OR991

Faculty of Electrical and Computer Engineering

Operations Research

University of Tehran

Teacher Assistants

Under supervision of Mohammad Rabiei

January 2, 2021

OR991

Colophon
This document was typeset with the help of KOMA-Script and LaTeX using the kaobook class.
The source code of this book is available at:
`https://github.com/fmarotta/kaobook`
(You are welcome to contribute!)

Version
This e-book is the first version of itself and is written during Fall Semester of 1399.

"The linear-programming was – and is – perhaps the single most important real-life problem.".

– Keith Devlin

Contents

List of Figures

List of Tables

Preamble

Abet | 1

Report this file's problems to me via this email address.

University of Tehran
Faculty of Electrical and Computer Engineering

Course	Operations Research <8101145>
Course type	Obligatory for control cluster of electrical engineering Optional for other branches
Level	Undergraduate ■ Graduate ☐
Co-requisite(s)	–
Prerequisite(s)	Introduction to Computer Systems and Programming <8101347>
Required Topics	Calculus, Algorithm and Programming
Textbook(s)	1. Linear and Nonlinear Programming, David G. Luenberger, Yinyu Ye, Springer, 3rd Edition, 2008 2. Introduction to Operations Research, F.S. Hillier, G.J. Lieberman, Eighth Ed., McGraw-Hill, 2008.
Coordinator	Mohammad Shokri
Goals	The main goal of this course is using analytical methods to decide better. To reach this goal, students will be introduced with modeling optimization problems, optimization algorithms, and analysis of the optimal solutions.
Outcome	Upon successful completion of this course, students will be able to understand different steps of operations research and how formulate constrained optimization problems in a formal mathematical perspective and finally how to solve and analyze some of basic problems in this context.
Topics	1. Introduction to operations research 2. Linear programming 3. Integer programming 4. Logical constraints 5. Applications 6. Unconstrainted nonlinear optimization 7. Constrainted nonlinear optimization
Computer usage	Students must be familiar with basic programming concepts and how to apply them in Python3 language.

Table 1.1: ABET Table

Assignments	5 homework assignments are given over term to assist students with basic mathematic backgrounds of course context and 3 computer assignments to work with relevant python3 libraries.
Projects	This course has a final project in which students cover different phases of operations research in a simplified environment.
Grading	Homeworks 3.75 points Assignments 2.00 points Quizzes 1.50 points Midterm 4.00 points Final 7.00 points Project 2.25 points
Further readings	Stephen Boyd and Lieven Vandenberghe. 2004. Convex Optimization. Cambridge University Press, USA.
Date	January 2, 2021

Programming in Python | 2

2.1 What is Python?

Python is an interpreted [1] , high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

1: An interpreted language is a type of programming language for which most of its implementations execute instructions directly and freely, without previously compiling a program into machine-language instructions. The interpreter executes the program directly, translating each statement into a sequence of one or more subroutines, and then into another language (often machine code).

Figure 2.1: Guido van Rossum
Creator of Python programming Lanuage

2.2 Why we need it?

In the context of this course, we'll need some of python libraries to integrate them with internal or external solvers and solve some of our pre-modelled problems numerically. So we don't need advanced python knowledge. Basic workflow suffices and further understanding of libraries' syntax may be accomplished during the semester.

Linear Programming

Linear Programming | 3

3.1 Simplex Algorithm

$$\begin{array}{ll}
\text{min} & c_1x_1 + c_2x_2 + \ldots + c_nx_n + c_0 \\
\text{subject to} & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
 & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\
 & \vdots \qquad\qquad\qquad\qquad\qquad \vdots \\
 & a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \\
\text{and} & x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0
\end{array}$$

1. Create a table as following:

$$\left[\begin{array}{c|cccc|c}
 & x_1 & x_2 & \ldots & x_n & \\
\hline
-Z & c_1 & c_2 & \ldots & c_n & -c_0 \\
\hline
x_1 & a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\
x_2 & a_{21} & a_{22} & \ldots & a_{2n} & b_2 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
x_m & a_{m1} & a_{m2} & \ldots & a_{mn} & b_m
\end{array}\right] \tag{3.1}$$

2. Convert it to canonical form:

   a) Choose an internal variable of which b_j and a_{ij} have the same polarity
   b) We should convert the inner variable coefficients in the corresponding constraints to 1 and otherwise to 0, using Gauss elimination theorem. In other words the matrix below inner constraints should become an identity matrix, Like the table below. $x_1, x_2, ..., x_m$ are called internal variables while the rest of variables are called external variables.

$$\left[\begin{array}{c|ccccc|ccc|c}
 & x_{i1} & x_{i2} & x_{i3} & \ldots & x_{im} & x_{o1} & \ldots & x_{o(n-m)} & RHS \\
\hline
-Z & 0 & 0 & 0 & \ldots & 0 & c'_1 & \ldots & c'_{1(n-m)} & c'_0 \\
\hline
x_{i1} & 1 & 0 & 0 & \ldots & 0 & a'_{11} & \ldots & a'_{1(n-m)} & b_1 \\
x_{i2} & 0 & 1 & 0 & \ldots & 0 & a'_{12} & \ldots & a'_{2(n-m)} & b_2 \\
x_{i3} & 0 & 0 & 1 & \ldots & 0 & a'_{13} & \ldots & a'_{3(n-m)} & b_3 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
x_{im} & 0 & 0 & 0 & \ldots & 1 & a'_{1m} & \ldots & a'_{m(n-m)} & b_m
\end{array}\right] \tag{3.2}$$

   c) If any of b_js were negative: change the internal variable respective to this constraint with another variable, swapping their columns (select a variable with negative coefficient)
   d) Stop condition. We pursue previous steps until every b_j becomes positive.

special situations:
if in step C we do not have any negative coefficient, the problem is not feasible
if in step B in any row all of its elements become 0, The constraint is dependent to other constraints.
if all of the elements of any row except one element become 0, the problem is not feasible

3. Solving the problem:
   a) Select most negative element of C'_j as incoming element

   b) Select all of the positive elements of column j and choose the element with the least b_k/a'_{jk} as coming out element
   c) And move two columns
   d) By row operation at first: identify the coefficient matrix and convert all of the coefficients of cost function to zero
   e) Repeat A, B, C until all of the C_js are positive
      if in step B there is no element to be selected as an exit element, the answer is infinity.
      if at the end, any C'_j become zero , we have infinite answers
   f) finally the answer is : $-C'_0$, $x^*_{inj} = b'_j$, $x^*_{outj} = 0$

3.2 Simplex Homework

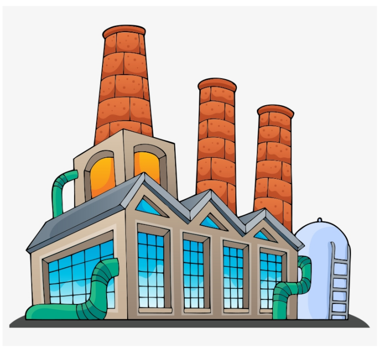
Ask your questions on this assignment from me!

Problem 1

A dietician wishes to mix two types of foods in such a way that vitamin contents of the mixture contain at least 8 units of vitamin A and 10 units of vitamin C. Food 'I' contains 2 units/kg of vitamin A and 1 unit/kg of vitamin C. Food 'II' contains 1 unit/kg of vitamin A and 2 units/kg of vitamin C. It costs Rs 50 per kg to purchase food 'I' and Rs 70 per kg to purchase food 'II'. Formulate this problem as a linear programming problem to minimize the cost of such a mixture.

Problem 2

A manufacturing company makes two models A and B of a product. Each piece of model A requires 9 labour hours for fabricating and 1 labour hour for finishing. Each piece of model B requires 12 labour hours for fabricating and 3 labour for finishing. For fabricating anf finishing, the maximum labour hours available are 180 and 30 respectively. The company makes a profit of Rs 8000 on each piece of model A and Rs 12000 on each piece of model B. Formulate this problem as a linear programming problem to minimize the cost of such a mixture.



Problem 3

Solve the following problem graphically. [1]

$$
\begin{aligned}
\text{maximize and minimize } \; Z &= 3x + 9y \\
\text{subject to } \; x + 3y &\le 60 \\
x + y &\ge 10 \\
x &\le y \\
x &\ge 0 \\
y &\ge 0
\end{aligned}
$$

1: By solving graphically, we mean you should draw feasible set on a 2D plane and evaluate corner points. Explain why we would do this. Don't use simplex algorithm in this problem!

Problem 4

Use simplex method to solve this problem:

$$
\begin{aligned}
\text{max} \quad & 2x - y + 2z \\
\text{subject to} \quad & 2x + y \le 10 \\
& x + 2y - 2z \le 2 \\
& y + 2z \le 5 \\
\text{where} \quad & x, y, z \ge 0
\end{aligned}
$$

Problem 5

A manufacturer produces three types of plastic fixtures. The time required for molding, trimming, and packaging is given in Table 9.1 (Times are given in hours per dozen fixtures.)

Process	Type A	Type B	Type C	Total time available
Molding	1	2	$\frac{3}{2}$	12,000
Trimming	$\frac{2}{3}$	$\frac{2}{3}$	1	4,600
Packaging	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	2,400
Profit	\$ 11	\$ 16	\$ 15	—

How many dozens of each type of fixture should be produced to obtain a maximum profit?

Problem 6

Use the simplex method to find the maximum of:

$$\begin{aligned} \text{max} \quad & 2x_1 + x_2 + x_3 \\ \text{subject to} \quad & x_1 + 3x_2 + x_3 = 3 \\ & x_1 - 2x_2 - 2x_3 = 4 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Problem 7

Use the simplex method to find the maximum of:

$$\begin{aligned} \text{max} \quad & 40y_1 + 30y_2 \\ \text{subject to} \quad & y_1 + y_2 \leq 12 \\ & 2y_1 + y_2 \leq 16 \\ & y_1, y_2 \geq 0 \end{aligned}$$

Problem 8

Minimum number of buses needed in i^{th} hour of a day is x_i $(i = 1, 2, \ldots, 24)$. Each bus works for 6 consecutive hours. If number of buses in i^{th} hour exceeds the minimum required by b_i, then an additional cost of c_i per hour of additional bus service is considered. Formulate this task as a linear programming problem to minimize the cost. [2]

2: This problem is a little bit tricky!

3.3 Solving Simplex with Python

Problem 1

Research about different python libraries and commercial softwares with which you can solve a linear programming problem. Report a short comparison between these libraries and select the one you think is the best, then solve the following problem with it. [3]

Ask your questions on this assignment from me!

3: Using an external LP solver has bonus points.

$$\begin{array}{ll}
\min & 2x_1 + 7x_2 + 6x_3 + 4x_4 \\
\text{subject to} & 2x_1 + 7x_2 + 6x_3 + 0.5x_4 \leq 65 \\
 & 1.2x_1 + x_2 + x_3 + 1.2x_4 \leq 96 \\
 & 0.5x_1 + 0.7x_2 + 1.2x_3 + 0.4x_4 \leq 80 \\
\text{where} & x_1, x_2, x_3, x_4 \geq 0
\end{array}$$

3.4 Assignment: Sudoku

Ask your questions on this assignment from me!

Sudoku is known as a puzzle of numbers, is a puzzle based on logic. The goal of this game is to add numbers from 1 to 9 in a 9×9 table which includes 9, 3×3 tables called regions. At the start, numbers are pre-prepared in the table. Every region, Row, and column must include numbers from 1 to 9 without repetition. This puzzle was published for the first time in an American magazine in 1979 but continuously published in Japan in 1986 And in 2005 gained its global p popularity. So now that we got to know the division in this game its time to know the rules to solve the puzzle. There is always some Pre assumed numbers in the table for guidance. With the help of these numbers and these three simple rules we can complete the puzzle : First: in every row, there should be numbers from 1 to 9 without any repetition. Second: in every column, there should be numbers from 1 to 9 without any repetition. Third: in every region, there should be numbers from 1 to 9 without any repetition.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Figure 3.1: A sample of a Sudoku table

Problem Definition

Question

Explain if this problem is an "Optimization Problem" or not. Model the program and write a python script that solves a given incomplete Sudoku table.

Input (example)

The first line of the input is N which shows the number of lines we are going to give the solver. For each of the rest N lines you will give three variables numbers key, row, and $column$. For example we are going to give the Sudoku figure as the input:
input:
30
5 1 1
6 2 1
3 1 2
⋮

Output

The result of our code will look like this(or any thing similar):

```
+——-+——-+——-+
| 5 3 4 | 6 7 8 | 9 1 2 |
| 6 7 2 | 1 9 5 | 3 4 8 |
| 1 9 8 | 3 4 2 | 5 6 7 |
+——-+——-+——-+
| 8 5 9 | 7 6 1 | 4 2 3 |
| 4 2 6 | 8 5 3 | 7 9 1 |
| 7 1 3 | 9 2 4 | 8 5 6 |
+——-+——-+——-+
| 9 6 1 | 5 3 7 | 2 8 4 |
| 2 8 7 | 4 1 9 | 6 3 5 |
| 3 4 5 | 2 8 6 | 1 7 9 |
+——-+——-+——-+
```

Submission

Attach a python3 notebook (with extension .ipynb) with your report. Your report should contain a brief explanation of your code and a detailed modeling of the problem. Language of report and it's template is up to you, while it's recommended to use kaobook's report template.[4]

4: Kaobook supports both books and reports. This e-book uses its book template, but you can find report template in 'examples' directory of their github page.

Integer Programming | 4

Ask your questions on this assignment from me!

4.1 Integer Programming Problems

Problem 1: A Business Application; Media Selection

The advertising alternatives for a company include television, radio, and newspaper advertisement. The costs and estimates for audience coverage are given in the table below.

	Television	Newspaper	Radio
Cost per advertisement	\$ 2000	\$ 600	\$ 300
Audience per advertisement	100,000	40,000	18,000

The local newspaper limits the number of weekly advertisements from a single company to ten. Moreover, in order to balance the advertising among the three types of media, no more than half of the total number of advertisements should occur on the radio, and at least 100% should occur on television. The weekly advertising budget is \$ 18,200. How many advertisements should be run in each of the three types of media to maximize the total audience?

Problem 2

Use the simplex method to find the maximum value of Z:

$$\begin{aligned} \text{maximize}\quad & Z = -3x_1 - 2x_2 + 10 \\ \text{subject to}\quad & x_1 - 2x_2 + x_3 = \frac{5}{2} \\ & 2x_1 + x_2 + x_4 = \frac{3}{2} \\ & x_j \geq 0 \quad (j = 1, 2, 3, 4) \\ & x_2 \text{ and } x_3 \text{ are integers} \end{aligned}$$

Problem 3

Use the simplex method to find the minimum value of Z:

$$\begin{aligned} \text{minimize}\quad & Z = 2x_1 + 2x_2 + -2x_3 \\ \text{subject to}\quad & 0.7x_1 + 0.5x_2 + x_3 \geq 1.8 \\ & x_i \in \{0, 1\} \quad (i = 1, 2, 3) \end{aligned}$$

4.2 Assignment: OLED Lifetime

Ask your questions on this assignment from me!

We consider an optimization problem arising in the design of controllers for OLED displays. Our objective is to minimize the amplitude of the electrical current flowing through the diodes which has a direct impact on the lifetime of such a display. The optimization problem consist of finding a decomposition of an image into sub frames with special structural properties that allow the display driver to lower the stress on the diodes. For monochrome images, we present an algorithm that finds an optimal solution of this problem in quadratic time. Since we have to find a good solution in real time, we consider an online version of the problem in which we have to take a decision for one row based on a constant number of rows in the look ahead. In this framework this algorithm has a tight competitive ratio. A generalization of this algorithm computes near optimal solutions of real-world instances in real time. A (passive matrix) OLED display has a matrix structure with n rows and m columns. At any crossover between a row and a column there is a vertical diode which works as a pixel. . Consider the contacts for the rows and columns as switches. For the time the switch of row i and column j is closed, an electrical current flows through the diode of pixel (i, j) and it shines. Since high amplitudes of the electrical current or high peaks of intensity respectively, are the major issues with respect to the lifetime of the diodes we try to trade as much time as possible for it.
You can see paper Algorithms for longer OLED Lifetime for more details(This computer assignment is similar but not exact as the paper).

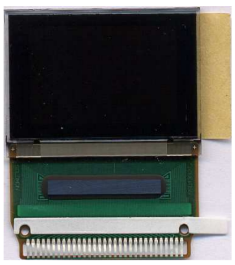


Figure 4.1: Sample of a commercial OLED device with integrated driver chip

Problem Definition

Questions

Assume that we have a 3×3 OLED matrix, which every element of this presents how long(number milliseconds) that diode must be on. We represent the 3×3 matrix with the sum of 3 other 3×3 which the first matrix close the circuit for every row for a certain amount of time, which the total amount of time for this task(first matrix) is the maximum value of the first matrix.
The second matrix is quite similar to the first matrix, instead of rows every column is closed for a certain amount of time, which the total amount of time for this task(second matrix) is the maximum value of the second matrix.
The third matrix, for every element of the matrix a certain row and column are closed therefore the total amount of time for this task(third matrix) is the total sum of all the third matirx elements.

Input(example)

The first three lines are the rows of the matrix input:
input:
7 4 9
8 6 4
4 8 7
where the result will look some thing like this:

$$\begin{bmatrix}8 & 7 & 10\\ 8 & 6 & 5\\ 6 & 8 & 7\end{bmatrix} = \begin{bmatrix}1 & 1 & 1\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} + \begin{bmatrix}6 & 6 & 5\\ 6 & 6 & 5\\ 6 & 6 & 5\end{bmatrix} + \begin{bmatrix}1 & 0 & 4\\ 2 & 0 & 0\\ 0 & 2 & 2\end{bmatrix}$$

The time needed for the first matrix in 1ms, second matrix is 6ms, and for the third matrix is (1 + 2 + 0 + 0 + 0 + 2 + 4 + 0 + 2)ms, therefore it will take 18ms overall.

Output

The result must be 18

Duality 5

Ask your questions on this assignment from me!

5.1 Duality Homework

Problem 1

Use the dual simplex method to find the maximum value of Z:

$$\begin{aligned} \text{maximize} \quad & Z = x_1 - 3x_2 - 2x_4 \\ \text{subject to} \quad & \frac{1}{2}x_1 - \frac{7}{2}x_2 - \frac{3}{2}x_3 + \frac{7}{2}x_4 \leq 0 \\ & \frac{1}{2}x_1 - \frac{3}{2}x_2 - \frac{1}{2}x_3 + \frac{1}{2}x_4 \leq 0 \\ & x_j \geq 0 \quad (j = 1, 2, 3, 4) \end{aligned}$$

Problem 2

Consider the linear programming problem

$$\begin{aligned} \text{maximize} \quad & Z = 5x_1 + 10x_2 \\ \text{subject to} \quad & x_1 + 3x_2 \leq 50 \\ & 4x_1 + 2x_2 \leq 60 \\ & x_1 \leq 5 \\ & x_j \geq 0 \quad (j = 1, 2) \end{aligned}$$

a) State the dual of the preceding LPP.
b) Given that $(5, 15)$ is an optimal solution to this LPP, use the Duality Theorem and the principle of complementary slackness to find optimal solution to the dual.

Problem 3

Use the dual simplex method to find the minimum value of Z:

$$\begin{aligned} \text{minimize} \quad & Z = 2x_1 + 3x_2 + 4x_3 + 5x_4 \\ \text{subject to} \quad & x_1 - x_2 + x_3 - x_4 \geq 10 \\ & x_1 - 2x_2 + 3x_3 - 4x_4 \geq 6 \\ & 3x_1 - 4x_2 + 5x_3 - 6x_4 \geq 15 \\ & x_j \geq 0 \quad (j = 1, 2, 3, 4) \end{aligned}$$

Logical Constraints 6

6.1 Kruskal's algorithm

Kruskal's algorithm finds a minimum spanning forest of an undirected edge-weighted graph. If the graph is connected, it finds a minimum spanning tree. (A minimum spanning tree of a connected graph is a subset of the edges that forms a tree that includes every vertex, where the sum of the weights of all the edges in the tree is minimized. For a disconnected graph, a minimum spanning forest is composed of a minimum spanning tree for each connected component.) It is a greedy algorithm in graph theory as in each step it adds the next lowest-weight edge that will not form a cycle to the minimum spanning forest.

This algorithm first appeared in Proceedings of the American Mathematical Society, pp. 48–50 in 1956, and was written by Joseph Kruskal. Algorithm:

- create a forest F (a set of trees), where each vertex in the graph is a separate tree.
- create a set S containing all the edges in the graph.
- while S is nonempty and F is not yet spanning.
- remove an edge with minimum weight from S.
- if the removed edge connects two different trees then add it to the forest F, combining two trees into a single tree.

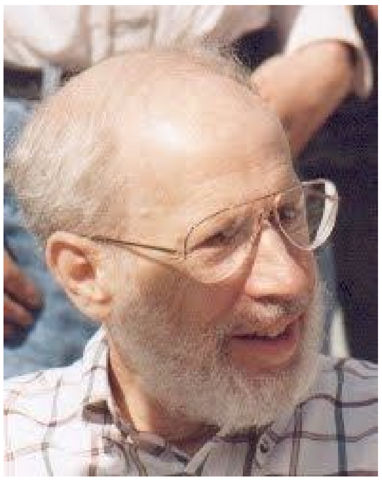


Figure 6.1: Joseph Kruskal (1928-2010)

6.2 Dijkstra's algorithm

Dijkstra's algorithm (or Dijkstra's Shortest Path First algorithm, SPF algorithm) is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later.

The algorithm exists in many variants. Dijkstra's original algorithm found the shortest path between two given nodes, but a more common variant fixes a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest-path tree. Algorithm:

Figure 6.2: Edsger W. Dijkstra (1930-2002)

- Mark all nodes un visited. Create a set of all the un visited nodes called the un visited set.
- Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes. Set the initial node as current.
- For the current node, consider all of its un visited neighbors and calculate their tentative distances through the current node. Compare the newly calculated tentative distance to the current

assigned value and assign the smaller one. For example, if the current node A is marked with a distance of 6, and the edge connecting it with a neighbor B has length 2, then the distance to B through A will be 6 + 2 = 8. If B was previously marked with a distance greater than 8 then change it to 8. Otherwise, the current value will be kept.

- When we are done considering all of the unvisited neighbours of the current node, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.
- If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the un visited set is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining un visited nodes), then stop. The algorithm has finished.
- Otherwise, select the un visited node that is marked with the smallest tentative distance, set it as the new "current node", and go back to step 3.

6.3 Logical Constraints Homework

Ask your questions on this assignment from me!

Problem 1

A company has agreed on supplying p1, p2, p3 tons of a certain product at the end of each of the following three months.
Each month the company can manufacture at most a ton to the cost of c dollars/ton. By working overtime, they can also manufacture another b tons to the cost of d dollars/ton. (a > b and d > c)
The company has the option of storing the product for future use. The storage cost is l dollars per ton per month.
If the company does not supply the agreed quantity in a certain month, they can instead deliver it at a later month, but no later than the third month. The agreed fee for late delivery is f dollars per ton per month.
In the beginning of the first month, the storage is empty and the storage must be empty at the end of the third month. We assume that p1+p2+p3 < 3a+3b.
Formulate the company's planning problem knowing that the goal is to minimize the cost of company during these three months.

Problem 2

Suppose you are planning to move to a new house. You have n items of size a_i $i = 1, ..., n$ that you need to move.
You have rented a truck that has the size Q and you also have m boxes. Box i has size b_i $i = 1, ..., m$.
Formulate an integer programming problem in order to decide whether the move is possible.

Problem 3

1. Formulate the Minimum Spanning Tree for the graph in Figure 6.3
2. Using the obtained formulations, solve the MST problem for the graph in Figure 6.4 using python.
3. Solve the previous part using Kruskal algorithm and compare your results.

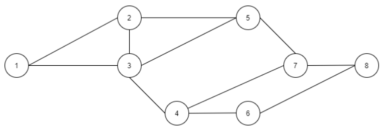


Figure 6.3: Problem 3 graph

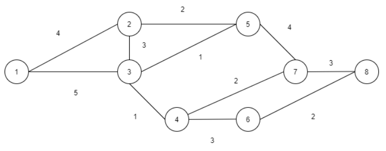


Figure 6.4: Problem 3 graph with edge values

Problem 4

1. Formulate the Shortest Path from the first vertex to the last one for the graph in Figure 6.5.
2. Using the obtained formulations, solve the Shortest Path problem with the values given in Figure 6.6.
3. Solve the previous part using Dijkstra algorithm and compare your results.

Problem 5

Describe the colored spaces of Figures 6.7 and 6.8 using optimization constraints.

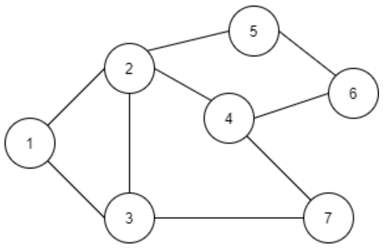


Figure 6.5: Problem 4 graph

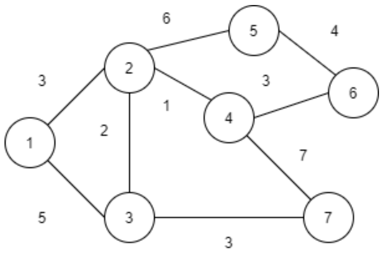


Figure 6.6: Problem 4 graph with edge values

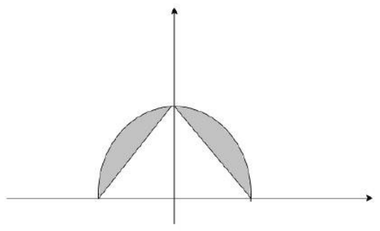


Figure 6.7: First graph of fifth problem, (assume r=1)

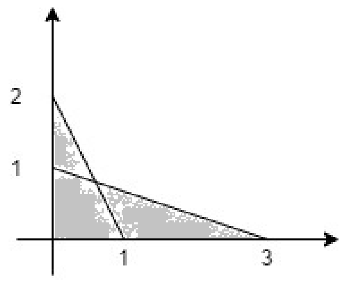


Figure 6.8: Second graph of fifth problem

6.4 Assignment: Graph Coloring

Ask your questions on this assignment from me!

The chromatic number of a graph G is the smallest number of colors needed to color the vertices of G so that no two adjacent vertices share the same color (Skiena 1990, p. 210), i.e., the smallest value of k possible to obtain a k-coloring. A k-coloring of a graph G is a vertex coloring that is an assignment of one of k possible colors to each vertex of G (i.e., a vertex coloring) such that no two adjacent vertices receive the same color.A vertex coloring is an assignment of labels or colors to each vertex of a graph such that no edge connects two identically colored vertices. The most common type of vertex coloring seeks to minimize the number of colors for a given graph. Such a coloring is known as a minimum vertex coloring, and the minimum number of colors which with the vertices of a graph G may be colored is called the chromatic number, denoted $X(G)$. For more details visit here.

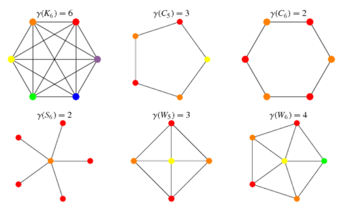


Figure 6.9: Chromatic Number of graphs

Problem

Question

The goal of this assignment is to find the chromatic number of a given graph. Submit your problem modeling in your report, and use one of the common python libraries for linear programming such as PulP.

Modeling of inequality constraints ($\neq$) should be done by students and not the library.

Input

Assume that N is for nodes and E is for edges, where $|N|$ is the number of nodes and $|E|$ is the number of edges in the graph. Where the i^{th} edge is between nodes u_i and v_i. The input must look like this:
$|N|$ $|E|$
u_0 v_0
u_1 v_1
u_2 v_2
$\vdots$
$u_{|N|-1}$ $v_{|N|-1}$
For example:
4 3
0 1
1 2
1 3

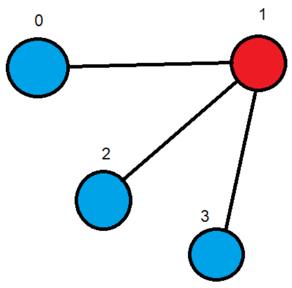


Figure 6.10: The examples result

Output

As you can see in the figure the result of this optimization problem will be 2.

Nonlinear Programming

Nonlinear Programming | 7

One of the most challenging issues of solving an optimization problem is choosing the right length of a step as an initial assumption of the matter. In choosing the initial step length, it is vital to keep in mind that under certain circumstances, if chosen wrong, a step length too small can elongate the time it takes for the problem to be solved, and a step too big could either cause the algorithm to halt in a certain distance from our optimized point or even stop us from ever getting to a convergent answer. There are several different methods of approaching this issue which will be discussed in the span of this assignment. One of the elementary solutions would be to start by choosing a fairly bigger step length and gradually shorten its length by the multiplication of a certain co-efficient or rather follow the changes in our answer and reduce the step size whenever there appears a setback in the descending course of our answers. By doing so and using such algorithms, we can assure the convergence of our points while also getting a satisfactorily quick descend in the course of our solution.

Analytical method.
In this method we will proceed to solve multiple single-variable problems by assigning a cost function to each one and solving them for the optimized step length.

$$\Phi(\alpha) = f(x + \alpha p)$$

It is notable that p is the direction in which the gradient is moving towards a more negative value.

Armijo rule.
For updating the step size and reducing it accordingly through the aforementioned method, the below rules are applied.

```
$\alpha^k = \alpha^0$
$While\ f(x^k + \alpha^k p^k) > f(x^k) + c\alpha^k \nabla f^{k^T} p^k :$
    $\alpha^k = \beta\alpha^k$
```

Constants a,b,c are decided on freely and b and c are to be chosen from a span of 0 to 1.

7.1 Nonlinear Programming Homework

Ask your questions on this assignment from me!

Problem 1 (∗)

Find the stationary points of the functions below and determine their types.

1. $f(x_1, x_2) = 1.5x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + 0.5x_1^4$
2. $f(x_1, x_2) = 3x_1^2 + 2x_2^2 - 3x_1x_2 + 4x_1^3 + x_1^4$

Solving the starred problems (1, 2, 3, 5) is mandatory. Beside that, select 2 other problems with your own choice. Solving extra problems does <u>not</u> have bonus points, so it's expected that everyone submit exactly 6 problems.

Problem 2 (∗)

1. Determine if the given function is convex or not then find its stationary points.

$$f(x) = x^T Ax + Bx + 10$$
$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix},\ A = \begin{pmatrix} 6 & -9 \\ -9 & 21 \end{pmatrix},\ B = \begin{pmatrix} 10 \\ -26 \end{pmatrix}$$

2. Suppose that if one of the the Eigen values of the hessian matrix has a negative value then the function would not be convex and applying optimization methods such as the Newton method would cause the divergence of the model. Present a solution in order to enable the appliance of the newton model in such problems.

Problem 3 (∗)

Use the gradient descent algorithm on the cost function below for two consecutive iterations. Assume an initial point of (1,1) and apply the Analytical method for calculating the step size. Then optimize the cost function in python using a gradient descent with a fixed step size of 0.01 and Newton method. Compare these methods.

$$f(x_1, x_2) = 3x_1^2 + 12x_1 + 8x_2^2 + 8x_2$$

Problem 4

Use the gradient descent algorithm on the cost function below for two consecutive iterations. Assume an initial point of (0,0) and apply the Armijo rule for calculating the step size.

1. $f(x_1, x_2) = 3x_1^2 + 2x_1 + 8x_2 + 4x_2^2$
2. $f(x_1, x_2) = exp(x_1 + 3x_2 - 0.1) + exp(x_1 - 3x_2 - 0.1) + exp(-x_1 - 0.1)$

Problem 5 (∗)

Solve the following optimization problem using Lagrange Method.

$$\begin{aligned}
\min \quad & x_1^2 + 2x_2^2 + x_3^2 - x_1x_2 + x_1x_3 - 3x_2x_3 + 5x_1 + 6x_2 + 2x_3 - 10 \\
s.t. \quad & \sum_{i=1}^{3} x_i = 5 \\
& x_1^2 + x_3^2 \leq 6
\end{aligned}$$

Figure 7.1: Joseph-Louis Lagrange (1736-1813)

Problem 6

Model the following optimization problem using Barrier Function Algorithm.
Plot the initial cost function and the cost function for Barrier Function Algorithm and approximately find the optimal point in $\mathbb{R}^2$.

$$\begin{aligned}
\min \quad & 3x_1 + 4x_2 && (7.1) \\
s.t. \quad & x_2 = -x_1 + 3 && (7.2) \\
& x_2 = -x_1 + 6 && (7.3) \\
& x_2 = x_1 - 3 && (7.4) \\
& x_2 = x_1 + 3 && (7.5) \\
& && (7.6)
\end{aligned}$$

Problem 7

1. Formulate the updating rule of the Projected Gradient algorithm for the unit circle in $\mathbb{R}^2$.
2. Using the Formula from the previous part, repeat the algorithm twice from starting point $x_0 = (1, 0)$ if the cost function is:

$$x_1^2 + x_1x_2 - x_2^2 + 10$$

Problem 8

1. Formulate the Conjugate Gradient Method updating rule for a quadratic problem.

$$\min \quad 1/2x^TQx - b^Tx$$

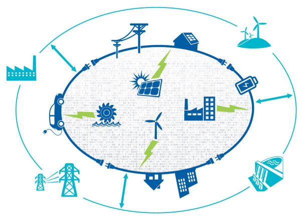
2. Then consider $x_0 = [1, 1]$ and calculate d_k, α_k and then write the updating rule accordingly.

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Final Project

Final Project | 8

The final project is done in groups of three and grouping is done by choosing the same country in ”Country Selection” module in elearn. This project is not that much harder from other ordinary computer assignments and yet it is done in groups; Because data gathering, data representation and ... are also considered as grading factors in the project.



Electricity is the vital energy source in both industrial production manufacturing and people’s daily lives so the importance of electric power generation and transmission is beyond question. As a link between electric power generation and transmission, electric power dispatching decides the effective and healthy operation of the whole power grid. The automation and intelligence of electric power dispatching is a long time issue concerning power grids deserving thorough research.

Collecting the Data Set

Cities and Power Plants

In the first phase, you need to find at least ten cities from your chosen county. You have to track the power lines passing through these ten cities and locate the power plants that provide electricity to them. Assume that these power plants are only at the service of your chosen cities. (Power plants are not necessarily in your chosen country). Power line routes can be found in here). Find out the maximum power that each of the plants can provide. Search in references and find average consumed power of these cities. If these information was not available, make a valid guess using factors like population and etc.

Distance

Now we need to model transmission cost from power plants to the cities. For the sake of simplicity, only consider the active power. Length of power lines can be measured with high precision using openinframap and openstreetmap data or can be estimated with euclidean distance between plants and cities, the former has bonus points. Estimation of resistance can be improved by adding other factors such as line type, number of cables and etc.

Generate the Dataset

Create a csv or json and save your nodes’ location (city and plant’s latitude and longitude). Also save the edge costs in a structure of your own choice. Edge cost should not be necessarily the exact resistance but it should be related to that. Also save your problem constraints.

Define Cost

Model the dispatching problem. Define your optimization variables and cost function.

Solve the Problem

Using the collected data and modeled problem, solve the dispatching problem in your simplified country.

Analysis

Provide appropriate representation for your solution. You can use maps, diagrams, tables and etc. Provide a full report of details of your work. Other than that, provide a recorded presentation and summarize your results.