Computer Engineering

Computer Engineering Advanced Algorithms

1. Explain the difference between PTAS and FPTAS, and give one example of a problem for which a FPTAS is known, and one example of a problem for which a PTAS is known but no FPTAS. [4 marks]
2. We consider an extension of the MAX-3-CNF problem, called MAX-4-CNF problem, where we are given a 4-CNF formula with m clauses, e.g., $(x_1 \, V \overline{x_3} \, V \, x_4 \, V \, x_5) \wedge (\overline{x_1} \, V \, \overline{x_2} \, V \, x_3 \, V \, \overline{x_5}) \wedge$ ...., and the goal is to find an assignment of the variables $x_1, x_2 \ldots\ldots x_n$ that satisfies as many clauses as possible.
   (i) Design a randomized approximation algorithm and analyses its approximation ratio. (For full marks, the approximation ratio must be smaller than 10/9.) [4 marks]
   (ii) Express the MAX-4-CNF problem as an integer program. [4 marks]
   (iii) Based on the construction from Part (b)(ii) or otherwise, describe an algorithm that performs randomized rounding on the solution of a linear relaxation. [3marks]
   (iv) Analyze the expected approximation ratio of the algorithm from Part (b)(iii). Hint: You may want to use the following two inequalities. Firstly, for any non-negative numbers aT, a2,..., ak, we have

$$\left(\prod_{i=1}^{k} a_i\right)^{1/k} \leq \frac{\sum_{i=1}^{k} a_i}{k}.$$

   Secondly, for any integer $k \geq 2$ and $0 \leq a \leq 1$,

$$1 - \left(1 - \frac{a}{k}\right)^k \geq \left(1 - (1 - \frac{1}{k})^k\right) \cdot a.$$

   [5 marks]
3. Into what three cases can a linear program in standard form be classified? [3 marks]
4. Consider the (unweighted) vertex cover problem for the graph G = (V, E) with V = {1, 2, 3} and E = {{1, 2}, {2, 3}, {1, 3}}.
   (i) Write down the linear program relaxation for the vertex cover problem and solve the linear program. [6 marks]

(ii) Based on the solution of the linear program in (b)(i), derive an integer solution using the rounding approach described in the lecture. [2 marks]

5. Consider the following randomized algorithm for the unweighted vertex cover problem:

```
Initialize S to be the empty set
For all edges e=(u,v) do
   If neither u nor v belongs to S
      Randomly choose u or v with probability 1/2
      and add the vertex to S
   End If
End For
Return S
```

Derive an upper bound, as tight as possible, on the approximation ratio of the algorithm.

Hint: Try to find an invariant that bounds from below the size of the intersection of the current solution S = S(i) with the optimum solution, where S(i) denotes the set S after the i-th iteration of the FOR loop. [9 marks]

6. Give two examples of greedy algorithms and state their approximation ratios. [4 marks]

7. Consider the Centre Selection Problem, defined as follows. The input consists of n points p1,p2,... ,pn in a metric space, and an integer k > 0. The goal is to find k centers C = (c1, c2,... , ck} (not necessarily from among the n points) so that r(C) = maxi<i<n dist (p^C), where dist (p^C) = min1<?-<fc dist (pi,c?-), is minimized.

   (i) Consider the standard greedy approach: solve the problem optimally for k =1 and then extend the solution to larger values of k by adding the optimal point to the current solution. Why is this likely to give a poor result? [4 marks]

   (ii) Consider the following algorithm to solve the Centre Selection Problem:

```
Let C be the empty set
Repeat k times
    Select a point p_i with maximum distance dist (p_i,C)
    Add point p_i to the set C
Return C
```

Derive a lower bound for this algorithm on the minimum pairwise distance among the chosen centers C. [4 marks]

(iii) Give an upper bound, as tight as possible, on the approximation ratio of the algorithm in part (b)(ii). [2 marks]

(iv) Give a detailed analysis in order to justify your answer for part (b)(iii). Hint: Exploit the lower bound derived in part (b)(ii) in order to construct disjoint balls around the center points. [6 marks]

8. Given an algorithm for the SET-COVER problem as a black box, how could you use this to solve the unweighted VERTEX-COVER problem? [4 marks]

9. Following the approach in Part (a), which approximation ratio for the VERTEX COVER problem do you achieve by applying the greedy algorithm for the SET-COVER problem? What happens if every vertex in the graph has at most 4 neighbors? [6 marks]

10. Consider the following greedy algorithm for the unweighted VERTEX-COVER problem:

    Compute a directed Depth-First-Search tree (DFS-tree) from every connected component in the graph, and output all nodes which are not leaves in the DFS-tree (a vertex is a leaf if it has no outgoing edges in the DFS-tree).

    (i) What is the running time of this algorithm? [2 marks]

    (ii) Why is the returned solution a valid vertex cover? [4 marks]

    (iii) Derive a bound, as good as possible, on the approximation ratio of this algorithm.

    Hint: You may use the fact that in any undirected graph G = (V, E), $\sum_{u \in V} \deg(u) = 2|E|$, where deg(u) denotes the number of neighbors of u. [4 marks]

11. What are the three possible cases for the solution of a linear program? For each of them, give an example of a linear program in standard form exhibiting this case. [6 marks]

12. What is the set of optimal solutions for the following linear program?

$$\text{Minimize} \quad -x_1 - x_2$$
$$-x_2 \geq -3$$
$$2x_1 + x_2 < 8$$
$$x_1 . x_2 \geq 0$$

[6 marks]

13. For a given linear program LP_1

$$\text{Maximize} \quad \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad (1 \leq i \leq m)$$

$$x_j \geq 0 \qquad (1 \leq j \leq n),$$

consider a new linear program $\mathbf{LP}_2$:

$$\text{Minimize} \quad \sum_{i=1}^{m} b_i y_i$$

$$\sum_{i=1}^{m} a_{ij} y_i \geq c_j \qquad (1 \leq j \leq n)$$

$$y_i \geq 0 \qquad (1 \leq i \leq m).$$

14. Prove that if x is a feasible solution for LP_1 and y is a feasible solution for $\mathbf{LP}_2$, then $c^T X \leq b^T y$ cT. [6 marks]
15. Using your answer in Part (c)(i), what can we conclude about LP_2 if we know that LP_1 is unbounded? [2 marks]
16. For each of the following claims, state whether it is true or not and give a brief justification.
    - (i) For any linear program with n variables and m constraints, there are at most $\binom{n+m}{m}$ different basic solutions. [2 marks]
    - (ii) The Simplex Algorithm has a worst-case polynomial runtime. [2 marks]
    - (iii) In each iteration of the Simplex Algorithm, the value of the objective function changes. [2 marks]
    - (iv) The auxiliary linear program in Initialize-Simplex always has a feasible solution. [2 marks]
    - (v) The fundamental theorem of linear programming also holds if linear constraints are allowed to be strict. [2 marks]
    - (vi) The set of feasible solutions of any linear program forms a convex set. [2 marks]
17. For the following linear program, write down the auxiliary linear program used by Initialize-Simplex in slack form: [3 marks]

$$\begin{aligned} \text{minimize} \quad & -4x_1 + x_2 \\ \text{subject to} \quad & -4x_1 + 2x_2 \geq -4 \\ & x_1 - 6x_2 \leq -3 \end{aligned}$$

18. Recall the algorithm for the unweighted vertex cover problem that is based on rounding the solution of a linear program.
    (i) What is the approximation ratio of this algorithm? [1 mark]
    (ii) Give an example of a graph and the corresponding linear program for which the gap between the linear program solution and optimal solution is as large as possible. [4 marks]
19. Consider the definition of an approximation algorithm.
    (i) Explain the meaning of approximation ratio in the case of a maximization problem. [2 marks]
    (ii) How is this definition adjusted to the case of a randomized approximation algorithm? [2 marks]
20. State the definition of PTAS and FPTAS. [4 marks]
21. Let G = (V, E) be an undirected graph. For any k > 1, define $G^{(k)}$ to be the undirected graph $\left(V^{(k)}, E^{(k)}\right)$, where $V^{(k)}$ is the set of all ordered k-tuples of vertices from V and $E^{(k)}$ is defined so that (v_1, v_2,..., v_k) is adjacent to (w_1,w_2,..., w_k) if and only if (v_1,v_2,..., v_k, w_1, w_2,..., w_k} forms a clique.
    a. Argue that the graph $G^{(k)}$ can be constructed in time polynomial in n (for any fixed value of k). [3 marks]
    b. Prove that the size of the maximum clique in $G^{(k)}$ is equal to the k-th power of the size of the maximum clique in G. [5 marks]
    c. Argue that if there is a polynomial-time approximation algorithm that has a constant approximation ratio for finding a maximum clique, then there is a polynomial-time approximation scheme (PTAS) for the problem.
       Hint: Your PTAS should be based on applying the given approximation algorithm with constant approximation ratio to $G^{(k)}$ for a proper choice of k > 0. Then use the equivalence in part (ii) to analyses its approximation ratio.
       [4 marks]

22. State the fundamental theorem of Linear Programming. [3 marks]

23. Consider the following linear program:

$$\begin{aligned} \text{minimise} \quad & 4 \cdot x_1 - x_2 \\ & -x_1 + 5x_2 \geq 4 \\ & x_1 - 0.5x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

a. Convert this linear program into slack form. [3 marks]

b. What is the number of different slack forms of the linear program in Part (b)(i)? [2 marks]

c. Give at least one non-feasible and one feasible basic solution of the linear program in (b)(i). [4 marks]

24. Consider the following separation problem. We are given m points $x^1 = (x_1^1, x_2^1), x^2 = (x_1^2, x_2^2), \ldots, x^m = (x_1^m, x_2^m) \in \mathbb{R}^2$ and n points $y^1 = (y_1^1, y_2^1), y^2 = (y_1^2, y_2^2), \ldots, y^n = (y_1^n, y_2^n) \in \mathbb{R}^2$. The goal is to find a "separating" vector $w = (w_1, w_2) \in \mathbb{R}^2$ (if it exists) such that:

$$\langle x^i, w \rangle = \sum_{j=1}^{2} x_j^i w_j > 0 \qquad \text{for } i = 1, 2, ..., m,$$

and

$$\langle y^i, w \rangle = \sum_{j=1}^{2} y_j^i w_j < 0 \qquad \text{for } i = 1, 2, ..., n.$$

(i) Create a new, equivalent system of inequalities by replacing each strict inequality by a suitable non-strict inequality. Justify why this new system has a solution if and only if the original system has one. [4 marks]

(ii) Based on your answer in Part (c)(i), how can you solve the above problem using linear programming? [4 marks]

25. What is the approximation ratio of an approximation algorithm? [2 marks]

26. State the definitions of PTAS and FPTAS. [4 marks]

27. Consider the two approximation algorithms for VERTEX-COVER from the lectures (one greedy algorithm and one based on rounding a linear program).

(i) What are the approximation ratios of these two algorithms? [2 marks]

(ii) Construct an input graph that demonstrates the tightness of the approximation ratio of the greedy algorithm (for full marks, your construction should work for any even number of vertices n). [3 marks]

28. Consider the following randomized algorithm to compute a solution of the VERTEX-COVER problem for an unweighted graph G = (V, E):

    Let C be the empty set While E not empty do

        Pick any edge e={u,v} from E

        Choose x from {u,v} uniformly at random

        Add x to C

        Remove all edges incident to x from E

    End While

    Return C

    (i) Explain briefly why the set C returned is a valid vertex cover. [2 marks]

    (ii) Find a lower bound on the probability that the algorithms return an optimal solution.

    Hint: For each edge e = {u,v} picked by the algorithm consider the event that the chosen vertex x E {u,v} added to C is also part of an optimal cover.

    [4 marks]

    (iii) Given a lower bound p E (0,1) on the probability that this algorithm returns an optimal solution, describe a new algorithm that returns an optimal solution with probability at least 5, for any given $\partial \in [P, 1)$. [3 marks]

29. Suppose you have a randomized approximation algorithm for a maximization problem such that, for any e > 0 and any problem instance of size n, the algorithm returns a solution with cost C such that

    Pr[C > (1 - 1/e) ■ C*] > 1/n ■ exp(-1/e),

    where C* is the cost of the optimal solution. Can you use your algorithm to obtain a PTAS or FTPAS? Justify your answer. [6 marks]

30. We consider the following optimization problem. Given an undirected graph G = (V, E) with non-negative edge weights w: E → R+, we are looking for an assignment of vertex weights x: V → R such that: (i) for every edge {u, v} ∈ E, x(u) + x(v) > w({u,v}), (ii) $\sum_{v \in V} x(v)$ is as small as possible.

    (i) Design a 2-approximation algorithm for this problem. Also analyses the running time and prove the upper bound on the approximation ratio. Note: For full marks, your algorithm should run in at most O (E^2) time. Hint: One way to solve this question is to follow the approach used by the greedy approximation algorithm for the VERTEX-COVER problem. [8 marks]

(ii) Can this problem be solved exactly in polynomial-time? Either describe the algorithm (including a justification of its correctness and why it is polynomial time) or prove that the problem is hard via a suitable reduction. [6 marks]

31. Assume you have a randomized approximation algorithm for a maximization problem, and your algorithm achieves an approximation ratio of 2. What can you deduce for E [C */C],
where C* is the cost of the optimal solution, C is the cost of the solution of the approximation algorithm, and E[.] denotes the expectation? [4 marks]

32. Consider the following optimization problem on graphs: Given an undirected, edge-weighted graph G = (V, E, w) with w: E ^ R+, we want to find a subset S C V such that w (S, V \ S) = $\sum_{e \in E}(S, V \backslash S)^{w(e)}$ (the total sum of weights over all edges between S and V \ S) is maximized.

(i) Design a polynomial-time approximation algorithm for this problem. Also analyses its running time and prove an upper bound on the approximation ratio. [8 marks]

(ii) Find a graph which matches your upper bound on the approximation ratio from Part (b)(i) as closely as possible. [4 marks]

(iii) Consider now the following generalization of the problem. Given an integer k > 2, we want to partition V into disjoint subsets s_1, s_2,..., s_k so that we maximize.

$$\sum_{i=1}^{k} w(S_1, V \backslash S_1).$$

Describe an extension of your algorithm in Part (b)(i). What approximation ratio can you prove for this algorithm? [4 marks]