

Project: Sowing plan

STMO

2020-2021

Project by: missing

fill in your name(s) below

Outline

In this project, we will implement a sowing plan for a mixed culture of three types of crops. At our disposal, we have fifteen plots of land, to be planted using a limited supply of seeds. The different lots have varying soil compositions, influencing plant growth. Furthermore, some combinations of the crops grow well together; other combinations show competition. These aspects make for a challenging optimization problem!

$n = 15$

• $n = 15$

A model for plant growth

We have $n = 15$ plots of land. The three types of plants are denoted with x , y , and z , respectively. Indices are used to indicate the amount of seed distributed to a particular plot, e.g., x_i is the amount of seed assigned to the i -th plot. So this optimization problem can be solved in a $15 \times 3 = 45$ -dimensional space.

Small-capped variables indicate the amount of seed assigned to a field. Large-cap variables represent the corresponding yield of that field, i.e., X_i is the yield (in kg) for field i . The yield for each type of seed can be computed using the following equations:

$$\begin{aligned} X_i &= \frac{A_i^x x_i^2}{120 + x_i^2 + 2y_i - 0.8x_i z_i + z_i^2}, \\ Y_i &= \frac{A_i^y y_i^2}{30 + 4x_i + y_i^2 + 7z_i}, \\ Z_i &= \frac{A_i^z z_i^2}{80 + 0.4x_i^2 + 0.2x_i z_i + 0.6y_i + z_i^2}. \end{aligned}$$

Here, A_i^x , A_i^y , A_i^z represent the maximum in yields in field i for the different seed types. It depends on the amount of nitrogen u_i and the water status v_i of the soil. The following equation can be used to compute these coefficients:

$$\begin{bmatrix} A_i^x \\ A_i^y \\ A_i^z \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 1 & 0.3 \\ -0.5 & 4 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \end{bmatrix} + \begin{bmatrix} 100 \\ 300 \\ 210 \end{bmatrix}.$$

```
(3×2 Array{Float64,2}[:, Int64[100, 300, 210]])
 4.0  2.0
 1.0  0.3
-0.5  4.0
```

```
• s, b = [4 2; 1 0.3; -0.5 4], [100, 300, 210] # soil effects
```

The growth model has several interesting facets:

- sowing more seed results in a larger yield, however, the effect saturates, increasing quantities have diminishing returns;
- the plants show competition;
- z positively influences x, while x negatively influences z (z produces nitrogen while x requires much nitrogen);
- y is a good producer but very sensitive to adverse conditions.

Below are the concentrations of nitrogen (u) and water status (v) for every field.

```
u = Float64[36.6776, 36.9967, 83.033, 50.3725, 43.4616, 55.5842, 44.8919, 99.6519
```

```
• u = [36.6776, 36.9967, 83.033, 50.3725, 43.4616, 55.5842, 44.8919, 99.6519, 20.158, 102.325, 96.8896,
33.7957, 26.6129, 38.7194, 60.1461]
```

```
v = Float64[34.5773, 24.3003, 24.3952, 28.462, 37.2912, 38.196, 36.4821, 30.1988,
```

```
• v = [34.5773, 24.3003, 24.3952, 28.462, 37.2912, 38.196, 36.4821, 30.1988, 20.9124, 35.207,
38.0924, 24.438, 28.3169, 20.3022, 24.8884]
```

The total amount of seed for every type is fixed by c_x, c_y, c_z :

```
(250, 175, 325)
```

```
• cx, cy, cz = 250, 175, 325
```

Finally, the yield for each crop can be sold at different prices (EUR/kg) w_x, w_y, w_z :

```
(0.7, 0.85, 0.6)
```

```
• wx, wy, wz = 0.7, 0.85, 0.6
```

So the objective is

$$f(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{i=1}^{15} w_x X_i(x_i, y_i, z_i) + w_y Y_i(x_i, y_i, z_i) + w_z Z_i(x_i, y_i, z_i),$$

but the constraints are

$$\sum_{i=1}^{15} x_i \leq c_x \quad \text{and} \quad x_i \geq 0 \text{ for } i = 1, \dots, 15,$$

$$\sum_{i=1}^{15} y_i \leq c_y \quad \text{and} \quad y_i \geq 0 \text{ for } i = 1, \dots, 15,$$

$$\sum_{i=1}^{15} z_i \leq c_z \quad \text{and} \quad z_i \geq 0 \text{ for } i = 1, \dots, 15.$$

Note: the sum constraints are inequalities because you don't need to use all the seed. However, the optimal solution will likely use all the available seed.

An example

Let us compute the yield for plot 1 when we sow $x_1 = 10, y_1 = 6, z_1 = 8$. First, we compute the maximum yields, then the resulting concrete yields and, finally the value.

```
(10, 6, 8)
```

```
• x1, y1, z1 = 10, 6, 8 # amount of seed per plot
```

```
Float64[315.865, 347.051, 329.97]
```

```
• (Ax1, Ay1, Az1) = S * [u[1], v[1]] .+ b
```

```
X1 = 136.14870689655172
```

```
• X1 = (Ax1 * x1^2) / (120 + x1^2 + 2y1 - 0.8x1 * z1 + z1^2)
```

```
Y1 = 77.12239777777779
```

```
• Y1 = (Ay1 * y1^2) / (30 + 4x1 + y1^2 + 7z1)
```

```
Z1 = 126.00301670644393
```

```
• Z1 = (Az1 * z1^2) / (80 + 0.4x1 + 0.2x1*z1 + 0.6y1 + z1^2)
```

```
revenue_plot1 = 236.45994296256367
```

```
• revenue_plot1 = wx * X1 + wy * Y1 + wz * Z1 # in EUR
```

Assignments

1. Give the formal optimization problem. Also provide the Lagrangian formulation.
2. Is the optimization problem of the optimal sowing plan concave? (since it is a maximization problem, I mean is minimizing the negative total revenue of a plan convex). You don't have to prove this formally, but you can make visual arguments. For a given field, make a the contour plots for (x,y), (x,z) and (y,z), always setting the third variable to 0.
3. Give a good/optimal sowing plan. You may solve this either using custom code or using a Julia package **Optim.jl**, **Convex.jl**, or **JuMP.jl**. Make plots of your solution and discuss it.
4. Given an optimal sowing plan found in the previous assignment, what is the price per kg for every seed you would be willing to buy, as to be break-even? How would you use this additional marginal quantity of seed. (HINT: you can obtain this from the Lagrangian).
5. Provide three alternative solutions, each in which you only use one of the types of seed, e.g. $\max_{\mathbf{x}} f(\mathbf{x}, 0, 0)$. Show and discuss the difference.

Submission

Hand in the solved Jupyter notebook by **14 November 2020**. Send your notebook to **me** both as Pluto notebook file (.jl) and as **PDF or HTML file**. Hand in a CSV file with your best **valid** solution for question 3. In this file, every row is a plot of land and the three columns represent x, y, z, respectively.

```
student_name = missing
```

```
• student_name = missing # fill in your name(s)
```

1. Formal description of the problem

COMPLETE

```
• md"""
• ### 1. Formal description of the problem
•
• COMPLETE
• """
```

2. Concavity

COMPLETE

```
• md"""
• ### 2. Concavity
•
• COMPLETE
• """
```

3. Making a sowing plan

Here, you have the space to solve the problem. Save your final solution in the variable `solution` and save it in a csv file using `save_solution`. This is done automatically when your name is filled in.

```
• Enter cell code...
```

```
• Enter cell code...
```

```
solution = 15×3 Array{Float64,2}:
```

```
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0
  ⋮
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0
 1.0  1.0  1.0
```

```
• solution = fill(1.0, 15, 3) # valid, but can be improved...
```

Check if the solution is valid.

```
true
```

```
• isvalidsolution(solution)
```

Save the solution.

```
• !ismissing(student_name) && save_solution("solution_$(student_name).csv", solution);
```

4. Price of seed

Compute how much you would pay for each additional unit of seed in your solution.

```
• Enter cell code...
```

5. One type of seed

Make three solutions in which you only use one type of seed. Analyse your solutions!

```
• Enter cell code...
```

```
• Enter cell code...
```

```
• Enter cell code...
```

Functions

.....

```
• using CSV, DataFrames
```

Main.workspace3.isvalidsolution

```
• """Check if a solution is valid."""
• function isvalidsolution(solution::Matrix)
•     return size(solution) == (15, 3) && all(solution .≥ 0.0) && all(sum(solution, dims=1) .≤ [cx cy
  cz])
• end
```

Main.workspace3.save_solution

```
• """Save the solution to a file for submission."""
• function save_solution(fname, solution::Matrix)
•     @assert isvalidsolution(solution) "Oh no, your solution is invalid!"
•     CSV.write(fname, DataFrame(solution))
• end
```