The University of Nottingham

SCHOOL OF COMPUTER SCIENCE

A LEVEL 2 MODULE, SPRING SEMESTER 2020-2021

ARTIFICIAL INTELLIGENCE METHODS

Maximum Time Allowed to Submit Answers in Moodle: 24 Hours

Open-book examination.

Answer ALL 4 QUESTIONS

Suggested time to complete the paper ~2 hours.

This open-book examination will be marked out of 100.

You may write/draw by hand your answers on paper and then scan them to a PDF file, or you may type/draw your answers into electronic form directly and generate a PDF file. Guidance on scanning can be found through the Faculty of Science Moodle Page Guidance for Remote Learning.

Your solutions should include complete explanations and should be based on the material covered in the module. Make sure your PDF file is easily readable and does not require magnification. Make sure that each page is in the correct orientation. Text/drawing which is not in focus or is not legible for any other reason will be ignored.

Submit your answers containing all the work you wish to have marked as a ***single PDF file****.*

Use the standard naming convention for your document: [Student ID] [Module Code] [Academic Year]. Write your student ID number at the top of each page of your answers.

Although you may use any notes or resources you wish to help you complete this open-book examination, the academic misconduct policies that apply to your coursework also apply here. You must be careful to avoid plagiarism, collusion or false authorship. Please familiarise yourself with the Guidance on Academic Integrity in Alternative Assessments, which is available on the Faculty of Science Moodle Page Guidance for Remote Learning. The penalties for academic misconduct are severe.

Staff are not permitted to answer assessment or teaching queries during the period in which your examination is live. If you spot what you think may be an error on the exam paper, note this in your submission but answer the question as written.

序号	Job	p_j	d_j	w_j	C_j	$w_j \cdot C_j$	L_j	Tardy?
1	5	2	3	3	2	3 × 2 = **6**	-1	❌
2	2	6	9	10	2+6 = 8	10 × 8 = **80**	-1	❌
3	3	5	14	5	8+5 = 13	5 × 13 = **65**	-1	❌
4	1	10	12	5	13+10 = 23	5 × 23 = **115**	11	✅
5	4	4	8	1	23+4 = 27	1 × 27 = **27**	19	✅

Question 1. AI Methods – Basics. This question consists of three parts. **[25 marks]**

(A) Consider 5 jobs having the processing times p_j, due-dates d_j and the weights w_j of the jobs j=1, ..., 5, given in the table:

Jobs	**1**	**2**	**3**	**4**	**5**
p_j	10	6	5	4	2
d_j	12	9	14	8	3
w_j	5	10	5	1	3

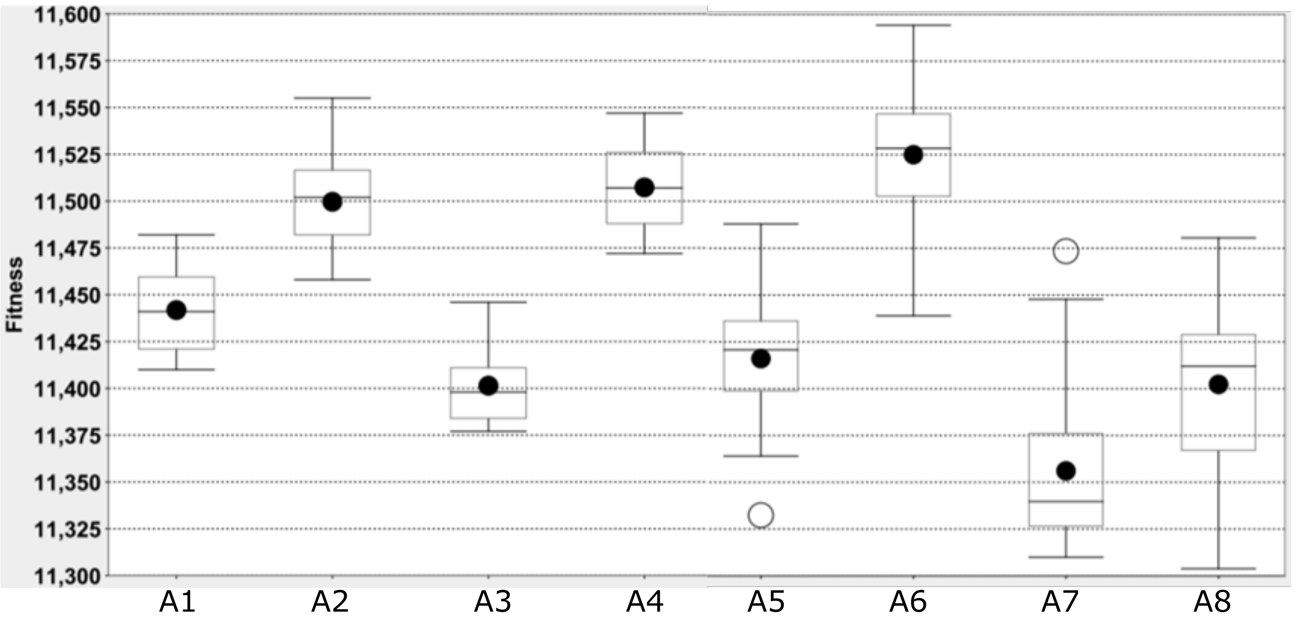
Calculate the following for the schedule: <5, 2, 3, 1, 4> showing your steps:

(i) the total weighted completion time 6 + 80 + 65 + 115 + 27 = 293 [4]
(ii) the maximum lateness \max(-1, -1, -1, 11, 19) = 19 [3]
(iii) the number of tardy jobs Job 1 和 Job 4 是逾期的（C_j > d_j）total=2 [3]

[10 Marks]

(B) Based on the following box-plots of the fitness (objective) values obtained after running 8 search algorithms on a given instance for a minimisation problem for 30 trials, indicate whether each of the five given statements below is TRUE or FALSE and provide your reasoning without exceeding **100 words**.

[15 Marks]



Statements:

(i) A3 is a hill climbing algorithm
(ii) The best performing algorithm is A7 on average for the instance
(iii) The best solution in a single trial is achieved by the algorithm A6 for the instance
(iv) The algorithm A3 performs better than A4 and this performance difference is statistically significant for the instance
(v) The algorithm A6 performs slightly better than A4 on average for the instance

(i) TRUE – A3 shows very low variance, typical of hill climbing.(
ii) TRUE – A7 has the lowest mean fitness.
(iii) FALSE – A5 achieves the best single-trial fitness.(
iv) TRUE – A3 clearly outperforms A4 with non-overlapping distributions.
(v) FALSE – A6 has higher average fitness than A4.

Question 2. Heuristics/Operators. This question consists of three parts. **[25 marks]**

(A) Consider a traveling salesman problem (TSP) instance consisting of 5 cities C1, C2, C3, C4 and C5 and answer the following two questions. Refer to the table below for the inter-city distances, which adopts generic units.

	C1	**C2**	**C3**	**C4**	**C5**
C1	0	9	2	4	10
C2	9	0	5	6	7
C3	2	5	0	12	3
C4	4	6	12	0	9
C5	10	7	3	9	0

2)

(i) **Firstly**, illustrate how the nearest neighbor heuristic works using path representation step by step, providing the final tour and its length that will be returned by the algorithm considering that <C2> is selected randomly as the starting city to construct a tour.
(ii) **Secondly**, provide 2 optimal/perfect solutions in path representation for this problem instance.

[10 marks]

(B) Order Crossover can be applied not only permutation encoding, but also other encodings as well, for example integer encoding. Provide the newly generated solution(s) after applying Order Crossover to the following solutions selected as parents: 5-3-1-2-4-1-5 and 3-4-9-2-1-5-3. In this representation, repetition is allowed and you should remove the first occurrence of an entry in the genetic material that will be inherited to an offspring, if that entry already exists in the offspring. You can ignore any remaining entries in the genetic material from the parent, once the offspring is complete, even if there are some left from the parent to be inserted into the offspring. Given that the first and fourth locations (e.g., x|x-x-x|x-x-x) are chosen as the crossover points. Show **all** your steps and provide any assumptions you make.

crossover

[5 marks]

(C) In the context of Simulated Annealing, assuming that the Geometric cooling schedule is used and the initial temperature (T_0) is fixed as 1, briefly explain how each one of the following settings for the cooling rate (α) influences the probability of acceptance of worsening solutions in time based on the temperature T (i.e., $e^{-\Delta/T}$). Each explanation should not exceed **100 words.**

α = 0.000001
α = 0.975
α = 1
α = 3

[10 marks]

温度下降非常快，几乎立刻趋近于 0，导致接受劣解的概率迅速降低为零。算法几乎变为贪心爬山法，容易过早陷入局部最优。

The temperature drops extremely quickly, almost immediately approaching zero. As a result, the probability of accepting worsening solutions rapidly becomes negligible. The algorithm behaves similarly to greedy hill climbing, getting stuck in local optima very early.

温度下降较缓，保留了在较长时间内接受劣解的适中概率，有助于前期的探索，并提高跳出局部最优的能力。

The temperature cools gradually. This allows the algorithm to maintain a moderate chance of accepting worse solutions for a longer time, which encourages exploration early on and improves the ability to escape local optima.

温度不下降，接受概率始终不变，算法表现高度随机，没有收敛趋势，难以收敛到优解。

The temperature remains constant, as no cooling occurs. The acceptance probability does not decrease over time, leading to excessively random behavior throughout the process. The search lacks convergence and may not settle to a good solution.

因 α > 1，温度随时间上升，违反退火原理，导致接受劣解概率不断增加，搜索行为混乱，不收敛，算法失效。

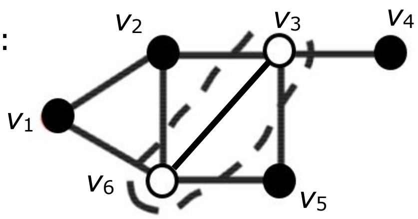
Since α > 1, the temperature increases over time instead of decreasing. This is invalid for simulated annealing, as it causes the acceptance probability of worse solutions to grow, leading to chaotic and divergent behavior.

Question 3. Single Point based Search. This question consists of three parts. **[25 marks]**

Optimal Split Problem (OSP): For a given undirected graph $G = (V, E)$ with N vertices and M edges, where $V=\{v_1, v_2,\ldots, v_N\}$ and $E=\{e_1, e_2,\ldots, e_M\}$, a **split** in G is a subset $S \subseteq V$. Let S' indicate the remaining vertices in V, that is, $V \setminus S$, so $S \cap S'=\{\}$ and $S \cup S'=V$. Hence, a "split" refers to the partition (S, S') of the vertices in V. The optimal split problem is to find a split S; such that the number of edges between the vertices in S and S' is maximised. An example optimal solution to the OSP instance of G_1 with 6 vertices ($V=\{v_1, v_2, v_3, v_4, v_5, v_6\}$) and 8 edges (i.e., N=6, M=8) is provided below. The vertices in **black** and **white** colours indicate the partitions $S=\{v_1, v_2, v_4, v_5\}$ and $S'=\{v_3, v_6\}$, respectively and the maximum number of edges between S and S' is 6.

Each vertex v_k 属于 V can independently be assigned to either partition S or S′. Since there are N vertices, and each vertex can be assigned in 2 ways, the total number of possible binary encodings is:
2^N

G_1:



2. Yes, there is redundancy. The split into sets S and S′ is symmetric, so flipping all bits (i.e., switching 1s and 0s in r) yields the same objective value. For example, [1,1,0,1,1,0] and its complement [0,0,1,0,0,1] represent the same cut. Hence, for any solution, its complement is equivalent, which reduces the number of unique solutions from 2^N to 2^{N-1}.

Answer the following questions.

(A) Assume that you decided to design a great deluge algorithm for solving the optimal split problem as described above, hence binary encoding will be used to represent a candidate solution $\boldsymbol{r}=[b_1, b_2, b_3,\ldots, b_k,\ldots b_N]$, where b_k is a binary variable indicating which set the k-th vertex in V is partitioned into, that is, if b_k =1, then the k-th number is partitioned into S, otherwise (which means b_k =0) the k-th vertex is partitioned into S'. The objective function $f(\boldsymbol{r})$ counts the number of edges between the vertices in S and S'. For example, the optimal solution to the instance G_1 as illustrated above can be represented as [1,1,0,1,1,0] and the objective value would be 6.

(i) How many different candidate solutions can be encoded with this representation scheme? Provide your answer in terms of N and/or M.

(ii) Is there any redundancy in this representation (that is for any given solution, is it possible to create an equivalent solution)? If yes, explain without exceeding **100 words.**

The minimum value of the objective function f(r) is 0, which occurs when there are **no edges** between S and S′. This can be achieved by assigning **all vertices to the same partition**. Therefore, two possible minimal solutions are:
r = [1,1,1,⋯,1] (all in S)
r = [0,0,0,⋯,0] (all in S′)

[6 marks]

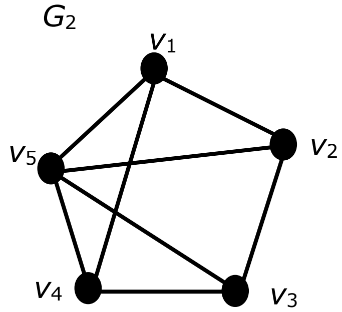
(B) Provide a general case for the solution $\boldsymbol{r}$ to an optimal split problem instance based on the formulation as in (A), for which the objective function, $f(\boldsymbol{r})$ evaluates to the minimum possible objective value.

[5 marks]

Both yield f(r) = 0 since there is no edge crossing between sets.

(C) Given the optimal split problem instance G_2 as illustrated on the right-hand side with $V=\{v_1, v_2, v_3, v_4, v_5\}$ and 8 edges (i.e., N=5, M=8) and a *current solution* of [1,1,1,0,0] with the objective value of 4 using the formulation as explained in part (A), what would be the resultant solutions returned by the ***first-descent*** (next-descent) and ***best-descent*** (steepest-descent) hill climbing methods, respectively? Assume that the bit-flip neighbourhood move is used, first-descent starts scanning a solution from the first bit returning as soon as a strictly better solution than the input solution is found, while best-descent makes a single complete pass over the current solution. Show the details of your computation for each step including the current solution, best solution and their objective values computed as described.

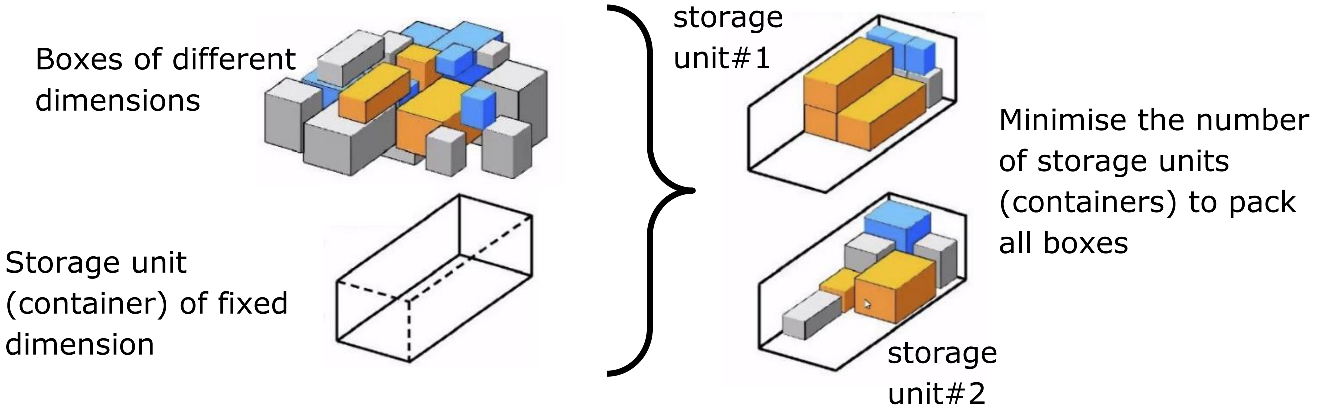
G_2



[14 marks]

Question 4. Population-based Search, Metaheuristics and Hyper-heuristics. This question consists of two parts. **[25 marks]**

Storage Packing Problem (SPP). This is the problem of orthogonally packing a given set of *M* rectangular-shaped boxes, each with different dimensions into the minimum number of rectangular storage units (containers) of predefined dimension as illustrated below. SPP is proven to be an NP-hard optimisation problem.



(A) Assume that there are three constructive packing heuristics, c#0, c#1 and c#2. Each heuristic uses a different strategy to choose from the remaining boxes and then to pack the chosen box into an available storage unit (if necessary opening a new storage unit). The details of those three heuristics are irrelevant.

Design a **genetic algorithm hyper-heuristic** for solving the Storage Packing Problem (SPP) using c#0, c#1 and c#2 as the low level heuristics in your hyper-heuristic algorithm design. Explain all your algorithmic choices, in particular chromosome length, (candidate solution) representation showing how a complete solution can be obtained with respect to the chromosome length, initialisation, genetic operators, replacement, termination and any other relevant parameter settings.

[15 marks]

(B) Which algorithm would perform better for this problem, a genetic algorithm hyper-heuristic or a simulated annealing metaheuristic? Provide your reasoning without exceeding **100 words**.

[10 marks]

1.**Chromosome Representation:**
Each chromosome is an array of integers of length M (number of boxes), where each gene g_i 属于 \{0,1,2\} represents the index of the constructive heuristic applied to the i-th box.
Initialisation:
Randomly generate an initial population of size P, with each gene assigned randomly from {0,1,2}.
Fitness Function:
Execute the heuristic sequence to pack boxes. Fitness = number of storage units used. Lower is better.
Selection:
Tournament selection or roulette wheel selection with elitism.
Crossover:
One-point or two-point crossover applied to heuristic sequences.
Mutation:
Small probability mutation (e.g., 5%) to replace a gene with another heuristic index.
Replacement and Termination:
Elitism with generational replacement. Terminate after G generations or if no improvement after G′ generations.

2. A genetic algorithm hyper-heuristic is better suited for the Storage Packing Problem because SPP is a complex combinatorial optimisation problem with multiple constructive heuristics. GA maintains a population of diverse heuristic sequences and explores the global search space more effectively than SA. Moreover, GA can exploit combinations of low-level heuristics over multiple boxes, allowing better packing patterns to evolve over generations. In contrast, simulated annealing operates with a single solution and may get stuck in local optima due to limited neighbourhood exploration.

GA hyper-heuristic 更适合 SPP，因为它可以全局搜索并组合多个低层启发式，而 SA 只能在单解局部邻域中搜索，容易陷入局部最优