380CT - Asis.pdf

by Asis Rai

Submission date: 14-May-2018 05:19AM (UTC+0100)
Submission ID: 89115140
File name: 71527_Asis_Rai_380CT_-_Asis_1685544_1475143125.pdf
Word count: 2742
Character count: 13190

380CT - Asis Rai

May 14, 2018

Name: Asis Rai
Student ID: 6528683
Coursework: 380CT Assignment
Module: Theoretical Aspects of Computer Science
Group Members: Asis Rai, Jubad Miah, Rakshak Nalukurthi & Muhammed Khan **[Shared Code and Pseudocode]**
GitHub Repository: https://github.coventry.ac.uk/raia10/380CT-CourseworkLateTeam

1 Notation

Let G be a directed graph, and let V be nodes of the graph G, $(V_1, V_2, \ldots, V_v)$ and E being the edges of G.

Hamilton Cycle is denoted by HC. We don't connect list entries with logical AND

Let P be *Path* being a list, given by $P = E_1 \wedge E_2 \wedge \cdots \wedge E_n$ symbols

Every *clause* in the path, E_n has the form $A \wedge B$, where A is the current node and B is the next node. Graph paths do not have clauses.

Number of nodes in the graph is be denoted by N. But 6 lines above you used little v for the number of nodes?

Degree of connectivity between nodes is denoted by C.

Distance between nodes is denoted by D. What distance? Distance is not relevant for DHCP.

2 Definition Of The Problem

...finding out if THERE EXISTS a...

The Hamiltonian Cycle problem is a problem of finding out if a Hamiltonian path which is a path ... in A ... in directed graph, in which a path visits every vertex of the graphs only once or a Hamiltonian cycle exists in a directed graph. (Johnson, 1979) graph singular or what?

3 Complexity Class Membership

3.0.1 Decisional Hamiltonian Cycle Problem

The decision problem takes input (G, V, E) and asks if there is a HC starting at V and ending at E, returns **YES** if there is a HC, **No** otherwise. (Johnson, 1979) correct

Hamilton Cycle Decisionional Problem in a directed path is **NP-Complete**. It is **NP-Complete** because DHP decision problem can be solved by a non-deterministic Turing Machine in polynomial Time. (West, 2017) It is NP complete. But the reason you gave is the definition of NP.

Optimisation problem not clearly defined

3.0.2 Computational/Search Hamiltonian Cycle

I think you mean that P is the cycle

If directed graph G has a HC, then return P to find the Hamiltonian Cycle. (www.ici.usi.edu, n.d.)

Computational/ Search Hamiltonian Cycle problem is **NP-Complete**. It is **NP-Complete** because it is a problem that can be solved by a non-deterministic Turning Machine in a polynomial Time. e.g. Finding the shortest P Hamiltonian cycle in a given G. (West, 2017)

Incorrect. This problem is a decision problem and so cannot be NP and hence is neither NP-Complete. It is NP hard though.

3.0.3 Optimization Hamiltonian Cycle

Find the HC, P with the shortest D by minising number of non-hamiltonian instances. (Johnson, 1979)

What is a non-Hamiltonian instance? This is not a clear optimisation problem

Optimization Hamiltonian Cycle is **NP-Hard** because Optimization probelms can only be **HP-Hard** as the task is to minimize D of nodes in the Hamiltonian P. (Jillian Beardwood, 2008)

Correct

4 Testing Methodology

How do you know the time increases before? Average over what?

Exhaustive Search: Average time increases for instances with increasing number of N.

Greedy Heuristic: Average time increases for instances with increasing number of N.

Why isn't greedy having its quality checked (it is also heuristic).

Meta-Heuristic: * Quality of approximation is calculated with increasing P, with the shortest D. * Average time increases for instances with increasing number of N.

The reason it wasn't implemented is not "deadlines". You had the whole Easter vacation to do it

NOTE: Since the code was not implemented by the team because of coursework deadlines leading to less time for the code to be developed, 3 algorithms will be compared with their O notation and their Big O notation instead. With suggestions on what kind of situation they will be best translated into. The accuracy and runtime performances of the three algorithms cannot be tested with inccreasing graph nodes and probablity of connection because the team could not develop the code on time.

You could have still presented a testing methodology though.

4.1 Random Graph

What is a Random Graph?

Random Graph is a graph in which properties, vertices, edges and connections between them are determined in random way. (Bollobas, 1985)

How to generate a Random Graph?

To generate a Random Graph, V is inputted from $(V_1, V_2, \ldots, V_v)$. Inputting number of nodes N, will generate a random number of edges and nodes.

GenerateGraphRandomly Function randomly generates a graph. N number of nodes, is given as vertex, it then creates the random graph, a new directed graph is created after that with the edges added into the new directed graph.

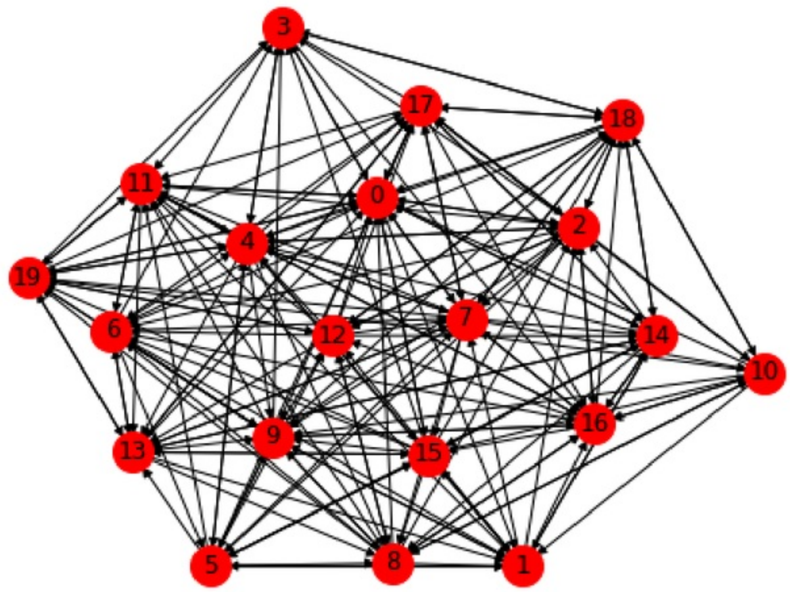
Yes, but how many graphs? What size and structure? How to avoid effect of outliers? etc.

4.2 Random Graph Generation - Code

```
In [2]: """Imported Modules"""
        import math
        import random
        import networkx as nx
        import matplotlib.pyplot as plt
        import sys
```

```python
"""Base Class Definition"""
class DirectedHamilton:
    #Variables for results #
    times = []
    start = None
    end = None

    #Setting Parameters #
    # Number of nodes, strategy #
    vertex = None
    structure = None
    graph = [[0 for i in range(20)] for j in range(20)]

    def __init__(self, n, g):
        self.vertex = n
        self.graph = g

"""Random Graph Generation"""
def GenerateGraphRandomly(self, n):
    # Generate a random directed graph of size n #
    # Probability between connectivity between nodes is defined #
    # 50% probability set, change it to whatever you want #
    G = nx.gnp_random_graph(n, 0.50, directed = True)
    # Setting directed graph which is empty #
    S = nx.DiGraph()

    # This will get the edgdes from the random graph #
    # The edges will be used to connect to the new one #
    S.add_edges_from(G.edges())

    # Creating a list which stores value of nodes #
    List = list(G.nodes())
    # Shuffle the list so that it's random #
    random.shuffle( List )

    # This wil make sure that there is a cycle in the graph #
    for i in range(len(List) - 1):
        # Adding the paths, so that every node is connected #
        S.add_path([List[i], List[i + 1]])

    # Making sure that final node is connected to final node #
    # This makes sure that the graph has a HC #
    S.add_path([List[-1], List[0]])

    # Setting up edges to be connected between nodes #
    # Default colour is black, therefore the colour chosen is black #
    edges = [edge for edge in S.edges()]
```

```
    # Setting the layout for the graph, through the module imported #
    # Force-directed graph drawing #
    # Kamada & Kawai (1989) #
    position = nx.kamada_kawai_layout(S)

    # Connected the nodes to the graph #
    # Setting the layout as 'Ocean' for the graph #
    # Setting the nodes colour to Red #
    nx.draw_networkx_nodes(S, position, cmap = plt.get_cmap('ocean'),
                           node_color = "Red", node_size = 400)

    # Set the number labels on the nodes, in the graph #
    nx.draw_networkx_labels(S, position)

    # Set the egdes with arrows in graph #
    nx.draw_networkx_edges(S, position, edgelist = edges, arrows = True)
    limits=plt.axis('off')

    return S

# Initialize GenerateGraphRandomly function #
DirectedHamilton.GenerateGraphRandomly = GenerateGraphRandomly

"""Giving values & Output for the graph generation"""

# Give number of nodes to start with #
vertex = 20

# Give maximum size of the layout #
plt.figure(figsize=(8, 6))
rand = DirectedHamilton(vertex, None)

# Calling the function to randomly generate graph #
graph = rand.GenerateGraphRandomly(vertex)
```



it's is short for "it is" which does not make sense in sentence below. You meant "its"

Generation of Hamiltonian graphs (yes-instances of DHCP)

The code is written to make sure that every node is connected with it's next node, leading connectivity to the final node to make sure that there is a connectivity between them and also to make sure that there is a Hamiltonian Cycle in the graph.

The generated graph from the code has nodes and arrows connected to each nodes and to the last, this makes sure that the generated graph is a directed graph and it has a Hamilton Cycle inside it.

5 Solution Methods

5.1 Exhaustive search - Pseudocode

This function header does not make sense. The job of exhaustive search is to find a path - so why would a path be given as input?

```
G = (V, E)
FUNCTION ExhaustiveSearch(P,position)
    HC = True;
    FOR each V in (V1, V2, ..., Vn)
        IF V not in P
            HC = False;
    IF HC:
```

```
        IF last V is adjacent to first N P from G
            Return True
        ELSE
            Return False
Else
    TRY different V in G
        IF current V and last V are adjacent in G
            Return False
        ELSE
            Check if V has been visited from G
        IF ExhaustiveSearch(P,P+1) = True
            Return True
        ELSE
            Remove current V if solution not found P[position] = 1
```

What does Try different V mean?

Undefined variables

Why give up search at this point?

I don't think this pseudocode is even close to a correct exhaustive search

5.1.1 Exhaustive search - Discussion

all possibilities of WHAT?

The Exhaustive search in practice, goes through all possbilities to check whether there is a Hamiltonian cycle in the graph. If it cannot find a Hamilton cycle then the search keeps going until it has traversed through every path in the graph. This means that if a Hamiltonian path is found then the Exhaustive search returns the most optimal path. However, if there is not Hamiltonian Cycle, then False is returned. (Kreinovich, 1987) The word I is always capitalised. You must know this?

It is exponential but this explanation does not justify it

If Exhaustive search was tested in practice, i believe the Exhaustive search would have exponential growth. This is because, the algorithm depends on the N of V given. If there are a large amounts of nodes V, the time to find an optimal solution would increase aswell, this means that the Exhaustive search will have to keep traversing around every single node in the graph. Therefore, in practice the algorithm depends on the number of nodes N inputted, if the size of the V is low then the algorithm would return an optimal path quickly however as the number of nodes N increases in the graph, the algorithms would take longer to find the optimal path and will have exponential growth.

Big O Nototation:

BigO Complexity: $O(n!)$

It is $O(n!)$ for this algorithm because every V is iterated and therefore will always be one remaining V, which gives notation O(n) and (n*(n-1)!) to find the optimal path. Therefore with every call, the worst factor is 0(n!) because every call reduces branch factor by 1.

5.2 Greedy Heuristic - Pseudocode

1. **FUNCTION** GreedyHeuristic()
2. FirstNode$V$ $\leftarrow$ First position
3. Path$P$ $\leftarrow$ []
4. **While** locations to travel
5. **For** neighbours, V of FirstNode$V$ do
6. **If** V has greatest number of neighbours then
7. FirstNode$V$ $\leftarrow$ V
8. **Append** FirstNode$V$ to Path$P$
9. **End While**
10. Return Path$P$

ok - this seems like a reasonable greedy heuristic for DHCP. The function header should probably include an input graph.

5.2.1 Greedy Heuristic - Discussion

The greedy algorithm traverses through the path by picking a random node and assining it as the startiting node, then it starts on the node to determine all the edges connected from the node, by traversing through one node at a time to find a path. It will traverse through the cheapest path, to find the cheapest fath.It will stop traversing and looking for paths when there are no nodes left to traverse to. It is an algorithm which tries to find a Hamiltonian path by visiting fewest number of nodes. (Stefan Voss, 1999)

Sometimes greedy algorithms fail to find the globally optimal solution because they do not consider all the nodes in the graph. The choice made by a greedy algorithm may depend on choices it has made so far, but it is not aware of future choices it could make. Therefore the path it returns is not always the optimal path. It is faster than Exhaustive but Exhaustive Search is more reliable. ok

Big O Nototation:

BigO Complexity: $O(N^2)$

Due to the fact that for each node, the near neighbour have to be located therefore giving the first n, computing distance between two nodes will get the factor 1 and it runs in $O(1)$. The travelling distance between one node to all the other cheapest node will give the factor of $O(n)$, making it ultimately $O(N^2)$. explanation a little confused, but O(N^2) reasonable

6 Meta-Heuristic - Tabu Search

1. **FUNCTION** TabuSearch() unevaluated Latex code
2. **While** \$Tabu < Tabu_m \$ do
3. $Tabu \leftarrow Tabu + 1$
4. **Search** Local Neighborhood
5. **Evaluate** Seek Best Solutions

Neither variable Tabu nor variable Tabu_m are defined or input. So what on earth does line 2 mean?

Define neighbourhood

Define best

6. **Update** Tabulist

7. **End while**

8. **Return** count of satisfied clauses

satisfied clauses has NOTHING to do with DHCP.

6.0.1 Tabu Search - Discussion

The Tabu Search algorithm maintains a tabu list of edges giving record of all the path visited. The algorithm begins with an initial *Path* P as a solution and then checks for other possible P solutions, comparing neighbouring solutions with the intitial solution. This is done by adding each solution in TabuList and evalutating the best one by removing the worst *Path* from the TabuList. The TabuList contains all the list of P visited, therfore aviding any illegal move and avioding groing around in circles.(Glover, 1986)

Big O Nototation:
BigO Complexity: $O(N^2)$

7 Special Cases

1. Singleton graph, where a graph which has just one V and no E. too trivial

2. Single V missing an E therefore not being able to connect to the neighbouring Vs'. too trivial

3. Complete Graph where each V is connected to another V in the graph.

4. Empty graph, it is a special case of the Singleton graph where the graph contains empty V and the V's have no edges. too trivial

5. Multiple V but zero E. too trivial

1,2,4,5 are all incredibly trivial. Not really worth discussing. 3 is quite trivial but I would have accepted it - however there is no discussion of how to check, cost of checking, likelihood of occurrence etc.

8 Conclusion

From the research, I can conclude that:

Your conclusion is meant to put a number on small/large. But obviously you cannot do that as no experiments

The Exhaustive Search: This algorithm is best when trying to solve the DHCP in a smaller size graph, with small number of nodes V, populated throughout the graph. If there are a large amounts of nodes V, the time to find an optimal solution would increase aswell, this means that the Exhaustive search will have to keep traversing around every single node in the graph. Therefore, the algorithm depends on the number of nodes V inputted, if the size of the V is low then the algorithm would return an optimal path quickly however as the number of nodes N increases in the graph, the algorithms would take longer to find the optimal path and will have exponential growth.

The Greedy Search: This algorithm is faster than Exhaustive search, this is self-explanatory since the BigO notation for greedy is $O(n^2)$ compared to $O(n!)$ of Exhaustive Search. The Exhaustive search is slower than the Greedy search but it is best for finding the best optimal path of a HC. The Greedy algorithm fail to find the optimal solution because it doesn't consider all the nodes in the graph. The choice made by a greedy algorithm depends on choices it has made so far and it is not aware of future choices it could make. Therefore, the path it returns is not always the optimal path. Therefore, Greedy Search is useful when a solution is not needed to be optimal but needed to be produced faster.

But big-O only kicks in after a certain point

The Tabu Search: The Tabu Search is the most efficient of the three because unlike the other two, Exhaustive and Greedy, The Tabu Search stores every P visited in the through into it's list, the TabuList contains all the list of P visited, therefore avoiding any illegal move and avoiding going around in circles, therefore finding the most best optimal path possible.

9 Reflection

Overall, working on this project has been very beneficial to me. I have a new understanding of Big O complexity can help algorithms grow and improve. It was particularly beneficial understanding Big O in-depth because last year, I only had basic knowledge of it. Furthermore, i also gained new knowledge in Pseudocode. Going into depth this year, fully understand Pseudocode, made me realise how wrongly i wrote Pseudocode for programming last year.

Explain where else these skills could be applied

i --> I

Last year we were only taught the basics of graph generation, this project has helped me learn to create new graph generation like Random graph with a HC inside it. To find the HC, algorithms had to be researched as a group. This helped me gain transferable skills, particularly social skills. I could communicate and learn better, leading me to find out that i work better in a group than individually.

However, I also faced some huge difficulties while working on this project. One in particular, was time-management. I felt like i sometimes spent too long learning things that shouldn't have taken me that long, such as Jupiter Notebook. It was a big hassle because i spent a long time researching about it when i could have asked my group peers all on about it, which would have made me learn quicker. Other problems were understanding the problem and coding to solve the problem. It was a huge challenge balancing loads between doing this coursework, doing dissertation, revising for the phase test, understand and learning theoretical aspects of the course.

You could have also come to labs and asked staff.

But how to IMPROVE. You cannot keep saying to improve time management skills - its the end of uni now - time to actually take some action to improve them

10 References

Bollobas, B., 1985. Random Graphs. First Printing edition ed. s.l.:Academic Pr; First Printing edition (September 1, 1985).

Glover, F., 1986. Tabu Search – Part 1. ORSA Journal on Computing, p. 190–206.

Jillian Beardwood, J. H. H. a. J. M. H., 2008. The shortest path through many points. Volume 55, Issue 4 ed. USA: s.n.

Why have you initialized surnames?

Johnson, M. R. G. a. D. S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. United States: W. H. Freeman and Company.

Kreinovich, V., 1987. Problems of Reducing the Exhaustive Search. Series 2, Volume 178 ed. s.l.:s.n.

Stefan Voss, S. M. I. H. C. R., 1999. Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. First Edition ed. France : Kluwer Science + Business Media New York .

West, D. B., 2017. Introduction to Graph Theory. Second Edition ed. USA: Pearson Education Heg .

www.ici.usi.edu, n.d. More NP-Complete and. [Online] Available at: http://www.ics.uci.edu/~goodrich/teach/cs162/notes/pnp3.pdf [Accessed 04 May 2018].

Some good references used in text. Citation style not quite correct though (lots of authors names lost?)

380CT - Asis.pdf

GRADEMARK REPORT

FINAL GRADE

47/100

GENERAL COMMENTS

Instructor

PAGE 1

Text Comment. We don't connect list entries with logical AND symbols

Text Comment. Graph paths do not have clauses.

Text Comment. But 6 lines above you used little v for the number of nodes?

Text Comment. What distance? Distance is not relevant for DHCP.

Text Comment. ...finding out if THERE EXISTS a...

Text Comment. ... in A ...

Text Comment. graph singular

Text Comment. or what?

Text Comment. correct

Text Comment. It is NP complete. But the reason you gave is the definition of NP.

PAGE 2

QM **Optimisation problem not clearly defined**

You have not clearly defined an Optimisation version of the DHCP. There were several

You have not clearly defined an Optimisation version of the DHCP. There were several possibilities:

- Maximise number of nodes reached by a cycle that never visits a node more than one

- Minimise number of nodes revisited by a cycle that visits all nodes.

- Minimise use of non-existent edges in a a Hamiltonian Cycle for a complete graph of same size

etc. Without providing a clear definition of the optimisation problem it is not possible to design or judge the performance of a heuristic algorithm.

Text Comment. I think you mean that P is the cycle

Text Comment. Incorrect. This problem is a decision problem and so cannot be NP and hence is neither NP-Complete. It is NP hard though.

Text Comment. What is a non-Hamiltonian instance? This is not a clear optimisation problem

Text Comment. Correct

Text Comment. How do you know the time increases before? Average over what?

Text Comment. Why isn't greedy having its quality checked (it is also heuristic).

Text Comment. The reason it wasn't implemented is not "deadlines". You had the whole Easter vacation to do it

Text Comment. You could have still presented a testing methodology though.

Text Comment. Yes, but how many graphs? What size and structure? How to avoid effect of outliers? etc.

PAGE 3

PAGE 4

PAGE 5

Text Comment. it's is short for "it is" which does not make sense in sentence below. You meant "its"

Text Comment. This function header does not make sense. The job of exhaustive search is to find a path - so why would a path be given as input?

PAGE 6

Text Comment. What does Try different V mean?

Text Comment. Undefined variables

Text Comment. Why give up search at this point?

Text Comment. I don't think this pseudocode is even close to a correct exhaustive search

Text Comment. all possibilities of WHAT?

Text Comment. The word I is always capitalised. You must know this?

Text Comment. It is exponential but this explanation does not justify it

PAGE 7

Text Comment. ok - this seems like a reasonable greedy heuristic for DHCP. The function header should probably include an input graph.

Text Comment. ok

Text Comment. explanation a little confused, but O(N^2) reasonable

Text Comment. unevaluated Latex code

Text Comment. Neither variable Tabu nor variable Tabu_m are defined or input. So what on earth does line 2 mean?

Text Comment. Define neighbourhood

Text Comment. Define best

PAGE 8

Text Comment. satisfied clauses has NOTHING to do with DHCP.

Text Comment. too trivial

Text Comment. too trivial

Text Comment. too trivial

Text Comment. 1,2,4,5 are all incredibly trivial. Not really worth discussing. 3 is quite trivial but I would have accepted it - however there is no discussion of how to check, cost of checking, likelihood of occurrence etc.

Text Comment. too trivial

Text Comment. Your conclusion is meant to put a number on small/large. But obviously you cannot do that as no experiments

Text Comment. But big-O only kicks in after a certain point

PAGE 9

Text Comment. i --> I

Text Comment. Explain where else these skills could be applied

Text Comment. You could have also come to labs and asked staff.

Text Comment. But how to IMPROVE. You cannot keep saying to improve time management skills - its the end of uni now - time to actually take some action to improve them

Text Comment. Why have you initialized surnames?

Text Comment. Some good references used in text. Citation style not quite correct though (lots of authors names lost?)

INTRODUCTION (10%) 40 / 100

0 (0)	No work done.
3 (30)	Poor definitions of the different versions of the problem. Poor discussion of complexity class membership. Notation not clearly identified and described.
4 (40)	Basic definitions of the different versions of the problem. Basic discussion of complexity class membership. Notation not clearly identified and described.
5 (50)	Good definitions of the different versions of the problem, but could be made more formal. Complexity class membership addressed, but need to add justification. Notation identified and described, but could be made clearer.
6 (60)	Very good definitions of the different versions of the problem, but could be made more formal. Very good discussion of complexity class membership, but could add more analysis. Notation identified and described, but could be made clearer.
7 (70)	Excellent formal definitions of the different versions of the problem. Excellent analytical discussion of complexity class membership, but could be more critical. Notation clearly identified and described.
8 (80)	Outstanding formal definitions of the different versions of the problem. Outstanding critical and analytical discussion of complexity class membership. Notation clearly identified and described mathematically.
10 (100)	Perfect work.

METHODOLOGY (10%) 50 / 100

0 (0)	No work done. Need to discuss the approach used to test and evaluate the different methods.
3 (30)	Little demonstration that material is understood. Need to discuss the approach used to test and evaluate the different solution methods.
4 (40)	Basic description of the testing methodology and random instances sampling strategy. Need to discuss the approach used to test and evaluate the different methods more thoroughly.
5 (50)	Good description of the testing methodology and random instances sampling strategy. Need to discuss the approach used to test and evaluate the different methods in more formality and technical detail.
6 (60)	Very good description of the testing methodology and random instances sampling strategy. Need to discuss the approach used to test and evaluate the different methods in more technical detail.

7 (70)	Excellent description of the testing methodology and random instances sampling strategy. Could be improved further by adding more technical detail and mathematical analysis.
8 (80)	Outstanding description of the testing methodology and random instances sampling strategy. Could be improved further by adding mathematical rigour and discussing possible bias and how to deal with it.
10 (100)	Perfect work.

EXHAUSTIVE (10%) 40 / 100

0 (0)	No work done.Need to discuss and investigate the exhaustive search solution method.
3 (30)	Little demonstration that material is understood. Algorithm may be incorrect in concept and/or implementation. Need to discuss and investigate the solution method.
4 (40)	Demonstrates a basic understanding of the theory but perhaps without succeeding in a complete implementation of all ideas. Need a better discussion and investigation of the solution method.
5 (50)	Good understanding of the theory but the practical investigation needs improvement. Results and conclusions are satisfactory, but could be improved.
6 (60)	Very good understanding of the theory with acceptable practical investigation. Results and conclusions are satisfactory. Could be improved with more technical analysis, insightful practical considerations, and references to the literature.
7 (70)	Excellent work demonstrating strong analytical skills and competent application of classification techniques. Software well implemented and tested, with well documented results and accurate conclusions. Could be improved with more insightful practical considerations, and references to the literature.
8 (80)	Outstanding work demonstrating strong analytical skills and competent application of classification techniques. Software well implemented and tested, with well documented results and accurate conclusions. Could improve by referencing relevant literature.
10 (100)	Perfect work.

GREEDY (10%) 50 / 100

0 (0)	No work done.Need to discuss and investigate the exhaustive search solution method.
3 (30)	Little demonstration that material is understood. Algorithm may be incorrect in concept and/or implementation. Need to discuss and investigate the solution method.

4 (40)	Demonstrates a basic understanding of the theory but perhaps without succeeding in a complete implementation of all ideas. Need a better discussion and investigation of the solution method.
5 (50)	Good understanding of the theory but the practical investigation needs improvement. Results and conclusions are satisfactory, but could be improved.
6 (60)	Very good understanding of the theory with acceptable practical investigation. Results and conclusions are satisfactory. Could be improved with more technical analysis, insightful practical considerations, and references to the literature.
7 (70)	Excellent work demonstrating strong analytical skills and competent application of classification techniques. Software well implemented and tested, with well documented results and accurate conclusions. Could be improved with more insightful practical considerations, and references to the literature.
8 (80)	Outstanding work demonstrating strong analytical skills and competent application of classification techniques. Software well implemented and tested, with well documented results and accurate conclusions. Could improve by referencing relevant literature.
10 (100)	Perfect work.

METAHEURISTIC (10%) 30 / 100

0 (0)	No work done.Need to discuss and investigate the exhaustive search solution method.
3 (30)	Little demonstration that material is understood. Algorithm may be incorrect in concept and/or implementation. Need to discuss and investigate the solution method.
4 (40)	Demonstrates a basic understanding of the theory but perhaps without succeeding in a complete implementation of all ideas. Need a better discussion and investigation of the solution method.
5 (50)	Good understanding of the theory but the practical investigation needs improvement. Results and conclusions are satisfactory, but could be improved.
6 (60)	Very good understanding of the theory with acceptable practical investigation. Results and conclusions are satisfactory. Could be improved with more technical analysis, insightful practical considerations, and references to the literature.
7 (70)	Excellent work demonstrating strong analytical skills and competent application of classification techniques. Software well implemented and tested, with well documented results and accurate conclusions. Could be improved with more insightful practical considerations, and references to the literature.
8 (80)	Outstanding work demonstrating strong analytical skills and competent application of classification techniques. Software well implemented and tested, with well documented results and accurate conclusions. Could improve by referencing relevant literature.

10 (100)	Perfect work.

SPECIAL CASES (10%) 40 / 100

0 (0)	No special cases identified or discussed. Need to correctly identify and discuss special cases that can be tackled in polynomial time.
3 (30)	Vague discussion without actually identifying any special cases, or the identified cases are all not correct. Need to correctly identify and discuss special cases that can be tackled in polynomial time.
4 (40)	1 special case correctly identified and discussed. Need to identify and discuss more special cases that can be tackled in polynomial time.
5 (50)	2 special cases correctly identified and discussed. Need to identify and discuss more special cases that can be tackled in polynomial time.
6 (60)	3 special cases correctly identified and discussed. Need to identify and discuss more special cases that can be tackled in polynomial time.
7 (70)	4 special cases correctly identified and discussed. Could be improved with a mathematical discussion, and by discussing more special cases that can be tackled in polynomial time.
8 (80)	5 special cases correctly identified and discussed. Could be improved with a mathematical discussion.
10 (100)	Perfect work.

CONCLUSION (10%) 50 / 100

0 (0)	No conclusion.
3 (30)	Little demonstration that material is understood. Results may be misreported or absent.
4 (40)	Demonstrates a basic understanding of the theory. Results may be misreported or absent.
5 (50)	Demonstrates a good understanding of the theory. Results are summarised but may be inaccurate/misreported.
6 (60)	Demonstrates a very good understanding of the theory. Results are well reported.
7 (70)	Demonstrates an excellent understanding of the theory. Results are well reported and discussed.
8	Demonstrates an outstanding understanding of the theory. Results are excellently

(80)	reported and discussed.
10 (100)	Perfect work.

REFLECTION (10%) 60 / 100

0 (0)	No reflection.
3 (30)	Poor reflection. Write about (highlight transferable skills): What you have learnt. What you could have done differently. If you collaborated on the coding then specify your contribution clearly.
4 (40)	Basic reflection. Write about (highlight transferable skills): What you have learnt. What you could have done differently. If you collaborated on the coding then specify your contribution clearly.
5 (50)	Good reflection. Write objectively about (highlight transferable skills): What you have learnt. What you could have done differently.
6 (60)	Very good reflection. Could improve by linking to other modules and/or future career.
7 (70)	Excellent reflection. Could be more specific about what could be done differently.
8 (80)	Outstanding reflection. Could be improved by writing a clear tangible plan for self-development.
10 (100)	Perfect work.

PRESENTATION (10%) 60 / 100

0 (0)	No work submitted. Need to consider structure and layout, flow of ideas and wording, grammar & spelling, and use & quality of graphs/tables.
3 (30)	Poor presentation. Need to improve structure and layout, flow of ideas and wording, grammar & spelling, and use & quality of graphs/tables.
4 (40)	Basic presentation. Need to improve structure and layout, flow of ideas and wording, grammar & spelling, and use & quality of graphs/tables.
5 (50)	Good presentation. Could improve flow of ideas and wording, grammar & spelling, and use & quality of graphs/tables.
6 (60)	Very good presentation. Could improve flow of ideas and wording, and grammar & spelling.
7 (70)	Excellent presentation. Could improve flow of ideas and wording.

8 (80)	Outstanding presentation.Could improve wording.
10 (100)	Perfect work.

REFERENCING (10%) 50 / 100

0 (0)	No referencing.
3 (30)	Inadequate referencing. Only websites.
4 (40)	Basic referencing. Mostly websites, and little use of the cited literature.
5 (50)	Good referencing. Some books cited, but little/shallow use of the cited literature.
6 (60)	Very good referencing. Some books and research papers cited, but little/shallow use of the cited literature.
7 (70)	Excellent referencing. Books and research papers cited, with use of the cited literature.
8 (80)	Outstanding referencing. Books and research papers cited, with good use of the cited literature.
10 (100)	Perfect referencing.