Optimization Method Final Project
Intensity Modulated Radiation Treatment for Cancer Therapy

Yutong Wang
yw2364

December 13, 2020

1 Introduction

Intensity Modulated Radiation Treatment, which directs radiation towards the tumor, is used by doctors to treat cancer patients. It would be optimal to have enough dose directed to the tumor area with a minimum dose delivered to the body's healthy parts. In the project, we aim to design the optimization component of a simplified 2-dimensional version of the treatment and find out three or four feasible plans with trade-offs.

In this project, we have a patient with the target area, denoted as CTV(clinical target volume), and the surrounding healthy region, classified into different structures: bladder, rectal solid, right and left femur head, and unspecified area. The patient is mapped into 400 voxels, and he or she will be treated from 6 beam angles, each consisting of 10 beamlets.

2 Formulation

We start with formulating and coding the problem. First, we need to load the data coefficients in and set up the solver. Then, we need to determine the decision variables, the objective functions, and the constraints.

```
dose_coeff = pd.read_csv('DoseMatrix.csv', index_col = 0)
solver = pywraplp.Solver('Final', pywraplp.Solver.CBC_MIXED_INTEGER_PROGRAMMING)
objective=solver.Objective()
objective.SetMaximization()
```

2.1 Decision Variables

To capture the intensity of dose delivered from each beamlet and received by each voxel, we define variables x_j and v_i, where:

$$x_j = \textit{the intensity delivered from beamlet } j$$
$$v_i = \textit{the dose received by voxel } i$$

as shown in the code below

```
# define the #s for voxel and beamlet
beamlet = 60
voxel = 400

# Decision Variables

# x[j] = dose delivered from beamlet j
x =[None for _ in range (beamlet)]
for j in range(beamlet):
    var_name = 'x_{0}'.format(j)
    x[j] = solver.NumVar(0, solver.infinity(), var_name)


# v[i] = dose received by voxel i
v = [None for _ in range (voxel)]
for i in range (voxel):
    var_name = 'v_{0}'.format(i)
    v[i] = solver.NumVar(0, solver.infinity(), var_name)
```

It is obvious that x_j and v_i can be linked using coefficients D_{ij} obtained from the "DoseMatrix.xlsx":

$$v_i = \sum_{j=1}^{60} D_{ij} * x_j \tag{1}$$

```
# v[i] = sum over j (d_unit[i][j] * x[j])
link_const = [None for _ in range(voxel)]
for i in range(voxel):
    link_const[i] = solver.Constraint(0,0,"y[i] - sum over j (d_unit[i][j] * x[j]) = 0")
    for j in range(beamlet):
        d = dose_coeff.iloc[i,j]
        link_const[i].SetCoefficient(x[j],d)
        link_const[i].SetCoefficient(v[i],-1)
```

There are 400 voxels in total, each belonging to a different part of the patient body. We need to classify the voxels into 6 different regions to put different constraints on them. As shown in the code below, the 400 voxels are mapped into CTV, Bladder, Rectum, Left Femur Head(left_f_head), Right Femur Head(right_f_head), and Unspecified Region(u_region).

```
# Classify each voxel as to which structure it belongs

# CTV
CTV = list(range(167, 174)) + list(range(186, 195)) + list(range(206, 215)) + list(range(227, 234))

# Bladder
Bladder = list(range(87, 94)) + list(range(106, 115)) + list(range(126, 135)) + list(range(147, 154))

# Rectum
Rectum = list(range(248, 253)) + list(range(267, 274)) + list(range(287, 294)) + list(range(308, 313))

# Left Femur Head
left_f_head = [102, 103, 122, 123, 124, 222, 223, 224, 242, 243] + list(range(141, 145)) + list(range(161, 165)) + list(range

# Right Femur Head
right_f_head = [116, 117, 136, 137, 138, 236, 237, 238, 257, 256] + list(range(155, 159)) + list(range(175, 179)) + list(rang

# Femur Head in total
f_head = left_f_head + right_f_head

# Unspecified region
u_region = list(range(0,400))
named_region = CTV + Bladder + Rectum + left_f_head + right_f_head
for n in named_region:
    u_region.remove(n)
```

Right Femur Head and Left Femur Head can further map into one region: Femur Head.

For the bladder, left and right femur head regions, we also need binary variables to realize the following constraints:

at most 10% of the bladder should receive a dose >65.0 Gy
at most 15 % of the left(right) femur head should receive >40.0 Gy

We define B_i and F_i, where:

$$B_i = \begin{cases} 1, & \text{if } v_i > 65.0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$F_i = \begin{cases} 1, & \text{if } v_i > 40.0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

```
# B[i] is a binary variable for the bladder area
B = [None for _ in range(0, 400)]
for i in range(0, 400):
    var_name = 'B_{0}'.format(i)
    B[i] = solver.IntVar(0, 1, var_name)

# F[i] is a binary varibale for the femur area
F = [None for _ in range(0, 400)]
for i in range(0, 400):
    var_name = 'F_{0}'.format(i)
    F[i] = solver.IntVar(0, 1, var_name)
```

And we need to link B_i and F_i with v_i:
to start with, we will link B_i and v_i:

$$v_i - 65 \leq M * B_i \qquad \text{where M is a large positive number} \tag{4}$$

$$v_i - 65 >= -M * (1 - B_i) \qquad \text{where M is a large positive number} \tag{5}$$

equation (4) make sure that when v_i is greater than 65, B_i has to be 1. (note: we have constraint B_i to be either 0 or 1.) And equation (5) make sure that when v_i is less than 65, B_i has to be 0. *We will pick $M = 80$ in the code.*

Similarly, we can construct the following two equation to link F_i and v_i :

$$v_i - 40 <= M * F_i \qquad \text{where M is a large positive number} \tag{6}$$

$$v_i - 40 >= -M * (1 - F_i) \qquad \text{where M is a large positive number} \tag{7}$$

We will pick $M = 60$ in the code.

```
# B[i]
for i in Bladder:
    solver.Add(v[i] - 65 <= 80 * B[i])
    solver.Add(v[i] - 65 >= -80 * ( 1- B[i] ))

# F[i]
for i in f_head:
    solver.Add(v[i] - 40 <= 60*F[i])
    solver.Add(v[i] - 40 >= -60 * ( 1- F[i] ))
```

2.2 Objective Function

Our objective in this project is to find a feasible plan that will fulfill all the constraints, which means targeting the CTV with enough dose and keeping the dose received by the healthy parts low. Therefore, we can start with optimizing the sum of all the v_i in CTV:

$$\max_{x_j, v_i} \sum_i v_i \qquad \forall \text{i in CTV} \tag{8}$$

```
# Objective Function
for i in CTV:
    objective.SetCoefficient(v[i],1)
```

We will change the objective function later to discuss possible trade-offs between different plans.

2.3 Constraints

We will formulate and code the constraints region by region.

2.3.1 CTV

The voxels in CTV should receives a approximately uniformed dose of 82.8 Gy, which means the doses received should be within 5% of each other. Moreover, any voxel receiving less than 79.0 Gy are called "cold spot.", which is something we would like to avoid. Note that the voxels' doses are within 5% of each other if they are all between 79.0 and 82.8 Gy. Therefore, we have the constraint:

$$79.0 \leq v_i \leq 82.8 \qquad \forall \text{i in CTV} \tag{9}$$

in the code we have:

```
# CTV: every voxel receives a uniform dose of 82.8

CTV_const = [None for _ in range(len(CTV))]
for i in range(len(CTV)):
    CTV_const[i] = solver.Constraint(79, 82.8, f"79 <= v[{i}] <= 82.8")
    CTV_const[i].SetCoefficient(v[CTV[i]],1)
```

This is obviously just one set of bonds that would work. We may change the bonds when facing trade-offs.

2.3.2 Bladder

There are three constraints on the bladder region:

1. **max dose to a voxel: 81.0 Gy**

$$0 \leq v_i \leq 81.0 \qquad \forall \text{i in Bladder} \tag{10}$$

```
# max dose to a voxel is 81.0
B_max_dose = [None for _ in range(len(Bladder))]
for i in range(len(Bladder)):
    B_max_dose[i] = solver.Constraint(0, 81, f"v[{i}] <= 81")
    B_max_dose[i].SetCoefficient(v[Bladder[i]],1)
```

2. **average dose should be $\leq$ 50.0 Gy**

$$0 \leq \frac{1}{32} * \sum_{\text{i in Bladder}} v_i \leq 50.0 \tag{11}$$

which is equivalent to:

$$0 \leq \sum_{\text{i in Bladder}} v_i \leq 50.0 * 32 = 1600 \tag{12}$$

```
# average dose
B_avg_dose = solver.Constraint(0, 1600, f"(1/32)*v[{i}] <= 50")
for i in Bladder:
    B_avg_dose.SetCoefficient(v[i],1)
```

3. at most 10% of the bladder should receive a dose greater than 65.0 Gy

$$\frac{1}{32} * \sum_{\text{i in Bladder}} B_i \leq 0.1 \tag{13}$$

which is equivalent to:

$$\sum_{\text{i in Bladder}} B_i \leq 0.1 * 32 = 3.2 \tag{14}$$

```
# at most 10% of the bladder should receive a dose > 65
B_most = solver.Constraint(-solver.infinity(), 3.2, "total B[i] <= 3.2")
for i in Bladder:
    B_most.SetCoefficient(B[i],1)
```

2.3.3 Rectum

Similar to what we did with the constraints in the Bladder region, we have the following two constraints:

1. max dose to a voxel: 79.2 Gy

$$0 \leq v_i \leq 79.2 \qquad \forall \text{i in Rectum} \tag{15}$$

```
# max dose to a voxel is 79.2
R_max_dose = [None for _ in range(len(Rectum))]
for i in range(len(Rectum)):
    R_max_dose[i] = solver.Constraint(0, 79.2, f"v[{i}] <= 79.2")
    R_max_dose[i].SetCoefficient(v[Rectum[i]],1)
```

2. average dose should be $\leq$ 40.0 Gy

$$0 \leq \frac{1}{24} * \sum_{\text{i in Rectum}} v_i \leq 40.0 \tag{16}$$

which is equivalent to:

$$0 \leq \sum_{\text{i in Rectum}} v_i \leq 40.0 * 24 = 960 \tag{17}$$

```
# average dose
R_avg_dose = solver.Constraint(0, 960, "(1/24)*v(rectum) <= 40")
for i in Rectum:
    R_avg_dose.SetCoefficient(v[i],1)
```

2.3.4 Unspecified Area

There is only one constraint on voxels in the Unspecified Area:

max dose to a voxel: 72.0 Gy

$$0 \leq v_i \leq 72.0 \qquad \forall \text{i in u\_region} \tag{18}$$

```
# max dose
U_max_dose = [None for _ in range(len(u_region))]
for i in range(len(u_region)):
    U_max_dose[i] = solver.Constraint(0, 72, f"v[{i}] <= 72")
    U_max_dose[i].SetCoefficient(v[u_region[i]],1)
```

2.3.5 Femur Head

Even though the femur head region are separated into left femur head and right femur head, they share the same constraints.

1. **max dose to a voxel: 50.0 Gy**

$$0 \leq v_i \leq 50.0 \qquad \forall \text{i in f\_head} \tag{19}$$

```
# max dose a voxel is 50
F_max_dose = [None for _ in range(len(f_head))]
for i in range(len(f_head)):
    F_max_dose[i] = solver.Constraint(0, 50, f"v[{i}] <= 50")
    F_max_dose[i].SetCoefficient(v[f_head[i]],1)
```

2. **at most 15% of the left(right) femur head should receive a dose >40.0 Gy**
For this constraint, unlike the max_dose constraint, we need to formulate separate constraints for the left and right femur head. Having 15% of femur head receive dose >40.0 Gy is different from having 15% of left femur head receive dose >40.0 Gy and 15% of right femur head receive dose >40.0 Gy. Therefore, we formulate the following two constraints:

At most 15% of the left femur head should receive >40.0 Gy:

For i in left(right)_f_head:

$$\frac{1}{26} * \sum_i F_i \leq 0.15 \tag{20}$$

which is equivalent to:

$$\sum_i F_i \leq 0.15 * 26 = 3.9 \tag{21}$$

```
# at most 15% of the left femur head should receive a dose > 40
LF_most = solver.Constraint(-solver.infinity(), 3.9, "total F[i] <= 3.9")
for i in left_f_head:
    LF_most.SetCoefficient(F[i],1)

# # at most 15% of the Right femur head should receive a dose > 40
RF_most = solver.Constraint(-solver.infinity(), 3.9, "total F[i] <= 3.9")
for i in right_f_head:
    RF_most.SetCoefficient(F[i],1)
```

3 Feasible Plans and Trade-offs

We run the following code with all the constraints above.

```
status = solver.Solve()

if status == solver.OPTIMAL:
    print('Problem solved in %f milliseconds' %solver.wall_time())
elif status == solver.FEASIBLE:
    print('Solver claims feasibility but not optimality')
else:
    print('Solver ran to completion but did not find an optimal solution')
```

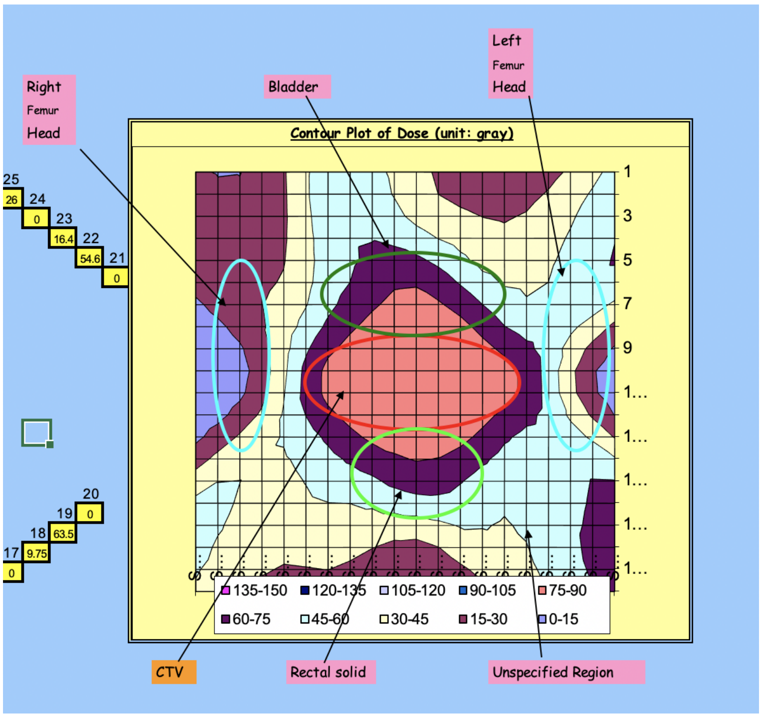
```
Solver ran to completion but did not find an optimal solution
```

As shown above, the solver can not find a feasible solution that satisfies all the constraints. Therefore, we need to make some trade-offs (sacrificing some healthy parts or reducing intensity of doses received by CTV) to get a feasible plan.

3.1 A Feasible Solution

One possible way to get the patient enough dose in CTV and to fulfill as many constraints as many as possible is only keeping the max_dose constraints in bladder, rectum, and unspecified region, and the "At most 15% of the left femur head should receive >40.0 Gy." The beamlet result and output contour is below:

beamlet	1	2	3	4	5	6	7	8	9	10
intensity	0	0	0	0	0	0	0	8.823849052	0	0
beamlet	11	12	13	14	15	16	17	18	19	20
intensity	0	0	35.64142182	0	5.275595718	53.26120561	0	9.748840566	63.47125484	0
beamlet	21	22	23	24	25	26	27	28	29	30
intensity	0	54.60722714	16.39504914	0	25.53563234	13.71269854	0	4.729511791	0	0
beamlet	31	32	33	34	35	36	37	38	39	40
intensity	0	0	4.479735347	10.78431854	0	22.24139432	13.06310523	0	32.57115757	0
beamlet	41	42	43	44	45	46	47	48	49	50
intensity	0	0	0	0	0	0	0	23.84204881	0	0
beamlet	51	52	53	54	55	56	57	58	59	60
intensity	0	7.658590598	49.28582888	0	0	26.98684854	0	0	0	0



We can, in an obvious way, find out how this plan is performing under the constraints that we did not use by running the code below:

```
b_count = 0
b_total = 0
for i in Bladder:
    if B[i].solution_value() == 1:
        b_count += 1
    b_total += v[i].solution_value()

percentrage_b = b_count/32

r_total = 0
for i in Rectum:
    r_total += v[i].solution_value()

left_f_count = 0
for i in left_f_head:
    if F[i].solution_value() == 1:
        left_f_count = left_f_count +1

percentage_l_f = left_f_count/26

right_f_count = 0
for i in right_f_head:
    if F[i].solution_value() == 1:
        right_f_count = right_f_count +1

percentage_r_f = right_f_count/26


print("the average dose on bladder", b_total/32)
print("the % of voxel receives dose > 65 in bladder", percentrage_b)
print("the average dose on rectum", r_total/24)
print("the % of voxel receives dose > 65 in left femur", percentage_l_f)
print("the % of voxel receives dose > 65 in rigth femur", percentage_r_f)
```

And the result we get:

```
the average dose on bladder 65.48713338308451
the % of voxel receives dose > 65 in bladder 0.5625
the average dose on rectum 65.18538968596867
the % of voxel receives dose > 65 in left femur 0.11538461538461539
the % of voxel receives dose > 65 in rigth femur 0.6923076923076923
```

If we can relax the out-of-bond constraints to the numbers shown above, the plan we have would be perfectly feasible for the problem. However, it is not the most ideal one that we can achieve. This plan ignores most of the constraints on the especially sensitive femur head area. We will adjust our plan towards making improvement in the femur head area in the next plan.

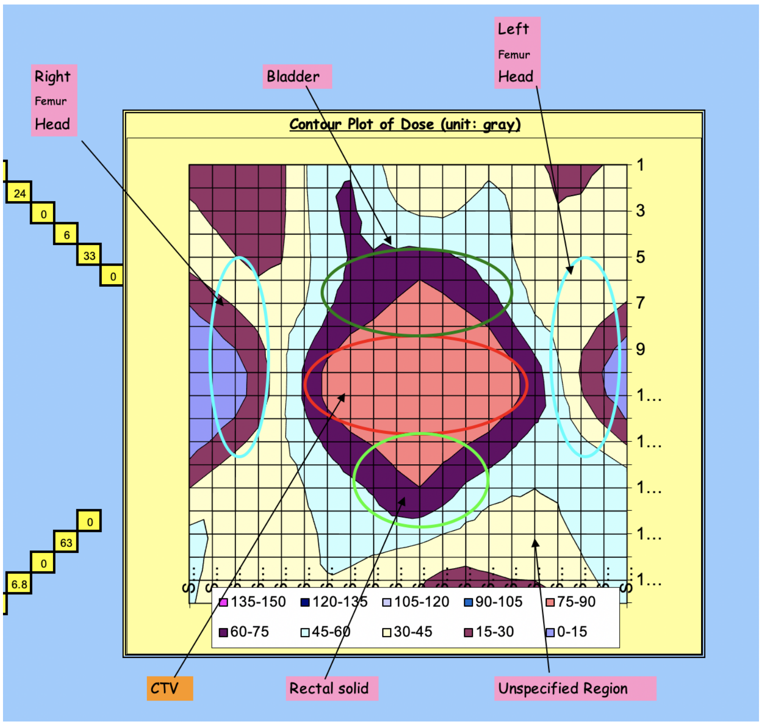
3.2 A Feasible Plan with Improvement on Femur Head

To improve Femur Head's condition under treatment, we will lose the constraints on femur head and keep the max_dose constraints in bladder, rectum, and unspecified region unchanged. Then we change our objective function to minimizing the number of doses over 40.0 Gy in femur head:

```
# Objective Function
for i in f_head:
    objective.SetCoefficient(F[i],-1)
```

The beamlet result and output contour:

beamlet	1	2	3	4	5	6	7	8	9	10
intensity	0	0	0	0	4.024288141	15.23438933	0	0	0	0
beamlet	11	12	13	14	15	16	17	18	19	20
intensity	0	0	16.19345576	0	0	42.61523624	6.751394118	0	63.35957601	0
beamlet	21	22	23	24	25	26	27	28	29	30
intensity	0	32.64124583	6.03686708	0	23.60268489	0.290403038	0	12.92891856	0	0
beamlet	31	32	33	34	35	36	37	38	39	40
intensity	0	28.33229568	2.498542794	0	18.98219447	5.540021281	0	26.53448553	23.43484293	0
beamlet	41	42	43	44	45	46	47	48	49	50
intensity	0	0	0	0	4.143284885	0	0	39.51805606	0	0
beamlet	51	52	53	54	55	56	57	58	59	60
intensity	0	0	44.10588748	0	6.896395161	25.4273971	0	2.990410971	23.98823698	0



We can clearly see improvement in both left and right femur head. This can also be confirmed by numbers:

```
the average dose on bladder 67.46330484889421
the % of voxel receives dose > 65 in bladder 0.53125
the average dose on rectum 67.05191214240503
the % of voxel receives dose > 65 in left femur 0.038461538461538464
the % of voxel receives dose > 65 in rigth femur 0.38461538461538464
```

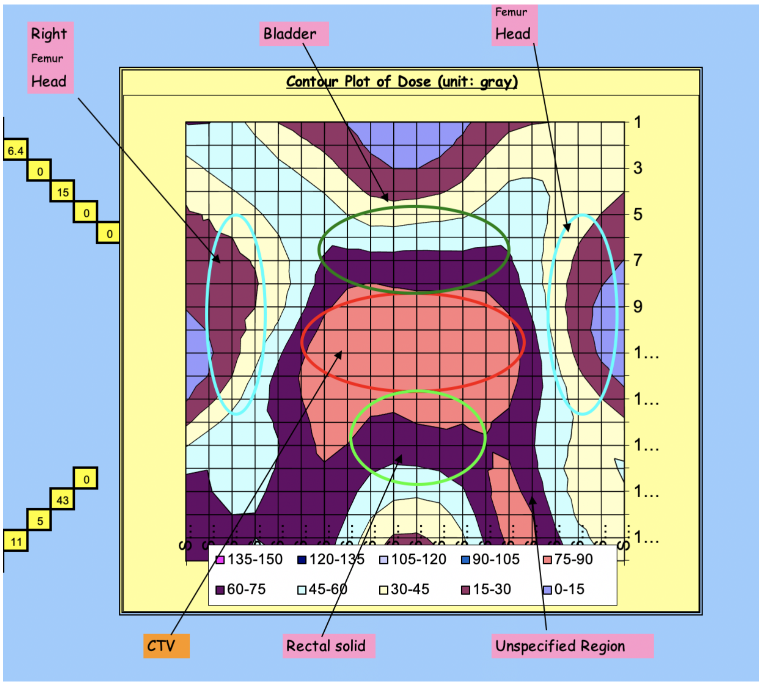
There are slight increases in the average dose in the bladder and the rectum. Meanwhile, we can also see large decreases in the percentage of the femur head with dose over 40.0 Gy.

In the next plan, we aim to reduce intensity received in the bladder.

3.3 A Feasible Plan with Improvement in Bladder

We lose the max_dose constraint on bladder and try to use the "at most 10% of the bladder should receive a dose >65.0 Gy" instead, keeping other constraints unchanged. However, the solver cannot find a feasible solution. That is why we need to relax the constraints. After several experiments, we can find a feasible solution if we relax the percentage constraint to 22% and the max_dose constraint on unspecified region to 82.0 Gy, with other constraints staying the same as before. Result is shown in the beamlet intensity chart and contour:

beamlet	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
intensity	0	29.06844731	0	1.389963266	1.99420301	3.758183159	11.63877567	0	0	45.74559307
beamlet	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
intensity	0	0	16.91729534	34.6502753	0	0	11.40112994	4.644695658	43.40673035	0
beamlet	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
intensity	0	0	14.60249728	0	6.423204262	23.34205879	0	0	0	0
beamlet	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**
intensity	0	0	0	0	0	0	0	0	0	0
beamlet	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**
intensity	0	0	0	12.68609887	44.05403691	0	10.86623798	0	0	30.95393862
beamlet	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**
intensity	0	8.40053761	59.73675475	0	0	49.67684468	0	18.10197885	18.45055591	15.87338952

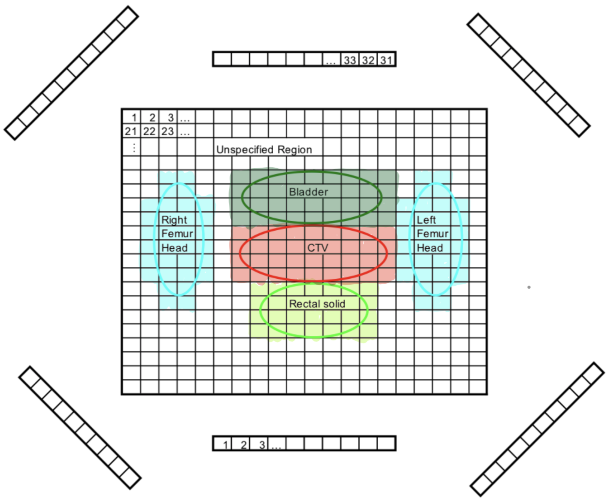


The improvement in bladder is significant, which is confirmed by the numbers:

```
the average dose on bladder 58.60444311405468
the % of voxel receives dose > 65 in bladder 0.21875
the average dose on rectum 71.666920891874
the % of voxel receives dose > 65 in left femur 0.23076923076923078
the % of voxel receives dose > 65 in rigth femur 0.2692307692307692
```

4 Delivering the Treatment

When delivering the treatment, there will be variations in the position of the patient, resulting in variation of the intensity received. One way to fix to that is to include nearby voxels to make sure small variation would not cause CTV to be under-dosed or other healthy parts to be over-dosed. For example, we can count all the newly-colored voxels into different regions to make sure all parts are treated properly. Voxels in the edges of each "new" region may be overdosed or under-dosed due to position variation, but the voxels in the bonds of real regions will still receive the doses intended.



5 Conclusion

In general, it is impossible for us to find a feasible plan that fulfill all the constraints. However, it is possible for us to find treatment plans with relaxing some constraints and make trade-offs.