

BIT 5201

A.I. as Representation and Search

Dr Matthew Montebello

CB 409

matthew.montebello@um.edu.mt



Lecture Schedule

1. Fuzzy Logic – FL 27/3
2. Genetic Algorithms – GA 3/4
3. Ant Colony Optimisation – ACO 8/5
4. Artificial Neural Networks – ANN 15/5

2008/9 Sem 2	Monday	Wednesday	Friday
MIT - Year 2 & 3 (Part 1)	BIT5208 OO Prog - Web App. Dr V.Nezval CB302 <u>Feb:</u> 2, 16, 23 <u>Mar:</u> 9, 16, 30 <u>Apr:</u> 20 <u>May:</u> 4, 11	BIT5205 / BIT5206 DB Impl./Digital Image Proc. C.Porter/Ing R Farrugia CB106 / Eng.PC Lab <u>Feb:</u> 4, 11, 25 <u>Mar:</u> 4, 18, 25 <u>Apr:</u> 22, 29 <u>May:</u> 13, 20	BIT5201 AI-Representation&Search Mr S.Spina/Matthew CB205 <u>Feb:</u> 13, 20 / - <u>Mar:</u> 6, 13 / 27 <u>Apr:</u> - / 3 <u>May:</u> - / 8, 15



Lecture / Topic Procedures

For each of the four topics the following is planned:

- Basic understanding
- Motivations
- History
- Applications
- Reading



Topic 1: Fuzzy Logic

What is it?

- Uncertainty!!!
- Not defining something exactly!!!
- A departure from classical two-valued sets and logic;
- A simple way to draw definite conclusions from information or input that is:
 - Vague;
 - Ambiguous;
 - Imprecise;
 - Noisy;
 - or Missing.

Fuzzy Logic basics ...

Traditional (crisp) logic categorically abides by the Aristotle's law (300BC) of the **Excluded Middle**.

Basically this law ... which is the principle foundation of Mathematics ... states that a statement can either be TRUE or FALSE.

$2B$ or $\neg 2B$

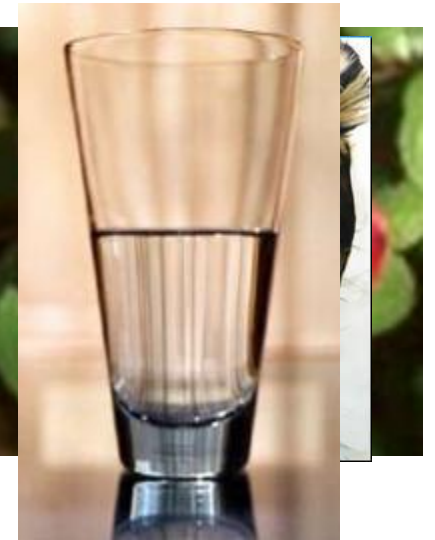
e.g. A Rose is either Red or not Red

e.g. What colour is the leopard?

e.g. Is the glass empty or full?

e.g. At what point short people become tall?

e.g. Is the lecture boring or not?



Proper Definition of FL

Fuzzy Logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth ... the truth values between “completely true” and “completely false”.

An approach to computing based on “degrees of truth” rather than the usual “true or false” ... boolean 1 or 0.

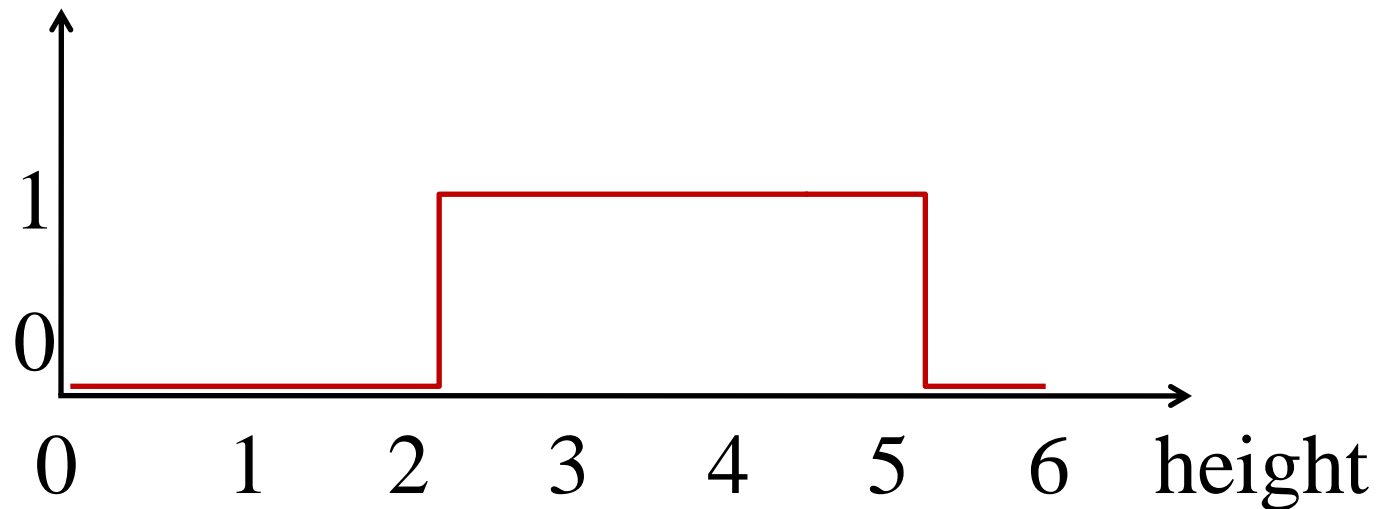
How can this work?

A mathematical technique for dealing with imprecise data and problems that have many solutions rather than one.

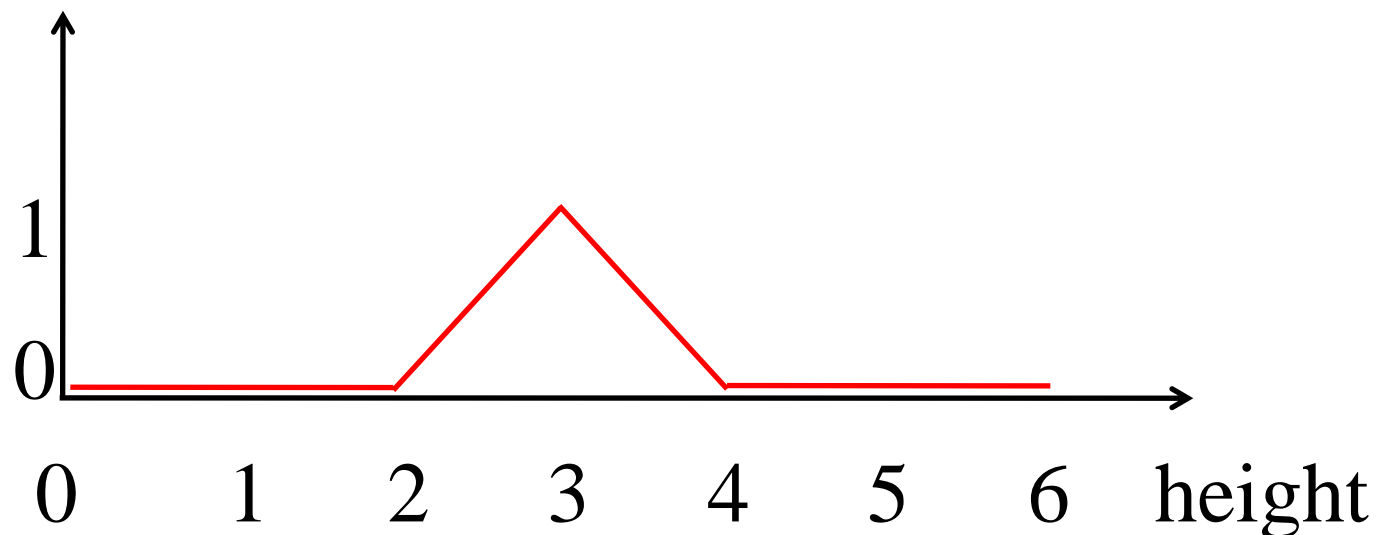
Although it is implemented in digital computers which ultimately make only Yes-No decisions, fuzzy logic works with ranges of values, solving problems in a way that more resembles human logic using a system that smoothes the edges.

Classical vs Fuzzy Logic

Crisp set: Set of heights from 2 to 5 feet



Fuzzy set: Set of heights in region around 3 feet



FL Motivations

- Alleviate difficulties in developing and analysing complex systems encountered by conventional mathematical tools.
- Observing that human reasoning can utilise concepts and knowledge that do not have well-defined, sharp boundaries.



More FL Motivations

Fuzziness is beneficial for

- Complex systems that are difficult or impossible to model;
- Simplified & reduced development cycle ;
- Ease of implementation;
- Can provide more "user-friendly" and efficient performance ;
- Systems controlled by human experts or systems that use human observations as inputs;
- Systems that naturally vague (behavioural and social sciences).



FL History

1964: Lotfi A. Zadeh, UC Berkeley,



Introduced the paper on fuzzy sets.

- Idea of grade of membership was born
- Sharp criticism from academia

- * Name!

- * Theory's emphasis on imprecision

- Waste of government funds!

1965-1975: Zadeh pursued foundations of fuzzy set theory;

- Multistage decision-making;
- Fuzzy similarity relations;
- Fuzzy restrictions;
- Linguistic hedges.

1970s: research groups were formed in Japan.



More recently ...

- 1974:** Mamdani, United Kingdom,
developed the first fuzzy logic
controller (steam engine control)
- 1982:** First commercial control system
using fuzzy logic
(cement kiln, Holmblad and Ostergaard)
- 1976-1987:** Industrial application of
fuzzy logic in Japan and Europe
- 1987:** Fuzzy Boom took off
- 2003:** First class on fuzzy logic is held
at Clarkson University
- Since then ...** Fuzzy Logic & Semantic Web

Fuzzy Operators

Operation

Definition

fuzzy AND (f-AND)

$$a \text{ f-AND } b = \min(a,b)$$

fuzzy OR (f-OR)

$$a \text{ f-OR } b = \max(a,b)$$

probability AND (p-AND)

$$a \text{ p-AND } b = a*b$$

probability OR (p-OR)

$$a \text{ p-OR } b = a+b-(a*b)$$

Unary NOT

$$\text{not } a = 1 - a$$

Algebraic laws

Apply the same

Hedges (example)

Linguistic Modifiers



FL Applications

Several heterogeneous applications like:

- Control (Robotics, Automation, Tracking, Consumer Electronics)
- Information Systems (DBMS, Info. Retrieval, Search Engines)
- Pattern Recognition (Image Processing, Machine Vision)
- Decision Support (Adaptive HMI, Sensor Fusion)
- Typical examples include:
 - ABS brakes
 - Expert Systems
 - Control units
 - Trains
 - Video Cameras
 - Automatic Transmission
 - Cement kilns
 - Home appliances (Washing machines / Fridges)



Application Examples

Image Stabilisation

- Detects hand jittering
- Works out direction of hand movement
- If same as direction of the all the motion vectors that are almost parallel with a small time differential

Aerospace

- Altitude control of spacecraft
- Satellite altitude control
- Flow and mixture regulation in aircraft deicing

Automotive

- Trainable fuzzy systems for idle speed control
- Shift scheduling method for automatic transmission
- Intelligent highway systems & traffic control
- Improving efficiency of automatic transmissions



More Application Examples

Business

- Decision-making support systems
- Personnel evaluation in a large company
- Data mining systems

Chemical Industry

- Control of pH, drying, chemical distillation processes
- polymer extrusion production
- coke oven gas cooling plant

Defense

- Underwater target recognition
- Automatic target recognition of thermal infrared images
- naval decision support aids
- control of a hypervelocity interceptor
- fuzzy set modeling of NATO decision making.



Even More Application Examples

Electronics

- Control of automatic exposure in video cameras
- Humidity in a clean room
- Air conditioning systems, Washing machine timing
- Microwave ovens, Vacuum cleaners.

Financial

- Banknote transfer control, Fund management
- Stock market predictions

Industrial

- Cement kiln controls (dating back to 1982)
- Heat exchanger control
- Activated sludge wastewater treatment process control
- Water purification plant control
- Quantitative pattern analysis for industrial quality assur.
- control of constraint satisfaction in structural design

Final Application Examples

Marine – Autopilot for ships, optimal route selection,

- Control of autonomous underwater vehicles, steering

Medical – Medical diagnostic support system e.g. radiology

- Control of arterial pressure during anesthesia
- Inference diagnosis of diabetes and prostate cancer

Mining and Metal Processing

- Sinter plant control, decision making in metal forming

Robotics – Fuzzy control for flexible-link manipulators

- Robot arm control

Securities – DSS for securities trading

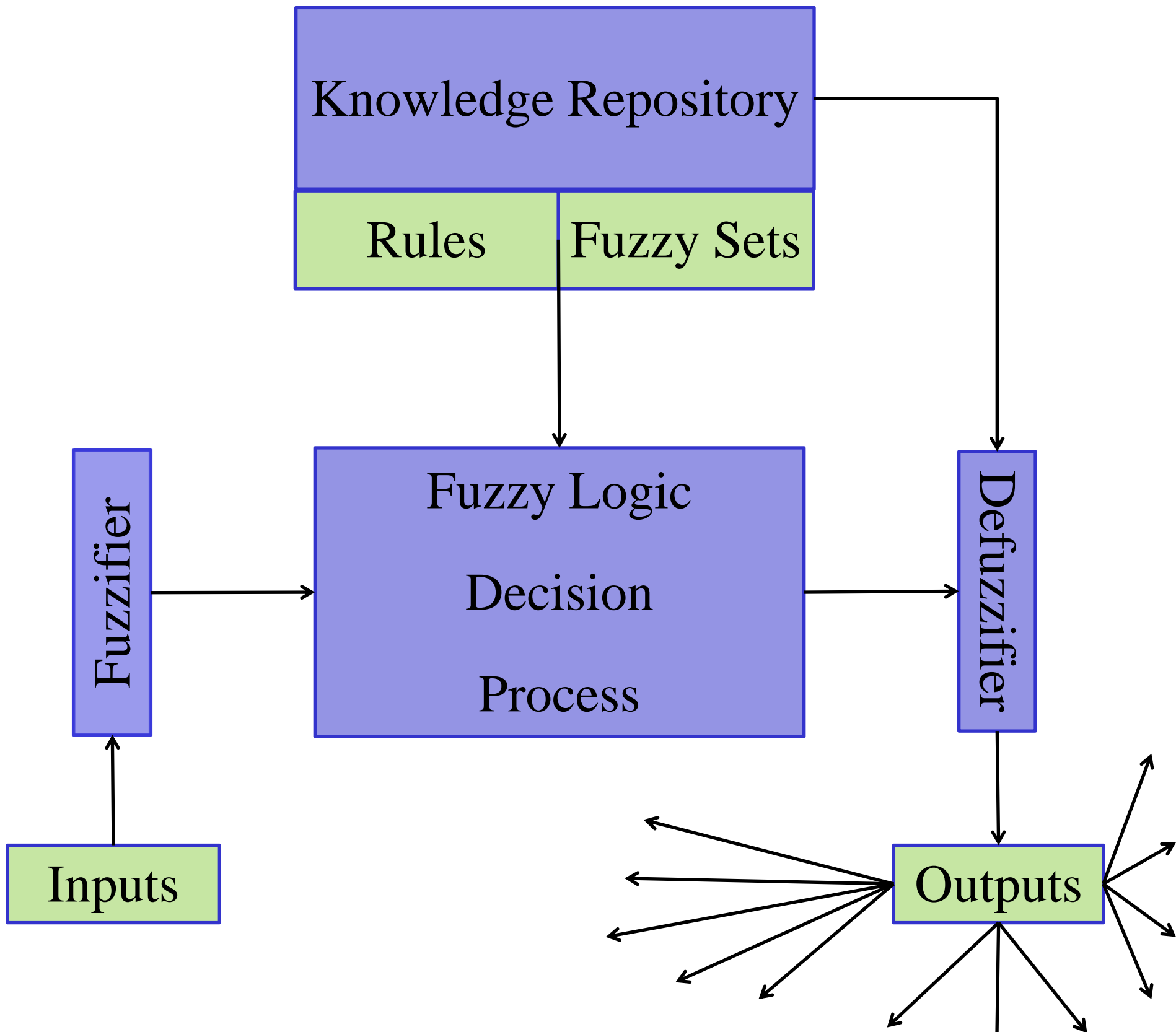
Signal Processing and Telecommunications

- Adaptive filter for nonlinear channel equalization control of broadband noise

Transportation

- Automatic underground train operation
- Train schedule control
- Railway acceleration, braking, and stopping





Links

- Important Topic Reading

<http://staff.um.edu.mt/mmon1/lectures/bit5201/FL.pdf>

<http://staff.um.edu.mt/mmon1/lectures/bit5201/FLintro.pdf>

- Very good introductory tutorial video

http://videlectures.net/acai05_berthold_fl/

- Gullu's Notes

<http://staff.um.edu.mt/kgui1/MIT/IntroFL.pdf>

- These notes

<http://staff.um.edu.mt/mmon1/lectures/bit5201/notes.pdf>



Hedges

- Hedges are linguistic modifiers on membership functions.
- Consider the membership function $\text{Is Tall}(x)$.
- A hedge would be:
 - Is **Very** Tall(x).
 - Is **Sort Of** Tall(x).
 - Is **Quite** Tall(x).
 - Is **Slightly** Tall(x).
 - Is **Somewhat** Tall(x).

Mathematical Representation of Hedges

- Consider a membership function $\mu(x)$:
 - A little = $[\mu(x)]^{1.3}$
 - Slightly = $[\mu(x)]^{1.7}$
 - Very = $[\mu(x)]^2$
 - Extremely = $[\mu(x)]^3$
 - More or Less, Somewhat = $\text{SQRT}[\mu(x)]$



Mathematical Representation of Hedges

- From the previous membership function for IsTall, consider $\text{IsTall}(\text{John}) = \text{IsTall}(1.75)$.
- This gives a fuzzy truth value of 0.8333.
- $\text{IsSomewhatTall} = \text{SQRT}(0.8333) = 0.913$.
- $\text{IsVeryTall} = 0.8333^2 = 0.694$.
- Etc...

[Back to slides ...](#)

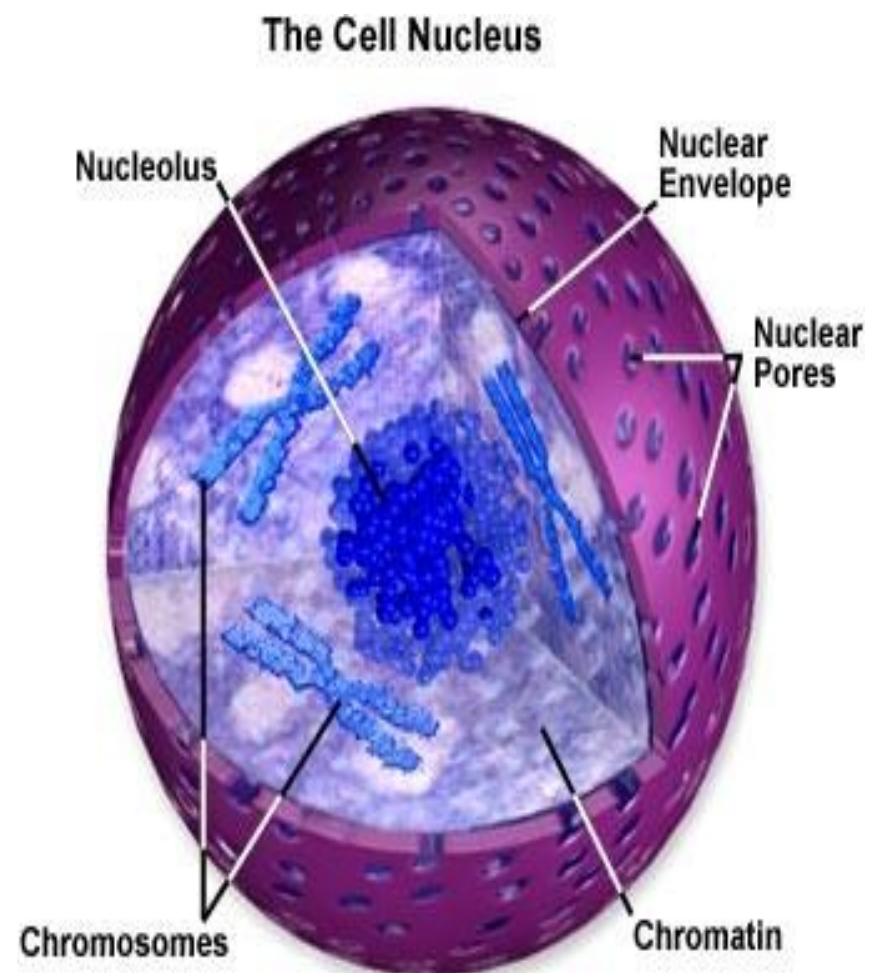
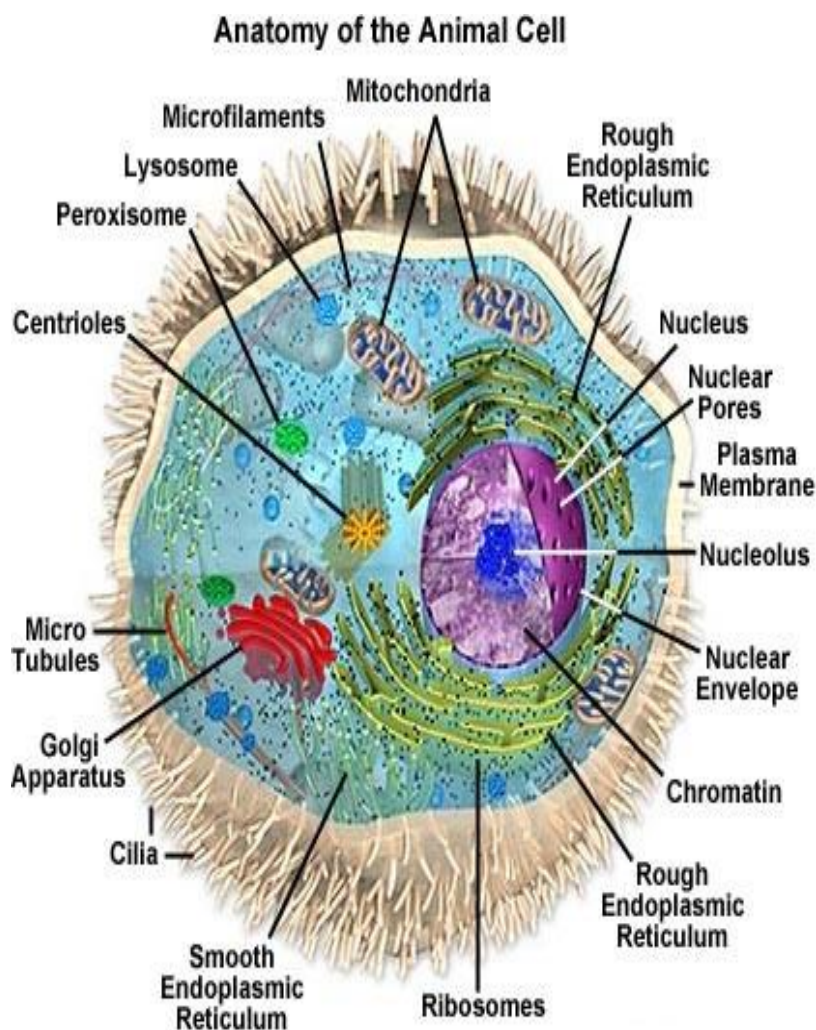
Topic 2: Genetic Algorithms

- What is a Genetic Algorithm (GA)?
- Biological background
 - Origin of species
 - Natural selection
- Genetic Algorithm details
 - Search space
 - Basic algorithm
 - Coding
 - Methods
 - Examples
 - Possibilities



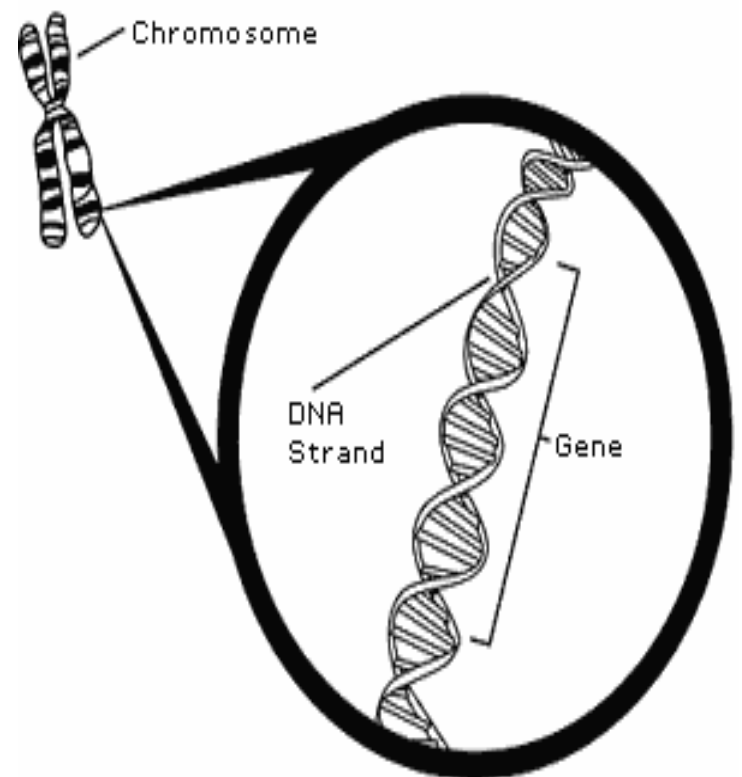
Biological detail – The cell

- Every animal cell is a complex of many small “factories” working together
- The center of this all is the cell nucleus
- The nucleus contains the genetic information



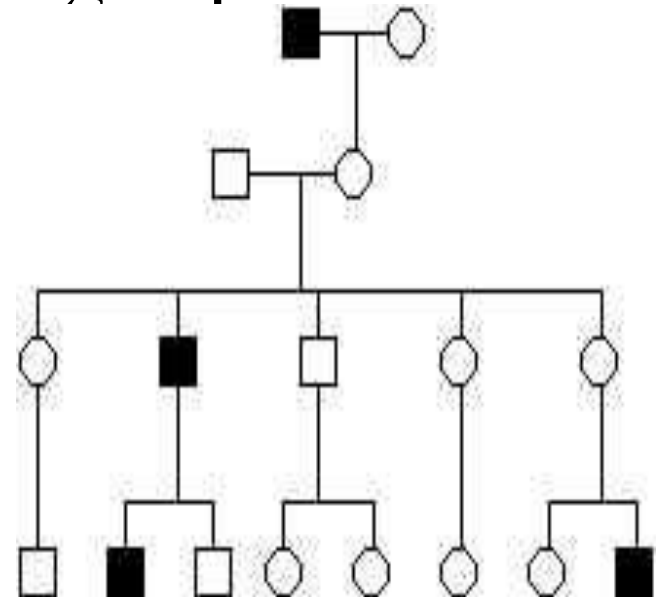
Biological detail – Chromosomes

- Genetic info is stored in the chromosomes
- Each chromosome is build of DNA
- Chromosomes in humans form pairs
- There are 23 pairs
- The chromosome is divided in parts: genes
- Genes code for properties
-



Biological detail – Genetics

- Entire combination of genes is called genotype
- A genotype develops to a phenotype
- Alleles can be either dominant or recessive
- Dominant alleles will always express from the genotype to the phenotype
- Recessive alleles can survive in the population for many generations, without being expressed.

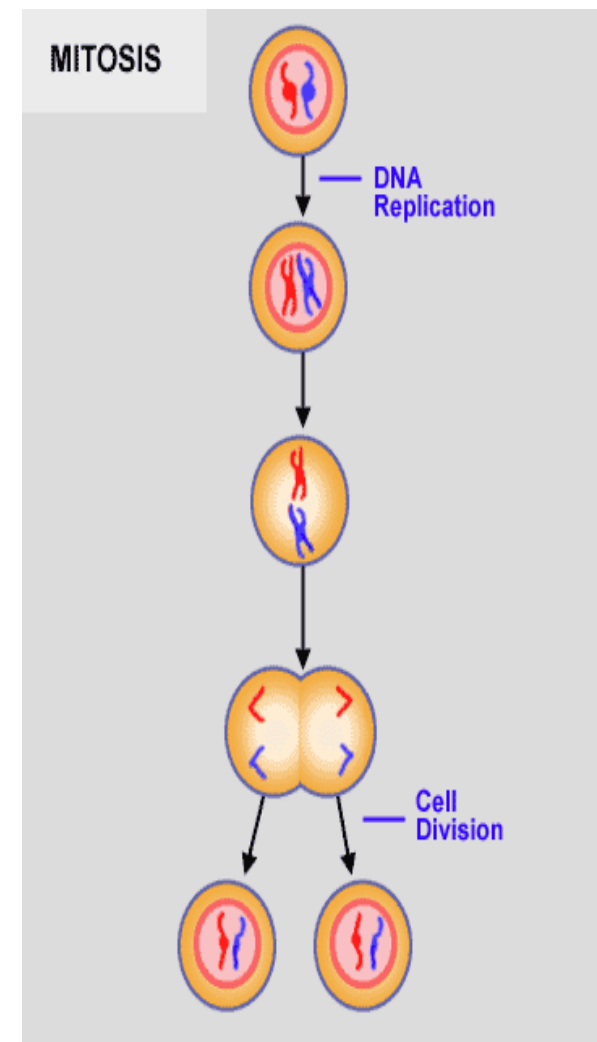


Biological detail – Reproduction 1

- Reproduction of genetical information

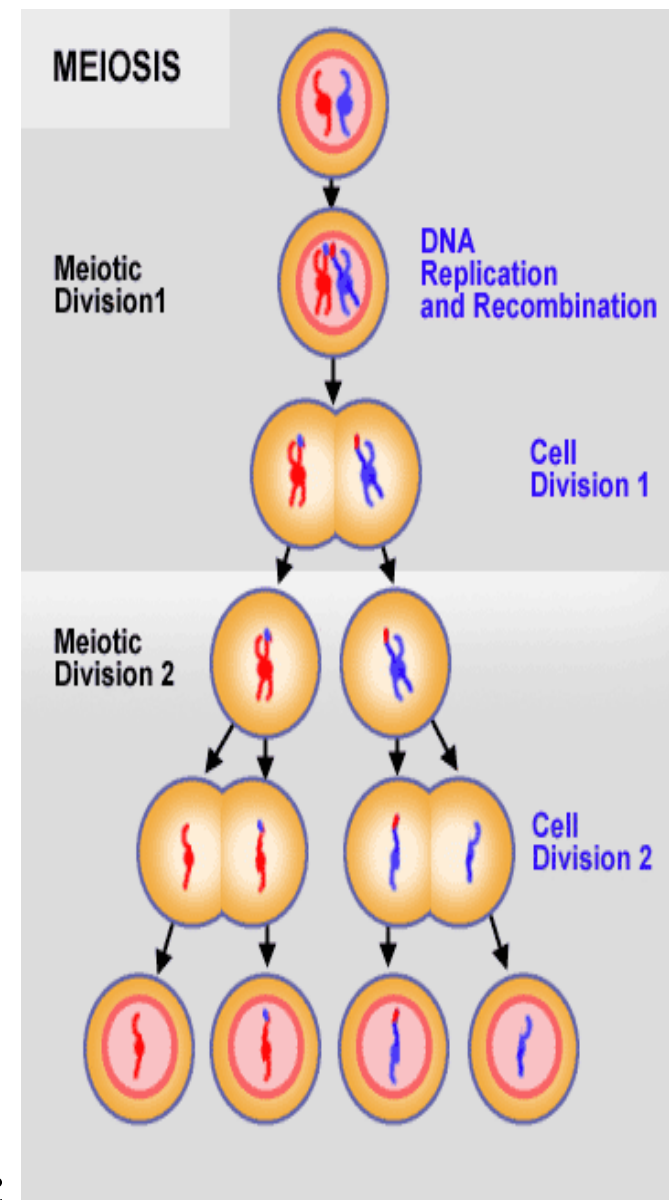
→ Mitosis & Meiosis

- 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.



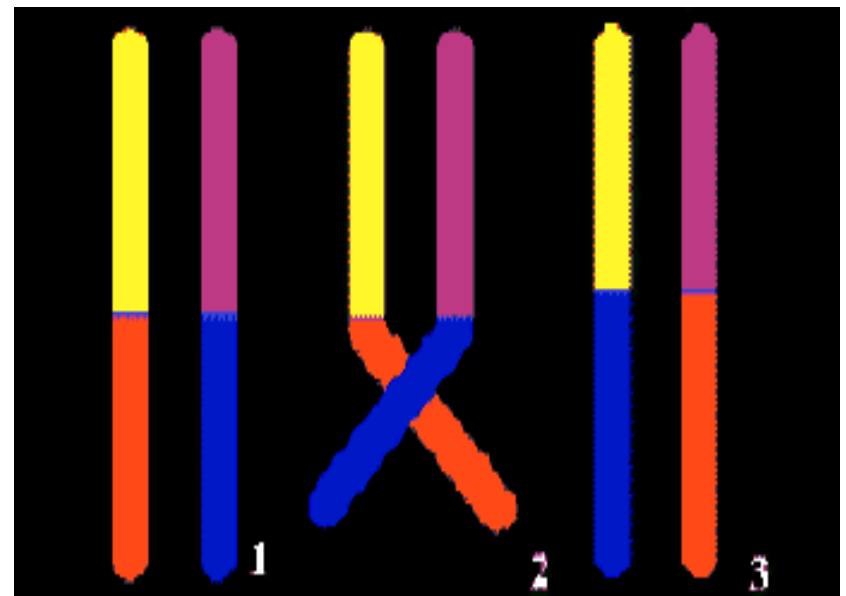
Biological detail – Reproduction 2

- Meiosis is the basis of sexual reproduction
- After meiotic division 2 gametes appear in the process
- In reproduction two gametes conjugate to a zygote which will become the new individual
- Hence genetic information is shared between the parents in order to create new offsprings



Biological detail – Reproduction 3

- During reproduction “errors” occur
- Due to these “errors” genetic variation exists
- Most important “errors” are:
 - Recombination (cross-over)
 - Mutation



Biological detail – Natural selection 2

- 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: survival of the fittest.

Summarising Evolution

- Organisms (animals or plants) produce a number of offspring which are almost but not entirely, like themselves
 - Variations may be genetic random changes
- Genetic Algorithms (GAs) use this same process to “evolve” better programs
 - Variations are introduced from each parent to produce offspring
 - The fittest programs are more likely to survive
 - Over time successive generations become better and better adapted

Genetic Algorithms in a nutshell

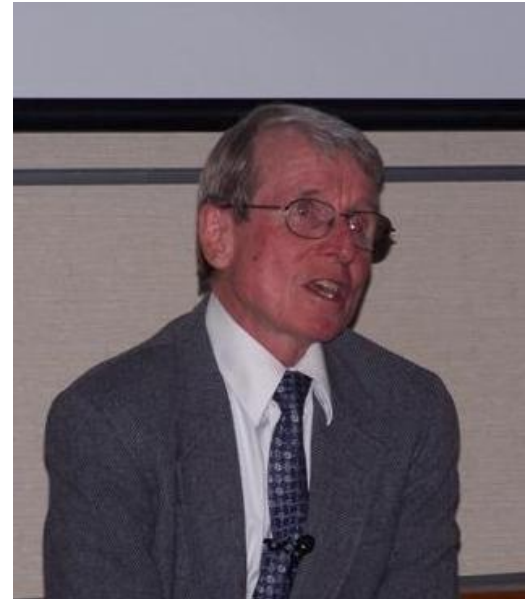
- Premise
 - Evolution worked once (it produced us!), it might work again
- Basics
 - Pool of solutions
 - Mate existing solutions to produce new solutions
 - Mutate current solutions for long-term diversity
 - Cull population



GAs History

Developed by John Holland 1975

- To understand the adaptive processes of natural systems;
- To design artificial systems software that retains the robustness of natural systems.



Made popular by John Koza 1992



How GA are Different than Traditional Search Methods

- GAs work with a coding of the parameter set, not the parameters themselves.
- GAs search from a population of points, not a single point.
- GAs use payoff information, not derivatives or auxiliary knowledge.
- GAs use probabilistic transition rules, not deterministic rules.

The Basic Genetic Algorithm

1. **[Start]** Generate random population of n chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness $f(x)$ of each chromosome x in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
 1. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
 2. **[Crossover]** With a crossover probability cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
 3. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
 4. **[Accepting]** Place new offspring in the new population
4. **[Replace]** Use new generated population for a further run of the algorithm
5. **[Test]** If the end condition is satisfied, stop, and return the best solution in current population
6. **[Loop]** Go to step 2



Genetic Algorithm – Search space

- Most often one is looking for the best solution in a specific subset of solutions
- This subset is called the search space (or state space)
- Every point in the search space is a possible solution
- Therefore every point has a fitness value, depending on the problem definition
- GA's are used to search the search space for the best solution, e.g. a minimum
- Difficulties are the local minima and the starting point of the search

GA terminology

- In the spirit – but not the letter – of biology
 - GA chromosomes are strings of genes
 - Each gene has a number of alleles; i.e., settings
 - Each chromosome is an encoding of a solution to a problem
 - A population of such chromosomes is operated on by a GA



Encoding

- A data structure for representing candidate solutions
 - Often takes the form of a bit string
 - Usually has internal structure; i.e., different parts of the string represent different aspects of the solution)

Crossover

- Mimics biological recombination
 - Some portion of genetic material is swapped between chromosomes
 - Typically the swapping produces an offspring
- Mechanism for the dissemination of “building blocks” (schemas)

Mutation

- Selects a random locus – gene location – with some probability and alters the allele at that locus
- The intuitive mechanism for the preservation of variety in the population

Fitness

- A measure of the goodness of the organism
 - Expressed as the probability that the organism will live another cycle (generation)
 - Basis for the natural selection simulation
 - Organisms are selected to mate with probabilities proportional to their fitness
 - Probabilistically better solutions have a better chance of conferring their building blocks to the next generation
- BIT blocks A. The as Representation (and Search



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



TSP 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)

TSP Crossover

Crossover combines inversion and recombination:

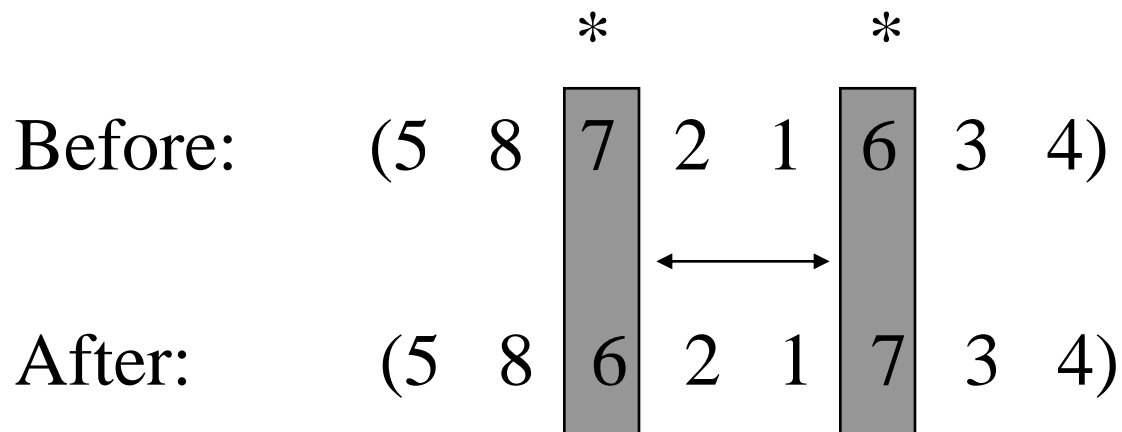
		*		*			
Parent1	(3	5	7	2	1	6	4 8)
Parent2	(2	5	7	6	8	1	3 4)
Child	(5	8	7	2	1	6	3 4)

This operator is called the *Order1* crossover.

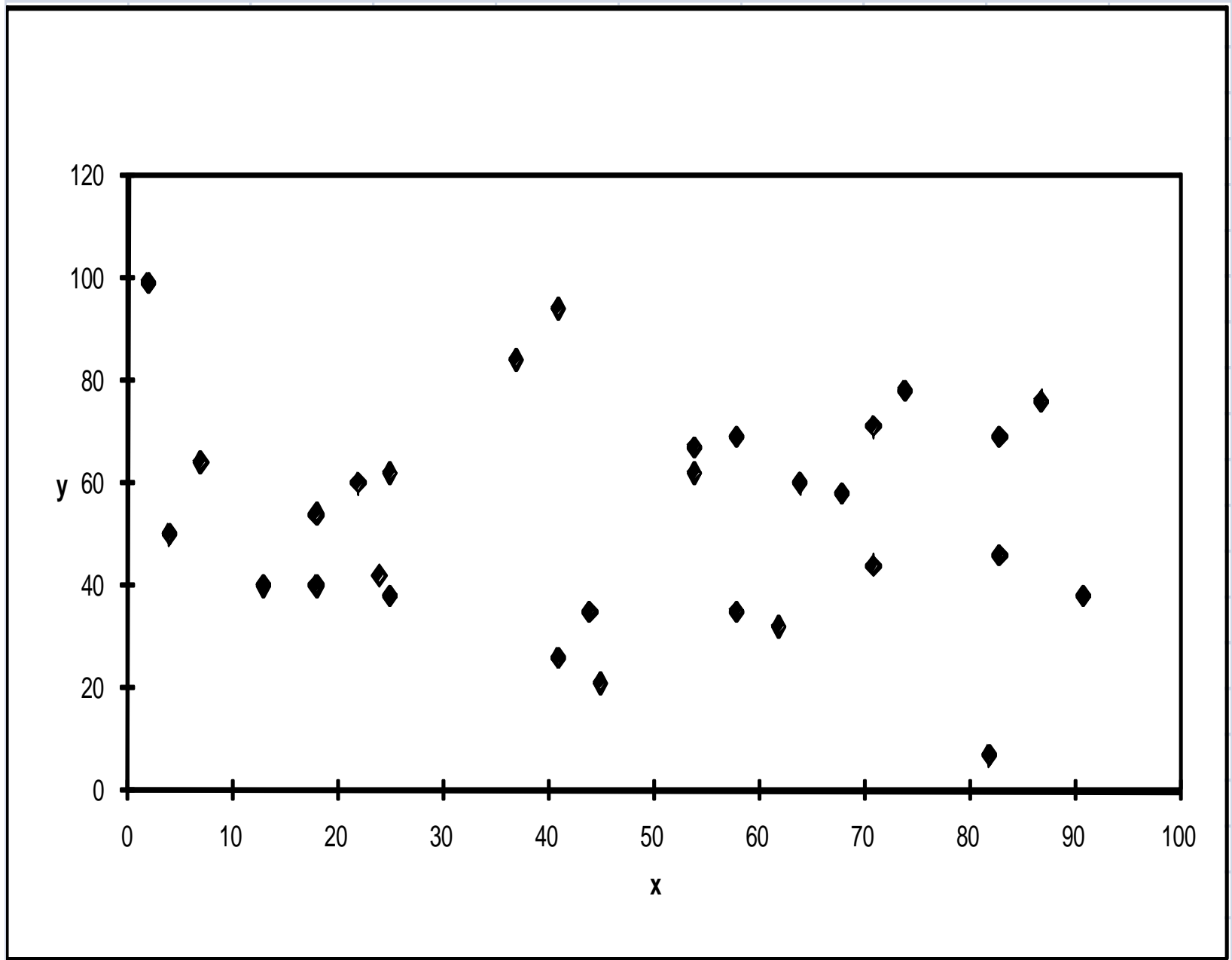


TSP Mutation

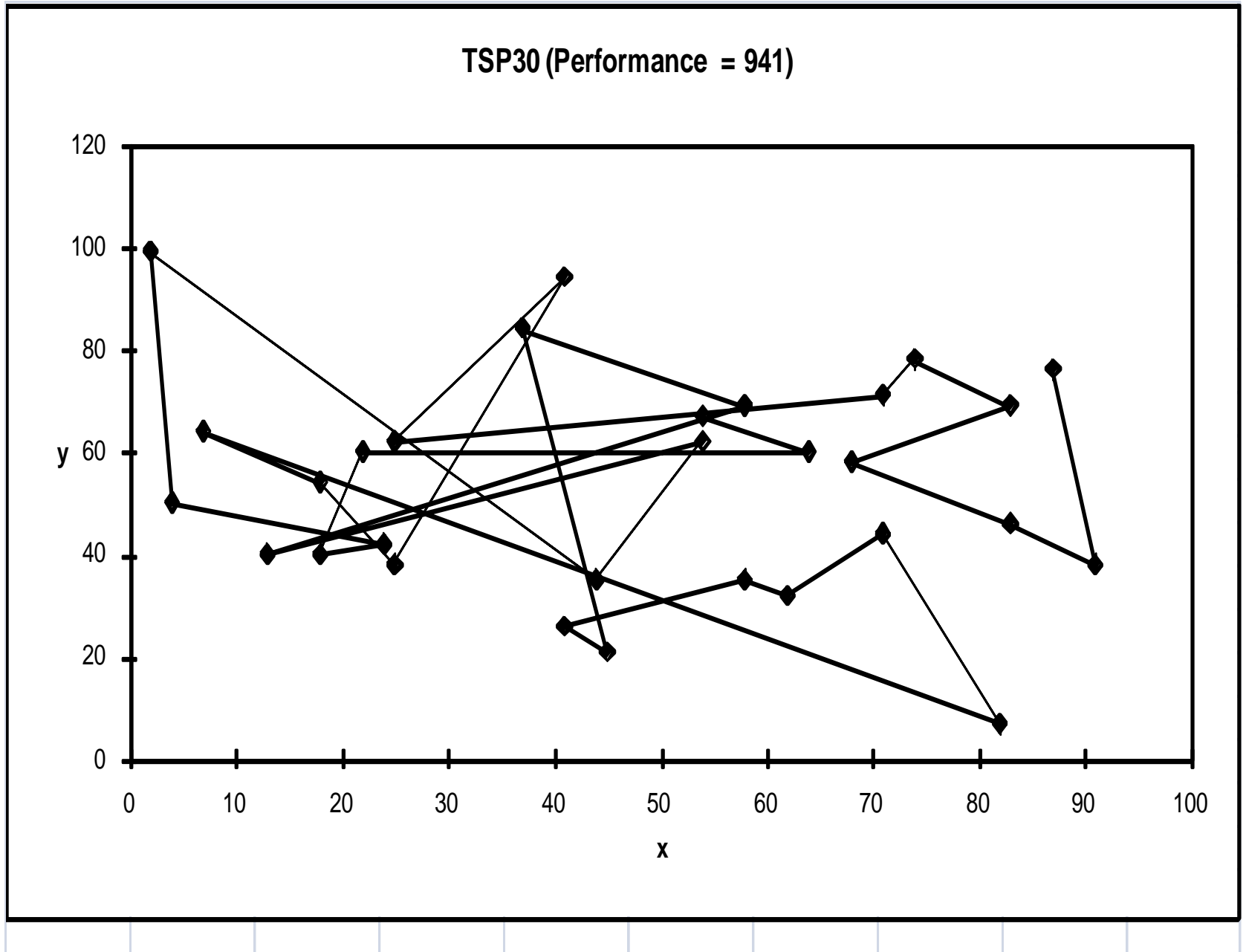
Mutation involves reordering of the list:



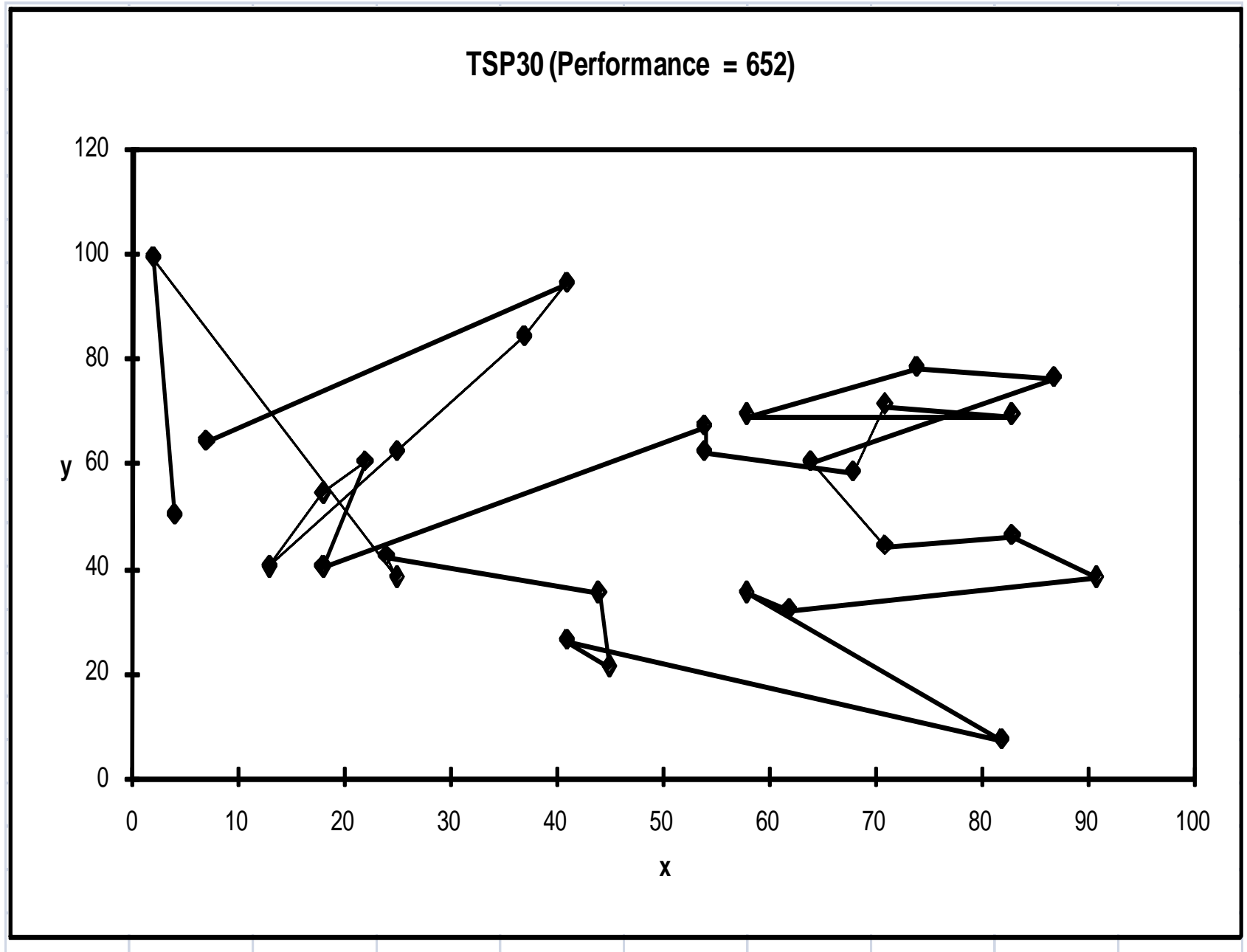
TSP Example: 30 Cities



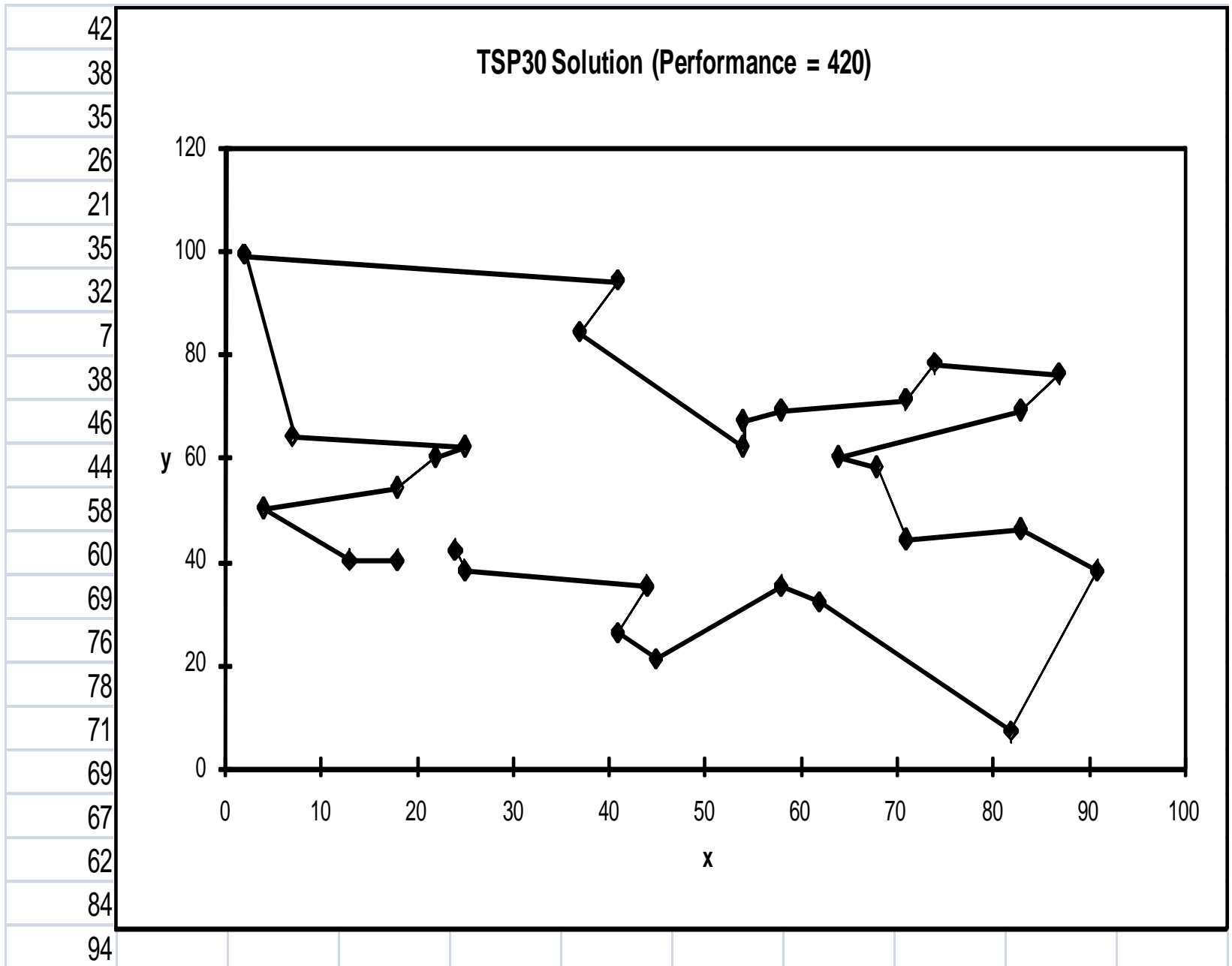
Solution_i (Distance = 941)



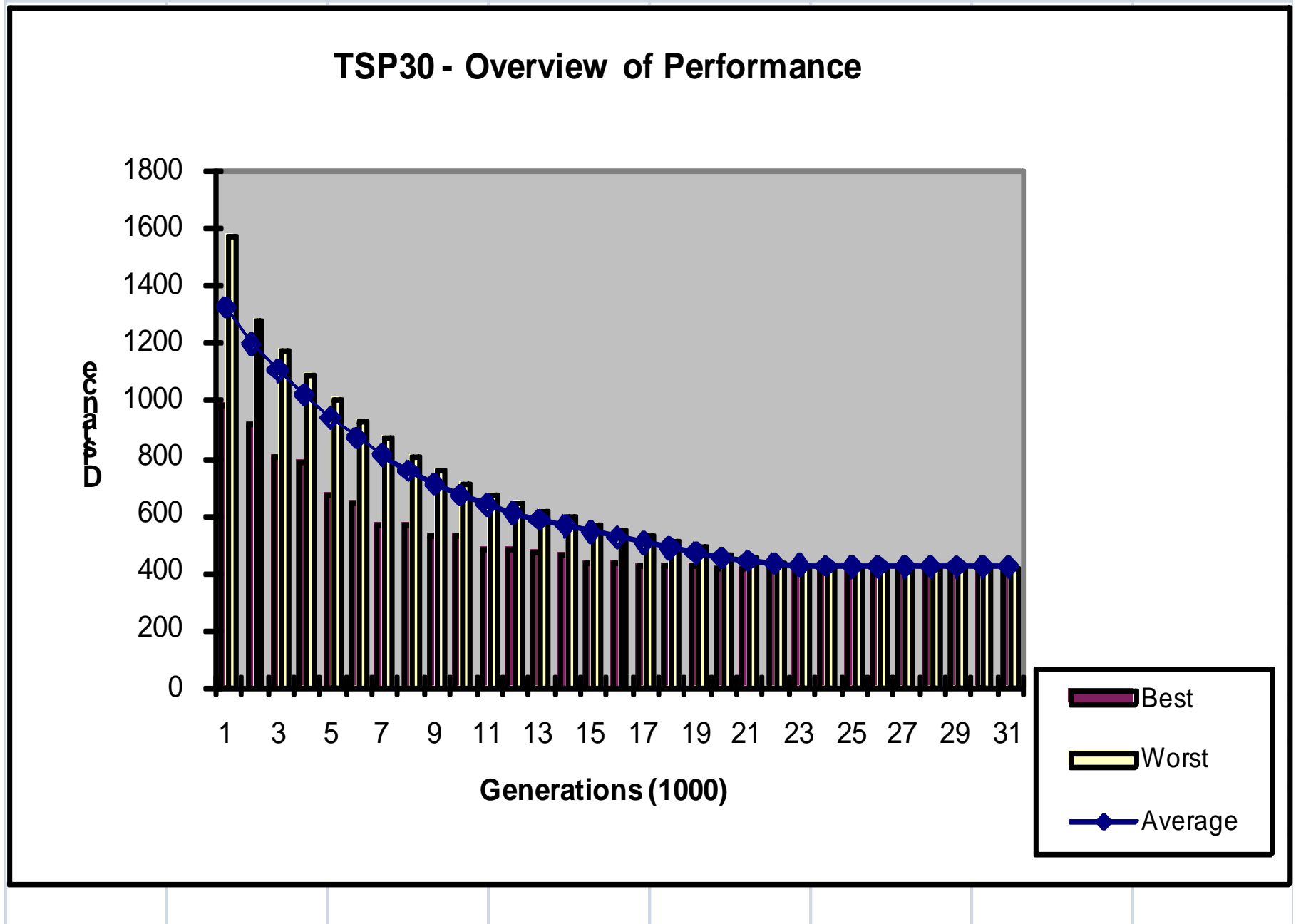
Solution_k (Distance = 652)



Best Solution (Distance = 420)



Overview of TSP Performance



Issues for GA Practitioners

- Choosing basic implementation issues:
 - representation
 - population size, mutation rate, ...
 - selection, deletion 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
- Modular, separate from application
- Supports multi-objective optimization
- Good for “noisy” environments
- Always an answer; answer gets better with time
- Inherently parallel; easily distributed

Benefits of Genetic Algorithms (cont.)

- Many ways to speed up and improve a GA-based application as knowledge about problem domain is gained
- Easy to exploit previous or alternate solutions
- Flexible building blocks for hybrid applications
- Substantial history and range of use



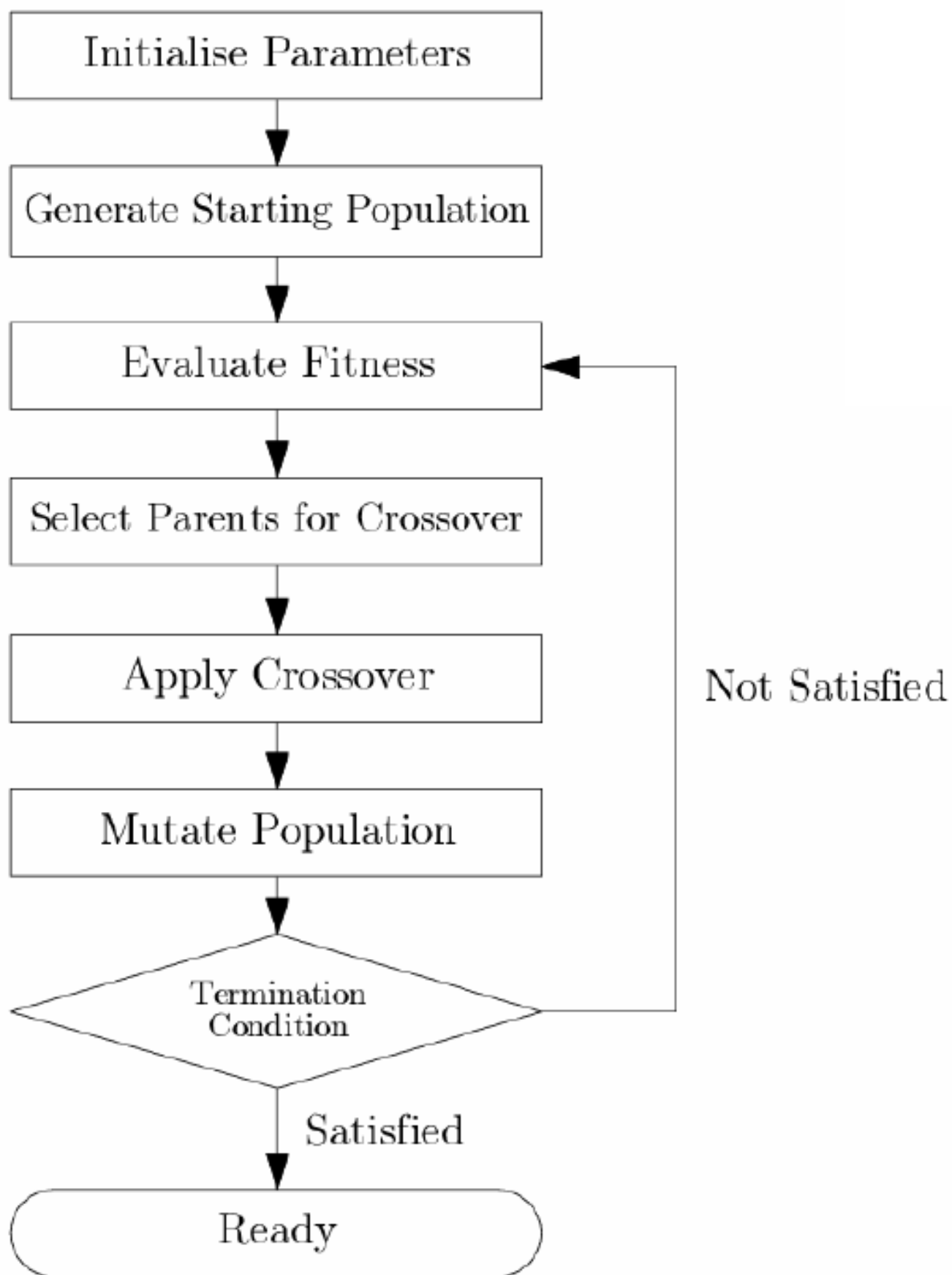
When to Use a GA

- Alternate solutions are too slow or overly complicated
- Need an exploratory tool to examine new approaches
- Problem is similar to one that has already been successfully solved by using a GA
- Want to hybridize with an existing solution
- Benefits of the GA technology meet key problem requirements

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





GAs Links

- Important Topic Reading

<http://www.geocities.com/francorbusetti/gaweb.pdf>

http://www.cs.iastate.edu/~honavar/ga_tutorial.pdf

- Funny GAs video

<http://www.youtube.com/watch?v=jfYo69UCQt4>

Gullu's Notes

<http://staff.um.edu.mt/kgui1/MIT/IntroGA.pdf>

These notes

<http://staff.um.edu.mt/mmon1/lectures/bit5201/notes.pdf>



Topic 3: Ant Colony Optimisation

- What is ACO?
- What motivated research in ACO?
- How did ACO research evolve?
- Applying ACO to practical situations



What is ACO?

Optimization algorithms drawn on various aspects of the behaviour of ants.

- Ants roam in a random manner;
- Eventually find food and inform the colony;
- When an ant traverses a path it marks a trail of pheromones behind;
- When a successful path to food is discovered, subsequent ants will not wonder at random but follow the trail.

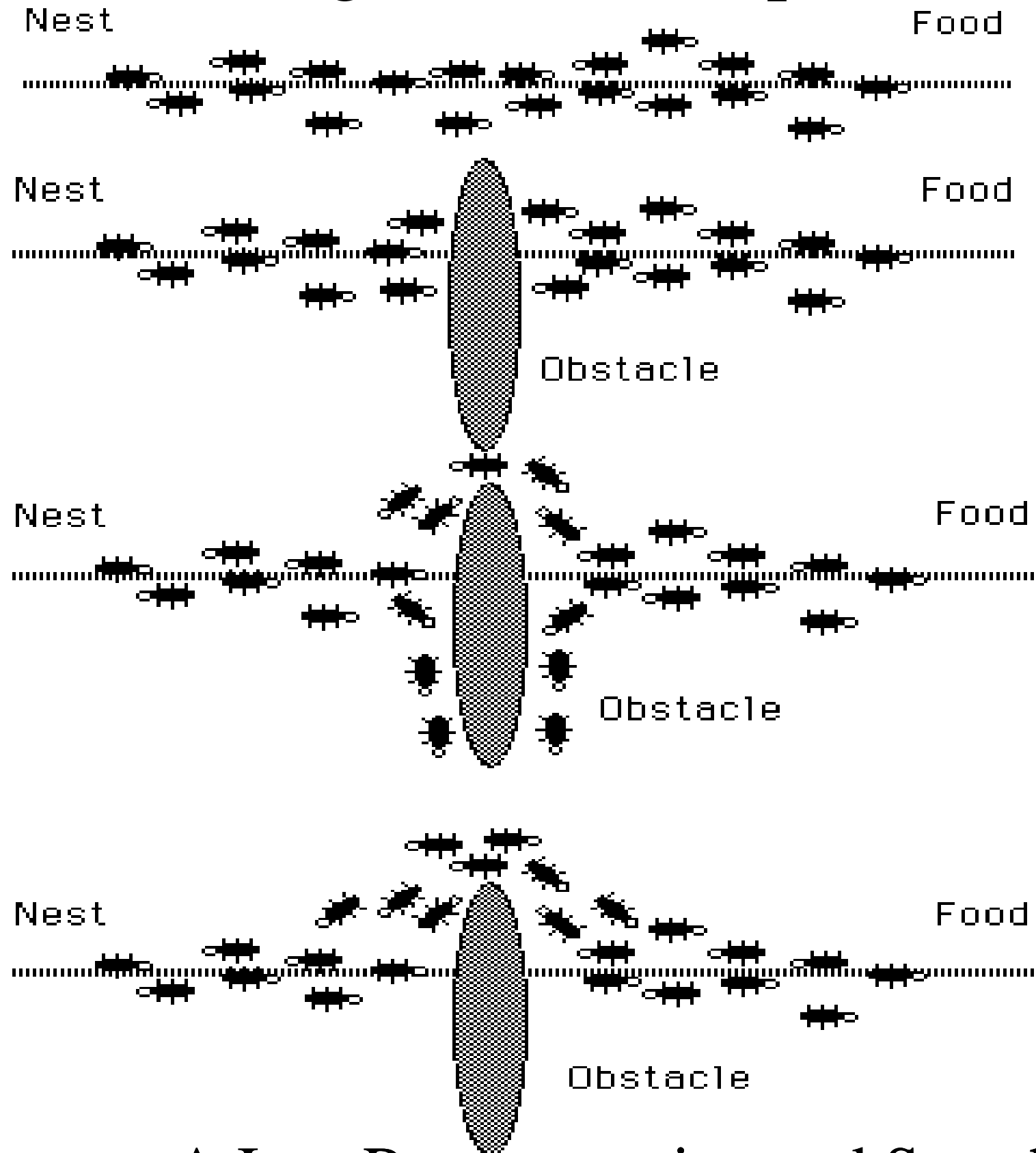


Its all about Pheromones!!!

Ants leave a trail of pheromones behind ...

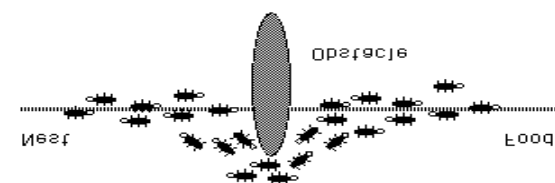
- Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength.
- The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate.
- A short path, by comparison, gets marched over faster, and thus the pheromone density remains high as it is laid on the path as fast as it can evaporate.

Finding the shortest path



Finding the shortest path

1. The first ant finds the food source via any way then returns to the nest leaving behind a trail pheromone;
2. Ants indiscriminately follow four possible ways, but the strengthening of the runway makes it more attractive as the shortest route;
3. Ants take the shortest route, long portions of other ways lose their trail pheromones;
4. In a series of experiments on a colony of ants with a choice between two unequal length paths leading to a source of food, biologists have observed that ants tended to use the shortest route



AC Algorithm

Ant colony looking for food

< == > Solving a problem

N Individual ants

< == > N Solutions

Each time the colony goes to
look for food and returns to
the nest

< == > Population of N solutions



... started off with ...

- Stigmergy in 1959 which is the study of behaviour in termites when building a nest;
- In 1991 Marco Dorigo proposed the Ant System in his Doctoral thesis;
- Five years later a technical report was published which laid the basis of Applying ACO to a plethora of problems.



Ant Farm Simulator



AntFarm.exe



ACO Research Applications

Academics have applied ACO to:

- Travelling Salesman Problem
- Network Routing
- Data Distribution
- Swarm Intelligence

ACO Real Applications

More Practically:

- Job Scheduling
- Graph Coloring
- Shortest Supersequence
- Vehicle Routing
- Call Routing



ACO Links

- Important Topic Reading

<http://iridia.ulb.ac.be/IridiaTrSeries/IridiaTr2006-010r003.pdf>

Gullu's Notes

<http://staff.um.edu.mt/kgui1/MIT/IntroACO.pdf>

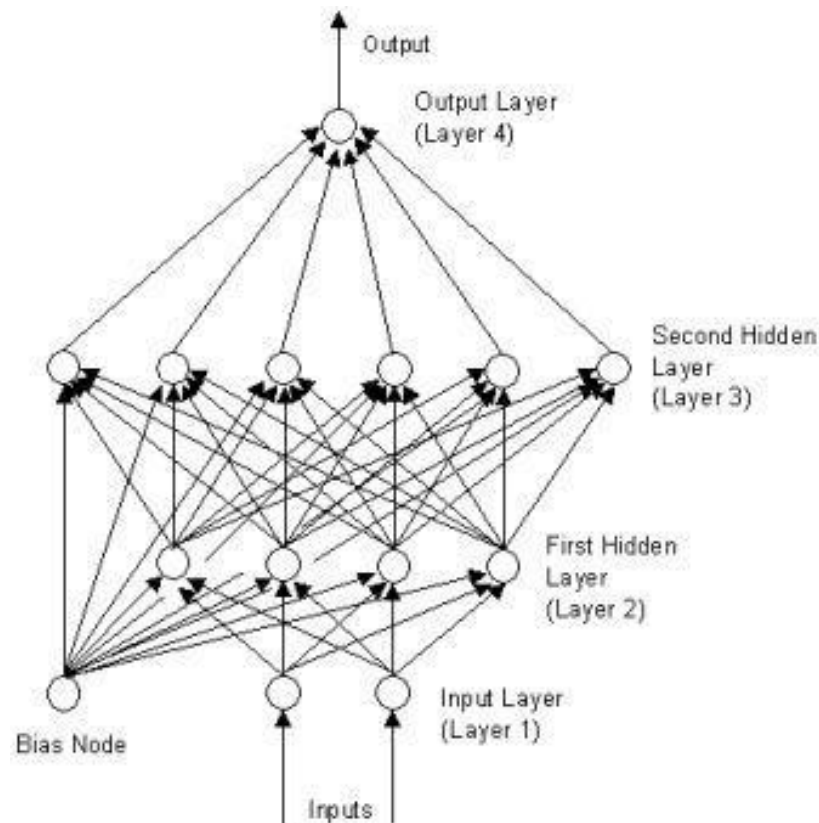
These notes

<http://staff.um.edu.mt/mmon1/lectures/bit5201/notes.pdf>



Topic 4: Artificial Neural Networks

- What are ANNs?
- Where can ANNs help?
- How do they work?
- Where are ANNs used and applied?



Neural networks to the rescue

- **Neural network:** *information processing paradigm inspired by biological nervous systems, such as our brain*
- Structure: large number of highly interconnected processing elements (*neurons*) working together
- Like people, they learn *from experience* (by example)

Neural networks to the rescue

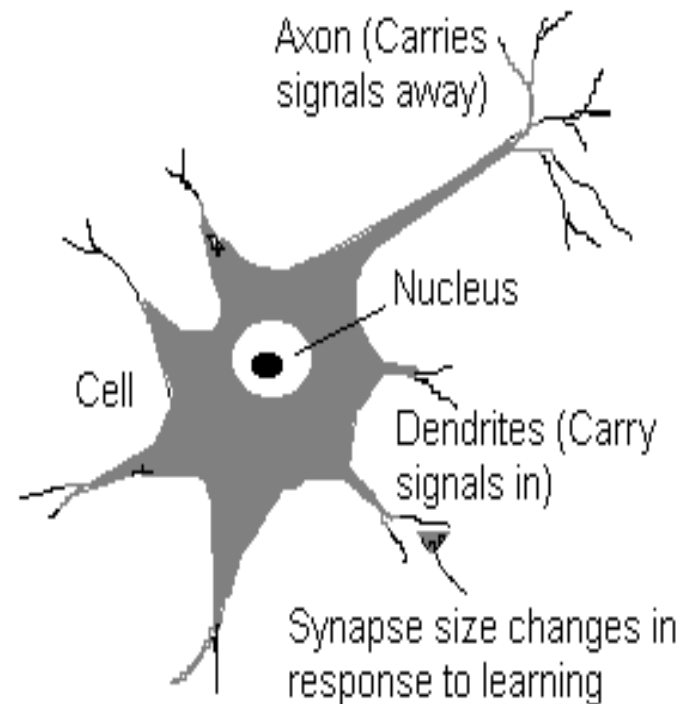
- Neural networks are configured for a specific application, such as pattern recognition or data classification, through a **learning process**
 - In a biological system, learning involves adjustments to the synaptic connections between neurons
- ➔ same for artificial neural networks (ANNs)

Where can neural network systems help

- when we can't formulate an algorithmic solution.
- when we *can* get lots of examples of the behavior we require.
 - ‘learning from experience’
- when we need to pick out the structure from existing data.

Inspiration from Neurobiology

- A neuron: many-inputs / one-output unit
- output can be *excited* or *not excited*
- incoming signals from other neurons determine if the neuron shall *excite* ("fire")
- Output subject to attenuation in the *synapses*, which are junction parts of the neuron



Synapse concept

- The synapse resistance to the incoming signal can be changed during a "learning" process [1949]

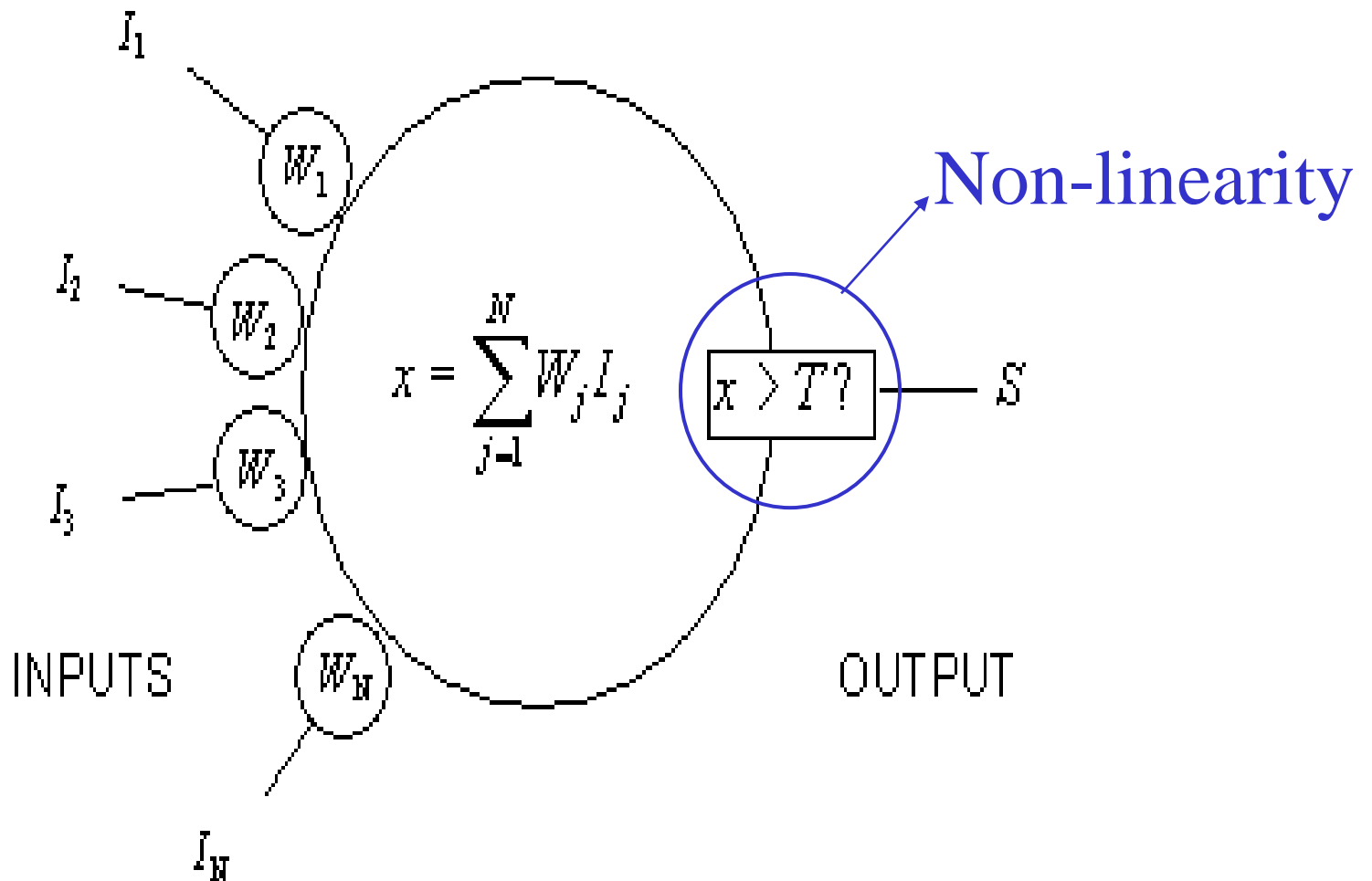
Hebb's Rule:

If an input of a neuron is repeatedly and persistently causing the neuron to fire, a metabolic change happens in the synapse of that particular input to reduce its resistance



Mathematical representation

The neuron calculates a weighted sum of inputs and compares it to a threshold. If the sum is higher than the threshold, the output is set to 1, otherwise to -1.



Learning

- From experience: examples / training data
- Strength of connection between the neurons is stored as a weight-value for the specific connection
- Learning the solution to a problem = changing the connection weights



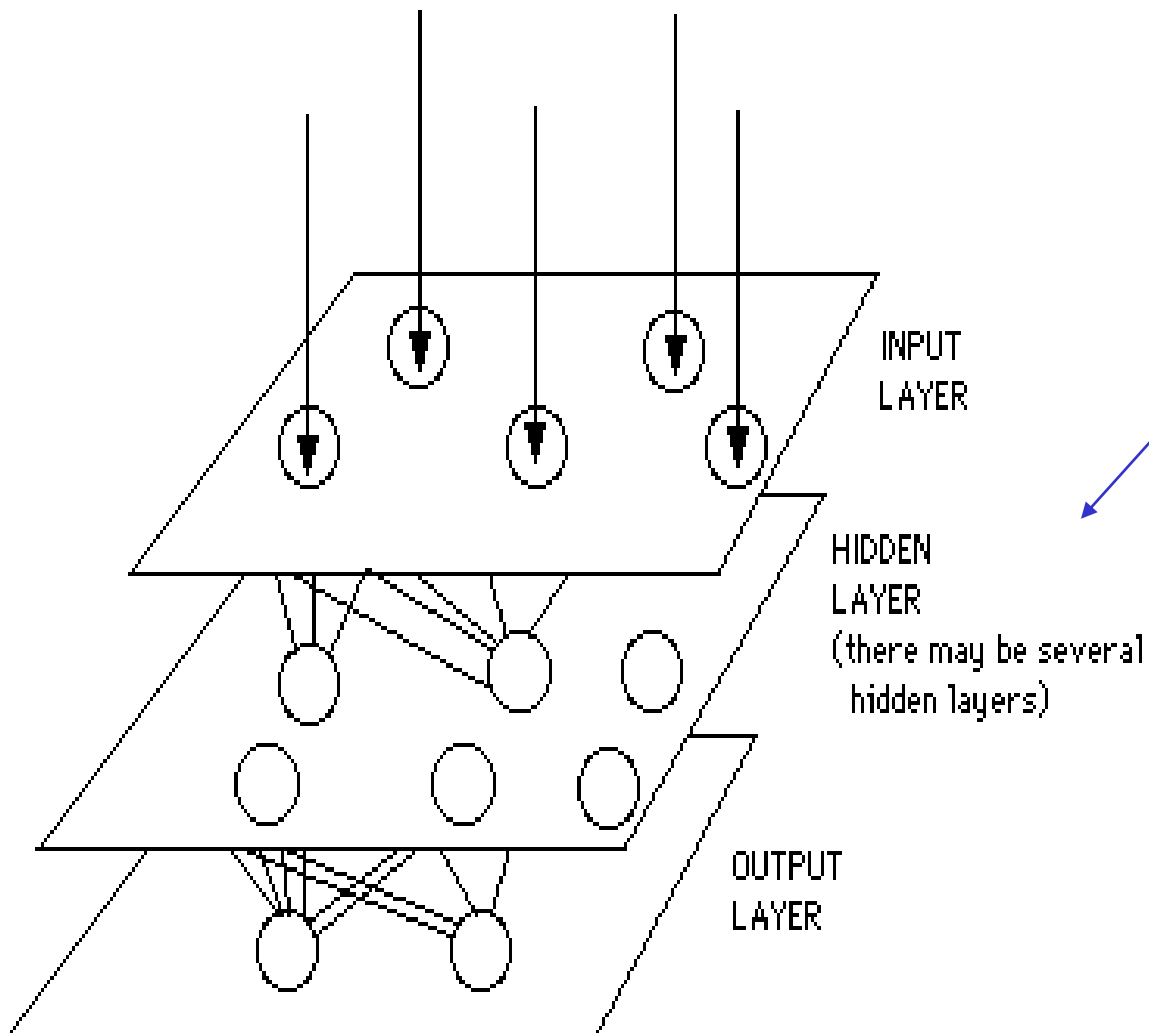
Operation mode

- Fix weights (unless in online learning)
- Network simulation = input signals flow through network to outputs
- Output is often a binary decision
- Inherently parallel
- Simple operations and threshold:
fast decisions and real-time response

ANNs physiology

Adaptive interaction between individual neurons

Power: collective behavior of interconnected neurons



The hidden layer learns to *recode* (or to *provide a representation* of) the inputs: associative mapping

Evolving networks

- *Continuous process of:*
 - Evaluate output
 - Adapt weights
 - Take new inputs
- ANN evolving causes stable state of the weights, but neurons **continue working**: network has ‘learned’ dealing with the problem
- Other issues involved:
 - Unsupervised
 - Reinforcement learning
 - Backpropagation
 - Number of Hidden layers

“Learning”



Where are ANNs used?

- Recognizing and matching complicated, vague, or incomplete patterns
- Data is unreliable
- Problems with noisy data
 - Prediction
 - Classification
 - Data association
 - Data conceptualization
 - Filtering
 - Planning



Applications (1/3)

- **Prediction: learning from past experience**
 - *pick the best stocks in the market*
 - *predict weather*
 - *identify people with cancer risk*
- **Classification**
 - *Image processing*
 - *Predict bankruptcy for credit card companies*
 - *Risk assessment*

Applications (2/3)

- **Recognition**

- *Pattern recognition: SNOOPE* (bomb detector in U.S. airports)
- *Character recognition*
- *Handwriting: processing checks*

- **Data association**

- *Not only identify the characters that were scanned but identify when the scanner is not working properly*



Applications (3/3)

- **Data Conceptualization**
 - *infer grouping relationships*
e.g. extract from a database the names of those most likely to buy a particular product.
- **Data Filtering**
 - *e.g. take the noise out of a telephone signal, signal smoothing*
- **Planning**
 - *Unknown environments*
 - *Sensor data is noisy*
 - *Fairly new approach to planning*

Practical Applications

- Process Modeling and Control - Creating a neural network model for a physical plant then using that model to determine the best control settings for the plant.
- Machine Diagnostics - Detect when a machine has failed so that the system can automatically shut down the machine when this occurs.
- Portfolio Management - Allocate the assets in a portfolio in a way that maximizes return and minimizes risk.
- Target Recognition - Military application which uses video and/or infrared image data to determine if an enemy target is present.
- Medical Diagnosis - Assisting doctors with their diagnosis by analyzing the reported symptoms and/or image data such as MRIs or X-rays.

More Practical Applications

- Credit Rating - Automatically assigning a company's or individuals credit rating based on their financial condition.
- Targeted Marketing - Finding the set of demographics which have the highest response rate for a particular marketing campaign.
- Voice Recognition - Transcribing spoken words into text.
- Financial Forecasting - Using the historical data of a security to predict the future movement of that security.
- Quality Control - Attaching a camera or sensor to the end of a production process to automatically inspect for defects.
- Intelligent Searching - An internet search engine that provides the most relevant content and banner ads based on the users' past behavior.
- Fraud Detection - Detect fraudulent credit card transactions and automatically decline the charge.



Strengths of a Neural Network

- **Power:** Model complex functions, nonlinearity built into the network
- **Ease of use:**
 - Learn by example
 - Very little user domain-specific expertise needed
- **Intuitively appealing:** based on model of biology, will it lead to genuinely intelligent computers/robots?

Neural networks cannot do anything that cannot be done using traditional computing techniques, **BUT** they can do some things which would otherwise be very difficult.

General Advantages

- Advantages
 - Adapt to unknown situations
 - Robustness: fault tolerance due to network redundancy
 - Autonomous learning and generalization
- Disadvantages
 - Not exact
 - Large complexity of the network structure
- For motion planning?

Status of Neural Networks

- Most of the reported applications are still in research stage
- No formal proofs, but they seem to have useful applications that work



ANNs Links

- Important Topic Reading

<http://arxiv.org/ftp/cs/papers/0308/0308031.pdf>

Gullu's Notes

<http://staff.um.edu.mt/kgui1/MIT/IntroANN.pdf>

These notes

<http://staff.um.edu.mt/mmon1/lectures/bit5201/notes.pdf>

