vrije Universiteit amsterdam

Faculty of Economics and Business Administration

Student name:	
Student number:	

Exam: Heuristic Optimization Methods

Code: E_EOR3_HOM

Examiner: Joaquim Gromicho & Daniele Vigo
Co-reader: Wout Dullaert & Leen Stougie

Date: October 28, 2016

Time: 15:15

Duration: 2 hours
Calculator allowed: Yes

Graphical calculator allowed: Yes

Number of questions: 3 main questions divided in a total of 10 sub-questions

Type of questions: Open

Answer in: English

Remarks:

- Write down your name and student number in the box on top of this front page.
- Exams without name or student number will **NOT** be graded!
- Answers to each question **MUST** be in the reserved space.
- Some questions include pre-drawn graphs for your convenience, use them to draw your answer.
- Draft paper is also provided in this bundle: this is the only paper you will receive!
- You **MUST** submit this set without taking out the staples or any of the papers!

Credit score: 10 = (0.5+0.5+1.0) + (0.5+1.5+1.0+0.5) + (1.5+1.5+1.5)

Grades: The grades will be made public on: November 11

Inspection: November 16 from 11:00 till 12:30 in room HG-01A41

Number of pages: (20 (including front page))

Good luck!

Dear Students,

The exam consists of three open questions, each one accounting for a given number of credits reported on the front page. Each question has a number of sub questions and also the partial credit score is clearly indicated in the same order as the sub questions.

As you know, this written exam accounts for 80% of the final grade, and the remaining 20% comes from your weekly assignments.

Keep in mind a few things:

a) you have roughly 40 minutes for each question, so manage your time wisely,
b) you should answer in the space provided after each sub question,
c) sometimes partial graphs are added in the answer space for your convenience: you may use them to draw your answer or the steps taken to find it,
d) if you cannot answer one of the sub questions, you can skip it and move on to the next,
e) you should be able to solve sub questions regardless of your success on other sub questions.

Note that four pages of draft paper are attached at the end of this document.

This set of 20 pages is the **only paper you are allowed to use** and you must **deliver it as a whole**.

Best of luck!

Daniele Vigo & Joaquim Gromicho

I

Consider the instance of the binary knapsack problem given by:
$p = (10, 10, 15, 10, 10)$, $w = (45, 40, 80, 5, 10)$ and $c = 95$.

I.1 Apply the Greedy heuristic to it. Which solution do you get?

I.2 Apply the modified Greedy heuristic to it (the heuristic that becomes an approximation guaranteed to deviate at most twice from the optimum). Which solution do you get?

I.3 Solve the fractional version of the problem. What can you say about the optimal value?

II

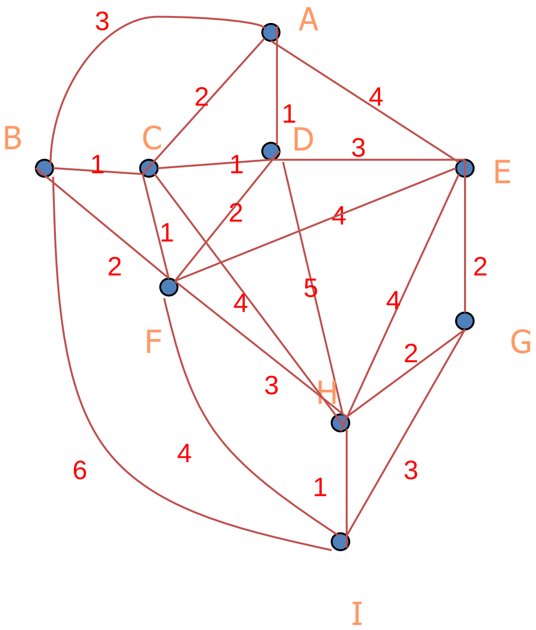
The Nearest Addition heuristic can be applied to the TSP. At each step a subtour is expanded by the addition of one node until this subtour becomes a tour visiting all vertices. The heuristic can be described as follows:

Initialization: start at one node (can be chosen arbitrarily) and initialize subtour T to include that node.

Iteration: until T does not include all nodes find nodes i in T and node j not in T which minimize the addition cost c_{ij} and add j to T after or before i depending on which gives the cheapest increase in cost $c_{ij} + c_{jk} - c_{ik}$ being k the node that was previously either the successor or the predecessor of i.

II.1 Determine the order of the running time of this algorithm.

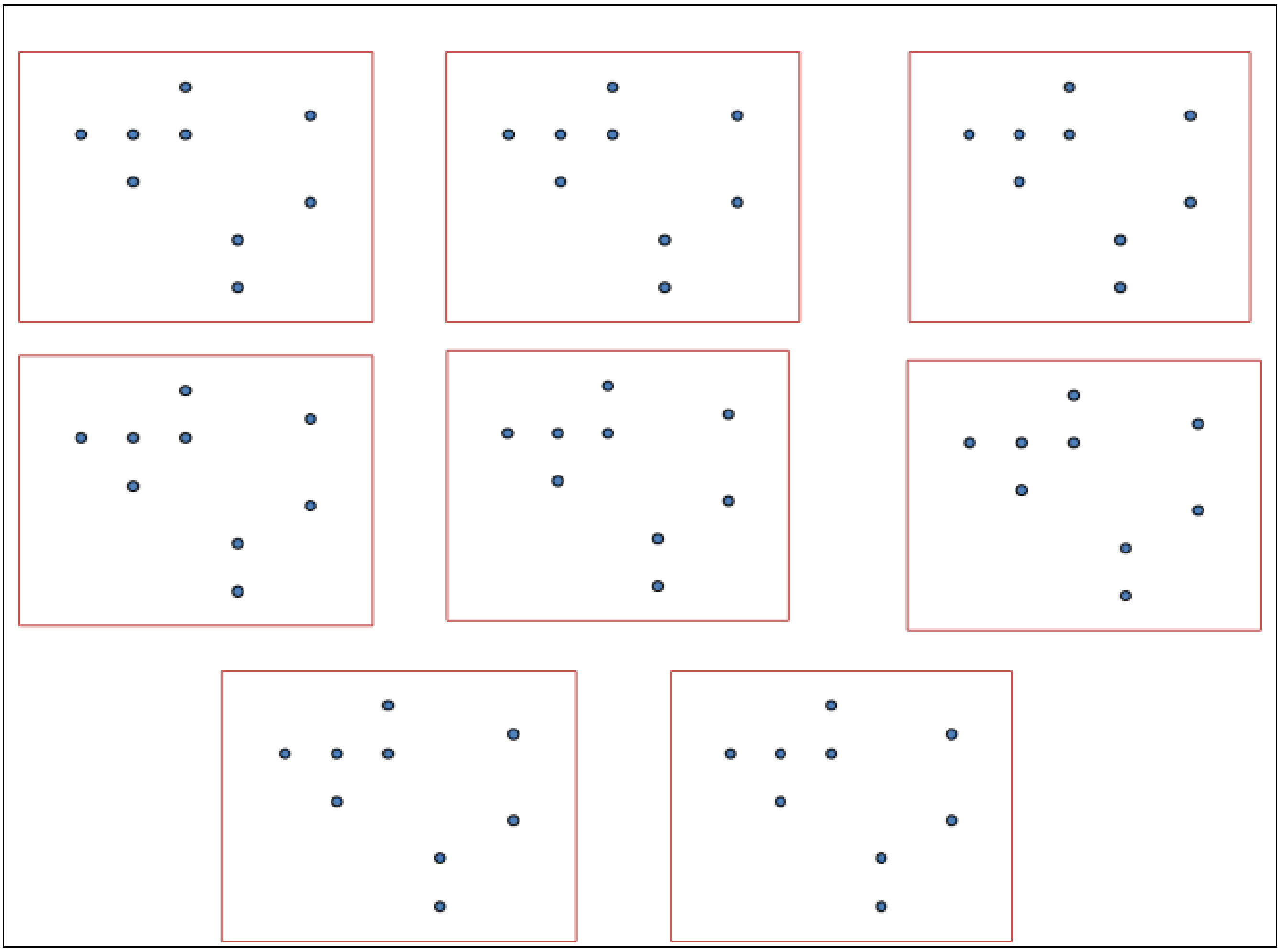
II.2 Consider the graph in the picture on the right and apply the nearest addition algorithm to it starting at node I.



You can consider the omitted edges as having cost equal to the length of shortest path connecting the corresponding nodes, but the given edges should suffice.

Note: when your subtour consists of only one node it plays the role of both i and k meaning that the insertion cost after finding the nearest addition node is $c_{ij} + c_{jk} - c_{ik} = c_{ij} + c_{ji} - 0$ in that case.

Additional note: you may use the graphs below to draw the iterations and supply any additional computations you may find relevant in subsequent page.



II.3 Prove that if the TSP instance is metric then the Nearest Addition heuristic can never produce a solution with a value higher than twice the optimum (i.e. it yields a 1-approximation)**.**

Note: you do not need to provide a very formal proof, the main arguments (if correct) suffice. Be, however, warned that you should need to make use of the fact that the instance is metric since you know that no efficient approximation is known for the general TSP.

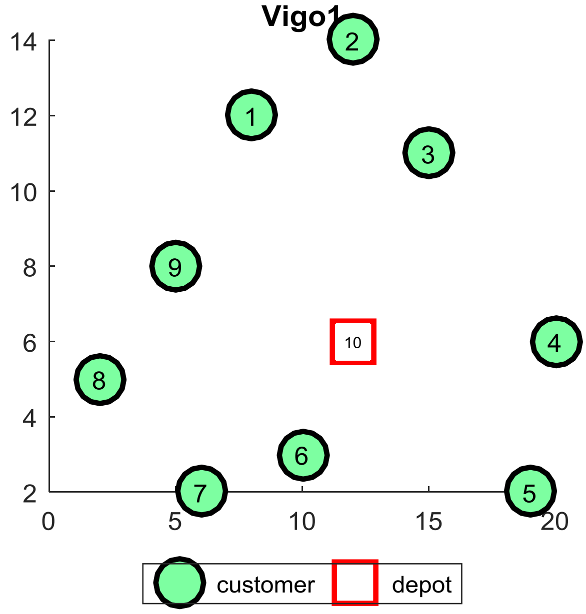
Hint: remember the proof that the algorithm of Prim is optimal for the Minimum Spanning Tree.

II.4 Show that the instance above is metric. What can you say about the optimal value with respect to the value of the solution found in **II.1**?

III

Consider your well-known Vigo1 instance of the Capacitated vehicle routing problem as depicted on the right. The demand vector is $q = (5,3,2,3,6,2,2,3,2,0)$ and the vehicle capacity is $Q = 10$. The (rounded) distances are:

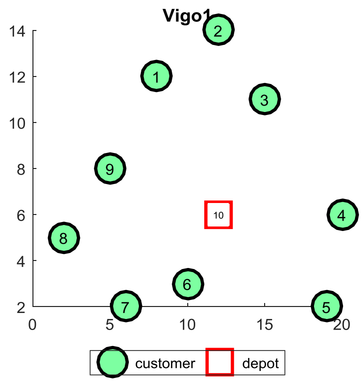
	1	2	3	4	5	6	7	8	9	10
1	0	4	7	13	15	9	10	9	5	7
2	4	0	4	11	14	11	13	13	9	8
3	7	4	0	7	10	9	13	14	10	6
4	13	11	7	0	4	10	15	18	15	8
5	15	14	10	4	0	9	13	17	15	8
6	9	11	9	10	9	0	4	8	7	4
7	10	13	13	15	13	4	0	5	6	7
8	9	13	14	18	17	8	5	0	4	10
9	5	9	10	15	15	7	6	4	0	7
10	7	8	6	8	8	4	7	10	7	0

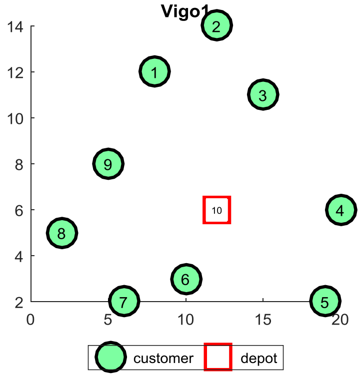


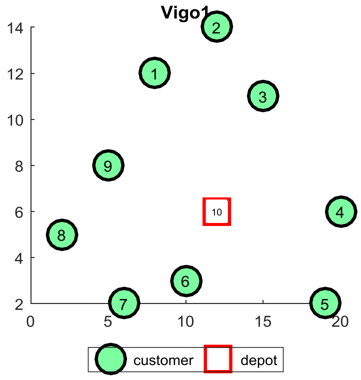
The angles of the position vector of each customer, in degrees, are:
$\alpha = (123.7, 90.0, 59.0, 0.0, -29.7, -123.7, -146.3, -174.3, 164.1)$.

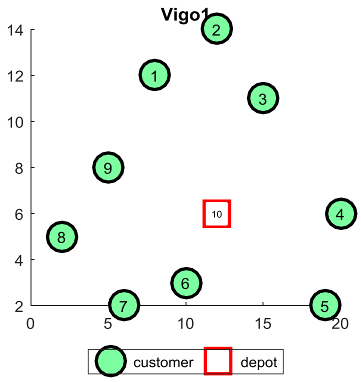
III.1 Apply the sweep algorithm to the Vigo1 instance. To this end, start at the angle -180.0. and take the customers in increasing order of their angles in steps (buckets) of 45 degrees, breaking ties within the same bucket by considering the demand in increasing order. (**Note:** This corresponds to the criteria `+(45)angle, +(0)quantity` applied to the function SweepSequence of your code repository.) Which solution do you obtain?

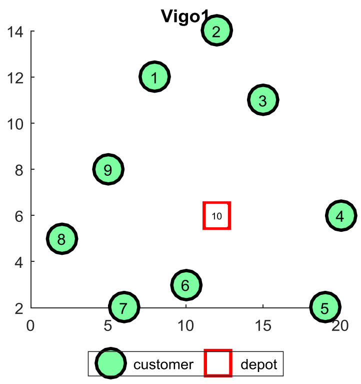
You may use the following pictures to represent the evolution of the sweep and/or report the detail of the computation.




Vigo1
customer
depot


Vigo1
customer
depot


Vigo1
customer
depot

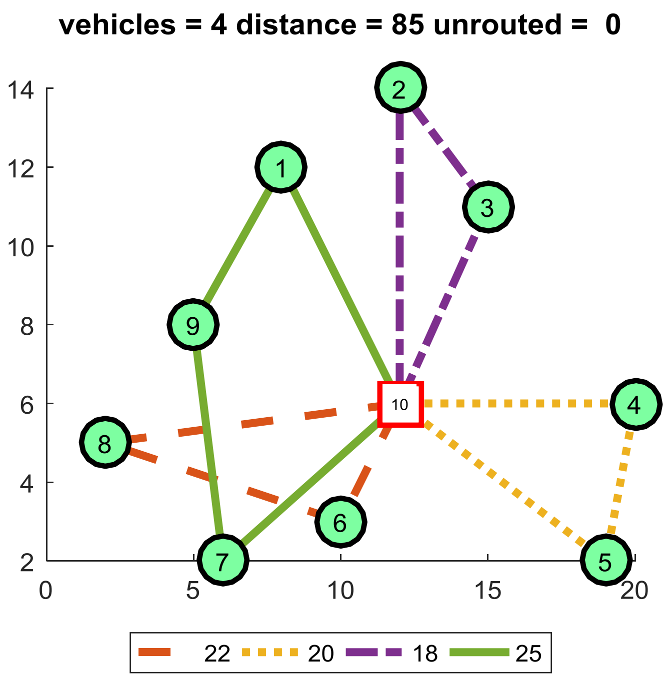

Vigo1
customer
depot

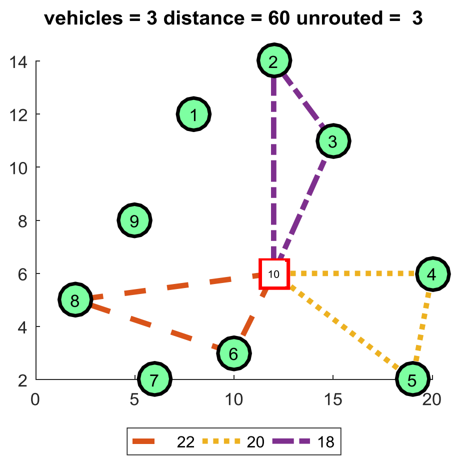
III.2 Consider the solution on the right and apply to it an iteration of the Ruin and Recreate algorithm.

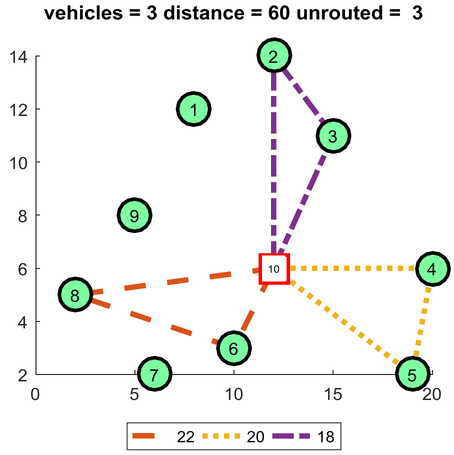
'Ruin' it by removing the route that visits customers 1, 7, 9 and 'recreate' it by means of cheapest insertion.

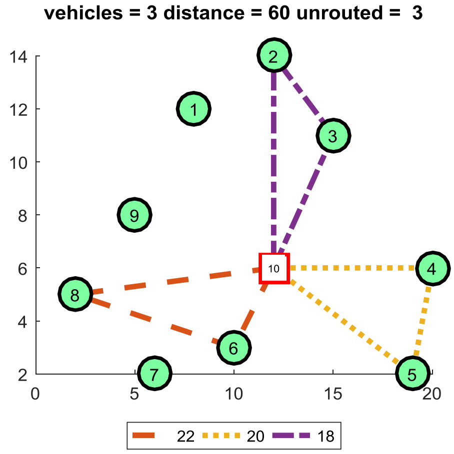
Which solution do you obtain?

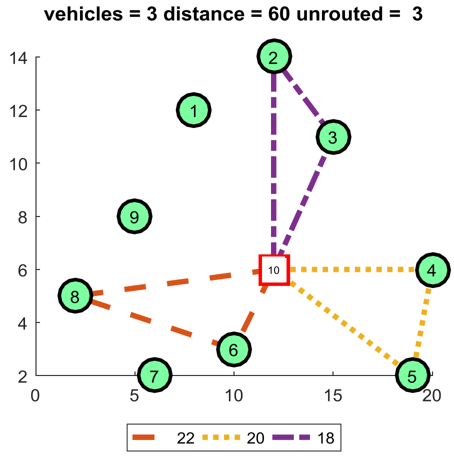
You may use the following pictures to represent the evolution of the Cheapest Insertion and/or report the detail of the computation.



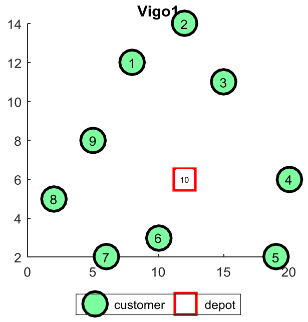


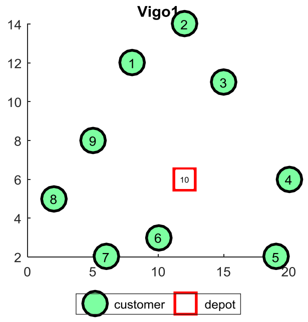

vehicles = 3 distance = 60 unrouted = 3
14
12
10
8
6
4
2
0
5
10
15
20
1
2
3
4
5
6
7
8
9
10
22
20
18

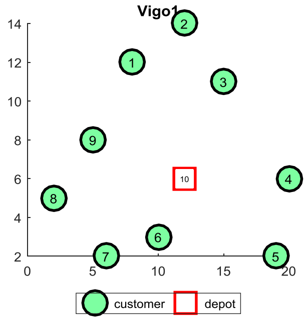

vehicles = 3 distance = 60 unrouted = 3
14
12
10
8
6
4
2
0
5
10
15
20
1
2
3
4
5
6
7
8
9
10
22
20
18

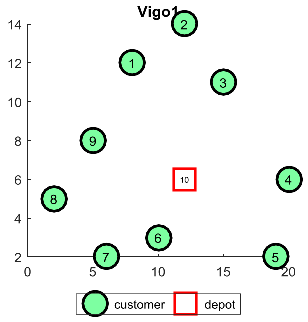

vehicles = 3 distance = 60 unrouted = 3
14
12
10
8
6
4
2
0
5
10
15
20
1
2
3
4
5
6
7
8
9
10
22
20
18

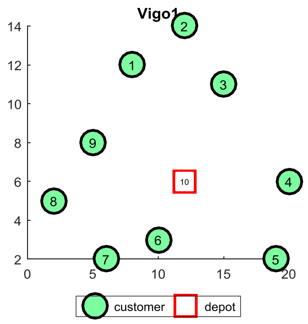
III.3 Suppose now that in instance Vigo1 some of your customers are backhauls. How would you adapt the sweep algorithm to cope with this situation? Apply your algorithm to Vigo1 but now with 2 and 8 being backhauls. Start at angle 0 and sweep counterclockwise. Again, you may use the figures below as an aid in your answer.

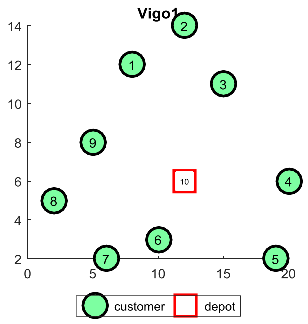


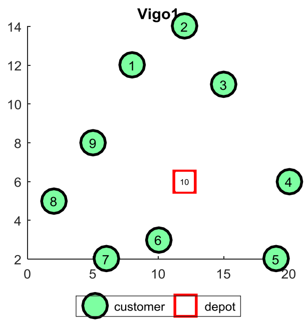


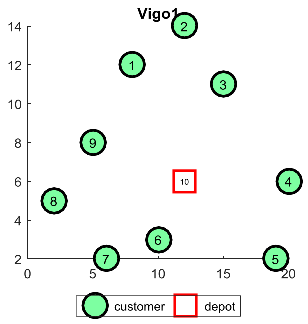












This is the end of the exam's question and answer area.

This and the following pages are your draft space.

This is the end of the exam booklet. Please keep it stitched and hand it in as a whole. Thank you!