Optimization Methods Final Project

Hongyi Nie

hn327

Dec.2020

Introduction:

We are given an Intensity Modulated Radiation Treatment for Cancer Therapy and its cross-section graph with defined region of each organ; our mission is to design the optimization component of a treatment planning system that best satisfied the constrains. We will divide the problem to multiple parts: 1. Formulation process and assumptions; 2. Tradeoffs & result (Question one: releasing the constrains); 3. Question two: delivered in fractions over multiple days. 4. implementation issues and Conclusion

1. Formulation process, assumptions

1.1 Define each region of organ and our assumption

From the graph, firstly labeled each organ's region with number from 1-400, and in the code index = the number labeled on graph – 1, which is the following:

ctv = [167, 168, 169, 170, 171, 172, 173, 186, 187, 188, 189, 190, 191, 192, 193, 194, 207, 208, 209, 210, 211, 212, 213, 214, 227, 228, 229, 230, 231, 232, 233]

bladder = [87, 88, 89, 90, 91, 92, 93, 106, 107, 108, 109, 110, 111, 112, 113, 114, 126, 127, 128, 129, 130, 131, 132, 133, 134, 147, 148, 149, 150, 151, 152, 153]

rectum = [248, 249, 250, 251, 252, 267, 268, 269, 270, 271, 272, 273, 287, 288, 289, 290, 291, 292, 293, 308, 309, 310, 311, 312, 312]

left = [116, 117, 135, 136, 137, 138, 155, 156, 157, 158, 175, 176, 177, 178, 195, 196, 197, 198, 215, 216, 217, 218, 235, 236, 237, 238, 256, 257]

right = [102, 103, 121, 122, 123, 124, 141, 142, 143, 144, 161, 162, 163, 164, 181, 182, 183, 184, 201, 202, 203, 204, 221, 222, 223, 224, 242, 243]

union =
list(set(ctv).union(set(bladder)).union(set(rectum).union(set(left).union(set(right)))))
unspecific = list(set(list(range(400))).difference(set(union)))

The assumption I made here was the edge of each organ region was as important as the center of organ, so even if some area didn't take a full voxel in the graph, I count it as a full voxel.

1.2 Create variables

Let i represent the index of beamlet, I $\in$ {0, 59}, I $\in$ Z
Let j represent the index of voxel, j $\in$ {0, 399}, j $\in$ Z

x_i = number of dose from each beamlet, x_i $\in$ {0, +∞},
y_j = binary number,

$$y_j = \begin{cases} 1 & if\ total\ beamlet\ of\ j\ th\ voxel\ \geq\ threshhold \\ 0 & if\ total\ beamlet\ of\ j\ th\ voxel\ \leq threshhold \end{cases}$$

```
#Loading data
#set variables
df = pd.read_excel("DoseMatrix.xlsx")
variables = [0 for i in range(60)]
y = [0 for i in range(400)]
dose = df.iloc[:,1:].to_numpy()
solver = pywraplp.Solver.CreateSolver('final project', 'GLOP')
objective=solver.Objective()
objective.SetMinimization()
#add variables
total_dose = []
for i in range(60):
    variables[i] = solver.IntVar(0, solver.infinity(), "x_"+str(i))
    for j in range(400):
        total_dose.append(variables[i]* dose[j][i])
for i in range(400):
    y[i] = solver.IntVar(0, 1, "y_"+str(i))
```

1.3 Set the constrains:

Set constrains specifically for each organ, I wrote all constrains that the solver supposed to satisfied, and in the future experiment I will try testing individual organ region or regions combination by comment other region's part and only adjust useful regions.

For CTV: Given uniform 82.8 Gy, calculate 95% - 105% range = 78.66 to 86.94 Gy, considering cold (low) and hot (high) limitation, set the uniform range to [79.1 , 86.94] Gy. Create an empty voxel list (total_ctv), use for loop to append each beamlet to the total voxel list, then for each index (each voxel) in the voxel list, add constrain to solver that total_ctv[i] $\in$ [79.1,86.94] Gy.

To conclude:

- for j in ctv:

  $\sum_{i=0}^{i=59} x_i, j \leq 86.94,$

  $\sum_{i=0}^{i=59} x_i, j \geq 79.1$

```
#strict constrains that will lead problem infeasible.
#CTV
total_ctv = []
for i in ctv:
    voxel_ctv = []
    for j in range(60):
        voxel_ctv.append(variables[j]* dose[i][j])
    total_ctv.append(sum(voxel_ctv))
for z in range (len(total_ctv)):
    solver.Add(total_ctv[z] <= 86.94)
    solver.Add(total_ctv[z] >= 79.1)
```

For Bladder: except using same method to add upper limit 81, I also added upper limit = 50 to average beamlet = mean(total); and at most 15% condition can be achieve by y_j: assign each y_j to its corresponding voxel and create a list of y. Sum if y list = number of total voxels that receive beamlet that above 65 Gy, adding the constrain that this sum should less or equal to 15%* total voxel that belongs to bladder. Meanwhile, adding the constrain of total_voxel_j $\leq$ 65 + M*y_j, M is a large number, I set M = 100 so that if y_j = 0, j $\leq$ 65; and if y_j = 1, j $\leq$ 81.

To conclude:

- for all j in bladder:

$$\sum_{i=0}^{i=59} x_i, j \leq 81,$$

$$\sum_{i=0}^{i=59} x_i, j \leq 65 + y_i * 100,$$

- $\frac{1}{len(bladder)} \sum_{j=0}^{j=399} \sum_{i=0}^{i=59} x_i, j \leq 50$
- $\sum_{j=0}^{j=399} y_j \leq 10\% * \frac{1}{len(bladder)}$

```
#Bladder
total_bladder = []
y_bladder = []
for i in bladder:
    voxel_bladder = []
    for j in range(60):
        voxel_bladder.append(variables[j]* dose[i][j])
    total_bladder.append(sum(voxel_bladder))
    y_bladder.append(y[i])
solver.Add(sum(y_bladder) <= math.floor(len(bladder)*0.1))
solver.Add(np.mean(total_bladder) <= 50)#50
for z in range (len(total_bladder)):
    solver.Add(total_bladder[z] <= 81) #81
    solver.Add(total_bladder[z] <= 65 + 100 * y_bladder[z]) #65
```

For rectum: same method to add constrains total_voxel[i] ≤ 79.2 and also added upper limit = 40 to average beamlet = mean(total).

To conclude:

- for all j in rectum:

$$\sum_{i=0}^{i=59} x_i, j \leq 79.2,$$

- $\frac{1}{len(rectum)} \sum_{j=0}^{j=399} \sum_{i=0}^{i=59} x_i, j \leq 40$

```
#rectum
total_rectum = []
for i in rectum:
    voxel_rectum = []
    for j in range(60):
        voxel_rectum.append(variables[j]* dose[i][j])
    total_rectum.append(sum(voxel_rectum))
solver.Add(np.mean(total_rectum) <= 40)#40
for z in range (len(total_rectum)):
    solver.Add(total_rectum[z] <= 79.2)#79.2
```

For left head: same method to add constrains total_voxel[i] ≤ 50 and most 15% femur head should receive > 40.0 Gy

To conclude:

- for all j in left:

$$\sum_{i=0}^{i=59} x_i, j \leq 50,$$

$$\sum_{i=0}^{i=59} x_i, j \leq 40 + y_i * 100,$$

- $\sum_{j=0}^{j=399} y_j \leq 15\% * \frac{1}{len(left)}$

```
#left
greater_left = []
total_left = []
y_left = []
for i in left:
    voxel_left = []
    for j in range(60):
        voxel_left.append(variables[j]* dose[i][j])
    y_left.append(y[i])
    total_left.append(sum(voxel_left))
solver.Add(sum(y_left) <=math.floor(len(right)*0.15))
for z in range (len(total_left)):
    solver.Add(total_left[z] <= 50)
    solver.Add(total_left[z] <= 40 + 20 * y_left[z])
```

For right head: same method to add constrains total_voxel[i] ≤ 50 and most 15% femur head should receive > 40.0 Gy

To conclude:

- for all j in right:

$$\sum_{i=0}^{i=59} x_i, j \leq 50,$$

$$\sum_{i=0}^{i=59} x_i, j \leq 40 + y_i * 100,$$

- $\sum_{j=0}^{j=399} y_j \leq 15\% * \frac{1}{len(right)}$

```
#right
greater_right = []
total_right = []
y_right = []
for i in right:
    voxel_right = []
    for j in range(60):
        voxel_right.append(variables[j]* dose[i][j])
    y_right.append(y[i])
    total_right.append(sum(voxel_right))
solver.Add(sum(y_right) <=math.floor(len(right)*0.15))
for z in range (len(total_right)):
    solver.Add(total_right[z] <= 50)
    solver.Add(total_right[z] <= 40 + 20 * y_right[z])
```

For unspecific: same method to add constrains total_voxel[i] $\leq$ 72 a

To conclude:

- for all j in unspecific:

$$\sum_{i=0}^{i=59} x_i, j \leq 72,$$

```
#unspecific
total_unspecific = []
for i in unspecific:
    voxel_unspecific = []
    for j in range(60):
        voxel_unspecific.append(variables[j]* dose[i][j])
    total_unspecific.append(sum(voxel_unspecific))
for z in range (len(total_unspecific)):
    solver.Add(total_unspecific[z] <= 72) #72
```

1.4 Set Objective

The objective function is $\sum_{j=0}^{j=399} \sum_{i=0}^{i=59} x_i, j *$ dose_matrix [i] [j]

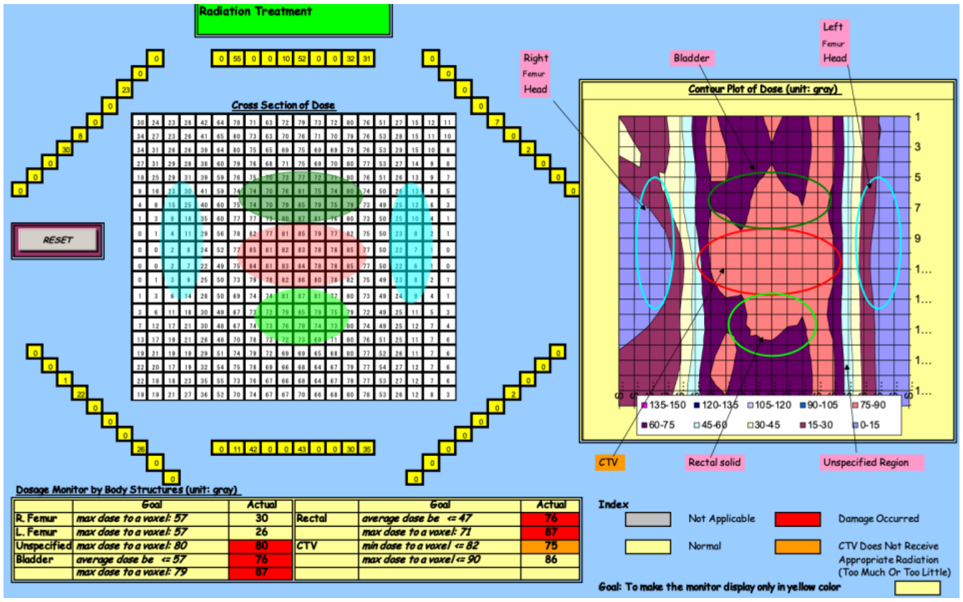
We want to minimize the objective since we want to make sure patient get as less as possible extra radiation than requirement, too much radiation will cause many problems, and we also want to save the unnecessary energy to the machine.

```
#add variables
total_dose = []
for i in range(60):
    variables[i] = solver.IntVar(0, solver.infinity(), "x_"+str(i))
    for j in range(400):
        total_dose.append(variables[i]* dose[j][i])
for i in range(400):
    y[i] = solver.IntVar(0, 1, "y_"+str(i))
#set objective function: to maximize total dose within the constrain
solver.Minimize(sum(total_dose))
```

2. Tradeoffs & result (Question one: releasing the constrains)

- Plan A: Only care about the CTV, Left, right and unspecific region. I started by only these areas because: firstly, CTV is our target region; secondly, left and right head are particularly sensitive since it is related to the immune system, it should not be changing in further experiment; lastly, the unspecific region is very large, and I want to make sure it be considered first.

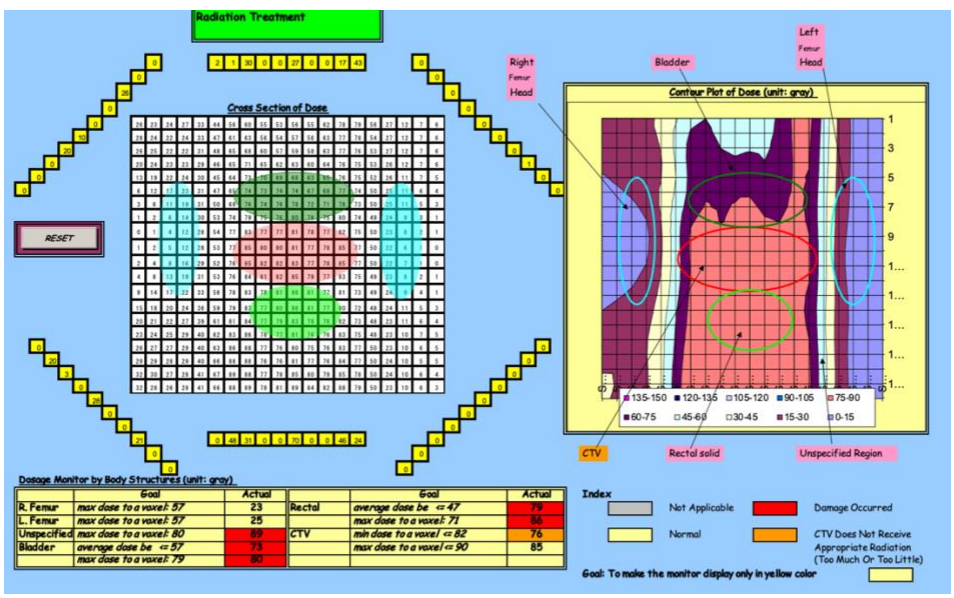
  Beside ignore Bladder and rectum condition first, we increase the max dose of unspecific from 72 to 90, and the other constrains unchanged. Thus, we are having the following graph and chart:



Graph1: Plan A visualization

As we expected for this plan, all the constarins of left, right head and CTV holds while the unspecific region constrains hold after we increase the upper bound, and the constrains bladder and rectum is not satisfied since we ignored it for now.

- Plan B: Base on plan A, adding bladder constrains.
  It is very hard to satisfy all constrains of bladder, so I release the upper limit from 81 to 82, and average from 50 to 75, while ignore the 10% conditions. The graph is shown below:

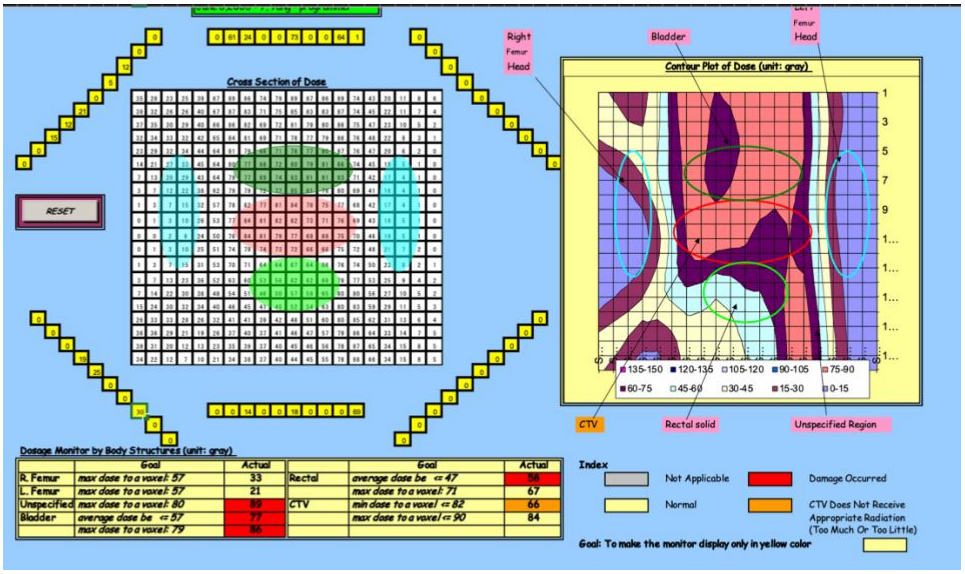


Graph2: Plan B visualization

As we expected for this plan, all the constrains of left, right head and CTV holds while the unspecific region and bladder constrains hold after we increase the upper bound.

- Plan C: Base on plan A, adding rectum constrains.

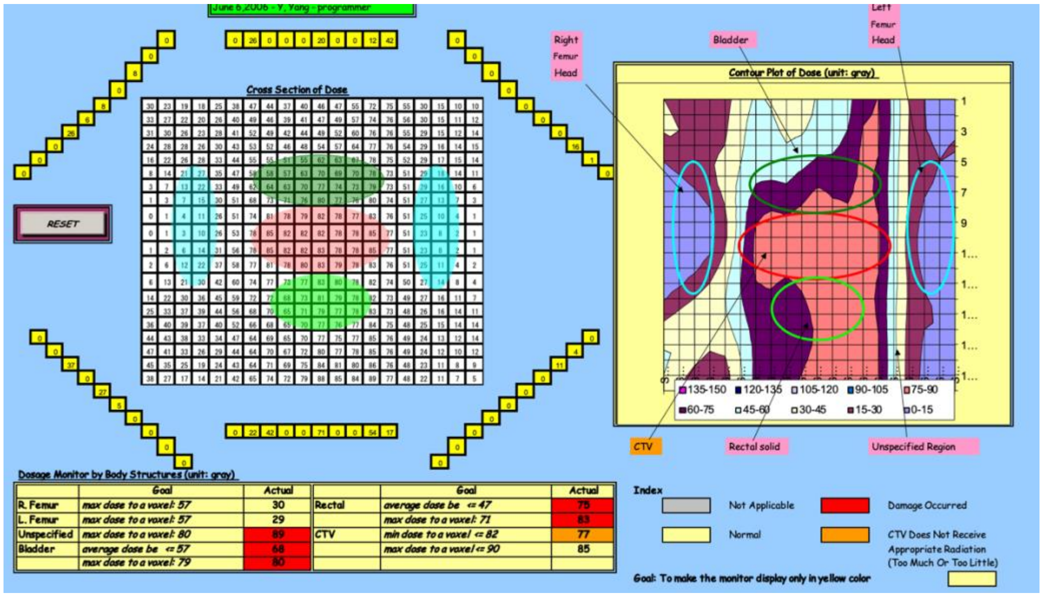
It is very hard to satisfy all constrains of rectum, so I release the upper limit from 79.2 to 82, but the actual max it reaches is 67. and average from 40 to 70, but the actual average it reaches is 58, while ignore the 10% conditions. The graph is shown below:



Graph3: Plan C visualization

As we expected for this plan, all the constrains of CTV, left, right head and holds while the unspecific region and rectum constrains hold after we increase the upper bound.

- Plan D: Putting all regions together.
  Based on the previous experiment, I set the final constrains as the following:
  CTV
  every voxel receives a uniform dose of [79.1, 86.94]
  Bladder
  max dose to a voxel: 82.0 Gy. (original = 81)
  average dose should be <= 70.0 Gy (original = 50)
  Rectum
  max dose to a voxel: 84 Gy (original = 79.2)
  average dose should be <= 78.0 Gy (original = 40)
  Left femur head
  max dose to a voxel: 52.0 Gy (original = 50)
  At most 15% of the left femur head should receive > 40.0 Gy
  Right femur head
  max dose to a voxel: 502.0 Gy (original = 50)
  At most 15% of the right femur head should receive > 40.0 Gy
  Unspecified
  max dose to a voxel: 90.0 Gy



Graph4: Plan D visualization

As we expected for this plan, all the released constrains satisfied.

3. **Question two: delivered in fractions over multiple days.**

There are two ways we can use:

1. ensure patients have same positions by drawing the position graph on the device.
2. Change the step 1.1 identify the region labels, simply change the region labels before use the solver, then the solver will change according to the updated voxel labels.

4. **implementation issues and conclusion:**

First challenge: I create 400 binary variables y_j because when comes to the “at most xxx%” constrains, since I want to avoid the conditional statement (If- Else statement), and instead, using a linear problem constrain that the solver can handle. This was a key component to the problem and it also adding complexity to the implementation process.

Second Challenge: It is very hard to satisfy all conditions, even all conditions for each organ region, so I have to release the constrains to ensure all constrains is satisfied. The average constrains is the hardest one to achieve, so it changed a lot.

To conclude: We use plan A as the initial experiment to ensure key organ’s safety, and Generally, we use plan D to treat patients, but if a patient come with the problem of bladder specifically, use plan B; if patients come with the problem of rectum specifically, use plan C.