Heuristic Optimisation

Problem sheet 5

Brief solutions

1. Consider the tabu search algorithm for a Travelling Salesperson Problem with 5 cities based on the neighbourhood defined by the 2-swap mapping. The frequency based memory is given by the following table.

2	3	4	5	
0	2	3	3	1
	2	1	3	2
		2	3	3
			1	4

How many times was city 4 swapped with another city in the last 20 iterations? Justify your answer.

Solution

Each entry (i, j) of the table shows how many times has city i swapped with city j during the number of iterations for which the frequency based memory was constructed called horizon. The horizon can be calculated by summing up the element of the table. Thus, the horizon is $0 + 2 + 3 + 3 + 2 + 1 + 3 + 2 + 3 + 1 = 20$ iterations. To see how many times was city 4 swapped with another city we have to sum up the numbers in the column of 4 and the line of 4. Hence, city 4 was swapped $7 = (3 + 1 + 2) + 1$ times with another city.

2. Consider the basic representation of genetic algorithms and the bit strings $Parent_1 = 1010101010$ and $Parent_2 = 1001001001$. Apply a one-point crossover between the fifth and sixth bit. What are the children (or offsprings) of this crossover?

Solution

$Children_1$ is obtained by taking the first five bits from $Parent_1$ and the next five bits from $Parent_2$. $Children_2$ is obtained by taking the first five bits from $Parent_2$ and the next five bits from $Parent_1$. Since

$$Parent_1 = \mathbf{10101\,01010}$$

and

$$Parent_2 = 10010\,01001,$$

we get

$$Children_1 = \mathbf{10101}\,01001$$

and

$$Children_2 = 10010\,\mathbf{01010}.$$

3. Consider the genetic algorithm for a Traveling Salesperson problem with 7 cities and the tours

position	1	2	3	4	5	6	7
tour	3	6	4	5	7	1	2

position	1	2	3	4	5	6	7
tour	5	2	7	3	4	6	1

given by the path representation. Find an offspring of these tours for the alternative edge crossover of their adjacency representation. Show the offspring by using both the adjacency and path representations.

Solution:

In the path representation a tour is the same as its representation. Therefore, below **tour** means path representation of a tour and **representation** means the adjacency representation of a tour. The *s mean the random changes due to premature cycles.

position	1	2	3	4	5	6	7
p_1 representation	2	3	6	5	7	4	1
p_2 representation	5	7	4	6	2	1	3
o representation	2	7	4	5	6*	1*	3*
p_1 tour	3	6	4	5	7	1	2
p_2 tour	5	2	7	3	4	6	1
o tour	1	2	7	3	4	5	6

4. Consider the genetic algorithm for a Traveling Salesperson problem with 7 cities and the tours

position	1	2	3	4	5	6	7
tour	5	3	4	1	2	7	6

position	1	2	3	4	5	6	7
tour	2	1	7	6	5	4	3

given by the path representation. Do a one-point crossover with the cutting point between positions 3 and 4 of the ordinal representations of the tours. Show the offsprings of the tours by using the path representation.

Solution:

In the path representation a tour is the same as its representation. Therefore, below **tour** means path representation of a tour and **representation** means the ordinal representation of a tour.

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

5. Consider the genetic algorithm for a Travelling Salesperson problem with 7 cities and the tours

position	1	2	3	4	5	6	7
tour	3	6	4	5	7	1	2

position	1	2	3	4	5	6	7
tour	5	2	7	4	3	6	1

given by the path representation. Find two offspring of these tours for the cycle crossover.

Solution:

CX preserves the absolute position of elements in the parent sequence, that is, each city and its position comes from one of the parents:

$$\begin{aligned} p_1 &= (3645712) \\ p_2 &= (5274361) \\ o_1 &= (3\text{xxxxxx}) \\ o_1 &= (3\text{xx}5\text{xxx}) \\ o_1 &= (3\text{x}45\text{xxx}) \\ o_1 &= (3\text{x}457\text{xx}) \end{aligned}$$

Here x denotes the unfilled positions. Thus, we have completed a cycle. The remaining cities together with their positions are filled in from the other parent p_2:

$$o_1 = (3245761)$$

Starting with parent p_2, we get:

$$\begin{aligned} p_1 &= (3645712) \\ p_2 &= (5274361) \\ o_2 &= (5\text{xxxxxx}) \\ o_2 &= (5\text{xxx}3\text{xx}) \\ o_2 &= (5\text{x}743\text{xx}) \end{aligned}$$

Here “x” denotes the unfilled positions. Thus, we have completed a cycle. The remaining cities together with their positions are filled in from the other parent p_1:

$$o_2 = (5674312)$$