ERX: Inputs, Output, and Notation

Inputs

- Two parent permutations P_1, P_2 over the same city set.
- A starting city (fixed, random, or heuristic).
- A tie-break rule (random or heuristic).

Output

- One child permutation C (a valid tour order).

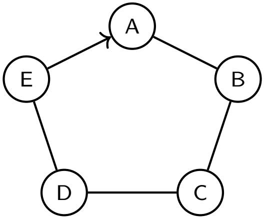
Neighbors in a tour

- For a city v in parent P:

$$N_P(v) = \{\text{left neighbor}, \ \text{right neighbor}\}$$

- Wrap-around is included (first and last are neighbors).

Example tour



Each city has exactly two neighbors in a tour

Note

ERX starts by extracting **adjacency information** from parent tours.

ERX Edge Table: Concrete Example (Two Cities)

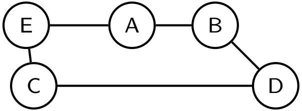
Edge table definition

$$N(v) = N_{P_1}(v) \cup N_{P_2}(v)$$

Interpretation

- Each parent contributes **two neighbors** per city (left/right with wrap-around).
- ERX takes the **union** of these neighbors.
- Duplicate neighbors are removed automatically.

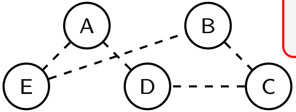
Parent $P_1 = [A\ B\ D\ C\ E]$



$N_{P_1}(A) = \{B, E\}$ $N_{P_1}(B) = \{A, D\}$

Union

Parent $P_2 = [A\ E\ B\ C\ D]$



$N_{P_2}(A) = \{E, D\}$ $N_{P_2}(B) = \{E, C\}$

Union (ERX)
$N(A) = \{B, D, E\}$
$N(B) = \{A, C, D, E\}$

Note

The edge table collects all adjacencies seen in the parents.

ERX: Using the Edge Table to Choose the Next City

Given (from previous slide)

- Edge table entries have been constructed: $N(A) = \{B, D, E\}$, $N(B) = \{A, C, D, E\}$

Step 1: start the child

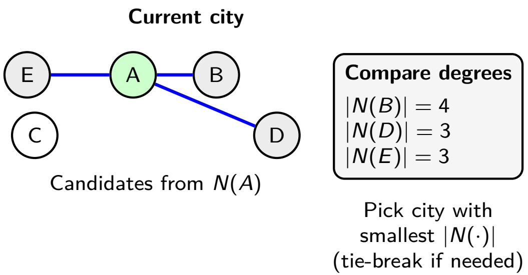
- Choose a starting city (here: A).
- Initialize the child: $C = [A]$

Step 2: candidate set

- The next city must come from the neighbors of the current city: Candidates $= N(A) = \{B, D, E\}$.

Note

ERX chooses the next city by consulting the edge sets of the candidates.



Step 3: apply the ERX rule

- Compare the sizes of the neighbors' edge sets:

$$|N(B)|,\ |N(D)|,\ |N(E)|.$$

- Select the city with the **smallest remaining degree** (most constrained).

ERX: Extending the Child (First Update)

Decision from previous slide

- From the candidate set:

$$N(A) = \{B, D, E\},$$

cities D and E have the smallest degree.

- Apply a tie-break rule (here: choose D).

Step 4: extend the child

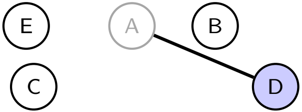
- Append the selected city:

$$C = [A, D].$$

Step 5: update the edge table

- Mark A as visited.
- Remove A from all neighbor sets $N(\cdot)$.

Chosen next city



Partial child: $A \to D$

After removing A

$N(B): \{C, D, E\}$
$N(C): \{B, D\}$
$N(D): \{B, C, E\}$
$N(E): \{B, C, D\}$

Next candidates will come from $N(D)$

Effect of the update

- Remaining degrees shrink.
- Some cities become more constrained.

Note

After each choice, ERX updates the edge table and continues the walk.

ERX: Continuing the Walk (Second Extension)

Current state

- Partial child from previous step:

$$C = [A, D].$$

- Current city is D.

Step 6: new candidate set

- Candidates must come from the neighbors of D: $N(D) = N_{P_1}(D) \cup N_{P_2}(D) = \{B, E\} \cup \{C, E\} = \{B, C, E\}$.

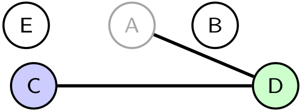
Step 7: apply ERX rule again

- Compare remaining degrees:

$$|N(B)| = 3, \quad |N(C)| = 2, \quad |N(E)| = 3.$$

- City C is the most constrained.

Current city D



Partial child: $A \to D \to C$

After removing D

$B: \{C, E\}$
$C: \{B\}$
$E: \{B, C\}$

Next candidates will come from $N(C)$

Decision

- Select C as the next city.
- Extend the child:

$$C = [A, D, C].$$

Note

ERX repeatedly chooses the most constrained neighbor of the current city.

ERX: Forced Move When Only One Neighbor Remains

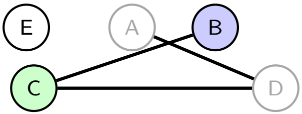
Current state

- Partial child:

$$C = [A, D, C].$$

- Current city is C.

Current city C



After removing visited
$B: \{E\}$
$E: \{B\}$

Partial child: $A \to D \to C \to B$

Candidate set from the edge table

- Neighbors of C after updates: $N(C) = N_{P_1}(C) \cup N_{P_2}(C) = \{B, D\} \cup \{B, A\} = \{A, B, D\}$.
- After removing already visited cities $\{A, D\}$: $N(C) = \{B\}$.

Only one valid continuation remains

textbfStep 8: forced choice

- Only one candidate remains.
- ERX must choose B:

$$C = [A, D, C, B].$$

Note

When $|N(v)| = 1$, ERX makes a deterministic (forced) move.

ERX: Completing the Tour

Current state

- Partial child from previous slide:

$$C = [A, D, C, B].$$

- Remaining unvisited city: E.

Final step

- From the updated edge table:

$$N(B) = \{E\}, \qquad N(E) = \{B\}.$$

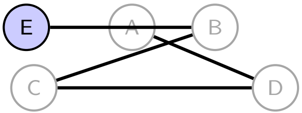
- Only one unvisited city remains.

Complete the child

- Append the final city:

$$C = [A, D, C, B, E].$$

Final step



Final child: $A \to D \to C \to B \to E$

Result

- The child is a **valid permutation**.
- Many edges are inherited from one or both parents.

Note

ERX always terminates with a valid tour once all cities are visited.