Discrete Optimization Take Home Exam [1]

March 08	Take Home, Due March 09, 10 AM

Ansvarig:
Devdatt Dubhashi   Tel. 772 1046   Rum 6479 EDIT

Max points:	60	
Grade criteria:	Chalmers	5:48, 4:36, 3:24
	GU	VG:48, G:24
	Doktorander	G:36
Helping material:	Course book, material on course page.	

- You are required to **work alone**.
- Recommended: First look through all questions and make sure that you understand them properly. In case of doubt, do not hesitate to ask.
- Answer concisely and to the point. (English if you can and Swedish if you must!)
- Code strictly forbidden! Motivated pseudocode or plain but clear English/Swedish description is fine.

Lycka till!

[1] 2010 LP3, TDA206/DIT370.

Problem 1 Helping Malmö Aviation [10] Malmö Aviation flies between Stockholm (yuppie city), Göteborg (your friendly city) and Malmö (Zlatan city). For this problem, focus on the Friday afternoon flight that departs from Stockholm, stops in Göteborg and continues to Malmö. There are three types of passengers:

(a) Those going from Stockholm to Göteborg (S–G).

(b) Those going from Göteborg to Malmö (G–M).

(c) Those going from Stockholm to Malmö (S–M).

The aircraft has a maximum capacity of 80 passengers. Malmö Aviation offers three fare classes:

(a) Super–flex (fully refundable).

(b) Flex (can change booking subject to availabilty).

(c) Economy (no rebooking, 3 week advance purchase).

Ticket prices (largely determined by external influences like competitors) have been set and are advertised as follows:

	S–G	G–M	S–M
SuperFlex	400	350	530
Flex	600	450	870
Economy	800	650	980

Based on past experience, demand forecasters at Malmö Aviation have determined the following upper bounds on the number of potential customers in each of the 9 possible origin–destination/fare–class combinations:

	S–G	G–M	S–M
SuperFlex	15	12	10
Flex	35	23	18
Economy	45	38	43

The goal is to decide how many tickets from each of the 9 origin–destination/fare–class combinations to sell. The constraints are that the plane cannot be overbooked on either of the two legs of the flight and the number of tickets made available cannot exceed the forecasted maximum demand. The objective is to maximize the revenue. Formulate the problem as a ILP.

Problem 2 Vertex Cover [10] Show that the weighted vertex cover problem can be solved in polynomial time for bipartite graphs using LP. Give a full justification.

Problem 3 Vertex Cover again [10] Show that in any basic feasible solution (BFS) of the LP relaxation (equation (3.3) in MG) of the ILP for the vertex cover problem in a general graph (not necessarily bipartite) $x_v \in \{0, 1/2, 1\}$ for all $v \in V$. Hence deduce another 2–approximation algorithm for the weighted vertex cover problem in general graphs. (HINT: recall that $\mathbf{x}$ is a BFS iff it cannot be written as $\lambda\mathbf{y} + (1 - \lambda)\mathbf{z}$ for $\lambda \geq 0$ and any $\mathbf{y} \neq \mathbf{x} \neq \mathbf{z}$.)

Problem 4 Set Cover [10] The Set Cover problem is an abstraction that occurs repeatedly in problems faced by Jeppesen in airline scheduling. Given a universal set $U := \{1, \cdots, n\}$ and subsets $S_1, S_2, \cdots, S_m \subseteq U$ with non–negative weights $w_1, w_2, \cdots, w_m$, the problem is to find a sub-collection $S_{i_1}, S_{i_2}, \cdots, S_{i_k}$ such that the sub-collection covers U i.e. $\bigcup_{i_\ell} S_{i_j} = U$ and the total cost $\sum_{i_\ell} w_{i_\ell}$ is minimized.

(a) Show that the weighted vertex cover problem is a special case of the set cover problem.

(b) Formulate the Set Cover problem as an ILP. Give a brief justification. Describe the constraint matrix: what are its dimensions and what is the (i, j) entry?

(c) Pass to the LP relaxation, and suggest a rule to round the optimal LP solution to a solution to the ILP.

(d) Give an analysis for an approximation that your rule guarantees.

(e) What is the integrality gap of your (ILP) -(LP) pair and what does it imply for approximation algorithms?

Problem 5 How to Solve LP Without GLPK [10] Without using GLPK or any LP solver, find the optimum value and the optimum solution $(x_1^*, \cdots, x_5^*)$ for

$$\min 2x_1 + 3x_2 + 5x_3 + 2x_4 + 3x_5$$

subject to

$$\begin{aligned} x_1 + x_2 + 2x_3 + x_4 + 3x_5 &\geq 4 \\ 2x_1 - 2x_2 + 3x_3 + x_4 + x_5 &\geq 3 \\ x_1, x_2, x_3, x_4, x_5 &\geq 0 \end{aligned}$$

HINT: Write the dual, solve it graphically and use the complementary slackness conditions.

Problem 6 Implementing Vertex Cover [10] Consider the primal–dual algorithm for vertex cover from class and discussed at the end of section 1 in David Williamson's survey available on the course webpage. (The algorithm he discusses is slightly different though.) This problem develops a version that would actually be implemented.
Recall the continuous time description: at time 0, no vertex is included in the cover and all dual variables have value $y_{i,j}^0 = 0$ and all edges are active. At any time $t \geq 0$, the dual variables corresponding to the active edges are raised simultaneously at unit rate (i.e. in 5 sec, they increase in value by 5 for example). This continues until a time $t' \geq t$ when a vertex i becomes tight i.e. the corresponding dual constraint is satisfied with equality: $\sum_{(i,j)\in E} y_{i,j}^{t'} = w_i$. At this time, vertex i enters the cover, and all active edges (i, j) incident on i are now made inactive and their dual variables are frozen to the values $y_{i,j}^{t'}$ (which remain the same at later times). This continues until we have a vertex cover i.e. all edges are inactive.

(a) What is the value of the dual variable $y^t_{i,j}$ for an edge (i, j) active at time t?

(b) What is the time $t+\delta t$ upto which the dual variables of active edges can be raised starting at time t i.e what is the value δt such that at time $t + \delta t$ the next vertex become tight? Your answer should involve the surplus capacity of a vertex i at time t given by $w^t_i := w_i - \sum_{(i,j)\in E} y^t_{i,j}$, and its active degree Δ^t_i which is the number of active edges incident on i at time t.

(c) Based on (a) and (b), write down a discrete version of the algorithm, where in the loop, you specify what is the next vertex to pick in the cover and the value at which the active dual variables corresponding to this vertex are frozen and become inactive. Specify also other necessary updates. Your algorithm must run in time $O(|V|^2)$.

(d) Can you implement the algorithm to run in time $O(|E| + |V| \log |V|)$? You must give a complete justification of the running time. (HINT: Check out the Fibonacci Heap data structure.)