Example: The Travelling Salesperman Problem

Travelling Salesman Problem (TSP)

- Problem: A salesman must visit every city in his territory exactly once and return to his starting city. Given the cost of travel between all cities, how should he plan his itinerary to minimize the total cost of his tour?

Representations and Operators

- binary representation and classical cross-over and mutation operators not suitable
  - cause illegal tours to form
    - not all cities visited
    - undefined city codes
    - cities visited more than once
    - loops formed within the tour
  - needs repair algorithms

Representations and Operators

Example: 5 cities $\Rightarrow$ 3 bits per city

individual 1: 0|10101011001100

individual 2: 1|01010001100011

After cross over:

individual 1: 001010001100011

individual 2: 110101011001100

Representations and Operators

- other representations:
  - vector representations
    - adjacency representation
    - ordinal representation
    - path representation
  - matrix representations
- require special cross-over and mutation operators

Adjacency Representation

- tour listed as a list of n cities
- city j is listed in position i if and only if the tour leads from city i to j.
- some may represent illegal tours
- repair algorithms may be necessary
- does not support classical cross-over operators

Example 1:

1 2 3 4 5 6 7 8 9

(2 4 8 3 9 7 1 5 6)

$\Downarrow$

1 - 2 - 4 - 3 - 8 - 5 - 9 - 6 - 7

Example 2:

1 2 3 4 5 6 7 8 9

(2 4 8 1 9 3 5 7 6)

$\Downarrow$

1 - 2 - 4 - 1

Adjacency Representation

- three cross-over operators defined:
  - alternating edges cross-over
  - subtour chunks cross-over
  - heuristic cross-over

Adjacency Representation

- alternating edges cross-over
  - builds an offspring by:
    - chooses (at random) an edge from the first parent
    - selects an appropriate edge from second parent
    - if the new edge (from one of the parents) causes a cycle, a random edge that does not introduce a cycle is selected
    - repeated until tour completed

Adjacency Representation

- (*alternating edges cross-over cntd.*)

Example: From the two parents given, a possible offspring can be as in o_1.

1 2 3 4 5 6 7 8 9

parent 1: (2 3 8 7 9 1 4 5 6)

parent 2: (7 5 1 6 9 2 8 4 3)

⇓

o_1:(2 5 8 7 9 1 6 4 3)

Note: (7,6) was introduced randomly instead of (7,8)

Ordinal Representation

- a tour: list of *n* cities
- ith element of list: number in range *1* to *n-i+1*
- C: ordered list of cities serving as reference point
- tour obtained using chromosome and reference list
- classical cross-over can be used

Example:

reference list: *C*: (1 2 3 4 5 6 7 8 9)

chromosome (list of references) *l*: (1 1 2 1 4 1 3 1 1)

represents the tour 1 2 4 3 8 5 9 6 7

Ordinal Representation

- classical cross-over can be used

Example: given C: (1 2 3 4 5 6 7 8 9)

p1: (1 1 2 1 | 4 1 3 1 1) p2: (5 1 5 5 | 5 3 3 2 1)

corresponding to the tours

1-2-4-3-8-5-9-6-7 and 5-1-7-8-9-4-6-3-2

gives as offspring

(1 1 2 1 5 3 3 2 1) and (5 1 5 5 4 1 3 1 1)

corresponding to

1-2-4-3-9-7-8-6-5 and 5-1-7-8-6-2-9-3-4

Path Representation

- the most natural representation

Example: (5 1 7 8 9 4 6 2 3) represents the path 5 - 1 - 7 - 8 - 9 - 4 - 6 - 2 - 3

- three cross-overs
  - partially mapped (PMX)
  - order (OX)
  - cycle (CX)

PMX

- builds offspring by
  - choose a subsequence of a tour from one parent
    - choose two random cut points to serve as swapping boundaries
    - swap segments between cut points
  - preserve the order and position of as many cities as possible from other parent

PMX

Example:

p1: (1 2 3 | 4 5 6 7 | 8 9)
p2: (4 5 2 | 1 8 7 6 | 9 3)
step 1: swap segments
o1: (x x x | 1 8 7 6 | x x)
o2: (x x x | 4 5 6 7 | x x)
this also defines mappings:
1⇔4, 8⇔5, 7⇔6, 6⇔7
step 2: fill in cities from other parents if no conflict
o1: (x 2 3 | 1 8 7 6 | x 9)
o2: (x x 2 | 4 5 6 7 | 9 3)
step 3: use mappings for conflicted positions
o1: (4 2 3 | 1 8 7 6 | 5 9)
o2: (1 8 2 | 4 5 6 7 | 9 3)

More Cross-Over Operators

- class of heuristic operators emphasizing edges rather than cities
  - randomly select a city to be the current city c of offspring
  - select edges (two from each parent) incident to current city c
  - define probability distribution over selected edges based on their cost (probability for an edge to a previously visited city is 0)

More Cross-Over Operators

  - select an edge. if at least one edge has non-zero probability, selection based on above random distribution, else selection is random (from unvisited cities)
  - city on the other end of selected edge becomes current city
  - repeat until tour completed
- tests report only 60% of edges come from parents, 40% are random!

Edge Recombination Cross-Over (ER)

- transfers more than 95% from parents to the single offspring

for tour (3 1 2 8 7 4 6 9 5) the edges are
(3,1) (1,2) (2,8) (8,7) (4,6) (9,5) (5,3)

- general idea is that an offspring should exclusively be built from edges present in both parents

ER

Example:

p1: (1 2 3 4 5 6 7 8 9)
p2: (4 1 2 8 7 6 9 3 5)
edge list:
city 1: edges to 9 2 4
city 2: edges to 1 3 8
city 3: edges to 2 4 9 5
city 4: edges to 3 5 1
city 5: edges to 4 6 3
city 6: edges to 5 7 9
city 7: edges to 6 8
city 8: edges to 7 9 2
city 9: edges to 8 1 6 3

select initial city from one parent
assume city 1 (connected to 9 2 4)
choose one with less connections
(2 or 4) assume city 4
cities 3 and 5 are candidates
5 selected (less edges)
o1: (1 4 5 x x x x x x)
continue with procedure
o1: (1 4 5 6 7 8 2 3 9)

Note: all edges from two parents

(more modifications exist in literature)

Mutation Operators

- displacement mutation
  Example: (1 2 3 4 5 6 7 8 9)
  (3 4 5) selected and inserted after 7
  new tour: (1 2 6 7 3 4 5 8 9)
- exchange (swap) mutation
  Example: (1 2 3 4 5 6 7 8 9)
  3rd and 5th selected randomly
  new tour becomes (1 2 5 4 3 6 7 8 9)
- insertion mutation
  Example: (1 2 3 4 5 6 7 8 9)
  4 is selected randomly and placed after 7
  new tour becomes (1 2 3 5 6 7 4 8 9)

Mutation Operators

- simple inversion mutation
  Example: (1 2 3 | 4 5 6 7 | 8 9)
  new tour becomes (1 2 3 7 6 5 4 8 9)
- inversion mutation
  Example: (1 2 3 4 5 6 7 8 9)
  (3 4 5) selected and inserted after 7
  new tour becomes (1 2 6 7 5 4 3 8 9)
- scramble mutation
  Example: (1 2 3 4 5 6 7 8 9)
  (4 5 6 7) selected
  new tour may become
  (1 2 3 5 6 7 4 8 9)