

Homework 1

June 16, 2020

Homework #1 (4 points)

Due June 24 @ 11:59pm

Submission requirements

Upload a **single PDF file** of your IJulia notebook for this entire assignment. Clearly denote which question each section of your PDF corresponds to.

Problem 1 - Getting Started with Julia/JuMP (1 point)

Model the following problem in JuMP:

$$\begin{aligned} \min \quad & 0.5x_1 - 7x_2 + (1/3)x_3 \\ \text{s.t.} \quad & x_4 = 5 \\ & x_1 + x_2 - 2x_3 + 0.5x_4 \geq 10 \\ & -x_1 + 3x_2 \leq -2 \\ & x_1, x_3 \geq 0, x_2 \leq 0 \end{aligned}$$

Solve this problem using Clp, ECOS, and SCS solvers. Compare the answers found by each solver by answering the following:

- Which solver is most accurate?
- Which is fastest (use the @time macro)? (Note: you should run each solver several times to get an average time.)
 - Can you speculate as to why?
- If there is no clear difference between the solvers, can you think of some factors that might contribute to solver speed differences?

Problem 2 - DnD (Decision Variables and Drawing) (1 point)

Prof. Smith is rolling up a new character for her optimization-based role-playing campaign and she wants to use linear programming to help make some important decisions. She is playing a 4th-level Dragonborn and is trying to decide how many pounds of iron and steel she wants to use to build a sword. Every pound of steel adds a “slowness” factor of 1, whereas every pound of iron adds a

“slowness” factor of -3 (reduces slowness by 3). She would like to combine iron and steel to meet the requirements given below while minimizing the total “slowness” of her character.

The table below summarizes several characteristics of each pound of iron and steel.

Material	Damage (per pound)	Protection (per pound)	Surface area/pound
Iron	-1 points	2 points	1 m^2
Steel	1 points	1 point	2 m^2

Prof. Smith wants to ensure that her total damage is at least -2 points and her total protection is at least 6 points. Furthermore, her character can carry a sword with a maximum surface area of 8 m^2 .

- Formulate a linear program to help Prof. Smith figure out how many pounds of each type of material (iron and steel) she should purchase to minimize her character’s “slowness.” State the math model, then code and solve the model using Julia.
- Code the same model once again, this time separating the parameters from the solution as we did in class (see Top Brass examples). Confirm that you obtain the same solution as in part (a).
- Solve the problem graphically by plotting the feasible set and at least two isoprofit lines for the objective function. Confirm that you obtain the same solution as in the previous parts.

Problem 3 - Standard Form (1 point)

Consider the following LP:

$$\begin{aligned}
 &\min (1/2)x_1 - 7x_2 + (1/3)x_3 \\
 &\text{s.t. } x_4 = 5 \\
 &\quad x_1 + x_2 - 2x_3 + 0.5x_4 \geq 10 \\
 &\quad -x_1 + 3x_2 \leq -2 \\
 &\quad x_1, x_3 \geq 0, x_2 \leq 0
 \end{aligned}$$

- Convert the problem to standard form.
- What are A , b , c , and x ? Be sure to explain how the decision variables of your transformed LP relate to those of the original LP.
- Solve the standard-form LP in Julia and report the objective value and the value of each decision variable in an optimal solution to the original LP.

Problem 4 - This problem is b-ORE-ing (1 point)

A mining company creatively named “MineCo” is trying to determine their daily mining plan. There are forty different locations within the mine where ore can be extracted, and at each of the forty locations the ore has slightly different percentages of 7 “quality attributes.” MineCo has to monitor each of these 7 attributes of the ore they sell. Assume all ore that is extracted is taken

to a processing center where it is perfectly blended before it is sent to the buyers, and quality is measured after the ore is blended. MineCo can extract up to 1000 total tons today.

The ore extracted from each site also has differing total quality, which means each location can be mined to produce different value (in \$) per ton extracted.

The full data on each mining location is provided in “mineco.csv” on Canvas. The data has the following structure, where each row shows a different mining site and each column shows the relevant ore characteristics. You can use the code snippet provided below to read the .csv file and load it into Julia, or you can read/load the data in any way you choose.

	Site	Value/ton (\$)	% Gold	% Carbon	% Iron	...
Minimum	%		3	2	4	...
Maximum	%		11	9	15	...
	1	2349	6	16	16	...
	2	2397	10	12	1	...
	⋮	⋮	⋮	⋮	⋮	⋮

- Formulate a **linear** program that MineCo can use to determine how many tons of ore to extract from each site today in order to maximize their total value. Give a general form (no numbers) of the math model.
- Implement and solve this instance of the model in Julia/JuMP. Display the optimal objective value and the optimal mining plan (in tons extracted from each site).

```
[2]: #You might need to run "Pkg.add(...)" before using these packages
using DataFrames, CSV, NamedArrays

#Load the data file
raw = CSV.read("mineco.csv");

# turn DataFrame into an array
mine_array = convert{Array{<T, N}, DataFrame}(raw);

# the names of the DataFrame (header) are the attributes
attributes = names(raw[3:end]);

# create a list of mining sites from the mine array
sites = mine_array[3:end, 1];

# create a dictionary of the value of each mining site's ore (per ton)
ore_val = Dict{String, Float64}(zip(sites, mine_array[3:end, 2]));

# create a dictionary of the value of min and max % of each attribute
min_percent = Dict{String, Float64}(zip(attributes, mine_array[1, 3:end]));
max_percent = Dict{String, Float64}(zip(attributes, mine_array[2, 3:end]));

# create a NamedArray that specifies the % of each attribute at each site
```

```
using NamedArrays
mine_attribute_matrix = mine_array[3:end,3:end]

# rows are sites, columns are attributes
mine_attribute_array = NamedArray(mine_attribute_matrix, (sites, attributes),  
↳("sites","attributes"))

# note this syntax uses some deprecated commands, so you'll get a warning  
↳message when you run it
# the code should still work (you can ignore the warning)
```