LUNDS TEKNISKA HÖGSKOLA
MATEMATIK

HOME EXAM
LINEAR AND COMBINATORIAL OPTIMIZATION
2016-03-07

The exam consists of six problems which will be awarded with a maximum of one point each. For a passing grade (3), at least 3 problems have to be solved correctly. Credit can be given for partially solved problems. Write your solutions neatly and explain your calculations.

You may use any books, but it is not permitted to get help from other persons. You are encouraged to use Matlab, except in problems 2 and 4, which have to be solved by hand. You need to write your own code, with the exceptions of files that were used in this course.

The exam can be picked up between 7/3 and 11/3 at the Student office. The exam should be handed in to the Student office in the Mathematics Department, LTH at the latest exactly 7 days after collecting the exam. Write your name, section-year (or subject for PhD-students), id-number and email address on the first page, and write your name on each of the following pages. Please also send an email to me (sara@maths.lth.se) with the Matlab code as attachments before the deadline. I will contact you about the oral part of the exam when the written exam papers have been marked.

1. Solve the following linear programming problem

$$\text{maximize } z = 2x_1 + x_2 + 3x_3 + 4x_4$$
$$\text{subject to } \begin{cases} 4x_1 + 2x_2 + 5x_3 + 5x_4 \leq 10, \\ 4x_1 + 2x_2 + 5x_3 + 5x_4 \geq 5, \\ 3x_1 + 5x_2 + 4x_3 + x_4 \geq 8, \\ 3x_1 + 5x_2 + 4x_3 + x_4 \leq 15, \\ x_1 + x_2 + x_3 + x_4 = 3. \end{cases}$$

using the two phase method. Present all the tableaus which are obtained in the process, and explain how you choose your incoming and outgoing variables. It is recommended that you use the function `checkbasic1.m` that you made in handin 1. Are there any feasible solutions of this problem? If there is an optimal solution, find it and present it together with the optimal value.

2. Suppose that $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a matrix with the property $\mathbf{A}^{\mathbf{T}} = -\mathbf{A}$, that $\mathbf{c} \in \mathbb{R}^n$, and that the following linear programming problem has a feasible solution:

$$\text{maximize } z = \mathbf{c}^{\mathbf{T}}\mathbf{x}$$
$$\text{subject to } \begin{cases} \mathbf{A}\mathbf{x} \leq -\mathbf{c}, \\ \mathbf{x} \geq \mathbf{0}. \end{cases}$$

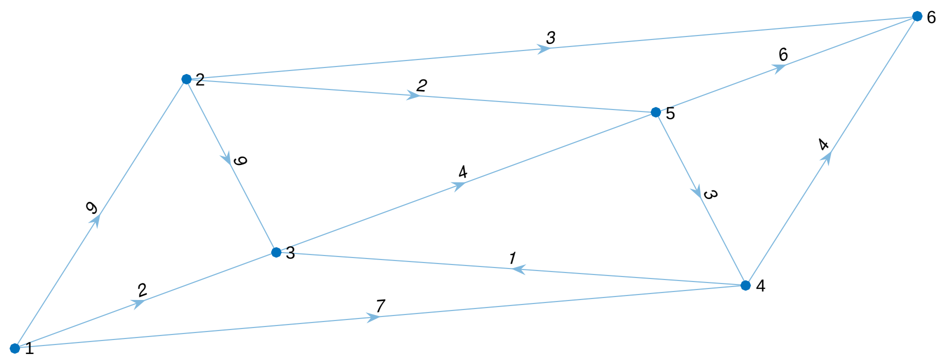
Prove that the problem has an optimal solution. What is the optimal objective function value of this problem?

3. Maximize $z = x_1 + 2x_2 + x_3 + x_4$, subject to

$$\begin{cases} 2x_1 + x_2 + 3x_3 + x_4 \leq 8, \\ 2x_1 + 3x_2 + 4x_4 \leq 12, \\ 3x_1 + x_2 + 2x_3 \leq 18, \\ x_j \geq 0, \\ x_j \in \mathbb{Z}, \qquad j = 1, 2, 3, 4, \end{cases}$$

using the branch and bound method with the bounds coming from linear programming problems, as was done in Lecture 6 for a different example. Present the initial and final tableau for each LP problem solved, and describe how the branches of the tree are constructed. Finally, present the optimal solution together with the optimal value if they exist.

4. This problem should be solved by hand, and not with Matlab. Consider the flow network below with source $S = 1$ and sink $T = 6$, with edge capacities indicated near each edge.



a) Find a maximum flow in this network. Explain how you came up with this solution. This should be done in a systematic way.

b) Find a minimum cut in this network, i.e. write down two sets of vertices each belonging to the two nonempty disjoint sets of vertices that define the minimum cut.

5. Sudoku is a logic-based, combinatorial number-placement puzzle. The objective is to fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 sub-grids that compose the grid contains all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a unique solution.
Instead of solving the full Sudoku puzzle, the task of this exercise will be to write a program in matlab which solves a smaller version of Sudoku, using a 4×4 grid. The 4×4 grid should be filled with numbers from 1 to 4 in such a way that each column, each row, and each of the four 2×2 subgrids that compose the grid contains all of the digits from 1 to 4. After doing this successfully, it will be easy to extend the algorithm to solve 9×9 Sudoku puzzles as well, but this will not be part of the exercise.

a) We start by deciding what the variables of the problem should be. For each small square (containing one digit), we are going to use four variables which can each take the values 0 or 1. This will give a total of $4 \times 16 = 64$ binary variables. All of the variables should be put in a column vector $\mathbf{x}$, taking the variables corresponding to small squares in order row-wise from left to right and top to bottom. Let $x_1, \ldots, x_4$ be the variables which correspond for the top left small square. The possible values for the variables (x_1, x_2, x_3, x_4) are $(1, 0, 0, 0)$ (corresponding to the number 1), $(0, 1, 0, 0)$ (corresponding to the number 2), $(0, 0, 1, 0)$ (corresponding to the number 3 and $(0, 0, 0, 1)$ (corresponding to the number 4). Note that the variable $x_j = 1$ if and only if the number in the first small square is j, and that $x_k = 0$ for $k \in \{1, \ldots, 4\} \setminus \{j\}$. Note that exactly one of the variables $x_1, \ldots, x_4$ will take the value 1, and so the coding of the variables for the first small square gives the constraints $x_1 + x_2 + x_3 + x_4 = 1$, $x_j \in \{0, 1\}$ for $j = 1, \ldots, 4$. Write down similar constraints for all the variables $x_1, \ldots, x_{64}$. Write them on the form $\mathbf{A_1 x} = \mathbf{b_1}$, where $\mathbf{A_1}$ is a matrix of size 16×64 and $\mathbf{b_1}$ is a column vector of length 16. Specify $\mathbf{A_1}$ and $\mathbf{b_1}$. Don't forget to state the integer constraints.

b) Each row in the grid should contain all the numbers from 1 to 4, and each number between 1 and 4 should appear exactly once in each row. Write down linear equality constraints for the variables $x_1, \ldots, x_{64}$ which are necessary and sufficient for this to happen. Write them on the form $\mathbf{A_2 x} = \mathbf{b_2}$, where $\mathbf{A_2}$ is a matrix of size 16×64 and $\mathbf{b_2}$ is a column vector of length 16. Specify $\mathbf{A_2}$ and $\mathbf{b_2}$.

c) Each column in the grid should contain all the numbers from 1 to 4, and each number between 1 and 4 should appear exactly once in each column. Write down linear equality constraints for the variables $x_1, \ldots, x_{64}$ which are necessary and sufficient for this to happen. Write them on the form $\mathbf{A_3 x} = \mathbf{b_3}$, where $\mathbf{A_3}$ is a matrix of size 16×64 and $\mathbf{b_3}$ is a column vector of length 16. Specify $\mathbf{A_3}$ and $\mathbf{b_3}$.

d) Each of the four sub-grids should contain all the numbers from 1 to 4, and each number between 1 and 4 should appear exactly once in each sub-grid. Write down linear equality constraints for the variables $x_1, \ldots, x_{64}$ which are necessary and sufficient for this to happen. Write them on the form $\mathbf{A_4 x} = \mathbf{b_4}$, where $\mathbf{A_4}$ is a matrix of size 16×64 and $\mathbf{b_4}$ is a column vector of length 16. Specify $\mathbf{A_4}$ and $\mathbf{b_4}$.

e) Each sudoku puzzle comes with a few clues in the sense that the grid is partially completed. This will give rise to additional constraints (one extra constraint for each number clue). Write down the constraints arising from the clues for the following puzzle:

1		3	
4			2

Write them on the form $\mathbf{A_5x} = \mathbf{b_5}$, where $\mathbf{A_5}$ is a matrix of size 4×64 and $\mathbf{b_5}$ is a column vector of length 4. Specify $\mathbf{A_5}$ and $\mathbf{b_5}$.

f) The objective of the puzzle is to find a feasible solution. For sudoku puzzles, each puzzle should be constructed so that there exists a unique feasible solution. We would like to solve the problem as an optimization problem, and so we can choose any linear function as our objective function. Pick an objective function of your choice, and prove by refering to a suitable theorem, that the integer linear programming problem, is equivalent to a linear programming problem. State this linear programming problem using the matrices and vectors that were defined above. Then rewrite the problem so that it is on standard form.

g) Since the feasible set is very small (it contains only one point), it may not be so easy to find a basic feasible solution directly. It will be much easier to find a feasible solution of the dual problem. Write down the dual of the problem that you stated in subproblem f), for example using the table on p. 160 in the book by Kolman and Beck. Be careful to take all the constraints of the primal problem into account. Then rewrite the dual problem on canonical form. If the objective function in f) was chosen in a good way, it will be easy to find a basic feasible solution of the dual problem. You may have to change the objective function if this is not the case.

h) Solve the sudoku puzzle using the simplex method, preferably using the function `simp` from lab 1. Test the method on some other instances of puzzles that you come up with yourself. Note that the matrices $\mathbf{A_1}, \dots \mathbf{A_4}$ will be the same for all puzzles. Only $\mathbf{A_5}$ will change.

i) How many variables would there be in a full Sudoku puzzle with a 9×9 grid solved with this method?

6. Using a genetic algorithm, come up with a solution to the knapsack problem. As we have seen, the problem is that we have a knapsack of capacity C, which represents a maximal total weight or size. Furthermore, there is a set of objects that can be put in the knapsack. Each object has a size and a value, which both are positive integers. To solve the problem, we need to decide which objects to put in the knapsack in a way so that the total size of the objects is less than or equal to the capacity C, and so that the value of the objects in the knapsack is maximized. Test your algorithm on knapsack problems of different sizes, and discuss what happens.