CS 486/686 Assignment 2

Due Date: 11:59pm on Thursday, October 25, 2018

Instructions

- Submit the assignment in the Dropbox labeled Assignment 2 Submissions on LEARN.
- No late assignment will be accepted.
- This assignment is to be done individually.
- For any programming question, you may use the language of your choice. We highly recommend using Python if you are familiar with it.
- Lead TA: Camilo Bravo (camunozb@uwaterloo.ca). Camilo's office hours will be posted on the course website.

1 Local Search (50 points)

Question 1 (50 points) (Local search on the Traveling Salesperson Problem)

You will solve the traveling salesperson problem using several local search algorithms including **hill climbing, hill climbing with sideway moves/tabu list, hill climbing with random restarts and simulated annealing**.

The setup of this problem is exactly the same as Question 2 in Assignment 1. Feel free to re-use any of your code from your Assignment 1 submission.

The data set for this problem is a subset of the data set for Assignment 1. This dataset only contains the 14-city, 15-city, and 16-city problem instances as well as the single 36-city problem instance. You can find the problem instances in the file `tsp_problems.zip` posted on the course website.

One way to measure the performance of a local search algorithm is to compare the quality of the best solution found by the local search algorithm to the best solution found by any alternative algorithm. For a given local search algorithm, let C_{LS} be the cost of the best solution found by a local search algorithm and let C_B be the cost of the best solution found by any alternative algorithm. The performance of the local search algorithm is given by

$$\frac{C_{LS}}{C_B}.$$

Since we are minimizing the cost of a solution, this fraction should be greater than or equal to one.

Here are a few examples of alternative algorithms.

- The Concorde online solver

  `https://neos-server.org/neos/jobs/6310000/6312670.html`
- Your A* algorithm from Assignment 1
- The best local search algorithm that you can come up with

For each problem instance, feel free to share the best solution you found on Piazza.

Please complete the following tasks.

1. (5 points) Formulate the TSP problem as a local search problem. Include the definitions of a state, the neighbor relation, and the cost function.

2. (10 points) **Consider the 14, 15, 16-city instances.** Solve the problem instances using **hill climbing (with no sideway moves, no tabu list and no random restarts)** and report your findings.

   Hill climbing is a randomized algorithm (since the initial state is chosen at random). Thus, we will run the algorithm multiple times and report its average performance across the repetitions.

   For each number of cities and each problem instance, run hill climbing for 100 times and report the following averages (where the averages is taken over the 100 repetitions of the algorithm)

   - The average number of steps it takes for hill climbing to reach a local optimum
   - The average quality of the best solution
   - The percentage of 100 repetitions where hill climbing found the same solution as the best solution found by the alternative algorithm

   Next, for each number of cities, calculate the average of each of the three numbers above over the ten instances. Report the three averages for each number of cities. How do the three average numbers change as the number of cities increases? Describe and discuss any trend that you can observe in no more than one paragraph.

3. (5 points) **Choose any instance with at least 5 cities from the data set for Assignment 1.** Choose one strict local optimum found by hill climbing for any problem instance. Confirm that it is a strict local optimum by providing a sample calculation. Recall that a state is a strict local optimum if and only if the best neighbor of the state has strictly higher cost than the current state.

4. (5 points) **Consider the 14, 15, 16-city instances.** Will the **hill climbing** algorithm perform better if it **allows sideway moves and/or maintains a tabu list**? Perform an investigation and report your findings.

   There are many ways to compare the performance of hill climbing with or without sideway moves and/or a tabu list. Feel free to use the examples in part 2 or come up with other performance measures. Be sure to clearly explain what criteria you use to measure the performance of the algorithm.

   If your answer is yes, provide evidence that adding sideway moves and/or a tabu list improves the performance of hill climbing.

   If your answer is no, provide evidence that adding sideway moves and/or a tabu list will not improve the performance of hill climbing.

5. (10 points) **Consider the 14, 15, 16-city instances.** Solve the problem instances using **hill climbing with random restarts** and report your findings.

   Consider the **first two instances for each number of cities**. For each instance, experiment with different numbers of random restarts. For each instance and each number of restarts, run hill climbing with a given number of random restarts for 100 times. For each instance, plot the average solution quality and the average execution time with respect to the number of random restarts (where the average is taken over the 100 repetitions).

   Then, answer the following questions.

   - For each number of cities, how many restarts is sufficient to ensure that the solution is within 1% of the best solution found by any alternative algorithm? (That is, the quality of the solution is less than or equal to 1.01.)
   - For each number of cities, is there a trade-off between the execution time and the average solution quality? Discuss your observations in no more than one paragraph.
   - Based on your findings, what number of restarts would you choose for each number of cities and why?

6. (10 points) **Consider the 14, 15, 16-city instances.** Solve the problem instances using **simulated annealing** and report your findings.

   Consider the **first two instances for each number of cities**. For each instance, experiment with at least three different annealing schedules. For each instance and each annealing schedule, run simulated annealing for 100 times. For each instance, plot the average solution quality and the average execution time with respect to the annealing schedule (where the average is taken over the 100 repetitions).

   Then, answer the following questions.

   - For each number of cities, is there a trade-off between the execution time and the choice of annealing schedule? Discuss your observations in no more than one paragraph.
   - For each number of cities, which annealing schedule would you choose for each number of cities and why?

7. (5 points) **Consider the 36-city instance.** Based on your findings above, which local search algorithm would be the best choice to solve the 36-city problem instance? Experiment with different local search algorithms and let each algorithm run for no more than 5 minutes. Report the best algorithm, the best solution found by the algorithm and the cost of the best solution found.

2 Probability (50 points)

Consider four Boolean random variables A, B, C, D. You are given the following information on the prior and conditional probabilities for these variables.

- The prior probabilities of A are defined as follows.
  $P(A) = 0.6$.
- The conditional probabilities of B given A are defined as follows.

  $P(B|A) = 0.8$ $\quad$ $P(B|\neg A) = 0.4$

- The conditional probabilities of C given A and B are defined as follows.

  $P(C|A \wedge B) = 0.7$ $\quad$ $P(C|A \wedge \neg B) = 0.9$
  $P(C|\neg A \wedge B) = 0.7$ $\quad$ $P(C|\neg A \wedge \neg B) = 0.9$

- The conditional probabilities of D given A, B, and C are defined as follows.

  $P(D|A \wedge B \wedge C) = 0.6$ $\quad$ $P(D|\neg A \wedge B \wedge C) = 0.6$
  $P(D|A \wedge B \wedge \neg C) = 0.6$ $\quad$ $P(D|\neg A \wedge B \wedge \neg C) = 0.6$
  $P(D|A \wedge \neg B \wedge C) = 0.2$ $\quad$ $P(D|\neg A \wedge \neg B \wedge C) = 0.2$
  $P(D|A \wedge \neg B \wedge \neg C) = 0.2$ $\quad$ $P(D|\neg A \wedge \neg B \wedge \neg C) = 0.2$

Please answer the following questions.

1. (8 points) Calculate the joint probability distribution over the four variables. Show the steps of your calculations for at least one probability in the joint distribution.
2. (12 points) Calculate the following probabilities. Round your final answers to three decimal places. For each calculation, be sure to show the steps of the calculations using only the probabilities provided or calculated in previous parts.
   (a) $P(D)$
   (b) $P(A \wedge C)$
   (c) $P(A|\neg B)$
   (d) $P(A|B \wedge \neg C)$
   (e) $P(A \wedge \neg B|C)$

(f) $P(\neg A \wedge C | \neg B \wedge D)$

3. (30 points) Based on the probabilities provided and calculated, answer the following questions. Justify your answers by providing probability calculations to prove/disprove unconditional/conditional independence using their definitions.

   (a) Are A and C independent?

   (b) Are A and C conditionally independent given another variable? If so, which variable is it?

   (c) Are C and D independent?

   (d) Are C and D conditionally independent given another variable? If so, which variable is it?

   (e) Are A and B independent?

   (f) Are A and B conditionally independent given another variable? If so, which variable is it?