Computational Intelligence

Unit # 3.2

Acknowledgement

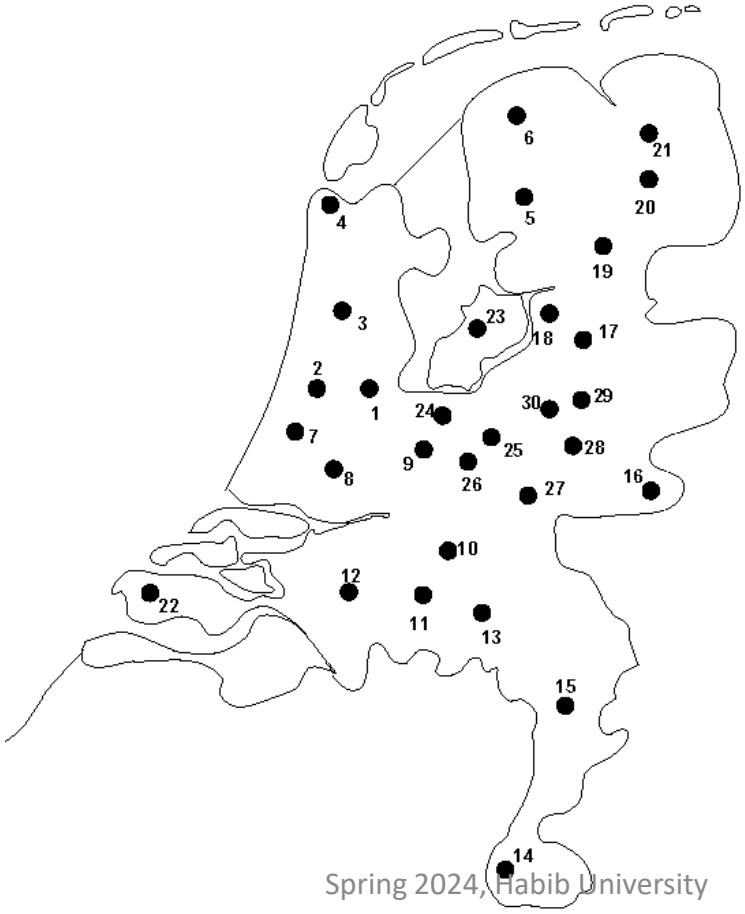
- The slides of this lecture have been taken from the lecture slides of “CSE659 – Computational Intelligence” by Dr. Sajjad Haider.

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.

Traveling Sales Person Problem

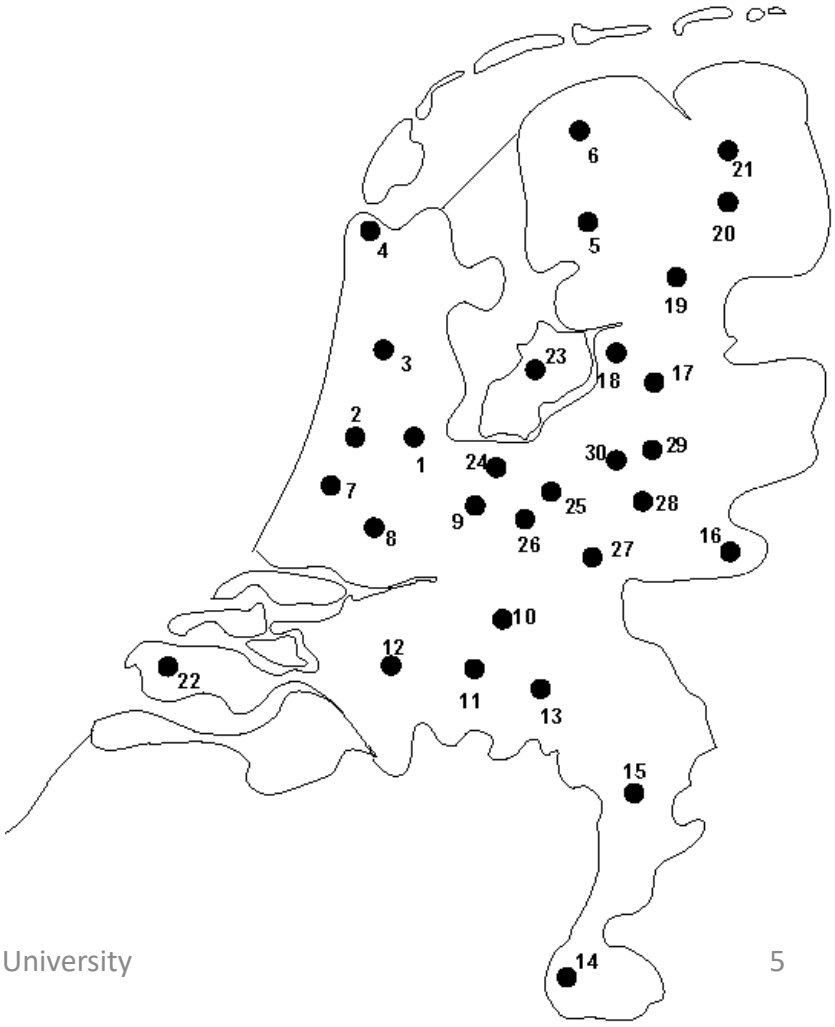
- Given a number of cities and the costs of traveling from one city to any other city, what is the cheapest round-trip route that visits each city exactly once and then returns to the starting city?



Permutation Representation: TSP

- Problem:
  - Given n cities
  - Find a complete tour with minimal length
- Encoding:
  - Label the cities 1, 2, … , *n*
  - One complete tour is one permutation (e.g. for n =4 [1,2,3,4], [3,4,2,1] are OK)
- Search space is BIG:

  for 30 cities there are 30! ≈ 10^{32} possible tours



TSP: Nearest Neighbor

	A	B	C	D	E	F	G	H
A	0	8	3	1	4	9	3	6
B	8	0	5	10	11	4	3	6
C	3	5	0	8	7	1	5	12
D	1	10	8	0	9	11	6	4
E	4	11	7	9	0	5	17	3
F	9	4	1	11	5	0	4	1
G	3	3	5	6	17	4	0	7
H	6	6	12	4	3	1	7	0

- Start with A: A – D – H – F – C – B – G – E Cost?
- Start with E: E – H – F – C – A – D – B – G Cost?
- Start with G: G – B – F – H – E – A – D – C Cost?

Initialize The Population

- Suppose the population size is 6

Candidate Solutions
A C B F H D E G
H B G E A C D F
A H G C B D F E
E G B C D H F A
F H A D C B E G
C D B A H E G F

Compute Fitness

Candidate Solutions	Fitness
A C B F H D E G	43
H B G E A C D F	52
A H G C B D F E	49
E G B C D H F A	47
F H A D C B E G	49
C D B A H E G F	56

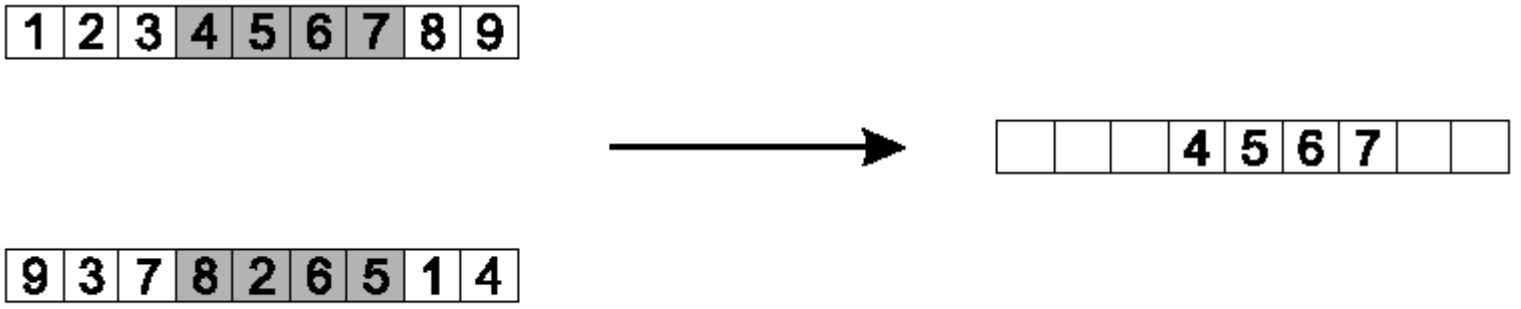
	A	B	C	D	E	F	G	H
A	0	8	3	1	4	9	3	6
B	8	0	5	10	11	4	3	6
C	3	5	0	8	7	1	5	12
D	1	10	8	0	9	11	6	4
E	4	11	7	9	0	5	17	3
F	9	4	1	11	5	0	4	1
G	3	3	5	6	17	4	0	7
H	6	6	12	4	3	1	7	0

Order 1 Crossover

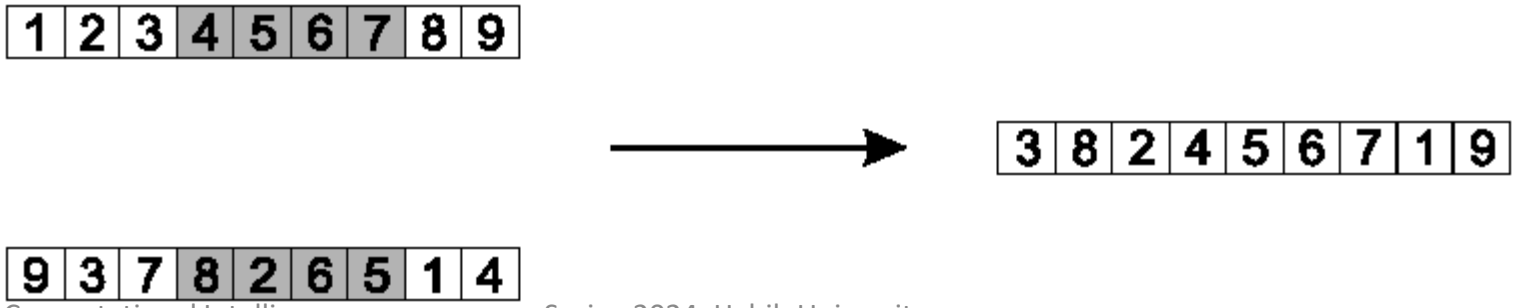
- Idea is to preserve relative order that elements occur
- Informal procedure:
  1. Choose an arbitrary part from the first parent
  2. Copy this part to the first child
  3. Copy the numbers that are not in the first part, to the first child:
     - starting right from cut point of the copied part,
     - using the **order** of the second parent
     - and wrapping around at the end
  4. Analogous for the second child, with parent roles reversed

Order 1 Crossover Example

- Copy randomly selected set from first parent



- Copy rest from second parent in order 1,9,3,8,2



Crossover Example

- The following parents have been selected during patent selection:

Parent 1	A C B F H D E G	43
Parent 2	F H A D C B E G	49

- Let's do crossover to produce two offspring. Suppose the crossover points are 3 and 5

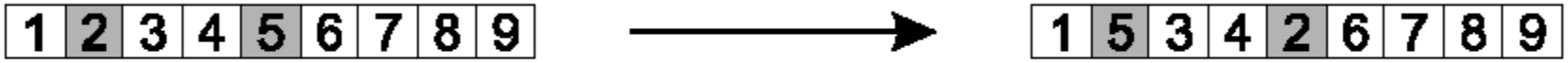
Offspring 1	D C B F H E G A
Offspring 2	F H A D C E G B

Insert Mutation for Permutations

- Pick two allele values at random
- Move the second to follow the first, shifting the rest along to accommodate
- Note that this preserves most of the order and the adjacency information

Swap Mutation for Permutations

- Pick two alleles at random and swap their positions
- Preserves most of adjacency information (4 links broken), disrupts order more



Mutation Example

- Perform swap mutation on the two offspring produced in the previous slide.

Offspring 1	D C B F H E G A
Offspring 2	F H A D C E G B

- For Offspring 1, swap 2nd and 4th gene.
- For Offspring 2, swap 2nd and 6th gene.

Offspring 1	D F B C H E G A	55
Offspring 2	F E A D C H G B	40

Thanks