Exam IN4301 Advanced Algorithms – Part II

January 27, 2014, 14:00–16:00

- Please answer questions 1 and 2 on a **separate sheet** from questions 3, 4 and 5.
- This is a closed book examination with 5 questions worth 50 points in total. It covers the material in the **second half of the course**.
- Use of books, readers, notes, slides and (graphical) calculators is not allowed.
- Your grade for this part of the exam will be the number of points awarded, divided by 5.
- Write your name, student number, degree program, and number of submitted sheets of paper on the first page.
- Write clearly, in correct English, and avoid verbose explanations. Giving irrelevant information may lead to a reduction in your score.
- The total number of pages of this exam is 2.

Questions

1. Let $G = (V, E)$ be an undirected graph. Recall that we have the following integer linear programming formulation for the maximum independent set problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{v \in V} x_v \\ \text{subject to} \quad & x_u + x_v \le 1 \quad \text{for} \quad \{u, v\} \in E, \\ & x_v \in \{0, 1\} \quad \text{for} \quad v \in V. \end{aligned}$$

   (a) (1 point) Give the linear programming relaxation.

   (b) (3 points) Show that you can find instances with arbitrarily small integrality gap (or arbitrarily large depending on the definition of integrality gap). Hint: use complete graphs.

   (c) (3 points) What does this tell you about the possibility of constructing an approximation algorithm based on the above linear programming relaxation? Explain your answer.

   (d) (1 point) For a perfect graph we can give a semidefinite programming relaxation whose optimal value $\vartheta(G)$ equals the independence number $\alpha(G)$ (the size of a largest independent set). Explain how we can use this to find $\alpha(G)$ in polynomial time.

   (e) (3 points) Use this to give a polynomial time algorithm to find a maximum independent set in a perfect graph. Explain why the algorithm is correct and why it runs in polynomial time.

   (f) (3 points) Now assume that each vertex $v \in V$ has a nonnegative weight w_v and that two integers $m < M$ are given. Consider the problem of finding an independent set S whose weight $\sum_{v \in S} w_v$ is as large as possible where we want the size (the number of elements) of the independent set S to be at least m and at most M. Write this problem as an integer linear program.

2. Given subsets $S_1, \ldots, S_m$ of $\{1, \ldots, n\}$ of size 3, we consider the problem of finding a smallest subset I of $\{1, \ldots, n\}$ such that the intersection $I \cap S_i$ is nonempty for every $i = 1, \ldots, m$.

   We can write this problem as the integer linear program (ILP)

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{n} x_j \\ \text{subject to} \quad & \sum_{j \in S_i} x_j \geq 1 \quad \text{for} \quad i = 1, \ldots, m, \\ & x_j \in \{0, 1\} \quad \text{for} \quad j = 1, \ldots, n. \end{aligned}$$

   The linear programming relaxation is obtained by replacing the constraint $x_j \in \{0, 1\}$ by $x_j \geq 0$.

   (a) (4 points) Show that the following algorithm generates a feasible solution to the ILP:
      - Obtain an optimal solution x^* to the linear programming relaxation.
      - Return the vector $x \in \mathrm{R}^n$ defined by

$$x_j = \begin{cases} 1 & \text{if } x_j^* \geq \frac{1}{3}, \\ 0 & \text{otherwise.} \end{cases}$$

   (b) (4 points) Show that the above algorithm is in fact a 3-approximation algorithm.

   (c) (3 points) Give the augmented form of the above linear programming relaxation.

Please answer the following questions on a separate sheet.

3. In his paper "Needed: An Empirical Science of Algorithms", Hooker writes: "It is symptomatic of the situation that in OR and computer science one cannot publish reports that an algorithm does *not* perform well in computational tests. Negative results are as important as positive results and are routinely reported in other empirical sciences."

   (a) (4 points) Describe what, in science *in general*, a 'negative result' is. Use the relevant notions from the empirical cycle (or steps in the scientific method) in your answer.

   (b) (3 points) Explain why *reporting* negative results is important.

   (c) (4 points) Give an example of a negative result from the empirical science of *algorithms*, and explain how it fits the notion of 'negative result'. (There are examples in the papers by Hooker & Vinay and by McGeoch.)

4. (a) (3 points) What does it mean to make "*heuristic* use of experimentation" (Hooker)?

   (b) (4 points) Which of the three studies (1) on branching rules for SAT by Hooker & Vinay, (2) on the FFD algorithm for Bin Packing by McGeoch, and (3) on GSAT by Gent & Walsh does *not* make "heuristic use of experimentation"? Explain how the other two studies *do* use experimentation in this way.

5. Given a set $E = \{e_1, \ldots, e_n\}$, subsets $S_1, \ldots, S_m \subseteq E$ and non-negative cost w_j for each S_j, the weighted set cover problem is to find a set $I \subseteq \{1, \ldots, m\}$ such that $\bigcup_{j \in I} S_j = E$ and $\sum_{j \in I} w_j$ is minimized. Gomes et al. plot the performance ratio of four approximation algorithms on the y-axis, for 30 instances on the x-axis. We focus here on one of the algorithms with approximation ratio $f = \max_{1 \leq i \leq n} |\{j \mid e_i \in S_j\}|$.

   (a) (3 points) Suppose it's not known whether the ratio f is tight, and you want to investigate this empirically. Why can you not use Gomes et al.'s plot for this purpose?

   (b) (4 points) If you know, for each instance i, the values for n, m, f, the optimal weight W, and the weight A of the solution found by the algorithm, how can you compute a metric $m(i) \in [0, 1]$ that supports your investigation? Make up some data to explain your solution, and illustrate when your metric would allow you to conclude that the ratio is tight.

End of exam questions