Dynamic Optimization and Traveling Salesman Problem

October 27, 2023

1 Information

- Due Date: 5th November 2023
- Upload solution in Moodle

2 The Problem, [10 Marks]

Consider a modified version of the four-city traveling salesman problem of

- Example 1.2.3 (page 24) in the book `https://web.mit.edu/dimitrib/www/RLCOURSECOMPLETE.pdf`
- Lecture 15 of class `https://courses.iiit.ac.in/pluginfile.php/168696/mod_resource/content/1/lec15.pdf`

where there is a fifth city E. The intercity travel costs are shown in Fig. 1.

■	5	1	20	10
20	■	1	4	10
1	20	■	3	10
18	4	3	■	10
30	10	0	10	■

Figure 1: Intercity cost

1. Use exact DP with starting city A to verify that the optimal tour is $ABDECA$ with cost 20.

2. Verify that the nearest neighbor heuristic starting with city A generates the tour $ACDBEA$ with cost 48.

3. Apply rollout with one-step lookahead minimization, using as base heuristic the nearest neighbor heuristic. Show that it generates the tour $AECDBA$ with cost 37.

4. Apply rollout with two-step lookahead minimization, using as base heuristic the nearest neighbor heuristic. This rollout algorithm operates as follows. For $k = 1, 2, 3$, it starts with a k-city partial tour, it generates every possible two-city addition to this tour, uses

the nearest neighbor heuristic to complete the tour, and selects as next city to add to the k-city partial tour the city that corresponds to the best tour thus obtained (only one city is added to the current tour at each step of the algorithm, not two). Show that this algorithm generates the optimal tour.

5. (Optional) Estimate roughly the complexity of the computations in parts (a), (b), (c), and (d), assuming a generic N-city traveling salesman problem.

   Answer: The exact DP algorithm requires $O(N^N)$ computation, since there are

$$(N-1) + (N-1)(N-2) + \ldots + (N-1)(N-2)\ldots 2 + (N-1)(N-2)\ldots 1$$

   arcs in the DP graph to consider, and this number can be estimated as $O(N^N)$. The nearest neighbor heuristic that starts at city A performs $O(N)$ comparisons at each of N stages, so it requires $O(N^2)$ computation. The rollout algorithm at stage k runs the nearest neighbor heuristic $N-k$ times, so it must run the heuristic $O(N^2)$ times for a total computation of $O(N^4)$. Thus the rollout algorithm's complexity involves a low order polynomial increase over the complexity of the base heuristic, something that is generally true for practical discrete optimization problems. Note that even though this may represent a substantial increase in computation over the base heuristic, it is a potentially enormous improvement over the complexity of the exact DP algorithm.