Evolutionary Algorithms for the Eulerian Cycle Problem

What is the Right Mutation Operator?

Outline

1. Eulerian Cycle Problem
   - Definition
   - Why this problem?

2. Different Mutation Operators
   - Exchange operations
   - Jump operations
   - One-sided jump operations

Eulerian Cycle Problem

- Input: Graph $G = (V,E)$
- Task: Find a Eulerian Cycle!

Def. (Eulerian Cycle): Cycle containing each edge exactly once.

Theorem: A Eulerian Cycle exists iff each vertex has even degree. Can be computed in time $O(m)$.

Why the EulerCycle Problem?

- Combinatorial optimization problem
  - as opposed to `artificial' problems.
- Theoretically well understood
  - can hope for results.
- Permutations instead of bit-strings
  - What is the right mutation operator?

General Framework: (1+1) Evolutionary Algorithms

1. Start with a random individual
2. While not happy
   - Mutate individual
   - Keep the better (fitter) of the two

(1+1) EAs for EulerCycle

1. Start with a random individual
2. While not happy
   - Mutate individual
   - Keep the better (fitter) of the two

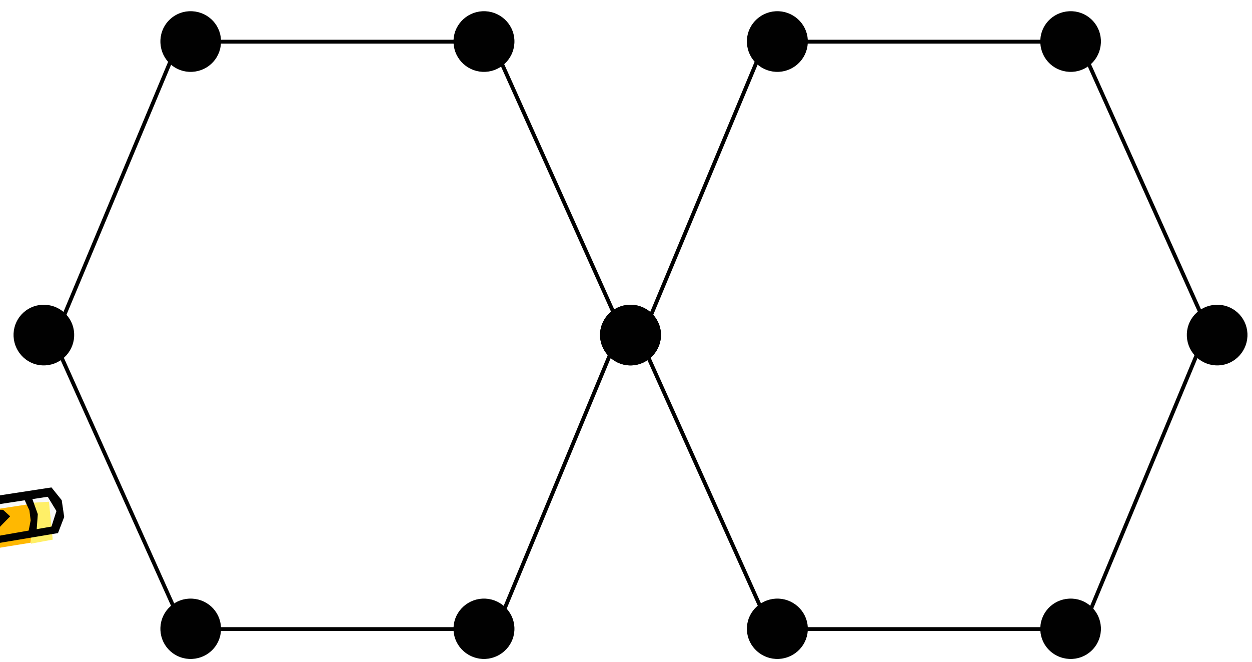
Individual: Permutation $(e_1, e_2, \ldots, e_m)$.
Fitness: Max. k such that $(e_1, e_2, \ldots, e_k)$ is path.
Mutation: Real topic of this talk!

Mutation: RLS vs. (1+1) EAs

- Randomized Local Search:
  - Perform exactly one elementary operation (bit-flip, exchange, etc.)
- (1+1) EA:
  - Pick s at random from Poi(λ=1).
  - Perform exactly s+1 elementary operations

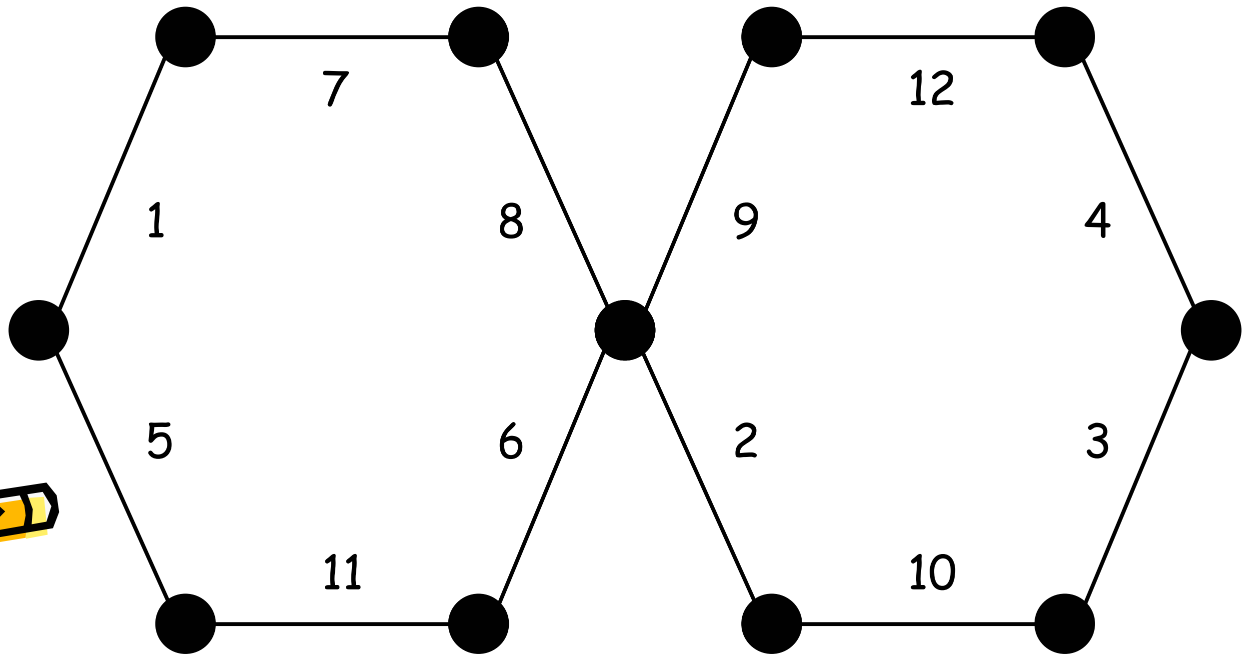
Mutation via Transpositions

- Pick two edges at random and exchange their position in the permutation.



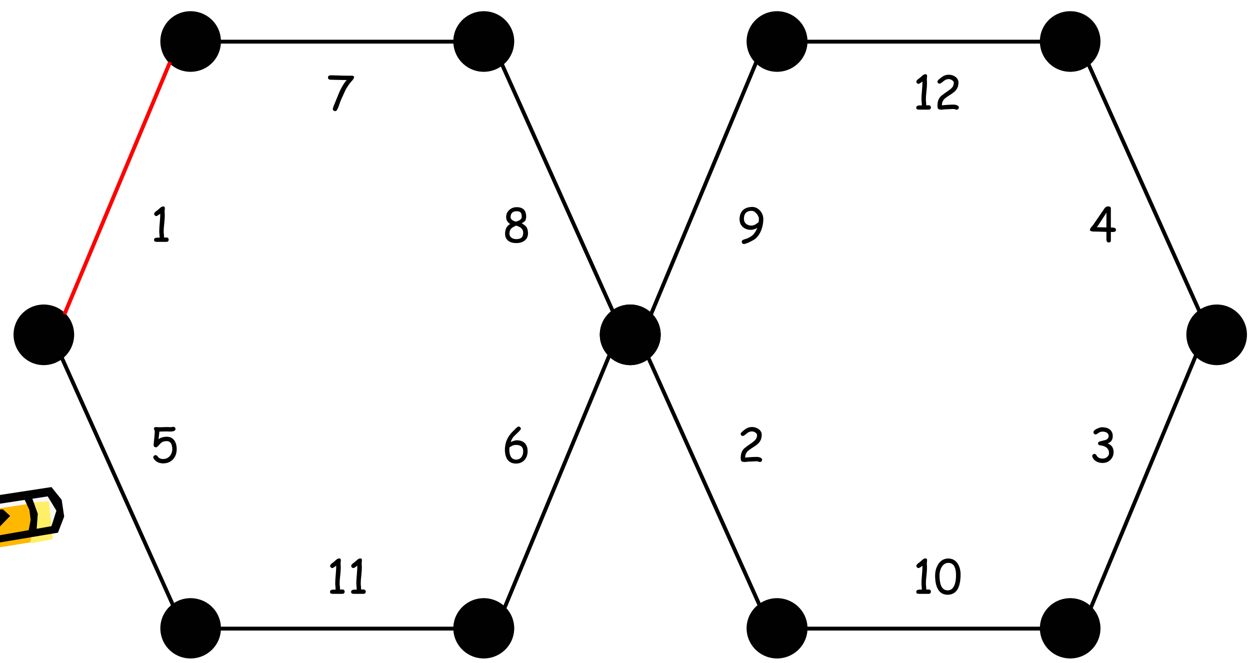
Mutation via Transpositions

- Pick two edges at random and exchange their position in the permutation.



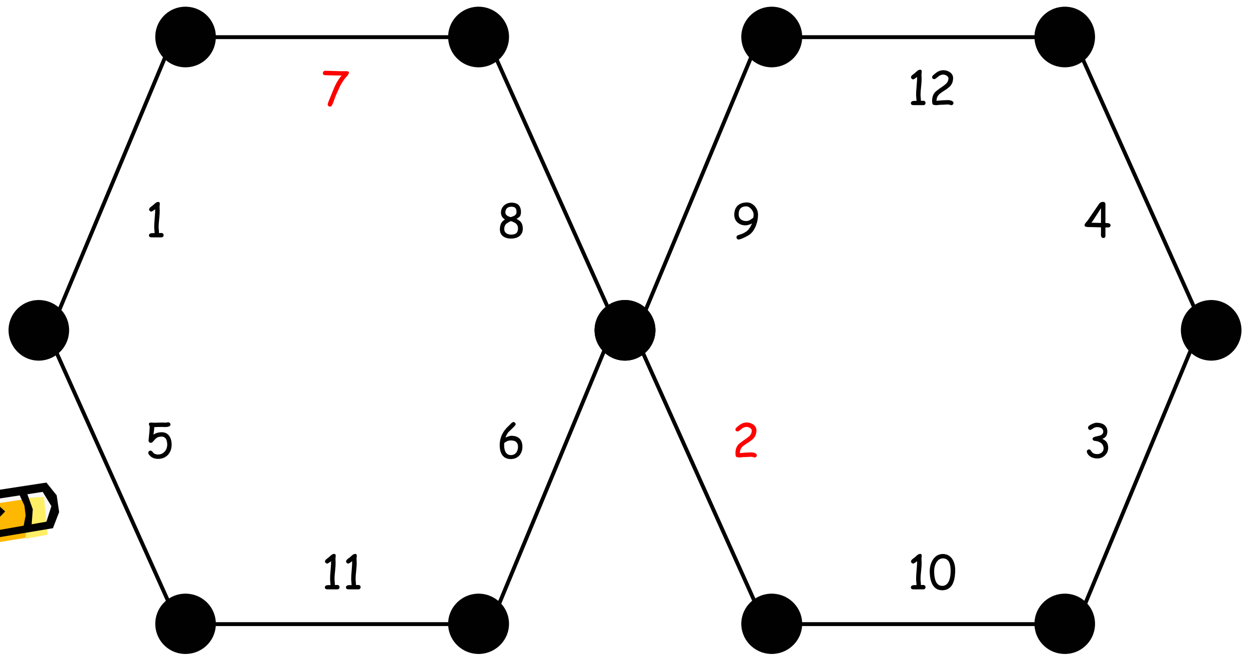
Mutation via Transpositions

- Current Fitness: 1.



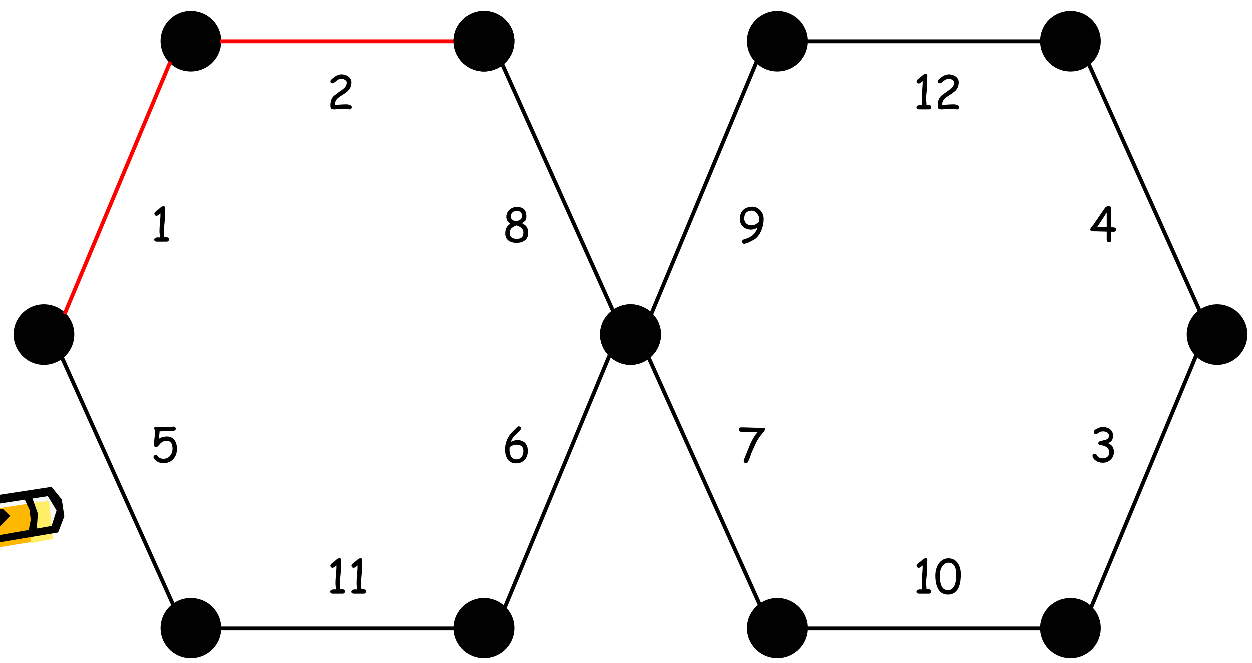
Mutation via Transpositions

- Current Fitness: 1.
- Random Transposition: (27). [Prob. $\Theta(m^{-2})$]



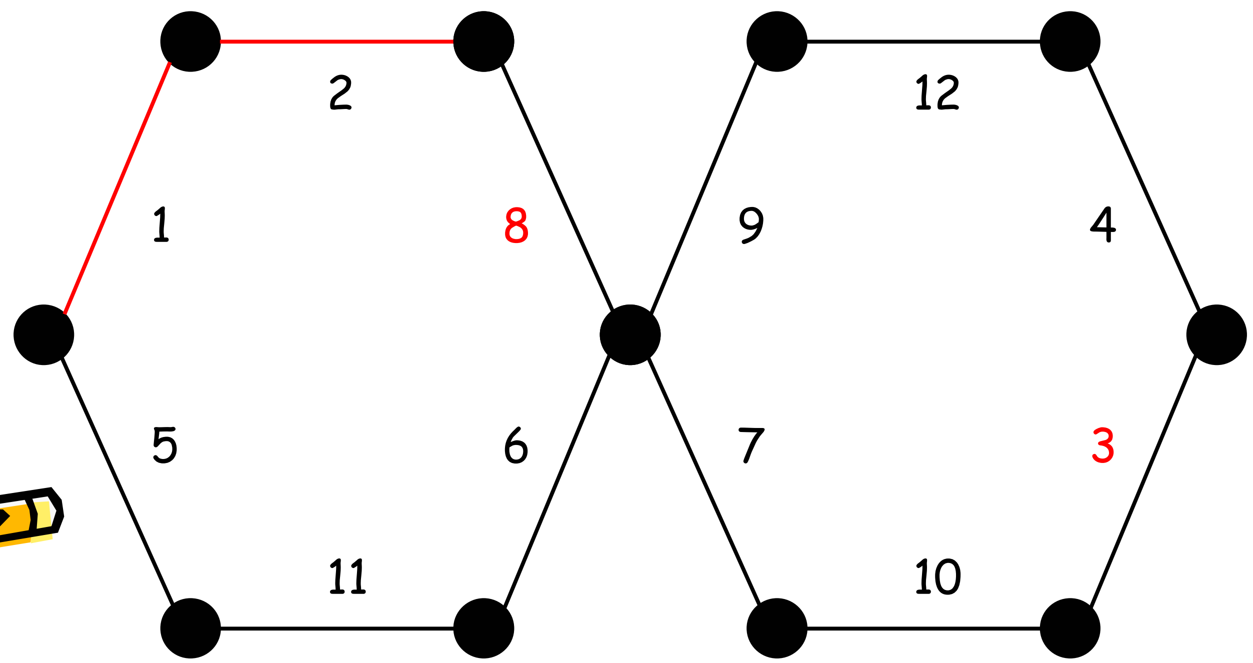
Mutation via Transpositions

- Current Fitness: 1.
- Random Transposition: (27).
- Resulting Fitness: 2.



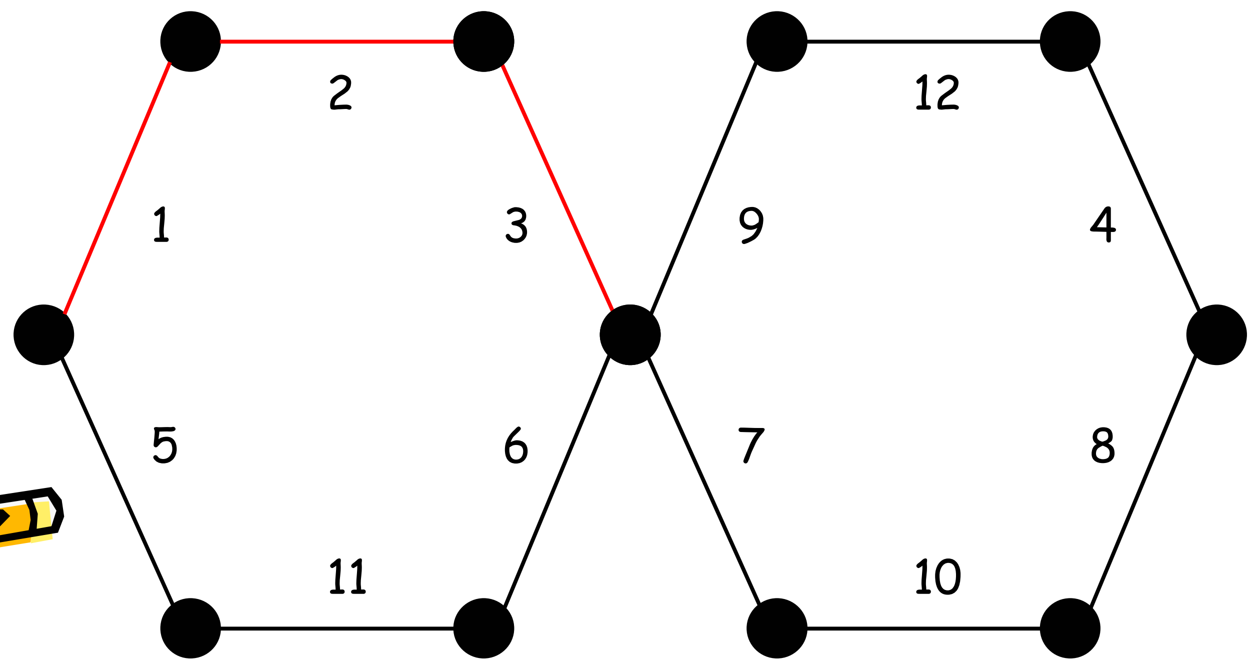
Mutation via Transpositions

- Only useful transposition: (83).



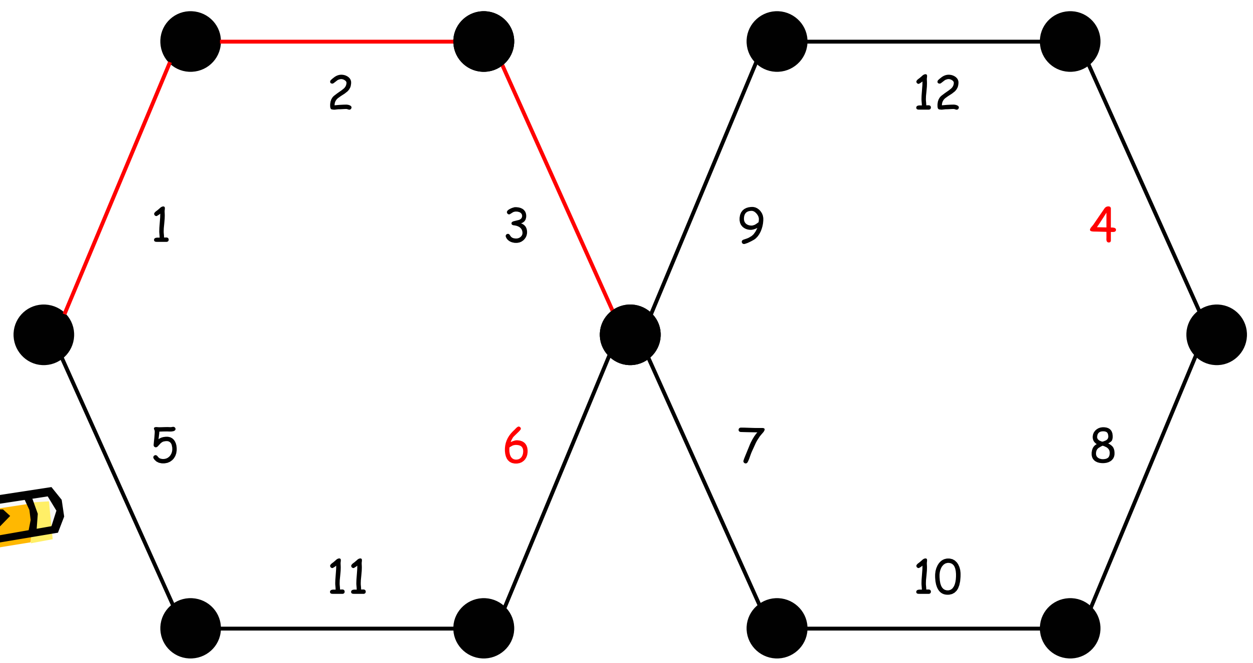
Mutation via Transpositions

- Only useful transposition: (83).
- Resulting Fitness: 3.



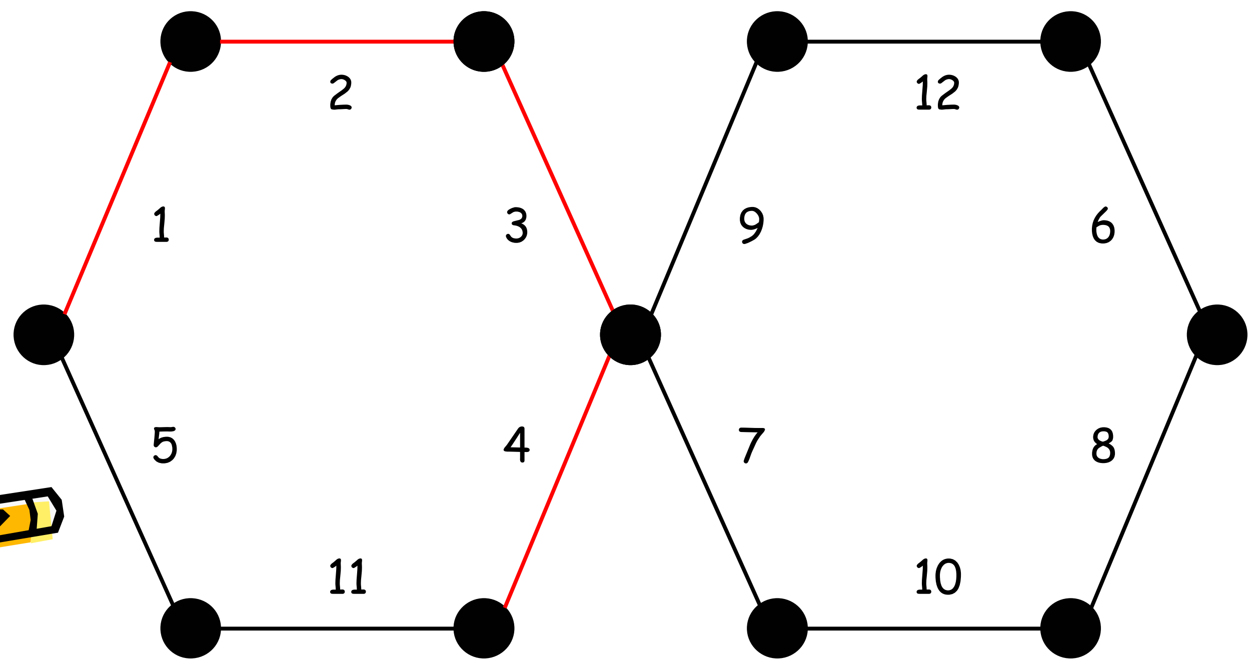
Mutation via Transpositions

- Useful random transpositions: (64), (74), (94).
- Here: (64).



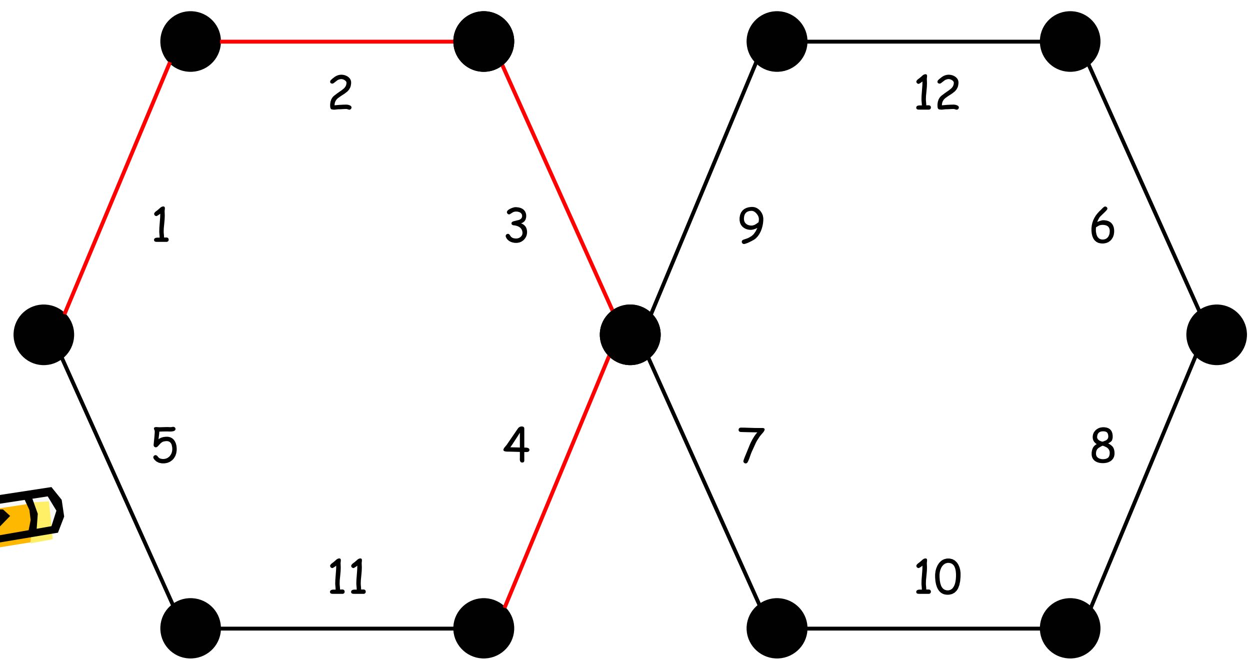
Mutation via Transpositions

- Useful random transpositions: (64), (74), (94).
- Here: (64).
- Resulting fitness: 4.



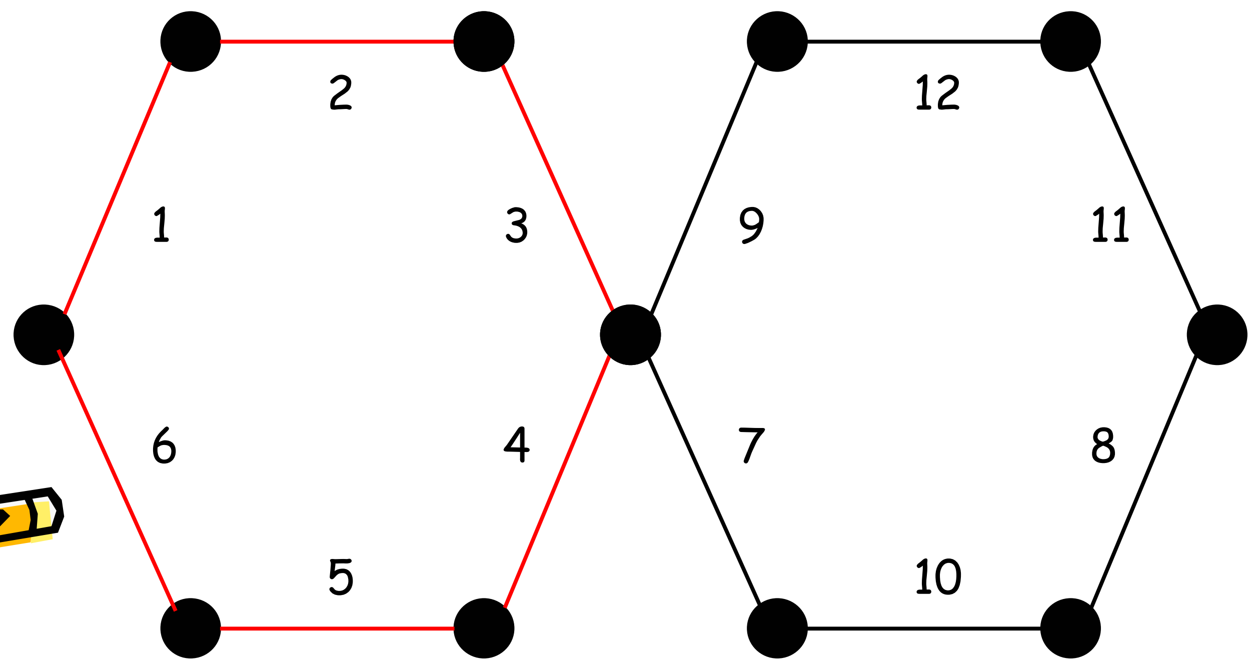
Mutation via Transpositions

- Two more steps: (11 5) and then (11 6).



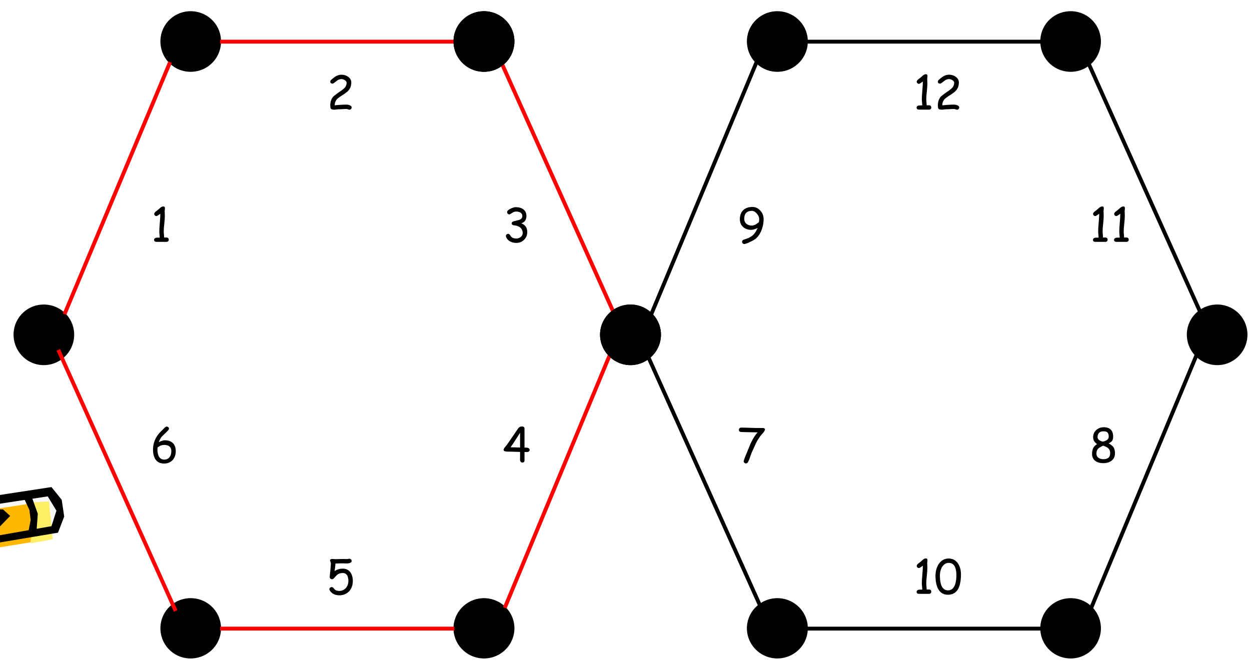
Mutation via Transpositions

- Two more steps: (11 5) and then (11 6).
- Resulting fitness: 6.
- Expected time up to here: $O(m^3)$.



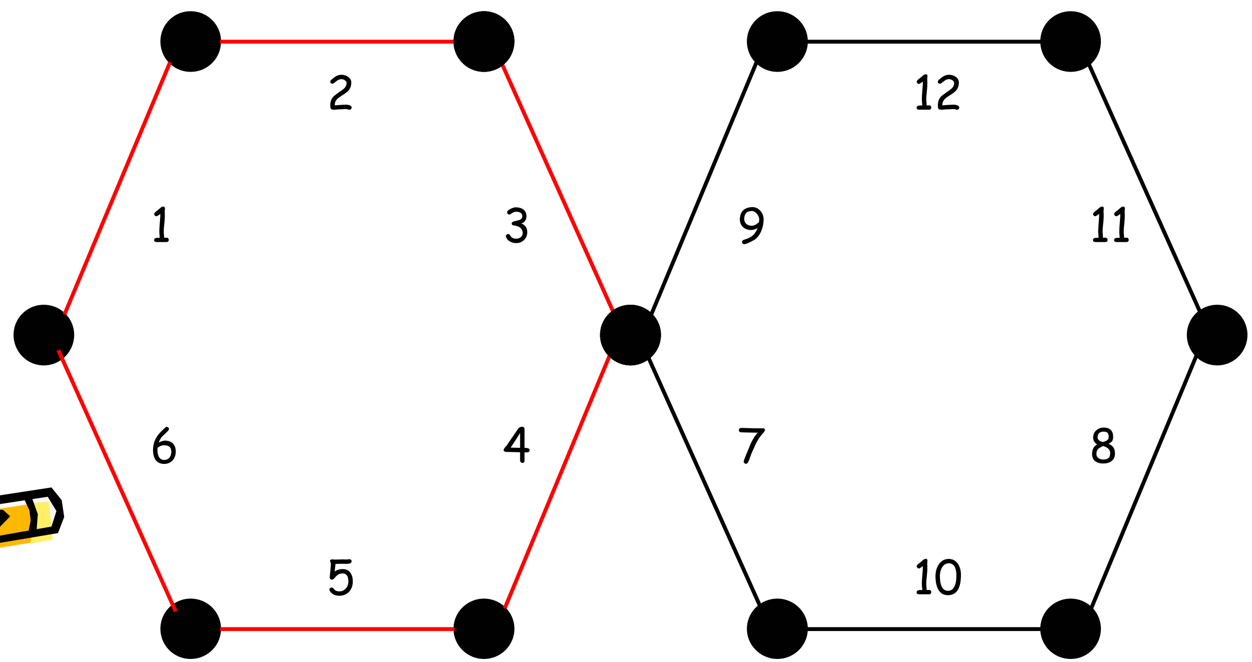
Mutation via Transpositions

- But now?



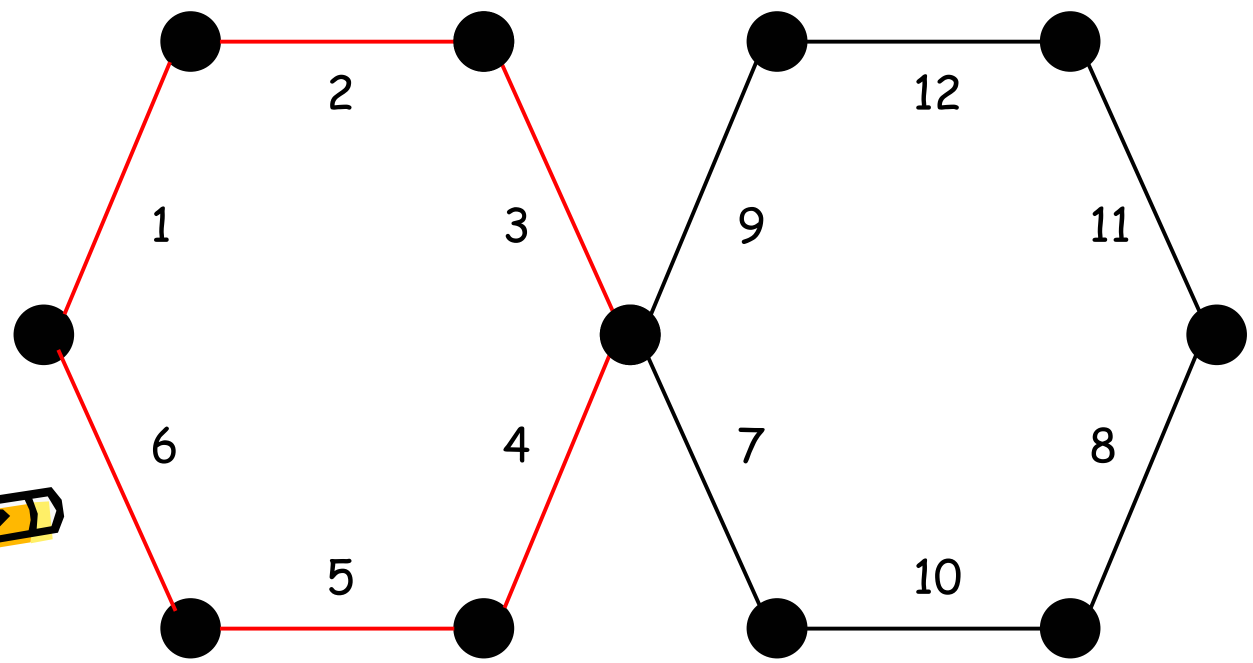
Mutation via Transpositions

- But now?
- RLS: Deadlock (no transposition increases fitness).
- EA: Need sth like (47)(5 10)(6 8) in one round!



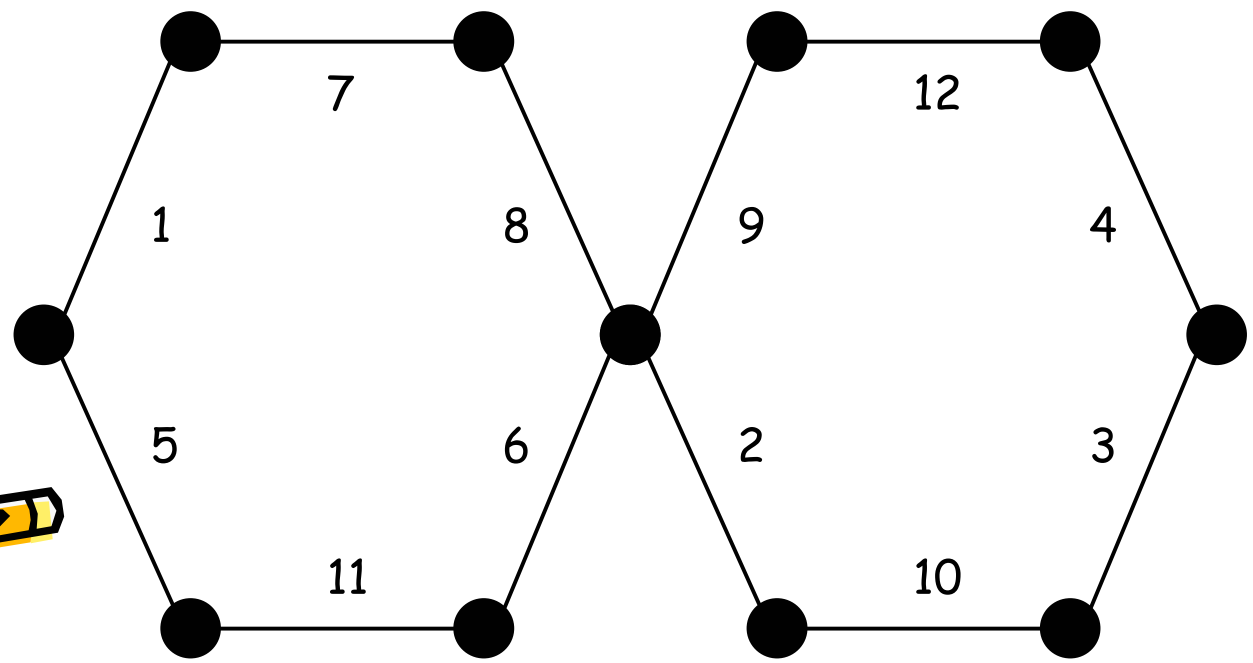
Mutation via Transpositions

- Transpositions yield exponential time!



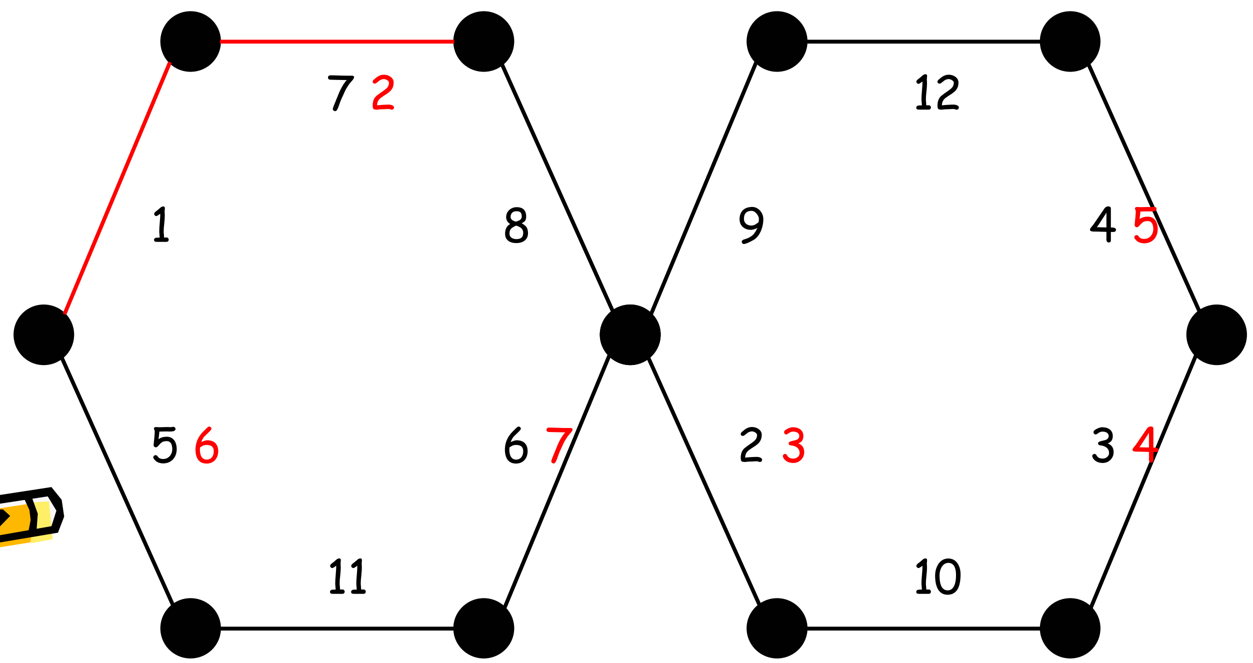
Mutation via "Jumps"

- Jump 7»2 means "7»2»3»4»5»6»7".



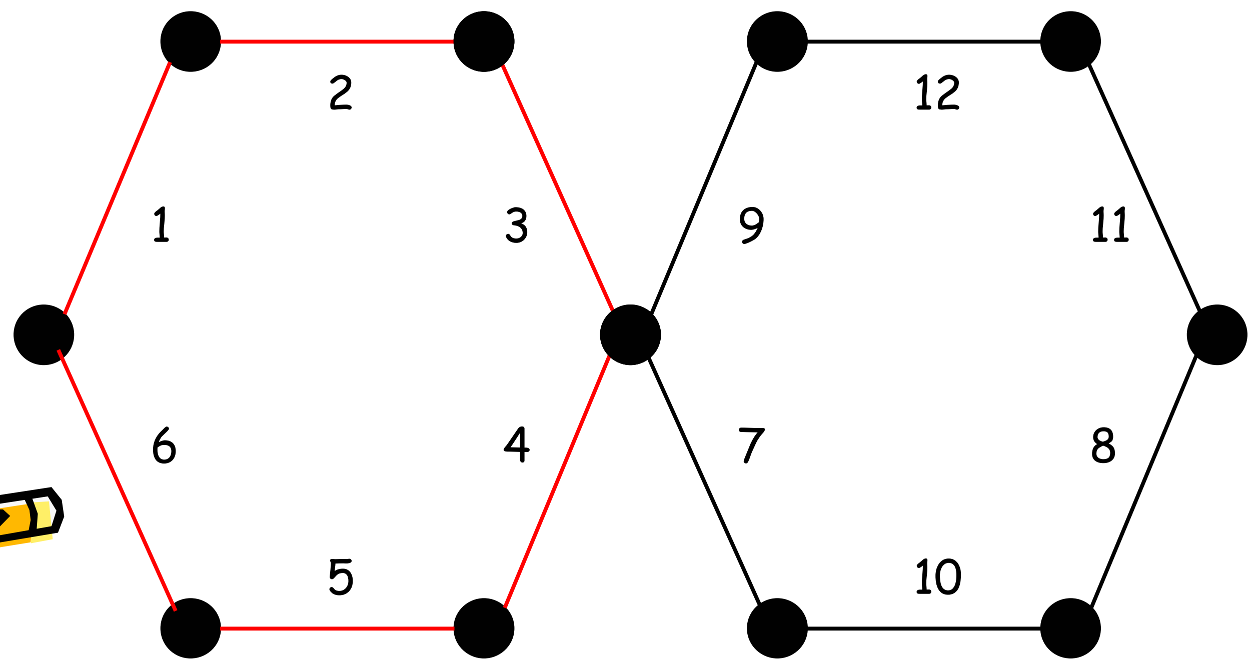
Mutation via "Jumps"

- Jump 7»2 means "7»2»3»4»5»6»7".
- Jump 7»2 `as useful as' exchange (27).
- Probability $\Theta(m^{-2})$.



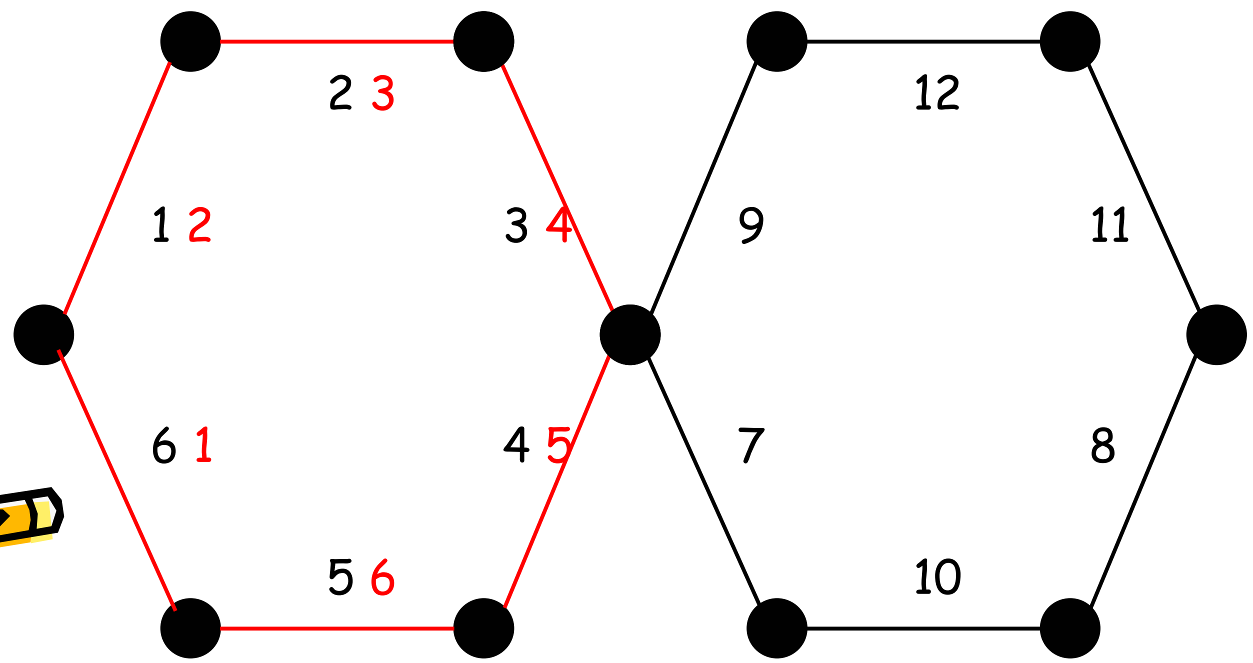
Mutation via Jumps

- Again after some operations...
- What now?



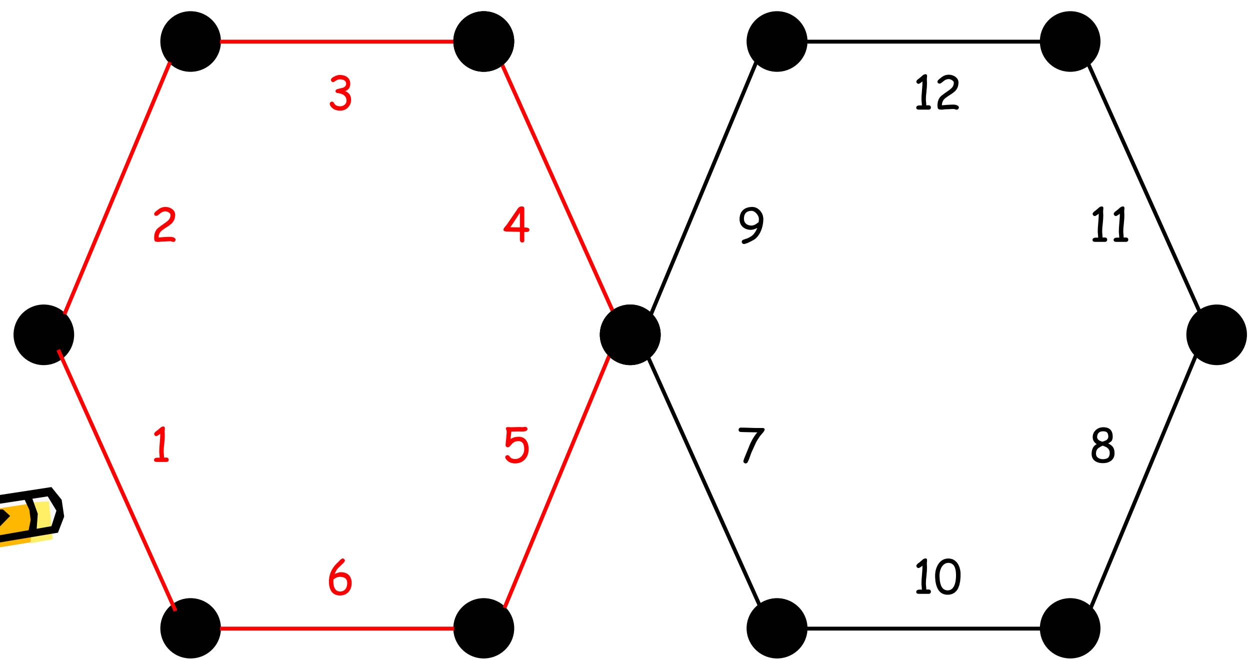
Mutation via Jumps

- Again after some operations...
- Jump 6»1 possible!



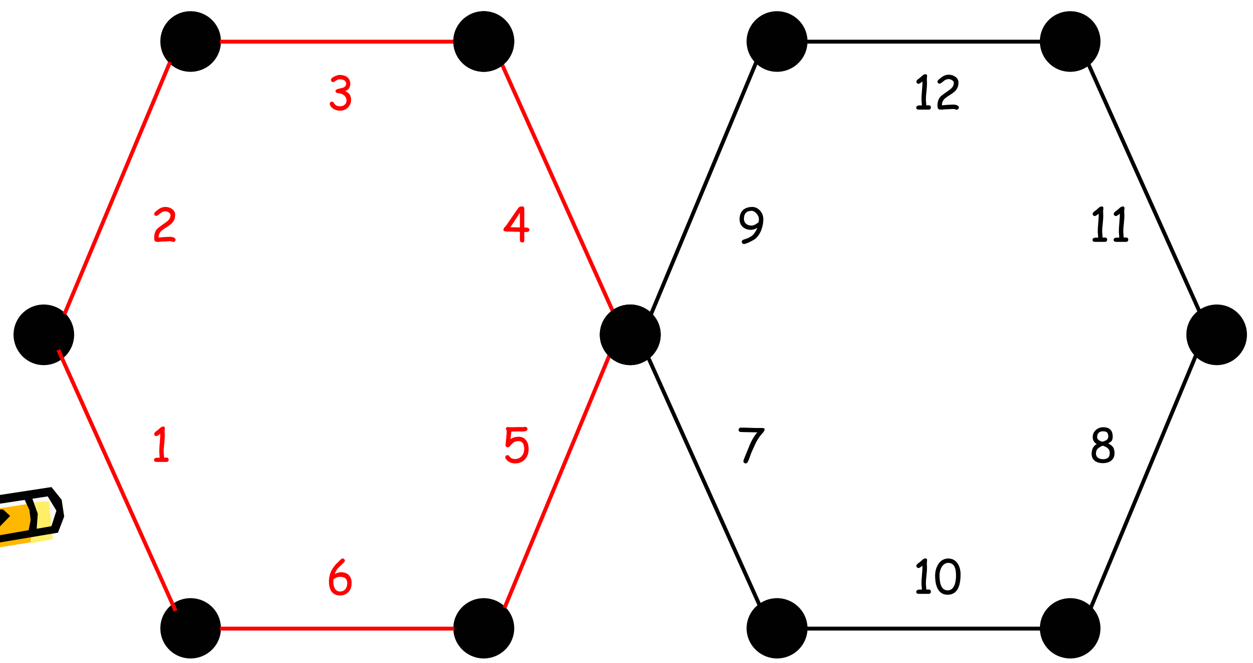
Mutation via Jumps

- Jump 6»1 possible!
- Shift the cycle left/right with probability $\Theta(m^{-2})$.
- Random Walks: $\Theta(k^2)$ random shifts needed for '+k'.



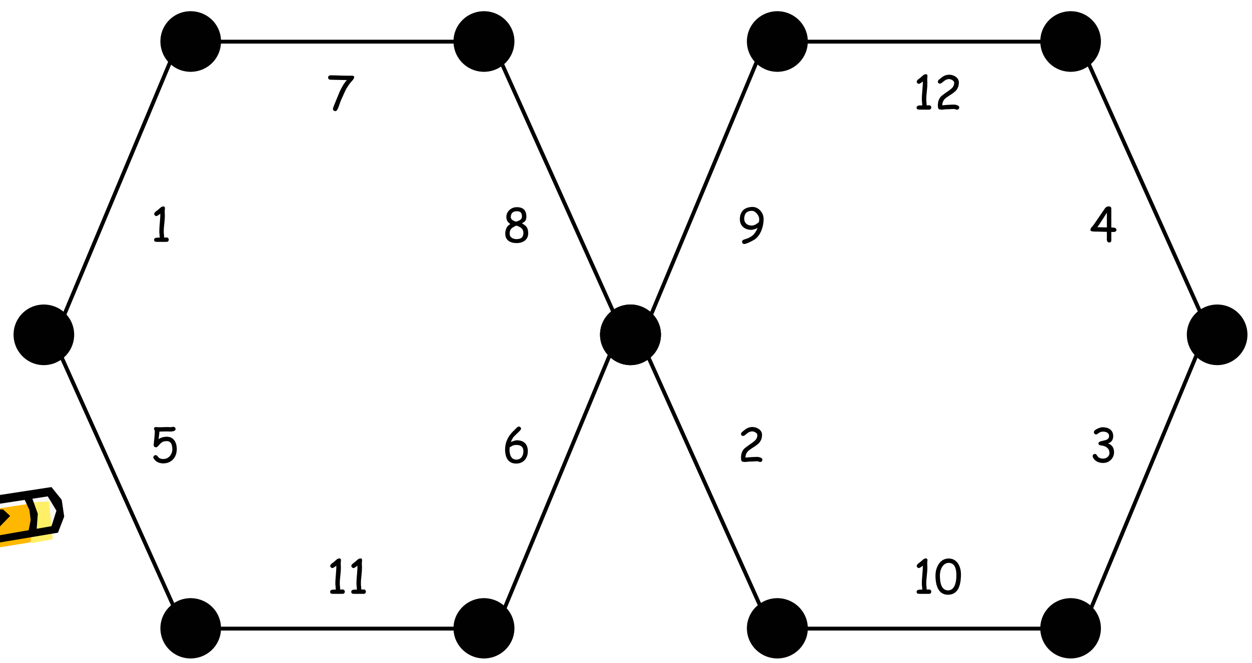
Mutation via Jumps: Summary

1. Add a fitting edge: $O(m^2)$ time.
2. Shift the cycle until 1. is possible: $O(m^2)$ time.
3. Do 1. and 2. m times: Total time $O(m^5)$.



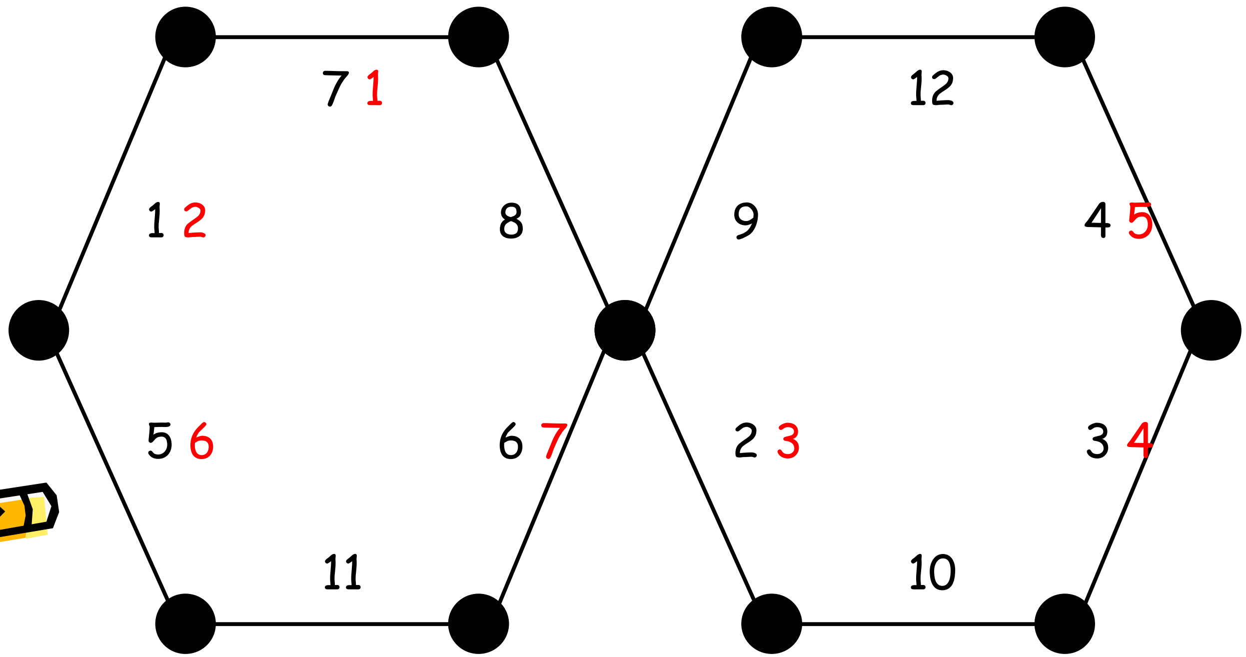
Restricted Jumps: …»1

- Before: Jump 7»2 means "7»2»3»4»5»6»7".
- Also possible: 7»1!



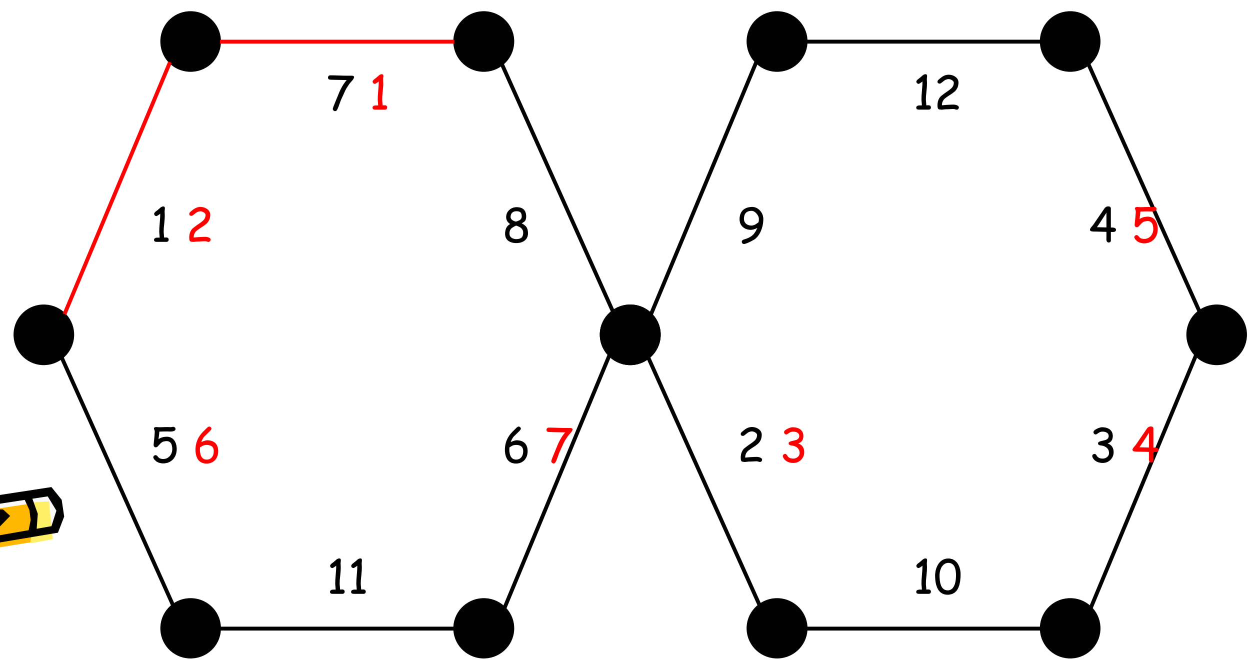
Restricted Jumps: …»1

- Before: Jump 7»2 means "7»2»3»4»5»6»7".
- Also possible: 7»1! [means 7»1»2»3»4»5»6»7]



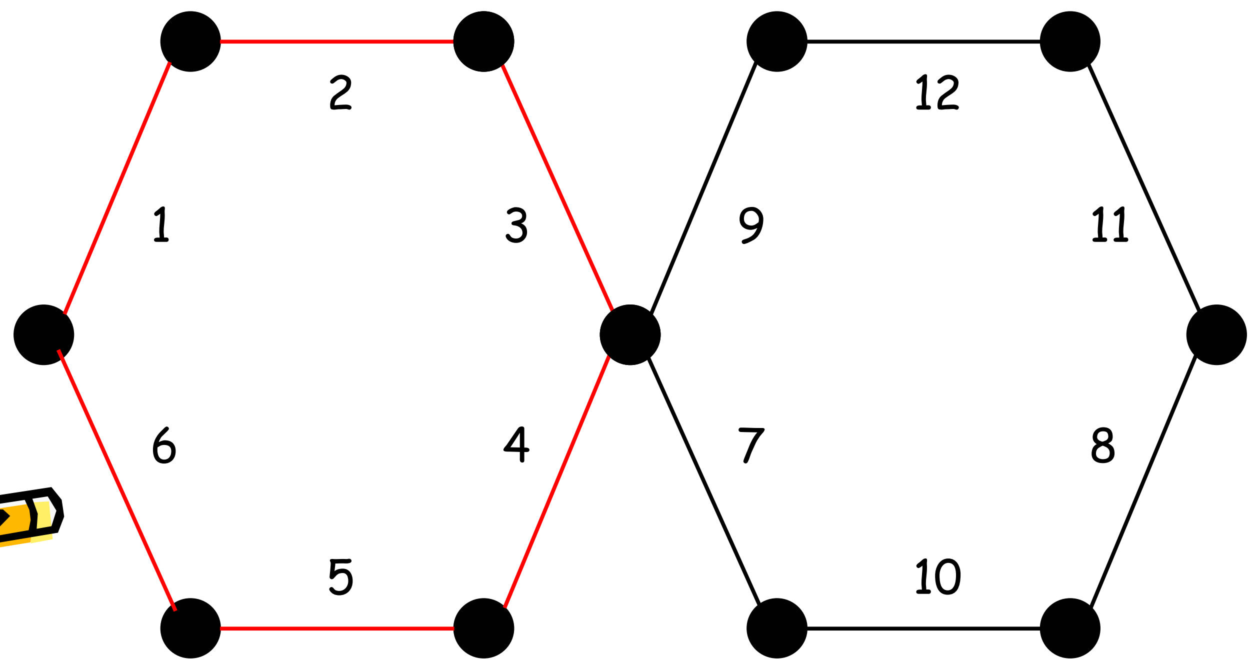
Restricted Jumps: …»1

- Before: Jump 7»2 means "7»2»3»4»5»6»7".
- Also possible: 7»1! [means 7»1»2»3»4»5»6»7]
- Add a fitting edge: O(m) time [before O(m^2)].



Restricted Jumps: …»1

- How about shifts?
- Before: 6»1 and 1»6 were possible.
- Now: Only shifts in one direction (no random walk).



Jumps and Restricted Jumps

Jumps [Neumann (2004)]:

1. Add a fitting edge: $O(m^2)$ time.
2. Shift the cycle until 1. is possible: $O(m^2)$ time.
3. Do 1. and 2. m times: Total time $O(m^5)$.

Restricted Jumps [D., Hebbinghaus, Neumann (2006)]:

1. Add a fitting edge: $O(m)$ time.
2. Shift the cycle until 1. is possible: $O(m)$ time.
3. Do 1. and 2. m times: Total time $O(m^3)$.

Outlook/Discussion

- Result: Restricted Jumps save $O(m^2)$ time.
- More clever mutation operators for other problems?
- Dichotomy:
  - Search for clever operators: Faster, ...
  - Concept of meta-heuristics: General, easy to use, no thinking required.