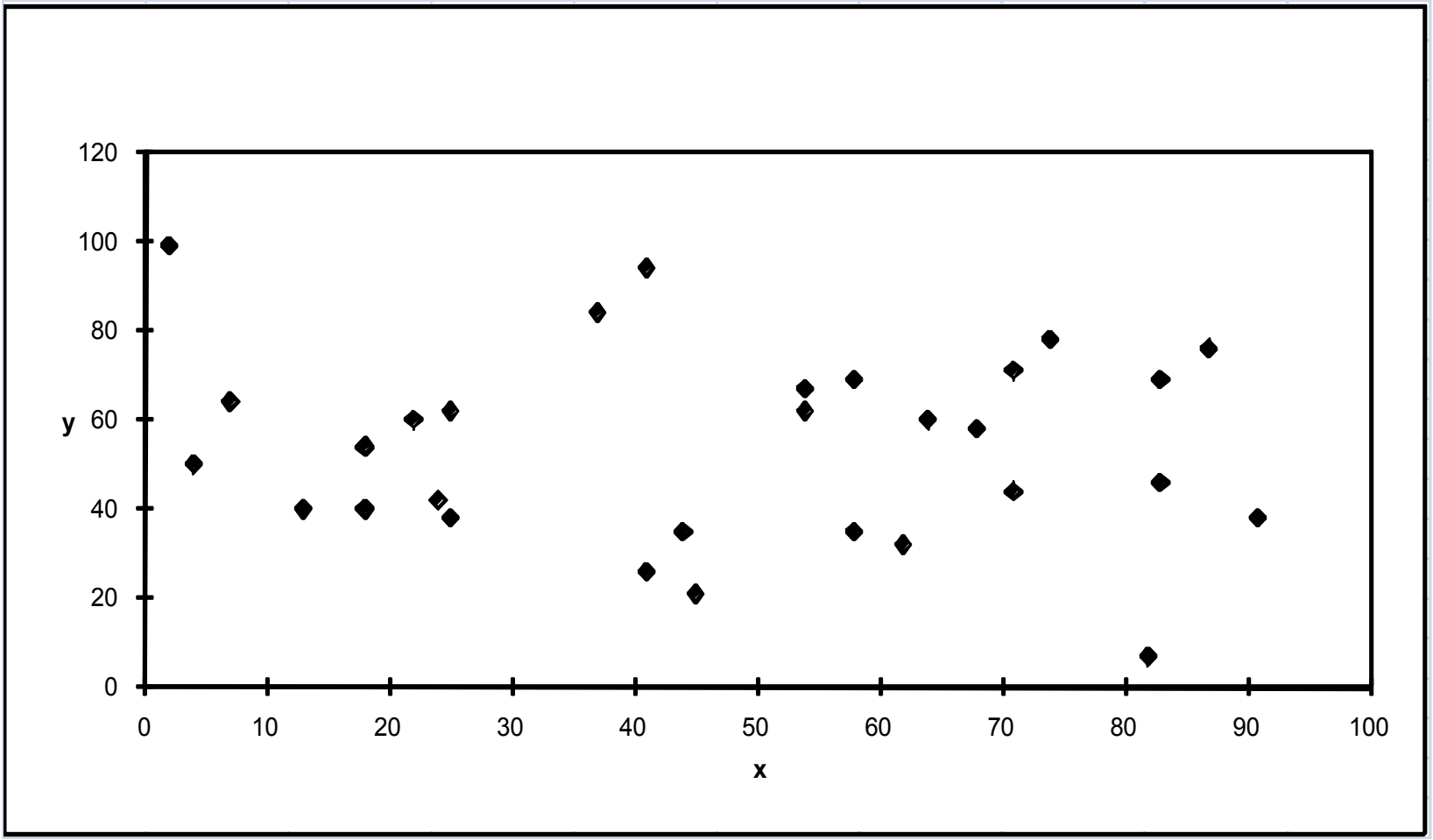
This operator is called the *Order1* crossover.

Mutation

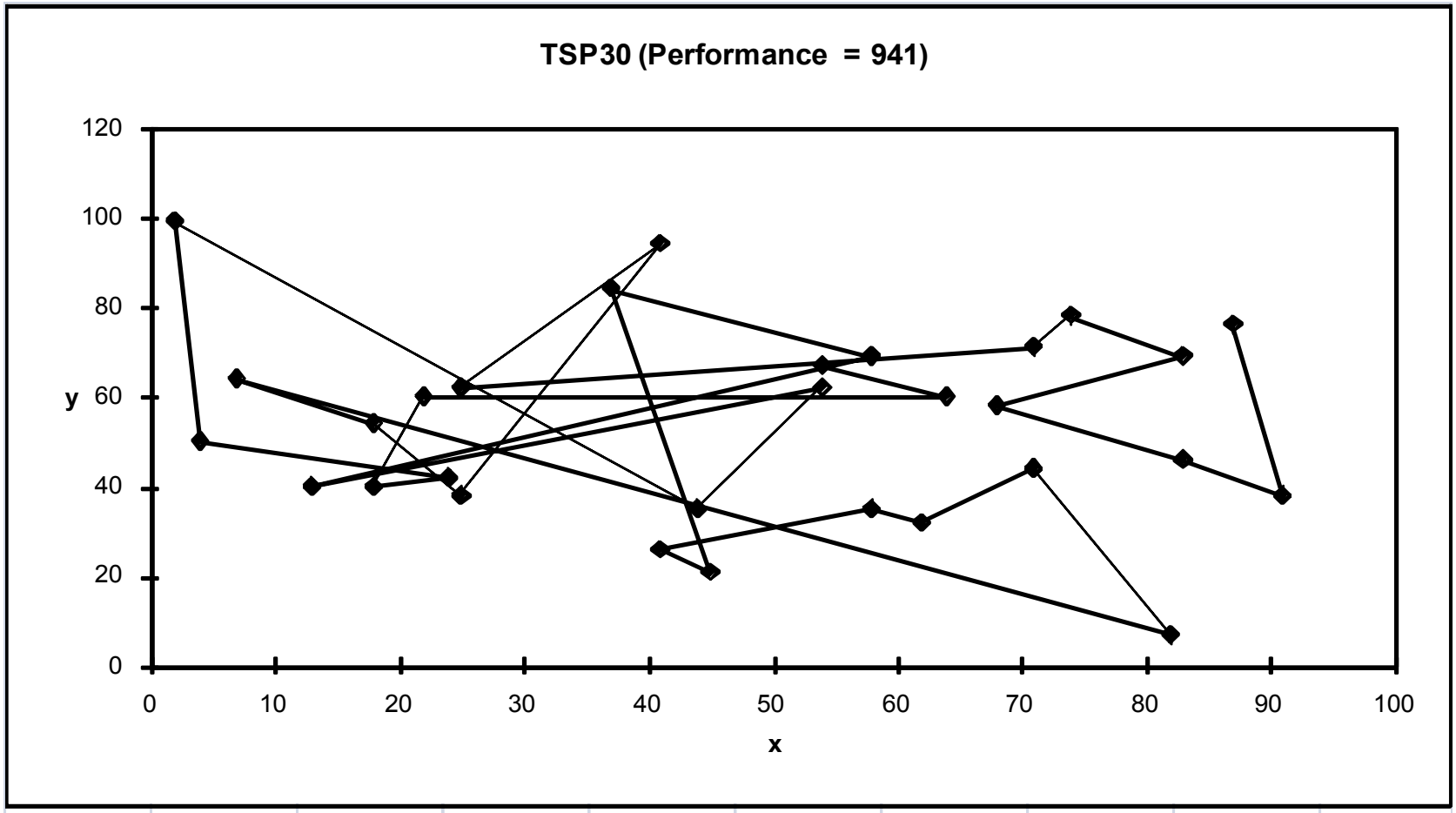
Mutation involves reordering of the list:



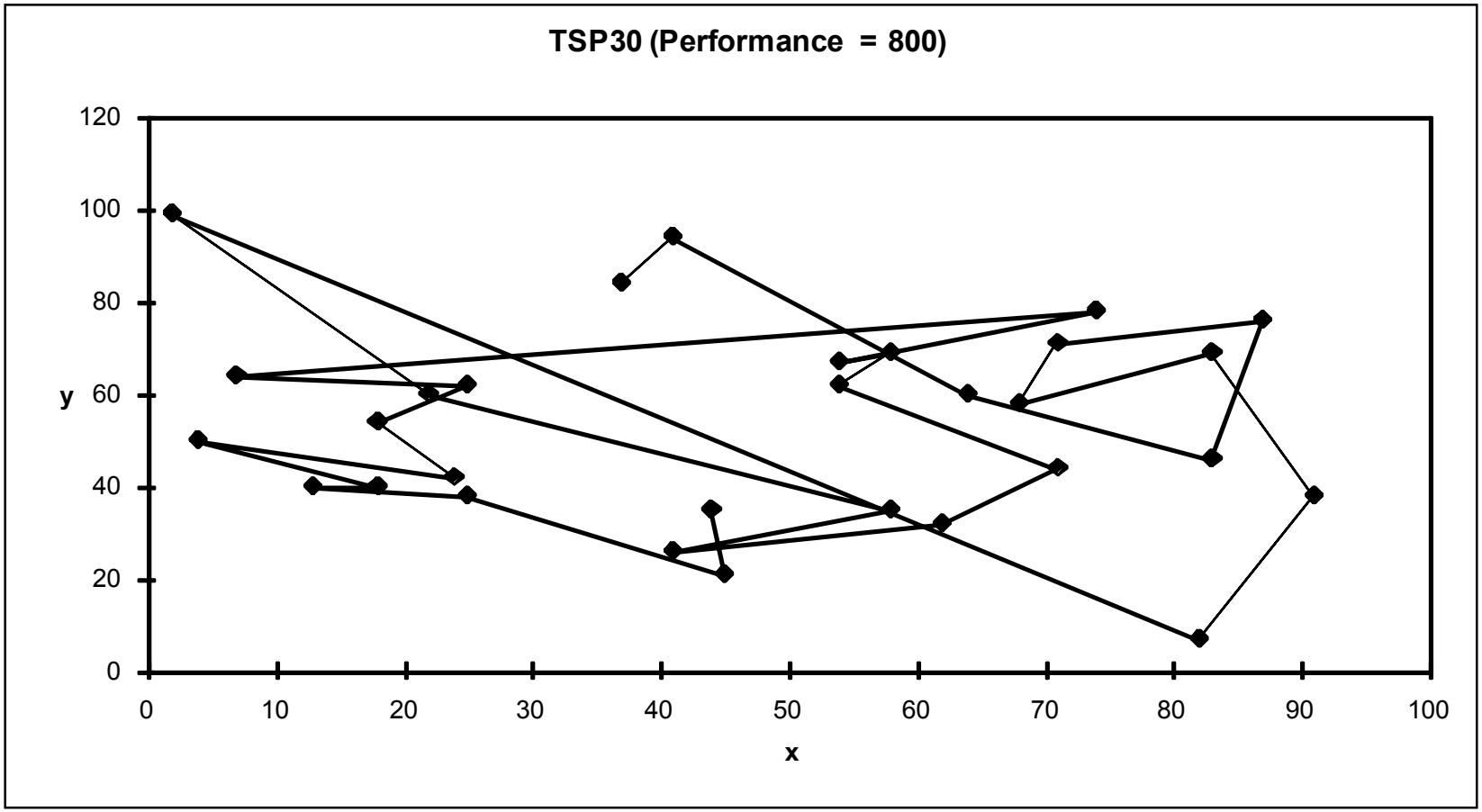
TSP Example: 30 Cities



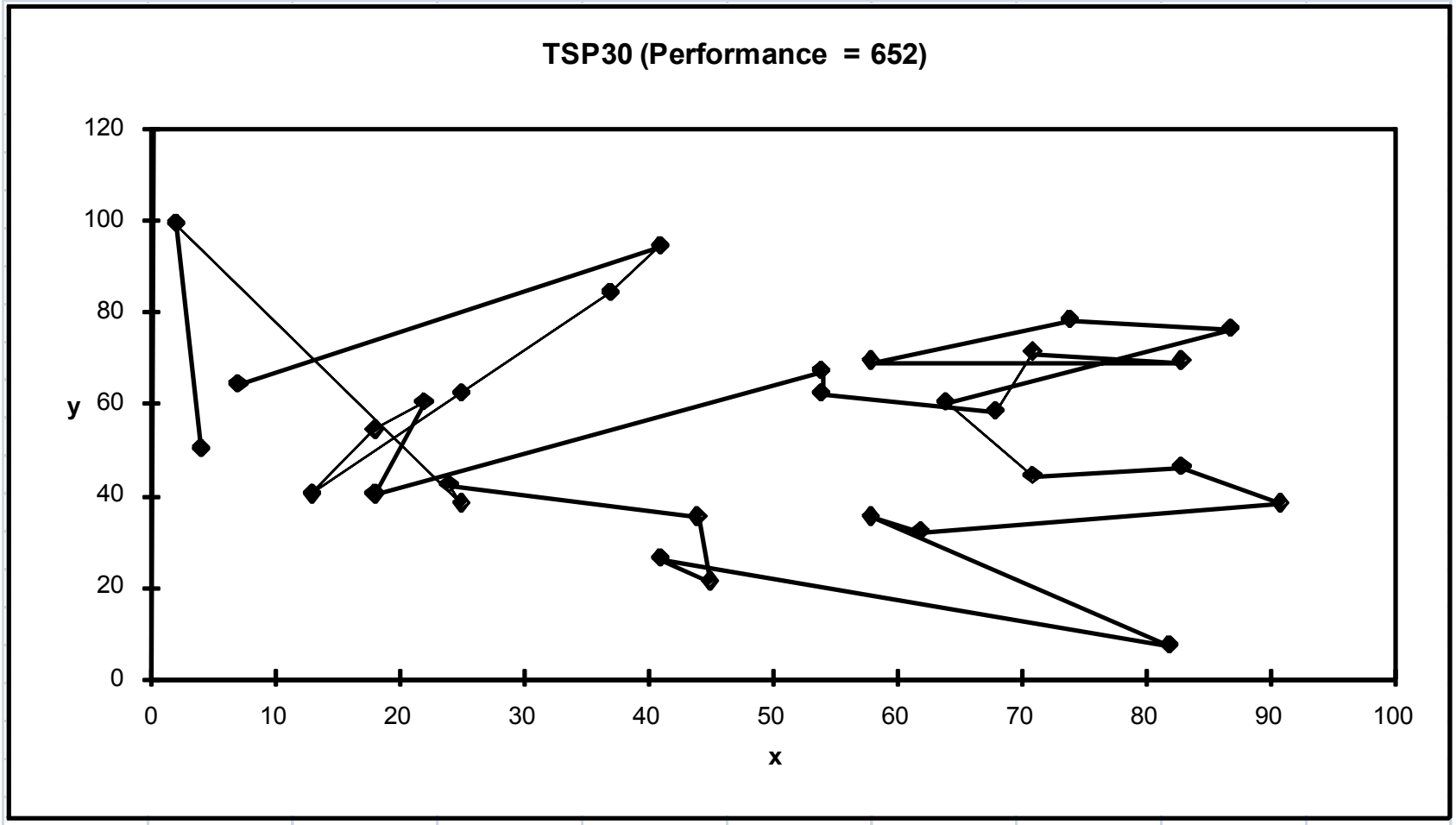
Solution $_{i}$ (Distance = 941)



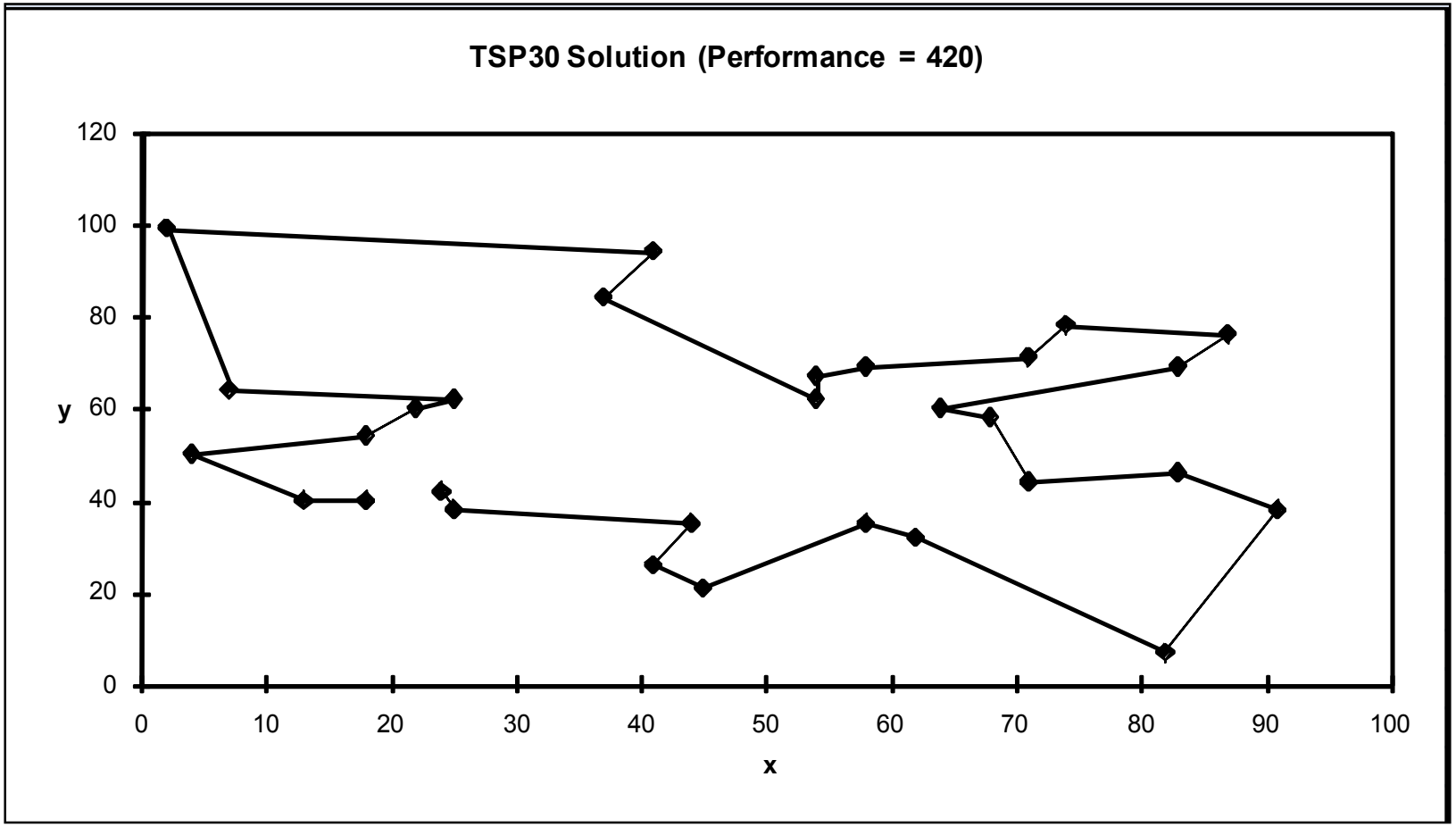
Solution $_j$(Distance = 800)



Solution $_k$(Distance = 652)



Best Solution (Distance = 420)



GA Applications

Domain	Application Type
Control	Gas pipeline, missile evasion
Design	Aircraft design, keyboard configuration, communication networks
Game playing	Poker, checkers
Security	Encryption and Decryption
Robotics	Trajectory planning

Benefits of Genetic Algorithms

- Concept is easy to understand
- Modular, separate from application
- Supports multi-objective optimization
- Always an answer; answer gets better with time.
- Easy to exploit previous or alternate solutions
- Flexible building blocks for hybrid applications.

Automatic design and manufacture of robotic lifeforms

Biological life is in control of its own means of reproduction... But this autonomy of design and manufacture has not yet been realized artificially… Here we report the results of a combined computational and experimental approach in which simple electromechanical systems are evolved through simulations from basic building blocks (bars, actuators and artificial neurons); the 'fittest' machines are then fabricated robotically... We thus achieve autonomy of design and construction using evolution in a 'limited universe' physical simulation coupled to automatic fabrication.

Nature 2000