Heuristic Optimisation

Part 11: Genetic algorithm for TSP

Sándor Zoltán Németh

http://web.mat.bham.ac.uk/S.Z.Nemeth

s.nemeth@bham.ac.uk

University of Birmingham

Overview

- Heuristic optimisation methods for solving TSP
- GA representations and operators:
  - Adjacency representation
  - Ordinal representation
  - Path representation
  - Matrix representation
- Incorporating local search methods
- A fast GA for TSP

Heuristic optimisation for TSP

An optimal solution: 5 hours on the best computer

A very close to optimal solution: seconds on a PC

Algorithms that generate approximate solutions:

nearest neighbour, greedy

Local optimisation:

2-opt, Lin-Kernighan

Evolutionary algorithms:

new methods still coming up,
no perfect algorithm has been found

GA for TSP

Approaches differ in the used representation and variation operators

Main problem: the GA is too slow to solve really big problems (10,000 cities)

TSP with 100 cities can be solved by global optimisation within hours

Comparisons are done on publicly available test problems with documented optimal or best-known solution (TSPLIB)

Variation operators

Evaluation for TSP : the length of the tour

Representation: list of cities

Binary representation?

Lot of work & time to repair the infeasible individuals generated by crossover, mutation

Adjacency representation

City j is in position $i \Leftrightarrow$ the tour contains (i, j)

position	1	2	3	4	5	6	7	8	9
representation	2	4	8	3	9	7	1	5	6
tour	1	2	4	3	8	5	9	6	7

Each tour has only one representation, but illegal tours can be represented

position	1	2	3	4	5	6	7	8	9
representation	2	4	8	1	9	3	5	7	6
tour	1	2	4	1					

Crossover for adjacency representation

Alternating edges:

position	1	2	3	4	5	6	7	8	9
parent 1	2	3	8	7	9	1	5	4	6
parent 2	7	5	1	6	9	2	8	4	3
offspring	2	5	8	7	9	1	6	4	3

Subtour-chunks: alternating random length subtours from the two parents

Templates associated with good solutions (schemata):
* * * 3 * 7 * * *

Poor results!

Ordinal representation

The *i*th element of the list is city *j* from the remaining cities, unvisited so far

position	1	2	3	4	5	6	7	8	9
representation	1	1	2	1	4	1	3	1	1
tour	1	2	4	3	8	5	9	6	7

$j \in \{1, 2, \ldots, n - i + 1\}$

Classical one-point crossover generates legal tours!

The partial tour to the left of the crossover point remains unchanged

Crossover for ordinal representation

position	1	2	3	4	5	6	7	8	9
p_1 representation	1	1	2	1	4	1	3	1	1
p_2 representation	5	1	5	5	5	3	3	2	1
o_1 representation	1	1	2	1	5	3	3	2	1
o_2 representation	5	1	5	5	4	1	3	1	1
p_1 tour	1	2	4	3	8	5	9	6	7
p_2 tour	5	1	7	8	9	4	6	3	2
o_1 tour	1	2	4	3	9	7	8	6	5
o_2 tour	5	1	7	8	6	2	9	3	4

Path representation

The most natural representation:

position	1	2	3	4	5	6	7	8	9
representation	5	1	7	8	6	2	9	3	4
tour	5	1	7	8	6	2	9	3	4

Partially-mapped crossover (PMX)

$$
\begin{aligned}
p_1 &= (123|4567|89) \\
p_2 &= (452|1876|93) \\
o_1 &= (xxx|1876|xx) \\
o_2 &= (xxx|4567|xx)
\end{aligned}
$$

Mappings: $1 \leftrightarrow 4, 8 \leftrightarrow 5, 7 \leftrightarrow 6, 6 \leftrightarrow 7$

Fill in the no-conflict xs:

$$
\begin{aligned}
o_1 &= (x23|1876|x9) \\
o_2 &= (xx2|4567|93)
\end{aligned}
$$

Use the mappings for the remaining xs:

$$
\begin{aligned}
o_1 &= (423|1876|59) \\
o_2 &= (182|4567|93)
\end{aligned}
$$

Order crossover (OX)

$$
\begin{aligned}
p_1 &= (123|4567|89) \\
p_2 &= (452|1876|93) \\
o_1 &= (xxx|4567|xx) \\
o_2 &= (xxx|1876|xx)
\end{aligned}
$$

Starting from the second cut point the cities from the other parent are copied in the same order, omitting the ones already present:

$$
\begin{aligned}
p_2 : 934521876 &\rightarrow 93218 \\
p_1 : 891234567 &\rightarrow 92345 \\
o_1 &= (218|4567|93) \\
o_2 &= (345|1876|92)
\end{aligned}
$$

Cycle crossover (CX)

$$
\begin{aligned}
p_1 &= (123456789) \\
p_2 &= (412876935) \\
o_1 &= (1xxxxxxxx) \\
o_1 &= (1xx4xxxxx) \\
o_1 &= (1xx4xxx8x) \\
o_1 &= (1234xxx8x) \\
o_1 &= (123476985)
\end{aligned}
$$

CX preserves the absolute position of elements in the parent sequence

Edges?

Most operators we discussed take into consideration cities and not links between cities, i.e. edges

But the linkage of a city with other cities is more important than its position in a tour (the distance is associated to the edge)!

An edge recombination operator would transfer the edges from parent to offspring most of the time

An edge list is constructed for each city in the two parents

Matrix representation

Precedence binary matrix M:

$m_{ij} = 1 \Leftrightarrow$ city i is before city j in the tour

	1	2	3	4	5	6	7	8	9
1	0	1	0	1	1	1	1	1	1
2	0	0	0	1	1	1	1	1	1
3	1	1	0	1	1	1	1	1	1
4	0	0	0	0	1	1	0	0	1
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	1
7	0	0	0	1	1	1	0	0	1
8	0	0	0	1	1	1	1	0	1
9	0	0	0	0	1	0	0	0	0

Tour (312874695)

Properties of matrix representation

- The number of 1s is $\frac{n(n-1)}{2}$
- $m_{ii} = 0$ for all $1 \leq i \leq n$
- If $m_{ij} = 1$ and $m_{jk} = 1$ then $m_{ik} = 1$

Intersection and union operators

Intersection of bits (1s) from two parents results in a matrix which can be completed to a legal tour

Union of bits can be applied to disjoint subsets from the two parents

The results are better than with path representation

Incorporating local search methods

A huge amount of effort was spent on inventing suitable myred representations and a myred recombination operator that preserves partial tours

GA results cannot compete with the Lin-Kernighan algorithm (neither quality nor time)

Local search can be incorporated into the genetic algorithm

Before evaluation, a local optimisation is applied to each individual

Results are much better than GA only

The inver-over algorithm

Each individual competes only with its offspring

There is only one adaptive variation operator

The number of times this operator is applied to an individual during a generation is variable

Quickest evolutionary algorithm for TSP

Only three parameters

Good precision and stability for relatively small problems

Recommended reading

Z. Michalewicz & D.B. Fogel
How to Solve It: Modern Heuristics

Chapter 8. The Traveling Salesman Problem