COMP523 Continuous Assessment 2

John Sylvester

Due at 17:00 on 05/05/2023

Assessment Information

Assignment Number	2 (of 2)
Weighting	15%
Assignment Circulated	20th April 2023
Deadline	5th May 2023, 17:00
Submission Mode	Please submit your solutions **electronically (PDF format)** on Canvas. You will find the submission upload in the assignments section.
Learning outcome assessed	3. Identify which of the studied design principles are used in a given algorithm taking account of the similarities and differences between the principles. 4. Apply the studied design principles to produce efficient algorithmic solutions to a given problem taking account of the strengths and weaknesses of the applicable principles. 5. Outline methods of analysing correctness and asymptotic performance of the studied classes of algorithms, and apply them to analyse correctness and asymptotic performance of a given algorithm.
Purpose of assessment	Design and analysis of algorithms similar to known ones. The aims are: to produce a solution to a related problem in order to obtain efficient solution to a given problem, based on known techniques.
Marking criteria	Based on the marking descriptors of the University's Code of Practice on Assessment
Submission necessary in order to satisfy Module requirements?	No
Late Submission Penalty	Standard UoL Policy.
Plagiarism and Collusion	Please be aware of the University guidelines on plagiarism and collusion.

A. Given a graph $G = (V, E)$ and integer d the *bounded degree spanning tree* problem is to determine if there is a spanning tree T of G such that the maximum degree of a node in T is at most d. Recall that a Hamiltonian path in G is a path that visits all vertices of G.

(i) Show that if a spanning tree has degree at most 2 then it is a Hamiltonian path. [5 marks]

We aim to show that the *bounded degree spanning tree* is NP-Complete. You may use that fact that deciding if a given graph has a Hamiltonian path is NP-Complete.

(i) Show that the *bounded degree spanning tree problem* is in NP. [5 marks]

(ii) Show that for $d = 2$ the *bounded degree spanning tree problem* is NP-Hard. [7 marks]

(iii) Show that for $d = 10$ the *bounded degree spanning tree problem* is NP-Hard. [9 marks]

B. A shop sells different flavour candies in bags each costing \$1. Each of the bags contains a mixture of candies with different flavours. The contents of each bag are known. Consider the following problem.

The \$k-Explorer Problem: You have \$$k$ to spend and wish to try as many different flavours as possible.

In this question we aim to show the algorithm below gives a $(1 - 1/e)$-approximation to this problem.

Algorithm: GREEDY \$$k$-EXPLORER
  Create a list of all candy flavours, label them all as 'new'.
  for $i = 1, \ldots, k$ **do**
    Buy the bag containing the most 'new' flavours.
    Mark all the flavours contained in this bag as 'old'.
  Eat all the candy.

Let x_i be the number of flavours in the bag brought in loop i that were 'new' at the time of purchase.

Let OPT be the number of different flavours tried in an optimal solution.

In what follows we define empty sums as 0, that is $\sum_{j=1}^{0} x_j = 0$.

(i) Show that, for any $1 \le i \le k$, in the i-th loop of GREEDY \$$k$-EXPLORER we have

$$x_i \ge \frac{OPT - \sum_{j=1}^{i-1} x_j}{k}.$$

[9 marks]

(ii) Show that, for any $0 \le i \le k$, in the i-th loop of GREEDY \$$k$-EXPLORER we have

$$OPT - \sum_{j=1}^{i} x_j \le \left(1 - \frac{1}{k}\right)^i \cdot OPT.$$

[9 marks]

For the following question: note that for any $k \ge 0$ we have $\left(1 - \frac{1}{k}\right)^k \le \frac{1}{e}$, where $e = 2.718...$

(iii) Show that the GREEDY \$$k$-EXPLORER algorithm is a polynomial time $(1 - 1/e)$-approximation algorithm for the \$$k$ Explorer Problem. [8 marks]

In the following use B_i denote the set of flavours of candies contained in bag i, x_i to denote the variable relating to buying bag i, and y_j for the variable relating to the appearance of flavour j.

(iv) Formulate \$$k$-*Explorer* as an Integer Linear Program and give its LP-relaxation. [7 marks]

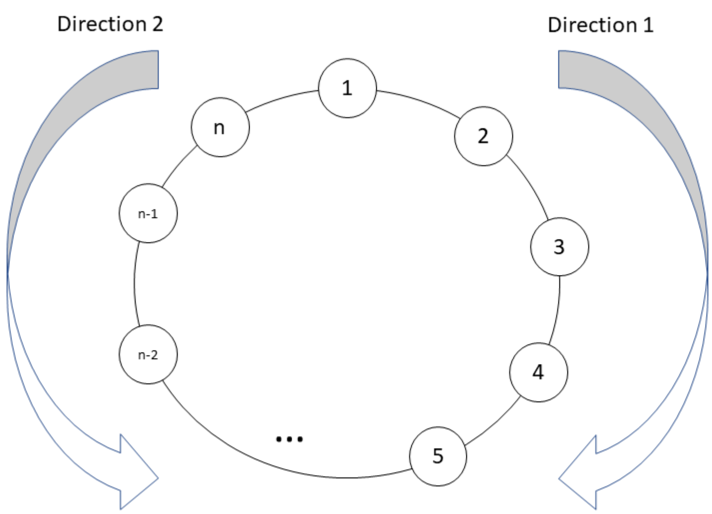
C. Given a graph $G = (V, E)$ and an integer $k \geq 2$, the problem is to assign labels $\{1, \ldots, k\}$ to V to maximise the number of edge whose endpoint have different labels. More formally, for every vertex $u \in V$, let $x_u \in \{1, \ldots, k\}$ be the number vertex u is assigned, and for each edge $\{i, j\} \in E$ we let $X_{ij} = 1$ if $x_i \neq x_j$ and $X_{ij} = 0$ otherwise. The problem is then to assign labels in a way which maximises $Z = \sum_{\{i,j\} \in E} X_{i,j}$. We seek an approximation algorithm for this NP-Hard problem.

(i) Design a randomised algorithm which returns a solution satisfying $\mathbb{E}[Z] \geq \left(1 - \frac{1}{k}\right) |E|$. You should prove that the solution produced by your algorithm satisfies this inequality. [9 marks]

(ii) Explain why your algorithm from part (i) gives a polynomial time randomised $\frac{k}{k-1}$-approximation algorithm for this problem. [7 marks]

D. There is a set of n cities (where n is even) located on a cycle graph $G = (V, E)$ and numbered as shown in the figure. There is also a set R of taxi ride requests, with each request $r_i \in R$ being represented by an (origin, destination) pair, where the origin and the destinations are two cities on the cycle. Each request can be driven via two alternative directions (Direction 1 or Direction 2), as shown in the figure.

The *congestion* C_e on an edge e of a cycle is the number of requests that are driven via that edge. The *makespan congestion* is the maximum congestion on any edge, i.e., $\max_{e \in E} C_e$. The objective is to find an assignment of the requests to the two directions (Direction 1 and Direction 2) such that the makespan congestion is minimised. We will find a polynomial time 2-approximation via LP-rounding.



In the following use indicator variables x_i to denote if a request was driven in Direction 1 or Direction 2. For each edge e, use R_e to denote the set of requests which would be driven via edge e *if they were driven via Direction 1* (regardless of the value of x_i). For example, in the figure, for edge $e = (4, 5)$, the requests $(4, n-2)$ and $(3, 2)$ are in R_e, but the request $(n-2, 3)$ is not.

(i) Formulate the problem as an integer linear program and give its LP-relaxation. [7 marks]

(ii) Provide a rounding scheme for the LP-relaxation to obtain a solution to the original problem. [9 marks]

(iii) Explain why the rounding scheme results in a polynomial time 2-approximation algorithm for the original problem. [9 marks]