Artificial Intelligence **2023/2024**

Student: LUCA PERNIGO StudentID:19-993-658

Assignment 4

Due date: 22 December 2023, 23:59

Monte Carlo Tree Search for Orienteering Problem

Monte Carlo Tree Search has been applied to the Orienteering Problem.

1. Target Problems

The instances considered are 0, 1 and 2. They all belong to the set of small size problems.

2. Implementation

2.1. treePolicy

The treePolicy method of the MCTS class expands a node if the selected node is not completely expanded, otherwise we move to its best child in the tree using the **selectBestChild** function.

2.2. selectBestChild

For every node, its children are stored inside the attribute **expanded**. For every node we update the UCT score for each of its child node then we return the best child according to such score.

2.3. backup

This function takes as input a node, then increases by one the number of times this node has been visited and we update its estimated reward. Finally we move back to its parent and repeat until we reach the root.

3. Evaluation Strategy

In order to evaluate the MCTS implementation the problem instances were run with different seeds. The results are reported in table 1

		TotalTime	Reward	Feasibility
ProblemInstance	Seed			
instance0	0	274.000000	2.130000	True
	42	274.000000	1.660000	True
	666	274.000000	1.430000	True
instance1	0	302.000000	1.680000	True
	42	302.000000	1.350000	True
	666	301.000000	1.740000	True
instance2	0	565.000000	2.070000	True
	42	565.000000	2.780000	True
	666	565.000000	2.530000	True

Table 1

Next, results were averaged along the seeds, they are reported in 2.

	TotalTime	Reward	Feasibility
ProblemInstance			
instance0	274.000000	1.740000	True
instance1	301.666667	1.590000	True
instance2	565.000000	2.460000	True

Table 2

4. Variant Implementation

In order to improve performance, the **calculateUCTscore** has been modified; what has been tried is giving more weight to the first element of the score function. That is the element corresponding to exploration has been multiplied by two. Doing so we get the following results.

		TotalTime	Reward	Feasibility
ProblemInstance	Seed			
instance0	0	274.000000	1.720000	True
	42	274.000000	1.470000	True
	666	274.000000	1.960000	True
instance1	0	302.000000	1.550000	True
	42	302.000000	1.600000	True
	666	302.000000	1.800000	True
instance2	0	565.000000	3.120000	True
	42	565.000000	2.270000	True
	666	565.000000	3.000000	True

Table 3

	TotalTime	Reward	Feasibility
ProblemInstance			
instance0	274.000000	1.716667	True
instance1	302.000000	1.650000	True
instance2	565.000000	2.796667	True

Table 4

Observing the results, we can conclude that weighting more the exploration factor of the UCT score results in higher rewards as the problem size increases. That is for instance1 and instance2 the variant achieves an higher reward.