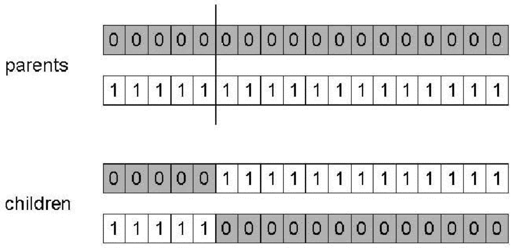
Jenis-jenis crossover

- 1-point crossover

  Choose a random point on the two parents

  Split parents at this crossover point

  Create children by exchanging tails



0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

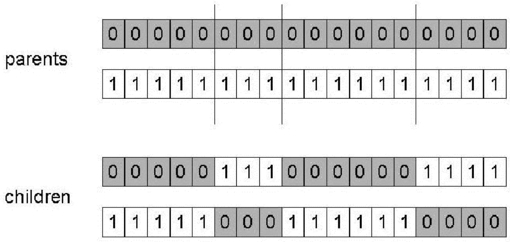
0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

- n – point crossover

  choose n random crossover points

  split along those point

  glue parts, alternating between parents



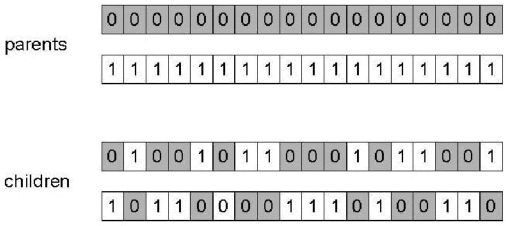
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1
1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0

- uniform crossover

  Flip a coin for each gene of the first child

  Make an inverse copy of the gene for the second child

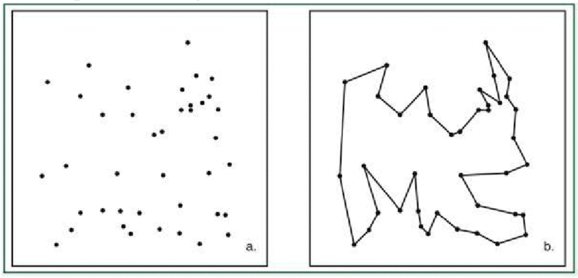


0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

0 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 1
1 0 1 1 0 0 0 0 1 1 1 0 1 0 0 1 1 0

TSP Theory

- Given a number of cities and the costs of travelling from any city to any other city, what is the cheapest round-trip route that visits each city exactly once ant then returns to the starting city?
- Finding a closed path that visits all cities



- Representation : (4, 1, 7, 2, 5, 3, 6) - list of cities visited
- Representation is a permutation, however standard crossover results in descendants that are not permutations

GA untuk kasus permutasi : TSP example

- Insert mutation for permutation
  Pick two allele values at random
  Move the second to follow the first, shifting the rest along to accommodate
  1 2 3 4 5 6 7 8 9 → 1 2 5 3 4 6 7 8 9
- Swap mutation for permutation
  Pick two alleles at random and swap their positions
  1 2 3 4 5 6 7 8 9 → 1 5 3 4 2 6 7 8 9
- Inversion mutation for permutation
  Pick two alleles at random and then invert the substring between them
  1 2 3 4 5 6 7 8 9 → 1 5 4 3 2 6 7 8 9
- Scramble mutation for permutation
  Pick a subset of genes at random
  Randomly rearrange the alleles in those positions
  1 2 3 4 5 6 7 8 9 → 1 3 5 4 2 6 7 8 9
- Crossover operator for permutation (informal procedure)
  Choose an arbitrary part from the first parent
  Copy this part to the first child
  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
  1 2 3 4 5 6 7 8 9
  → 3 8 2 4 5 6 7 1 9
  9 3 7 8 2 6 5 1 4