BEE 4750/5750 Homework 1

Jason Shao (jls647)

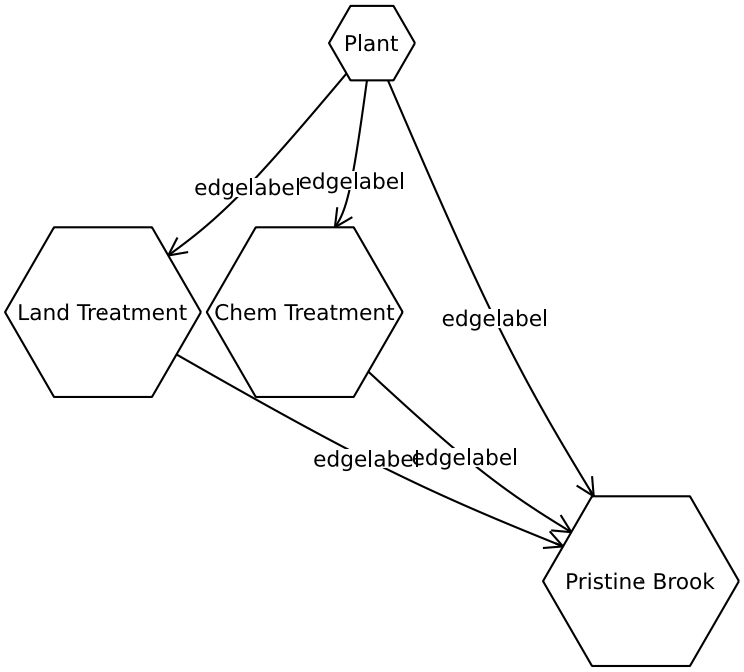
2022-09-08

Problem 1

Problem 1.1

```
julia> using GraphRecipes, Plots

julia> A = [0 1 1 1;
            0 0 0 1;
            0 0 0 1;
            0 0 0 0]
4×4 Matrix{Int64}:
 0  1  1  1
 0  0  0  1
 0  0  0  1
 0  0  0  0

julia> names = ["Plant", "Land Treatment", "Chem Treatment", "Pristine Brook"]
4-element Vector{String}:
 "Plant"
 "Land Treatment"
 "Chem Treatment"
 "Pristine Brook"

julia> shapes=[:hexagon, :rect, :rect, :hexagon]
4-element Vector{Symbol}:
 :hexagon
 :rect
 :rect
 :hexagon

julia> xpos = [0, -1, -0.25, 1]
4-element Vector{Float64}:
  0.0
 -1.0
 -0.25
  1.0

julia> ypos = [1, 0, 0, -1]
4-element Vector{Int64}:
  1
  0
```

```
  0
 -1

julia> edgelabels=Array{String}(undef,4,4)
4×4 Matrix{String}:
 #undef  #undef  #undef  #undef
 #undef  #undef  #undef  #undef
 #undef  #undef  #undef  #undef
 #undef  #undef  #undef  #undef

julia> for i=1:4
       for j=1:4
       edgelabels[i,j]=string("edgelabel")
       end
       end

julia> graphplot(A, names=names, markersize=0.15, edgelabel=edgelabels,
smarkershapes=shapes, markercolor=:white, x=xpos, y=ypos)
```



Problem 1.2

The plant orignally starts off discharging 100 kg/day, so the rate of mass discharge starts off with this value

$$d(X_1, X_2) = 100 - ...$$

next you can subtract the amounts X_1 and X_2 since the orginal concentration is $1\frac{kg}{day}$

$$d(X_1, X_2) = 100 - X_1 - X_2...$$

then you have to take into account the efficiency of the treatments and add back mass

$$d(X_1, X_2) = 100 - X_1 - X_2 + .2X_1 + X_2(.005X_2)$$

This rearranges to

$$d(X_1, X_2) = -.8X_1 + .005{X_2}^2 - X_2 + 100$$

The cost is equal to the sum of both treatments

$$C(X_1, X_2) = \frac{{X_1}^2}{20} + 1.5X_2$$

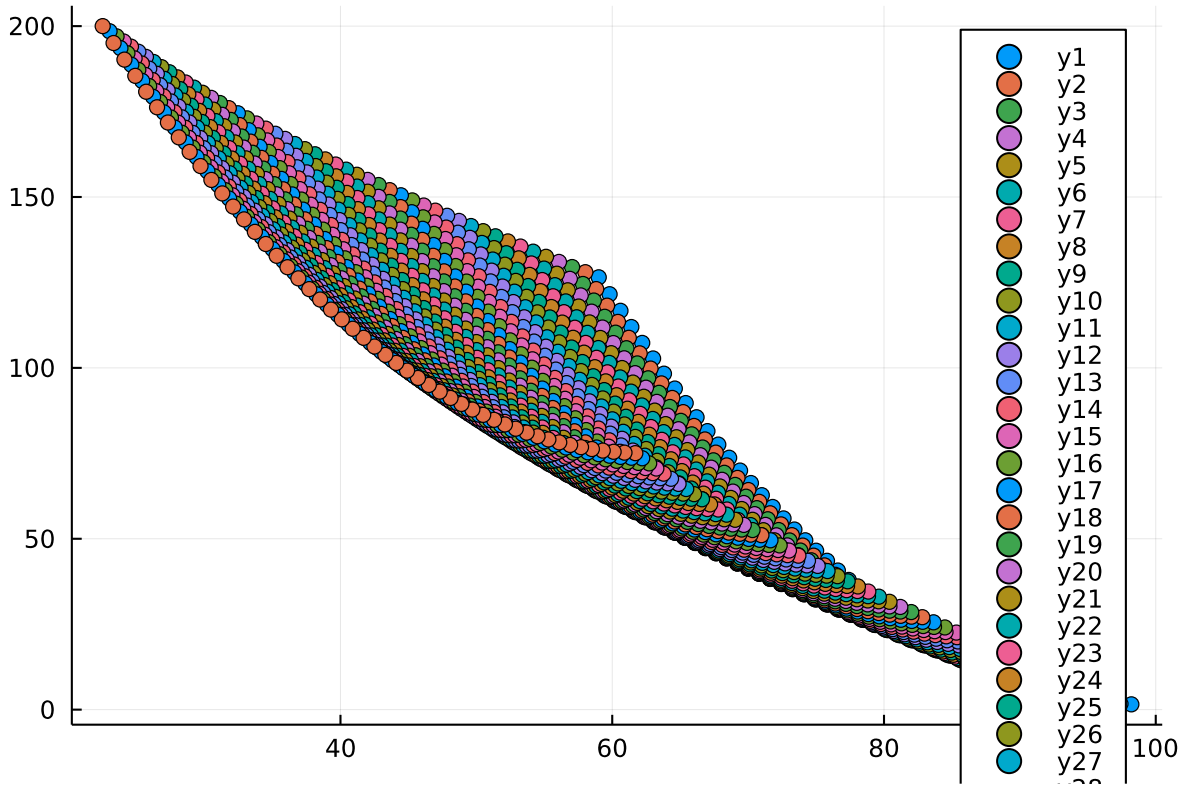
Problem 1.3

```
julia> function ConcCost(X1,X2)
       d=(-.8*X1)+(.005*X2^2)-(X2)+(100)
       C=X1^2/20+1.5*X2
       return d,C
       end
ConcCost (generic function with 1 method)
```

Problem 1.4

```
julia> using Plots, Distributions

julia> d=zeros(50,50)
50×50 Matrix{Float64}:
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 ⋮                    ⋮              ⋱           ⋮
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0

julia> C=zeros(50,50)
50×50 Matrix{Float64}:
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 ⋮                    ⋮              ⋱           ⋮
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0     0.0  0.0  0.0  0.0  0.0  0.0  0.0

julia> for i=1:50
       for j=1:50
       a,b = ConcCost(i,j)
       d[i,j]=a
       C[i,j]=b
       end
       end

julia> scatter(d,C)
```



Problem 1.5

Problem 1.6

Problem 1.7

References