

---

# Advanced Artificial Intelligence

## Lab 05

# Content

---

- The adjacency representation for TSP
- Mutation operators
- Exercise
- References

# The adjacency representation

---

- The adjacency representation is designed to facilitate the **manipulation** of edges.
- **The crossover operators based on this representation** generate offspring that inherit most of their edges from **the parent chromosomes**.

# The adjacency representation

---

The adjacency representation can be described as follows:

- city  $j$  occupies position  $i$  in the chromosome if there is an edge from city  $i$  to city  $j$  in the tour.
- An example
  - the chromosome 38526417 encodes the tour 13564287.
  - City 3 occupies position 1 in the chromosome because edge (1, 3) is in the tour.
  - Similarly, city 8 occupies position 2 because edge (2, 8) is in the tour, etc.

# The adjacency representation

---

- **Alternate edges crossover** (*Grefenstette et al. [1]*)
  - A starting edge  $(i,j)$  is **selected at random** in one parent.
  - Then, the tour is extended by selecting the edge  $(j, k)$  in the other parent.
  - The tour is progressively extended in this way by alternatively selecting edges from the two parents.
  - **When an edge introduces a cycle, the new edge is selected at random** (and is not inherited from the parents).

# The adjacency representation

---

- **Alternate edges crossover** (*Grefenstette et al. [1]*)
  - In the following figure, an offspring is generated from two parent chromosomes that encode the tours 13564287 and 14236578
  - Here, edge (1, 4) is first selected in parent 2, and city 4 in position 1 of parent 2 is copied at the same position in the offspring.
  - Then, the edges (4, 2) in parent 1, (2, 3) in parent 2, (3, 5) in parent 1 and (5, 7) in parent 2 are selected and inserted in the offspring.

|           |   |          |          |          |          |          |          |          |          |
|-----------|---|----------|----------|----------|----------|----------|----------|----------|----------|
| parent 1  | : | <b>3</b> | <b>8</b> | <b>5</b> | <b>2</b> | <b>6</b> | <b>4</b> | <b>1</b> | <b>7</b> |
| parent 2  | : | 4        | 3        | 6        | 2        | 7        | 5        | 8        | 1        |
| <hr/>     |   |          |          |          |          |          |          |          |          |
| offspring | : | 4        | 3        | <b>5</b> | <b>2</b> | 7        | 8        | 6        | 1        |

# The adjacency representation

---

- **Alternate edges crossover** (*Grefenstette et al. [1]*)
  - Then, edge (7, 1) is selected in parent 1. However, this edge introduces a cycle, and a new edge leading to a city not yet visited is selected at random. Assume that (7, 6) is chosen.
  - Then, edge (6, 5) is selected in parent 2, but it also introduces a cycle. At this point, (6, 8) is the only selection that does not introduce a cycle.
  - Finally, the tour is completed with edge (8, 1).
  - The final offspring encodes the tour 14235768, and all edges in the offspring are inherited from the parents, apart from the edges (7, 6) and (6, 8).

|           |   |   |   |   |   |   |   |   |   |
|-----------|---|---|---|---|---|---|---|---|---|
| parent 1  | : | 3 | 8 | 5 | 2 | 6 | 4 | 1 | 7 |
| parent 2  | : | 4 | 3 | 6 | 2 | 7 | 5 | 8 | 1 |
| <hr/>     |   |   |   |   |   |   |   |   |   |
| offspring | : | 4 | 3 | 5 | 2 | 7 | 8 | 6 | 1 |

# The adjacency representation

---

- **Edge recombination crossover (ER)** (*Whitley et al. [2]*)
  - The alternate edge operator introduces many random edges in the offspring, particularly the last edges, when the choices for extending the tour are limited.
  - **Since the offspring must inherit as many edges as possible from the parents,** the introduction of random edges should be minimized.
  - The edge recombination operator reduces the myopic behavior of the alternate edge approach with a special data structure called the "edge map".



# The adjacency representation

---

- **Edge recombination crossover (ER)** (*Whitley et al. [2]*)
  - The edge map maintains the list of edges that are incident to each city in the parent tours and **that lead to cities not yet included in the offspring.**
  - These edges are still available for extending the tour and are said to be **active.**
  - The strategy is to extend the tour by selecting the edge that leads to the city **with the minimum number of active edges.**
  - In the case of equality between two or more cities, one of these cities is selected at random.
  - The approach is less likely to get trapped in a "dead end", namely, a city with no remaining active edges **(thus requiring the selection of a random edge).**

# The adjacency representation

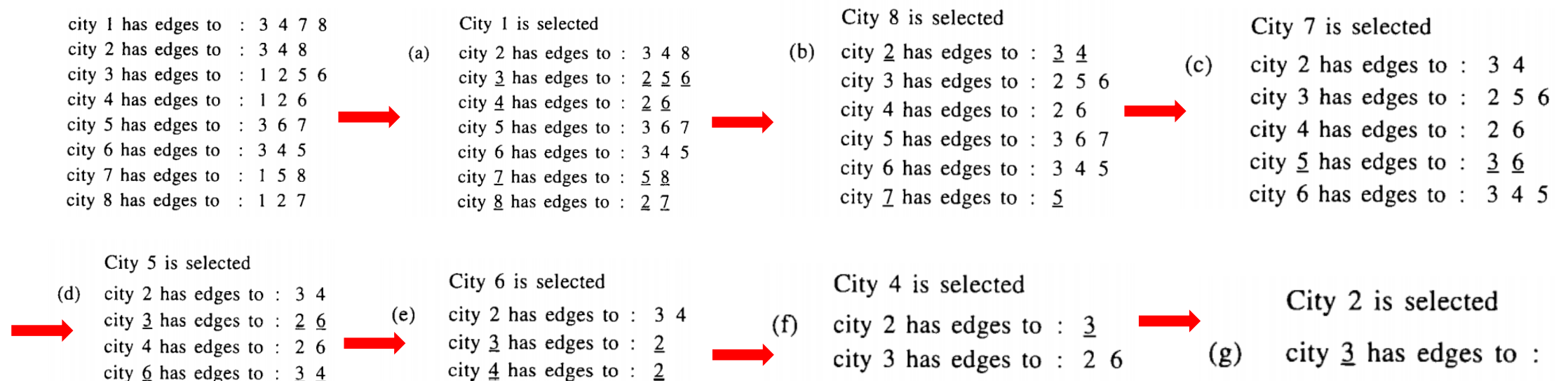
---

- Edge recombination crossover (ER) (*Whitley et al. [2]*)
  - For the tours 13564287 and 14236578 (path representation), the initial edge map is shown in the following figure.

```
city 1 has edges to : 3 4 7 8
city 2 has edges to : 3 4 8
city 3 has edges to : 1 2 5 6
city 4 has edges to : 1 2 6
city 5 has edges to : 3 6 7
city 6 has edges to : 3 4 5
city 7 has edges to : 1 5 8
city 8 has edges to : 1 2 7
```


# The adjacency representation

- Edge recombination crossover (ER) (*Whitley et al. [2]*)
  - The operation of the edge recombination crossover operator will now be illustrated on this initial edge map.



# The adjacency representation

- Edge recombination crossover (ER) (*Whitley et al. [2]*)
  - Assume that city 1 is selected as the starting city.
  - Accordingly, all edges incident to city 1 must first be deleted from the initial edge map.
  - From city 1, we can go to cities 3, 4, 7 or 8.
  - City 3 has three active edges, while cities 4, 7 and 8 have two active edges.
  - Hence, a random choice is made between cities 4, 7 and 8.

|                               |   |   |
|-------------------------------|---|---|
| city 1 has edges to : 3 4 7 8 |   | City 1 is selected                                      |
| city 2 has edges to : 3 4 8   | (a)   | city 2 has edges to : 3 4 8                             |
| city 3 has edges to : 1 2 5 6 |   | city <u>3</u> has edges to : <u>2</u> <u>5</u> <u>6</u> |
| city 4 has edges to : 1 2 6   |   | city <u>4</u> has edges to : <u>2</u> <u>6</u>          |
| city 5 has edges to : 3 6 7   |  | city 5 has edges to : 3 6 7                             |
| city 6 has edges to : 3 4 5   |   | city 6 has edges to : 3 4 5                             |
| city 7 has edges to : 1 5 8   |   | city <u>7</u> has edges to : <u>5</u> <u>8</u>          |
| city 8 has edges to : 1 2 7   |   | city <u>8</u> has edges to : <u>2</u> <u>7</u>          |

# The adjacency representation

---

- Edge recombination crossover (ER) (*Whitley et al. [2]*)
  - Assume that city 8 is selected.
  - From 8, we can go to cities 2 and 7.

City 1 is selected

(a) city 2 has edges to : 3 4 8  
city 3 has edges to : 2 5 6  
city 4 has edges to : 2 6  
city 5 has edges to : 3 6 7  
city 6 has edges to : 3 4 5  
city 7 has edges to : 5 8  
city 8 has edges to : 2 7



City 8 is selected

(b) city 2 has edges to : 3 4  
city 3 has edges to : 2 5 6  
city 4 has edges to : 2 6  
city 5 has edges to : 3 6 7  
city 6 has edges to : 3 4 5  
city 7 has edges to : 5

# The adjacency representation

---

- Edge recombination crossover (ER) (*Whitley et al. [2]*)
  - As indicated in edge map (b), city 2 has two active edges and city 7 only one, so city 7 is selected.
  - From city 7, there is no choice but to go to city 5.

City 8 is selected

(b) city 2 has edges to : 3 4  
city 3 has edges to : 2 5 6  
city 4 has edges to : 2 6  
city 5 has edges to : 3 6 7  
city 6 has edges to : 3 4 5  
city 7 has edges to : 5



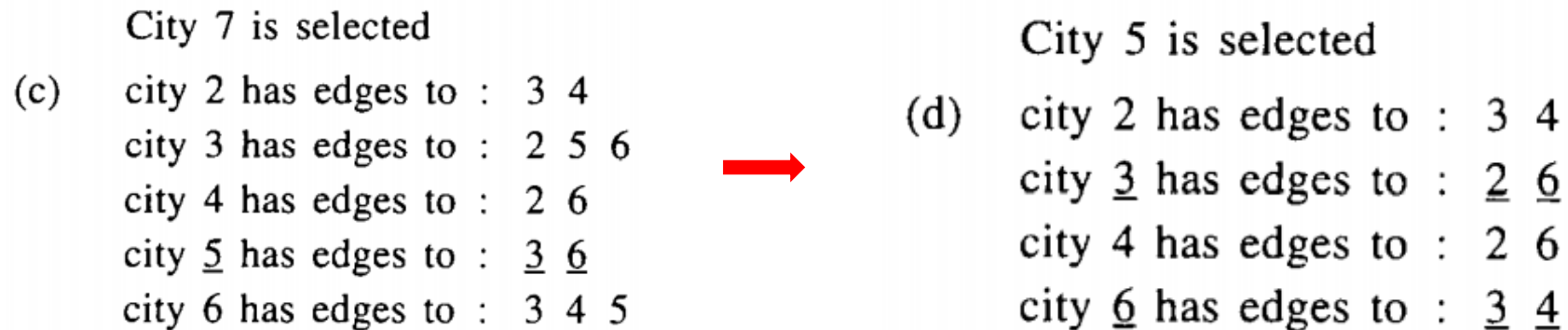
City 7 is selected

(c) city 2 has edges to : 3 4  
city 3 has edges to : 2 5 6  
city 4 has edges to : 2 6  
city 5 has edges to : 3 6  
city 6 has edges to : 3 4 5

# The adjacency representation

---

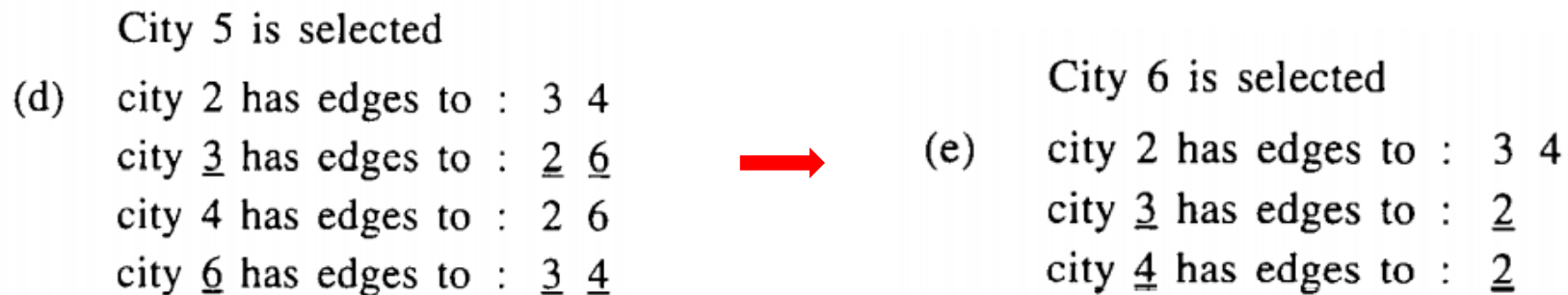
- Edge recombination crossover (ER) (*Whitley et al. [2]*)
  - City 5 is selected.
  - From this point, edge map (d) offers a choice between cities 3 and 6 with two active edges.



# The adjacency representation

---

- Edge recombination crossover (ER) (*Whitley et al. [2]*)
  - Assume that city 6 is randomly selected.
  - From city 6, we can go to cities 3 and 4, and edge map (e) indicates that both cities have one active edge.





# The adjacency representation

---

- Edge recombination crossover (ER) (*Whitley et al. [2]*)
  - assume that city 4 is randomly selected.
  - from city 4 we can only go to city 2.

City 6 is selected  
(e) city 2 has edges to : 3 4  
city 3 has edges to : 2  
city 4 has edges to : 2



City 4 is selected  
(f) city 2 has edges to : 3  
city 3 has edges to : 2 6

# The adjacency representation

---

- Edge recombination crossover (ER) (*Whitley et al. [2]*)
  - City 2 is selected.
  - From city 2 we must go to city 3.
  - The final tour is 18756423 and all edges are inherited from both parents.

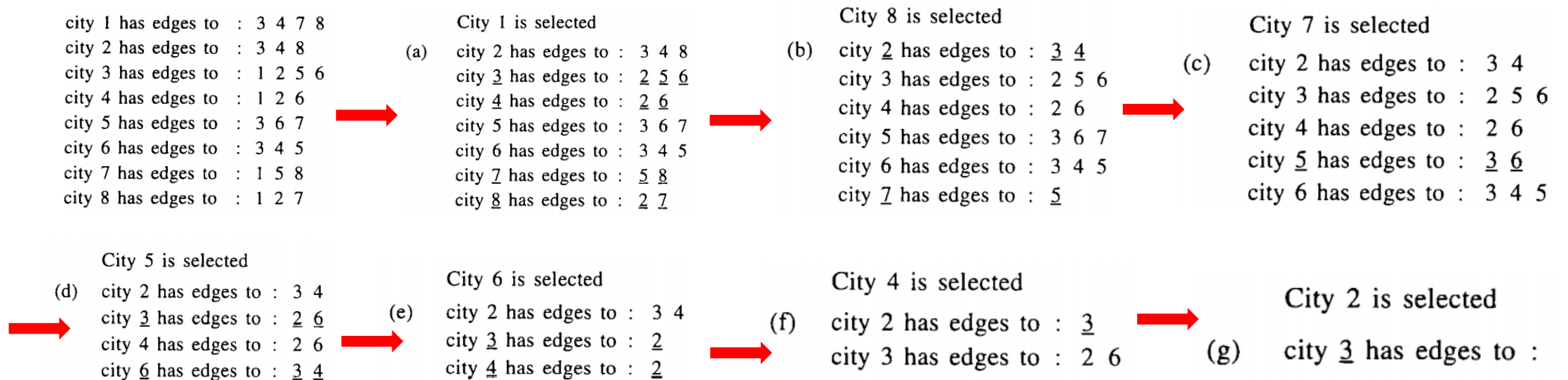
City 4 is selected  
(f) city 2 has edges to : 3  
city 3 has edges to : 2 6



City 2 is selected  
(g) city 3 has edges to :

# The adjacency representation

- Edge recombination crossover (ER) (*Whitley et al. [2]*)
  - To this end, the edge map will be updated after each city selection. In these edge maps, cities of particular interest are underlined.



# The adjacency representation

---

- **Heuristic crossover (HX)** (*Grefenstette et al. [1], Grefenstette [3]*)
  - The previous crossover operators did not exploit the distances between the cities (i.e., the length of the edges).
  - It is a characteristic of the genetic approach to avoid any heuristic information about a specific application domain, apart from the overall evaluation or fitness of each chromosome.
  - This characteristic explains the **robustness** of the genetic search and its wide applicability.

# The adjacency representation

---

- Heuristic crossover (HX) (*Grefenstette et al. [1], Grefenstette [3]*)
  - Step 1. Choose a **random** starting city from one of the two parents.
  - Step 2. Compare the edges leaving the current city in both parents and **select the shorter edge.**
  - Step 3. If the shorter parental edge introduces a cycle in the partial tour, then **extend the tour with a random edge that does not introduce a cycle.**
  - Step 4. Repeat steps 2 and 3 **until all cities are included in the tour.**

# Mutation operators

---

- Mutation operators for the TSP aim at randomly generating new permutations of the cities.
- Permutation operators for the TSP often greatly modifies the original tour.
  - Swap
  - Local hill-climbing
  - Scramble

# Mutation operators

---

- Swap
  - Two cities are randomly selected and swapped (i.e., their positions are exchanged).
- Local hill-climbing
  - A local edge exchange heuristic is applied to the tour (e.g., 2-opt).
- Scramble
  - Two cut points are selected at random on the chromosome, and the cities within the two cut points are randomly permuted.

# Exercise

---

For the tours 13564287 and 14236578 (path representation), the initial edge map is shown in the following figure. Here, we consider the Edge recombination crossover (ER) in The adjacency representation.

- Exercise 1:

What is the final tour if city 2 is selected as the starting city? Write down the derivation process and justify your conclusion.

- Exercise 2:

What is the final tour if city 3 is selected as the starting city? Write down the derivation process and justify your conclusion.

```
city 1 has edges to : 3 4 7 8
city 2 has edges to : 3 4 8
city 3 has edges to : 1 2 5 6
city 4 has edges to : 1 2 6
city 5 has edges to : 3 6 7
city 6 has edges to : 3 4 5
city 7 has edges to : 1 5 8
city 8 has edges to : 1 2 7
```



# References

---

- [1] J. Grefenstette, R. Gopal, B.J. Rosmaita and D.V. Gucht, Genetic algorithms for the traveling salesman problem, in: Proc. 1st. Int. Conf. on Genetic Algorithms (ICGA '85), Carnegie-Mellon University, Pittsburgh, PA (1985) pp. 160-168.
- [2] D. Whitley, T. Starkweather and D. Fuquay, Scheduling problems and traveling salesmen: The genetic edge recombination operator, in: Proc. 3rd Int. Conf. on Genetic Algorithms (ICGA '89), George Mason University, Fairfax, VA (1989) pp. 133-140.
- [3] J. Grefenstette, Incorporating problem specific knowledge into genetic algorithms, in: Genetic Algorithms and Simulated Annealing, ed. L. Davis (Morgan Kaufmann, 1987) pp. 42-60.