Optimization Methods (ORIE 5380): Final Project

Due Dec. 13, 2020

Connor Young (cey23)

I. Introduction

I will begin by qualitatively describing the problem and solution formulation from a high level. Details of the decision variables, objective, and constraints in terms of precise descriptions and equations are given further below.

The overall objective of the program is to generate a set of 60 beamlet intensities for Intensity-Modulated Radiation Therapy (IMRT) such that the radiation doses received by various organs and structures within the body conforms as best as possible to the allowable and beneficial level. It is important to note that the problem is infeasible if all the dose recommendations given are enforced as “hard” constraints – there will have to be some relaxation and violation of these provided parameters. Obviously, this is not ideal, but it is unavoidable. Given this knowledge, the problem becomes less about a technical challenge of strict linear programming ability and requires a more nuanced consideration of which parameters to relax and by how much. In determining this, one must think quantitatively about the amount of objective value increase for a given relaxation amount of some constraint, but also qualitatively about the biology and medical considerations of these constraints.

For example, imagine that with one beamlet plan, both the right femur head and bladder doses exceed their average dose requirements by 10 Gy. Now, by relaxing the requirement on either one of them, we can achieve an improvement on the other. Say that allowing the bladder to be 5 Gy higher brings the femur head dose down to only 8 Gy above while allowing the femur head to be 5 Gy higher brings down the bladder dose to only 6 Gy above. If one were merely trying to minimize total excess, one would obviously relax the femur head parameter because doing so gives a greater return. However, we know that the femur heads are extremely important because of their link to the immune system, so with this qualitative consideration we may opt to improve their dose by relaxing the bladder requirements instead.

II. Formulation Discussion

In terms of solution method, I formulated this problem as a mixed-integer linear program, with three different types of decision variables. Before getting into the variable types, though, it is important to understand the objective function since they are closely related.

Note: this was not my original formulation when beginning the problem. I will describe the steps and process of building to this final formulation later but will just give the end result here for sake of clarity.

My objective was to minimize the total amount of “error,” where error is defined as any violation of the given dose parameters. For example, if the maximum dose allowed to a voxel in the unspecified region of the body is 72 Gy and it receives 76 Gy, that is an error of four. It is natural to see how minimizing these differences is synonymous with creating a plan that best conforms to the given parameters. There are eleven different dose parameters total for the six defined regions, and all eleven have the potential to contain error. Therefore, the objective function is the sum of all eleven of these errors for a given beamlet plan, where each error is also multiplied by a corresponding weight. Determining these weights is the crux of the problem, as adjusting them is how one controls what dose parameters are perceived as more important by the solver and which regions should be emphasized. It is also important to weight carefully because not all eleven of the parameters are of the same type or magnitude, they fall into four categories.

1. Maximum dose (ex. Maximum dose to a bladder voxel is 81 Gy)
2. Average dose (ex. Average dose to a rectum voxel is less than or equal to 40 Gy)
3. Percentage of region above a certain threshold dose (ex. At most 15% of the left femur head should receive a dose greater than 40 Gy)
4. Uniformity of dose (only for CTV, every voxel should receive a uniform dose of 82.8 Gy and the range of doses should be kept within 5% of each other)

Now, imagine that one bladder voxel receives a dose of 85 Gy and the average dose to the rectum voxels is 44 Gy. Both of these are an error of four, but in reality the rectum voxel discrepancy is probably a bigger issue because it means that many of the voxels in that region are being irradiated at too high of a level while the bladder error only signifies an issue with a single voxel. This is just a simple example of why the weighting of these different error types must reflect the severity of their respective errors.

Going back to the decision variables, one of the variable types naturally emerges due to the objective function: we need variables to measure the amount of error for each of the eleven parameters. For example, we will need a variable called bladder_overshoot that records the amount by which each voxel in the bladder region exceeds its maximum allowed dose (if at all). Each of these decision variables corresponds 1-to-1 with a given parameter except for the CTV uniformity of dose. Since this is the only region where we worry about having too low of a dose instead of just too high, we will need two different variables to measure both undershoot and overshoot for each voxel. These twelve* variables will all be summed up and multiplied by their respective weights to form the objective function.

*There are actually more than 12 because, for example, bladder_overshoot is a list of variables the length of the number of voxels the bladder contains, but we will just call it one for simplicity.

The second type of decision variable is fairly obvious – continuous ones that output values for what we are actually trying to solve for. Specifically, there is a list of length 60 where each variable corresponds to a selected beamlet intensity, and there is a list of length 400 where each variable corresponds to a voxel's radiation dose. The relationship between beamlet intensity and radiation provided to each voxel is given for all beamlets and voxels in the provided DoseMatrix.xlsx spreadsheet. The mathematical relationship contained in this spreadsheet is enforced in the linear program via a linking constraint which I will touch on later.

The third type of decision variable is less obvious, but also arises out of necessity in enforcing certain parameters, specifically the "percentage of region above a certain threshold dose" type. There are three lists of binary variables that correspond to the voxels in a given region, specifically the bladder and both femur heads. For each of these, the binary variable will have a value of 1 if the dose to that voxel exceeds the threshold allowed for that region and a value of 0 otherwise. Since linear programming tools do not function properly with 'if' statements, using binary variables in this way is a proxy to doing so in order to classify the voxels as above or below the threshold and enforce the percentage requirements.

In terms of constraints, there was again more than one "type": linking constraints and constraints to actually enforce parameters. (Again, constraint formulas are provided further down, this is just a general description). The linking constraints simply exist to control some of our decision variables in the way that we know they should behave. For example, as mentioned earlier, there was one to maintain the relationship between voxel dose and beamlet intensity. This just iterates through each voxel and says that its dose must be equal to the dose computed from the chosen beamlet intensities and the Dose Matrix. Similarly, the other linking constraints control the relationship between voxel dose and our binary threshold variables. As mentioned, we cannot use 'if' statements, so the equations in these constraints are how we force the binary variable to take on the correct 1/0 value given the selected dose. There are six of these, two each for the bladder, left femur head, and right femur head. Two are needed for each because one will force the binary to be 1 if the dose is too high and the other will force the binary to be 0 if the dose is low enough.

The constraints to enforce the parameters are very straightforward, each one corresponds 1-to-1 with the eleven types of error discussed earlier. For example, for maximum dose allowed to the bladder, there is a constraint created for each voxel in the bladder that says that the dose to it must be less than 81 Gy. The only tricky part relates to the fact that the problem is infeasible with all constraints applied, so we do not want to strictly enforce that limit. What the constraint actually says in my formulation is that for each voxel in the bladder, the dose applied must be less than 81 minus the "overshoot" for that voxel. This is linking in its own way for the error variables since it causes their values to be mathematically in-line with the voxel doses and dose requirements. As a result, it is possible to exceed 81 Gy (as it should be) but we can measure how much we exceed by and use that information to adjust the weighting of different errors in the objective function.

Below this you will see the formalized version of what I have described thus far with details provided in terms of variable description and constraint equations. One important thing to note here is the discrepancy between how I indexed the voxels and how Python list indexing works. The voxels are named from 1-400, while Python indexing starts at 0. Thus, for the voxel_doses list that stores the dose to each voxel, voxel_doses[i] gives the dose for voxel i+1 (ex. voxel_doses[0] gives dose for voxel 1). I note this below as well but want to be clear that when I use the word "corresponding" anywhere I am referring to the same one voxel even if their index is technically not the same.

III. Detailed Formulation

Decision Variables

Continuous Output Variables:

beamlet_intensities – list of length 60 where the i^{th} element corresponds to the intensity of beamlet i+1 (ex. beamlet_intensities[0] = beamlet 1 intensity)

voxel_doses – list of length 400 where the i^{th} element corresponds to the dose received by voxel i+1 (ex. voxel_doses[0] = dose delivered to voxel 1)

- **beamlet_intensities** and **voxel_doses** linked via constraint to conform to the mathematical relationship given in DoseMatrix.xlsx

Binary Threshold Variables:

bladder_above_threshold – list of length equal to the number of voxels in the bladder region, each element equals 1 if the corresponding voxel receives a dose greater than 65 and equals 0 otherwise (ex. bladder_above_threshold[0] = 1 if voxel_dose[87] > 65 since voxel 88 is the first in the bladder region)

left_femur_above_threshold – list of length equal to the number of voxels in the left femur region, each element equals 1 if the corresponding voxel receives a dose greater than 65 and equals 0 otherwise

right_femur_above_threshold – list of length equal to the number of voxels in the bladder region, each element equals 1 if the corresponding voxel receives a dose greater than 65 and equals 0 otherwise

- All of these are linked to **voxel_doses** by constraints to ensure that the binary values accurately reflect the dose values

Error Variables:

unspecified_overshoot – list of length equal to the number of voxels in the unspecified region, the i-th element of the list corresponds to the amount by which the i^{th} voxel dose in that region exceeds its maximum allowed dose (ex. unspecified_overshoot[0] = 2 if voxel_dose[0] = 74 since voxel 1 is the first in the unspecified region and the max allowed dose is 72)

rectum_overshoot – list of length equal to the number of voxels in the rectum region, the i^{th} element of the list corresponds to the amount by which the i^{th} voxel dose in that region exceeds its maximum allowed dose

rectum_average_overshoot – single variable equal to the amount by which the average dose for the rectum region exceeds its recommended value (ex. rectum_average_overshoot = 13 if average dose to rectum voxels is 53 since the recommended average dose is maximum 40)

bladder_overshoot – list of length equal to the number of voxels in the bladder region, the i^{th} element of the list corresponds to the amount by which the i^{th} voxel dose in that region exceeds its maximum allowed dose

bladder_average_overshoot – single variable equal to the amount by which the average dose for the bladder region exceeds its recommended value

bladder_percentage_overshoot – single variable equal to the amount by which the percentage of voxels in the bladder region over a given dose threshold exceeds the recommended percentage (ex. bladder_percentage_overshoot = 13 if 23% of the voxels in the bladder receive a dose above 65 because only a maximum of 10% are supposed to receive a dose that large)

left_femur_overshoot – list of length equal to the number of voxels in the left femur region, the i^{th} element of the list corresponds to the amount by which the i^{th} voxel dose in that region exceeds its maximum allowed dose

left_femur_percentage_overshoot – single variable equal to the amount by which the percentage of voxels in the left femur region over a given dose threshold exceeds the recommended percentage

right_femur_overshoot – list of length equal to the number of voxels in the right femur region, the i-th element of the list corresponds to the amount by which the i^{th} voxel dose in that region exceeds its maximum allowed dose

right_femur_percentage_overshoot – single variable equal to the amount by which the percentage of voxels in the right femur region over a given dose threshold exceeds the recommended percentage

CTV_overshoot – list of length equal to the number of voxels in the CTV region, the i^{th} element of the list corresponds to the amount by which the i^{th} voxel dose in that region exceeds the target dose + 2.5% (ex. CTV_overshoot[0] = 1 if voxel_dose[167] = 85.87 since voxel 168 is the first in the CTV region and the target dose + 2.5% is 82.8 + 2.07 = 84.87)

CTV_undershoot – list of length equal to the number of voxels in the CTV region, the i^{th} element of the list corresponds to the amount by which the target dose – 2.5% exceeds the i^{th} voxel dose in that region (ex. CTV_undershoot[0] = 1 if voxel_dose[167] = 79.73 since voxel 168 is the first in the CTV region and the target dose – 2.5% is 82.8 – 2.07 = 80.73)

- All of these are associated with the required parameter constraints given below, causing their values to be an accurate representation of deviation from the parameters

Objective

The objective is to minimize the plan's deviation from the desired constraints. This is an easy objective function to formulate given how we have created decision variables that each capture deviation amount from each individual constraint, we simply sum them all and tell the solver to minimize. Where the trickiness and qualitative aspect comes into play is in figuring out how to weight the different deviations in the objective function to achieve a plan that makes the most sense. For example, without weighting, our program will put just as much emphasis on minimizing our unspecified_overshoot as it will on minimizing CTV_overshoot and CTV_undershoot. This does not make a lot of sense given that it is probably more important to precisely dose the CTV region as opposed to the unspecified one.

min. Objective = (weight 1)(unspecified_overshoot) + (weight 2)(rectum_overshoot)

+ (weight 3)(rectum_average_overshoot) + (weight 4)(bladder_overshoot)

+ (weight 5)(bladder_average_overshoot) + (weight 6)(bladder_percentage_overshoot)

+ (weight 7)(left_femur_overshoot) + (weight 8)(left_femur_percentage_overshoot)

+ (weight 9)(right_femur_overshoot) + (weight 10)(right_femur_percentage_overshoot)

+ (weight 11)(CTV_overshoot) + (weight 12)(CTV_undershoot)

<u>Constraints</u>

Linking:

intensity_to_dose_linking – links beamlet_intensities and voxel_doses to conform to the mathematical relationship in DoseMatrix.xlsx

For each voxel (all 400):

$$\text{voxel\_dose} = \sum_{\text{beamlets}} (\text{beamlet\_intensity} \times \text{fraction of beamlet to voxel})$$

bladder_threshold0 – forces each element of bladder_above_threshold to be 0 if the corresponding dose from voxel_dose is less than or equal to 65

For each voxel in bladder region:

$$\text{voxel\_dose} - 65(\text{bladder\_above\_threshold}) \geq 0$$

bladder_threshold1 – forces each element of bladder_above_threshold to be 1 if the corresponding dose from voxel_dose is greater than 65

For each voxel in bladder region:

$$\text{voxel\_dose} - 100(\text{bladder\_above\_threshold}) \leq 65$$

left_femur_threshold0 – forces each element of left_femur_above_threshold to be 0 if the corresponding dose from voxel_dose is less than or equal to 40

For each voxel in left femur region:

$$\text{voxel\_dose} - 40(\text{left\_femur\_above\_threshold}) \geq 0$$

left_femur_threshold1 – forces each element of left_femur_above_threshold to be 1 if the corresponding dose from voxel_dose is greater than 40

For each voxel in left femur region:

$$\text{voxel\_dose} - 100(\text{left\_femur\_above\_threshold}) \leq 40$$

right_femur_threshold0 – forces each element of right_femur_above_threshold to be 0 if the corresponding dose from voxel_dose is less than or equal to 40

For each voxel in right femur region:

$$\text{voxel\_dose} - 40(\text{right\_femur\_above\_threshold}) \geq 0$$

right_femur_threshold1 – forces each element of right_femur_above_threshold to be 1 if the corresponding dose from voxel_dose is greater than 40

For each voxel in right femur region:

$$\text{voxel\_dose} - 100(\text{right\_femur\_above\_threshold}) \leq 40$$

Required Parameters:

unspecified_dose_max – stores the amount by which an unspecified voxel dose exceeds the maximum allowed (72 Gy) in the corresponding spot in unspecified_overshoot

For each voxel in unspecified region:

$$\text{voxel\_dose} - \text{unspecified\_overshoot} \leq 72$$

rectum_dose_max – stores the amount by which a rectum voxel dose exceeds the maximum allowed (79.2 Gy) in the corresponding spot in rectum_overshoot

For each voxel in rectum region:

$$\text{voxel\_dose} - \text{rectum\_overshoot} \leq 79.2$$

rectum_dose_average – stores the amount by which the average rectum voxel dose exceeds the maximum average dose allowed (40 Gy) in rectum_average_overshoot

$$\sum_{\text{rectum voxels}} \frac{\text{voxel\_dose}}{\text{\# of rectum voxels}} - \text{rectum\_average\_overshoot} \leq 40$$

bladder_dose_max – stores the amount by which a bladder voxel dose exceeds the maximum allowed (81 Gy) in the corresponding spot in bladder_overshoot

For each voxel in bladder region:

$$\text{voxel\_dose} - \text{bladder\_overshoot} \leq 81$$

bladder_dose_average – stores the amount by which the average bladder voxel dose exceeds the maximum average dose allowed (50 Gy) in bladder_average_overshoot

$$\sum_{\text{bladder voxels}} \frac{\text{voxel\_dose}}{\text{\# of bladder voxels}} - \text{bladder\_average\_overshoot} \leq 50$$

bladder_percentage_lower – stores the amount by which the percentage of bladder voxel doses above the threshold (65 Gy) exceeds the allowable percentage (10%) in bladder_percentage_overshoot

$$\sum_{\text{bladder voxels}} \text{bladder\_above\_threshold} - (\text{bladder\_percentage\_overshoot})\left(\frac{\text{\# of bladder voxels}}{100}\right) \leq 0.1(\text{\# of bladder voxels})$$

left_femur_dose_max – stores the amount by which a left femur voxel dose exceeds the maximum allowed (50 Gy) in the corresponding spot in left_femur_overshoot

For each voxel in left femur region:

$$\text{voxel\_dose} - \text{left\_femur\_overshoot} \leq 50$$

left_femur_percentage_lower – stores the amount by which the percentage of left femur voxel doses above the threshold (40 Gy) exceeds the allowable percentage (15%) in left_femur_percentage_overshoot

$$\sum_{\text{left femur voxels}} \text{left\_femur\_above\_threshold} - (\text{left\_femur\_percentage\_overshoot})\left(\frac{\text{\# of left femur voxels}}{100}\right) \leq 0.15(\text{\# of left femur voxels})$$

right_femur_dose_max – stores the amount by which a right femur voxel dose exceeds the maximum allowed (50 Gy) in the corresponding spot in right_femur_overshoot

For each voxel in right femur region:

$$\text{voxel\_dose} - \text{right\_femur\_overshoot} \leq 50$$

right_femur_percentage_lower – stores the amount by which the percentage of right femur voxel doses above the threshold (40 Gy) exceeds the allowable percentage (15%) in right_femur_percentage_overshoot

$$\sum_{\text{right femur voxels}} \text{right\_femur\_above\_threshold} - (\text{right\_femur\_percentage\_overshoot})\left(\frac{\#\text{ of right femur voxels}}{100}\right) \leq 0.15(\#\text{ of right femur voxels})$$

CTV_dose_target – stores the amount by which a CTV voxel dose differs from the acceptable range of target dose (82.8 +/- 2.5% → (80.73, 84.87)) in the corresponding spot in either CTV_overshoot or CTV_undershoot. If actual dose is larger than 84.87 the difference is stored in CTV_overshoot, if actual dose is less than 80.73 the difference is stored in CTV_undershoot

For each voxel in CTV region:

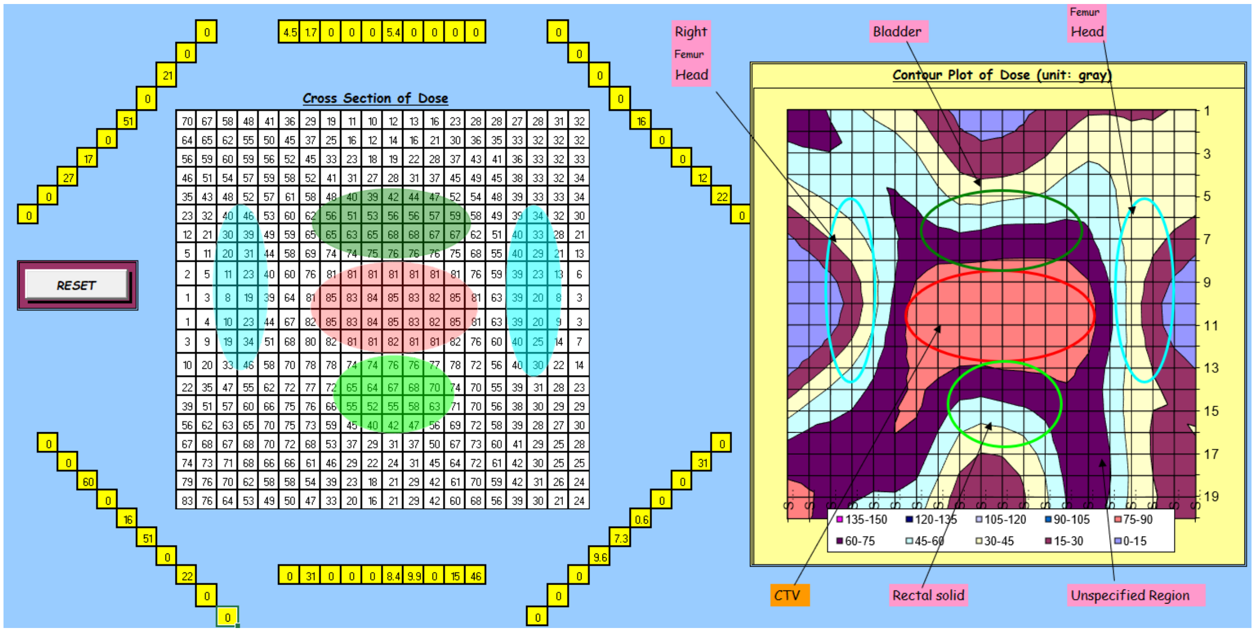
$$80.73 \leq \text{voxel\_dose} - \text{CTV\_overshoot} + \text{CTV\_undershoot} \leq 84.87$$

IV. Proposed Treatment Plans

I generated three different treatment plans to be selected from based on the severity of the cancer and the amount of other health risk the doctor and patient are willing to take on in eliminating the CTV. The first plan is an aggressive one for a case where the CTV must be eliminated, even at the expense of high damage to other healthy structures. The second is a more balanced plan that may make small sacrifices in terms of efficacy of CTV elimination for the sake of preserving healthy structures, but still prioritizes the CTV. In the first two plans, there is also higher priority placed on meeting the femur head parameters as opposed to the bladder, rectum, and unspecified region. The third is a relaxed plan for cases where the doctor would ideally like to eliminate the cancer, but it is not severe enough to warrant any significant damage to other healthy structures. Here, the parameters are all relaxed to about the same magnitude, without any significant priority placed on meeting the requirements of one structure vs. another.

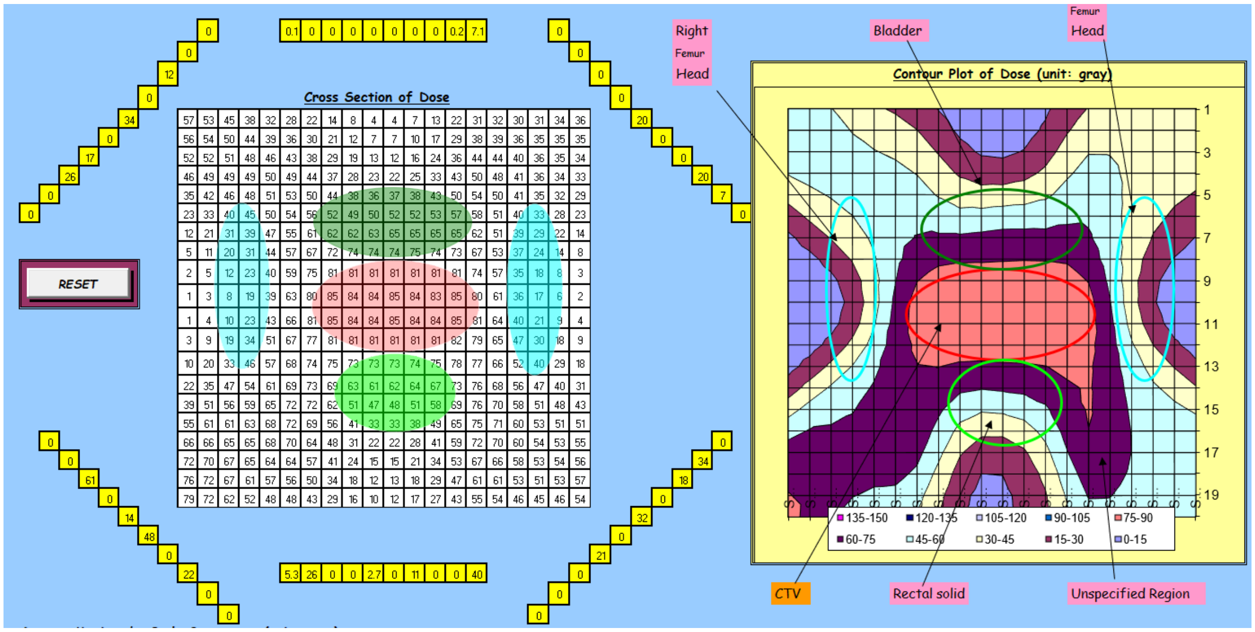
The three plans are shown below. Each one contains a visualization of the applied radiation and a table of the 60 beamlet intensity values to provide it. At the bottom there is a comparison of the treatment plans in terms of their performance on the provided parameters.

Plan 1: Aggressive



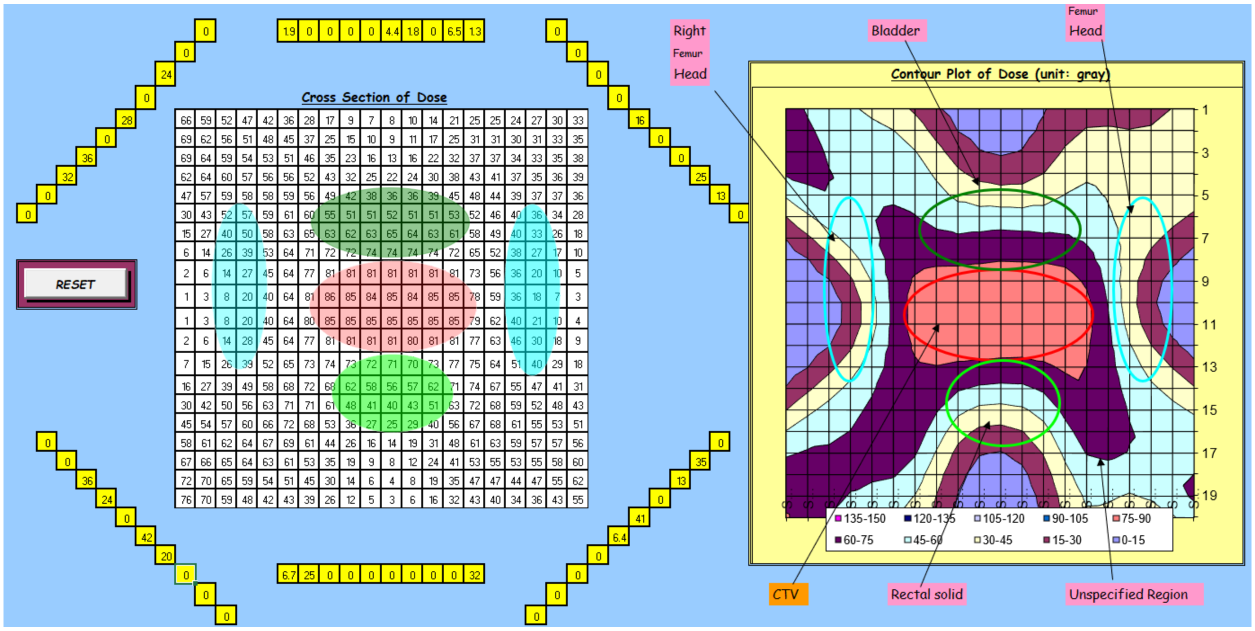
Beamlet	Intensity	Beamlet	Intensity	Beamlet	Intensity
1	0	21	0	41	0
2	30.7426	22	22.3379	42	0
3	0	23	11.5175	43	21.2564
4	0	24	0	44	0
5	0	25	0	45	51.3490
6	8.4173	26	16.3254	46	0
7	9.8981	27	0	47	16.7579
8	0	28	0	48	27.1642
9	14.9549	29	0	49	0
10	45.5434	30	0	50	0
11	0	31	0	51	0
12	0	32	0	52	0
13	0	33	0	53	60.4962
14	9.6304	34	0	54	0
15	7.2657	35	5.3731	55	15.8552
16	0.6284	36	0	56	50.8540
17	0	37	0	57	0
18	0	38	0	58	22.1047
19	30.8545	39	1.7281	59	0
20	0	40	4.5350	60	0

Plan 2: Balanced



Beamlet	Intensity	Beamlet	Intensity	Beamlet	Intensity
1	5.3286	21	0	41	0
2	26.2805	22	7.4537	42	0
3	0	23	19.5011	43	11.7591
4	0	24	0	44	0
5	2.7288	25	0	45	33.6450
6	0	26	19.9430	46	0
7	10.9139	27	0	47	17.3375
8	0	28	0	48	26.0253
9	0	29	0	49	0
10	39.8201	30	0	50	0
11	0	31	7.0991	51	0
12	0	32	0.2043	52	0
13	0	33	0	53	60.6502
14	21.2176	34	0	54	0
15	0	35	0	55	13.5240
16	32.0719	36	0	56	47.5783
17	0	37	0	57	0
18	18.4007	38	0	58	22.0803
19	34.0834	39	0	59	0
20	0	40	0.1167	60	0

Plan 3: Relaxed



Beamlet	Intensity	Beamlet	Intensity	Beamlet	Intensity
1	6.6500	21	0	41	0
2	25.1211	22	13.0542	42	0
3	0	23	25.1499	43	23.9862
4	0	24	0	44	0
5	0	25	0	45	27.6276
6	0	26	16.3968	46	0
7	0	27	0	47	36.3530
8	0	28	0	48	31.6058
9	0	29	0	49	0
10	31.6706	30	0	50	0
11	0	31	1.2657	51	0
12	0	32	6.4923	52	0
13	0	33	0	53	36.4216
14	0	34	1.7907	54	24.0801
15	6.3530	35	4.3738	55	0
16	40.5048	36	0	56	42.2200
17	0	37	0	57	19.9126
18	12.8673	38	0	58	0
19	34.9961	39	0	59	0
20	0	40	1.9378	60	0

Comparison of Treatment Plans:

Parameter	Aggressive Plan	Balanced Plan	Relaxed Plan
# Doses Outside CTV Acceptable Range	0	4	9
# Doses Above Bladder Maximum	0	0	0
Bladder Avg. Dose (recommended 50 Gy)	60.44 Gy	57.80 Gy	57.44 Gy
% of Bladder Above 65 Gy (rec. 10%)	34.28%	21.88%	21.88%
# Doses Above Rectum Maximum	0	0	0
Rectum Avg. Dose (recommended 40 Gy)	62.71 Gy	58.57 Gy	54.02 Gy
# Doses Above Unspecified Max.	16	12	7
# Doses Above Left Femur Maximum	6	7	6
% of Left Femur Above 40 Gy (15% rec.)	20%	26.67%	26.67%
# Doses Above Right Femur Maximum	1	1	5
% of Right Femur Above 40 Gy (15% rec.)	21.43%	21.43%	25.0%

V. Further Discussion and Application to Real World

If I had the ability to relax one or two of the parameters, I would choose to loosen the requirements on the bladder and rectum. Specifically, I would relax the average dose requirement for each because it is so much more restrictive than the maximum dose allowed for each or even the percentage below a threshold for the bladder.

Looking at the problem spatially with the orientation of beamlets and the various regions of the body, an obvious conundrum emerges. The CTV, the region that is supposed to receive by far the highest dose per voxel, is located in the middle of the patient and is surrounded by healthy regions that desire to have much lower doses applied to them. In order to get radiation to the CTV, it inherently must provide it to the other regions as well, except for potentially a couple beamlets coming in from the corners. The femur heads have the most restrictive requirements and are very important due to their impact on the patient’s immune system, so we do not want to irradiate them heavily from the diagonally oriented beamlet sets. Therefore, it would be excellent if we could provide most of the radiation from the top and bottom beamlet sets, which transmit the majority of their strength through the bladder and/or rectal solid and not the femur heads.

Unfortunately, the bladder and rectum parameters prevent us from using too large of intensities here, and as mentioned above the average dose requirements are the main culprit. The bladder and rectum have maximum allowed doses of 81 Gy and 79.2 Gy respectively, but they also demand average doses no larger than 50 Gy and 40 Gy. This means that for both of these regions, we could easily have all the doses well under the maximum allowed per voxel and still be far above the average allowable dose. We can easily see this effect manifesting itself in the proposed treatment plans, where the average bladder and rectum doses must be well above their allowable level in order to get anywhere close to the CTV and femur head requirements, which are more important. If I had the ability to relax a third parameter I would opt to increase the threshold that 10% of doses can be above for the bladder. This follows the same reasoning as the average dose requirements, except that the threshold is currently 65 Gy which is much more reasonable than the 50 Gy required for the average dose.

In order to make the formulation and proposed treatment plans most applicable to the real-world situation, we must consider the circumstances under which the plan is administered in a medical environment. These radiation treatments are not administered in one sitting, but occur in fractions over multiple days, which inherently means that there is some variability in the patient positioning from one day to the next. There are a few different ways that we can modify our formulation or further work that we can do to make our treatment proposal more robust in the face of this variability.

The first way involves classification of borderline voxels which contain a part of multiple regions. Some of these are a majority one region vs. the other while others are fairly evenly split. (Voxels 115 and 149 are examples of each of these, respectively). In both cases, the percentage of the voxel filled by each region will vary day-to-day due to the patient repositioning. In light of this, the simplest way to ensure that we meet the requirements of whichever region is dominant in that voxel on a certain day is to just count that voxel as part of both regions so that it must conform to both sets of parameters. This is something I have actually already implemented in my formulation, several voxels are “double-counted” for multiple regions.

In a very similar vein, we can double-count voxels that do not actually contain multiple regions in the given diagram, but very reasonably could given the positioning variability. For example, voxels 83 and 84, located just above the right femur head, do not currently contain any of the femur head region but very likely would if the patient repositions slightly. Since voxels like this only contain one region in the diagram, I only counted them once in my implementation so that there would be fewer parameters to meet. However, this is something that would be good to alter when trying to apply the formulation in a realistic setting.

Finally, there is a way that we can do further work to better estimate the positioning variance and use this to inform our voxel classification. We could run a simulation of the region positions over many iterations to reflect where they are according to long-run average. Then, instead of classifying each voxel based on a single-position diagram, we can classify them based on a generalization of that diagram. This will allow us to determine which voxels need to be double-counted and which do not. For example, if our simulation tells us that voxel 83 contains the right femur head 50% of the time, we should consider it both a part of the right femur head and the unspecified region. However, if the simulation tells us the voxel only contains the femur head 2% of the time, it is probably fine to only classify that voxel as the unspecified region. The benefit of not double counting a voxel is the fact that it has less parameters to meet which means that the solver can prioritize the parameters that do need to be met and improve on them. We can also just use the simulation to compute an "average" plan by solving for the best plan over all the different positioning cases and then taking the mean of each beamlet intensity over all trials. This is effectively the same thing but without manually examining and classifying each voxel. In other words, you can use a simulation to either find average voxel position and develop a plan for it, or you can directly find the average plan, which is a higher-level proxy for voxel position.

VI. Assumptions, Decisions, and Implementation Process

In beginning this project and developing my formulation, I started by following the suggestions given on the project description document. The first thing I did was classify each voxel according to which structure it belonged to. Already, I was facing my first question which I know many other people had as well: how should I classify voxels that are on the border of two regions? I originally thought about assigning the voxel to the region that held the majority of space in it but rethought this for two reasons. Primarily, just because a voxel may only be 20% filled by a femur head does not mean that that 20% is not still bound to the parameters given for a femur head. There may be a small amount there to get damaged, but that damage is still an issue and when added up across all partial-femur voxels it becomes quite significant. Secondly, I had already begun thinking about the patient positioning variability question, and as I discussed above, a voxel that may only be 5% femur head in the given diagram may very likely be 50% femur head when the patient comes in the next day. Due to these considerations, I opted to count any voxel that contained any portion of multiple regions as belonging to both those regions, no matter which region is dominant or how much so.

The next thing was to get the dose calculation working using the given spreadsheet and beamlet intensities, which was very straightforward, although I ended up having to change how I did this which I will discuss further in a moment.

After experimenting with the Excel dose visualization tool, I began trying to formally define my optimization problem starting with the most rudimentary formulation I could think of. The constraints were obvious, there should be one for each parameter that we are trying to enforce. I knew that ideally the objective would be to minimize the sum of the deviation from each parameter, but this seemed too complicated to implement at first in trying to get a working linear program. Therefore, my initial objective was to minimize the total dose applied to the patient, regardless of region. For decision variables, I made a list of length 60, where each entry would store the intensity for a given beamlet. I thought that I would simply choose 60 beamlet intensities and use the dose matrix to convert them to voxel doses, which I would then minimize the sum of.

Unfortunately, I ran into a problem with my objective function when I tried to assign coefficients to the computed doses. Since the computed doses were a function of the beamlet intensity decision variables and not decision variables in themselves, I was unable to assign coefficients to them. This made me realize that I needed to have decision variables for the dose to each voxel as well if I wanted to use the dose amount as a variable in my formulation. The logical next question was then how I would maintain the relationship between beam intensity and voxel dose so that the values chosen for the beamlet variable set aligned mathematically with the values chosen for the dose variables. I realized I needed to create a linking constraint and incorporate the dose matrix spreadsheet through that instead of the way that I initially set up the dose calculation.

This seemed to work well, and I started generating reasonable beamlet intensities and doses that mathematically aligned, which I verified with the spreadsheet. I also added all the parameters as constraints except for the percentage/threshold-based ones. This was working but was inconvenient because the problem is infeasible. Due to this fact, the only way to get a solution for the formulation was to manually adjust the parameter values, such as letting the left femur head maximum dose be 70 Gy instead of 50 Gy. It was impossible to tell how much I needed to adjust the values by to make the problem feasible, and if it was infeasible I did not receive any results to inform how close I was to feasibility.

After going to office hours, I learned how to restructure my formulation so that the problem was always feasible and I could just see how far off I was from the original required parameters instead of having to change their values to meet them. This involved adding the error/deviance measuring variables which function as I explained in the Formulation Discussion section. The added benefit of creating these variables is that they tie in perfectly with the objective function that I knew I wanted to eventually move to from my initial dose minimization one. Once I had the variables created, I modified my objective function to be the sum of them multiplied by their respective weights, which I initially left as 1 for all of them.

At this point, I had my final formulation almost complete, but I still had not implemented the percentage/threshold-based constraints. In order to enforce these, I had to have some way to count the number of dose values that exceed the allowed threshold so that I could compute the percentage. However, after trying an ‘if’ statement to do so, I quickly remembered that these conditionals are not allowed in a linear program. After going to office hours again, I learned that I could use lists of binary variables that would equal 1 if the corresponding voxel dose was above the threshold and 0 if it was below. I created the lists for the three percentage/threshold-based regions: bladder, left femur head, and right femur head. The last thing I had to do was link each of these to their corresponding voxel dose lists via constraints for each, as I explained in the Formulation Discussion section.

From here, with my formulation fully complete, all that was left to do was adjusting the weighting of the different error variables in order to find the best plans. Due to their sensitivity and importance, I put a lot of weight on minimizing error related to the CTV and femur heads at the expense of the bladder and rectum. Of course, this caused the average doses to the bladder and rectum to be far above the allowed level, which is very low, as I discussed when talking about which parameters I would choose to relax. One reason for this is because of the difference in magnitude of the various error types as I mentioned in my formulation discussion. If the average dose is 10 too high with a weight of 1, this has the same amount of impact on the objective function as a single voxel exceeding its maximum allowed dose by 10. In order to make the impact of average error on the objective reflect its impact in reality, I weighted the average value overshoots more heavily than the maximum value overshoots. Beyond these general weighting rules, I varied the weights slightly differently for each of the proposed plans that were laid out previously.

One thing I did to make the process of tuning weights more efficient and to get more quantitative feedback was to write a "metrics" cell in my code. This simply took the plan generated from my linear program, evaluated how well it matched the parameter requirements, and displayed the error on each parameter to me in an easily readable format so that I could see which parameters were generating the most error so that I could weight them more heavily to reduce it. An example output from this cell is shown below. This is what was used to create the treatment comparison table in the Proposed Treatment Plans section.

```
# of Unspecified Doses >72 (0):  12
Unspecified Max Error:  38.091843746673476


# of Rectum Doses >79.2 (0):  0
Average Rectum Dose (max 40):  58.56564872099147
Rectum Max Error:  0.0
Rectum Average Error:  18.565648720991454


# of Bladder Doses >81 (0):  0
Average Bladder Dose (max 50):  57.80364622556213
% of Bladder Doses >65 (max 10%):  21.875
Bladder Max Error:  0.0
Bladder Average Error:  7.803646225562133
Bladder Percentage Error:  11.875


# of Left Femur Doses >50 (0):  7
% of Left Femur Doses >40 (max 15%):  26.666666666666668
Left Femur Max Error:  53.03623993005005
Left Femur Percentage Error:  11.666666666666666


# of Right Femur Doses >50 (0):  1
% of Right Femur Doses >40 (max 15%):  21.428571428571427
Right Femur Max Error:  0.5647424930959559
Right Femur Percentage Error:  6.428571428571428


List of CTV Doses with >2.5% Deviation from 82.8:  [80.12384844839951, 85.24744065206639, 79.62783924476732, 84.91828755178321]
# of CTV Doses with >2.5% Deviation from 82.8 (0):  4
CTV Deviation Error:  2.134040510682766
```

I made one final change in my formulation while I was in the process of determining weights for my proposed treatment plans. I originally had my CTV error variables measuring deviation from 82.8 Gy, the ideal voxel dose for the CTV. However, while looking at my weights, I realized this presented an issue. The most important item for CTV dosing is uniformity of dose, not conformity to the exact ideal dose. In other words, I was computing an error when a voxel received a dose of 82.9 Gy, even though this is well within the +/- 2.5% range mentioned in the problem description document. With how heavily I was weighting CTV error because of its importance, it was punishing doses that were very marginally off from ideal at the expense of violating parameters of other regions by far more. I changed that constraint so that CTV_overshoot and CTV_undershoot would only record an error if the dose was less than 80.73 or larger than 84.87, a 5% range centered on the ideal dose of 82.8 Gy. After doing this, the solution the solver came up with was much kinder to the other healthy structures since it was not trying to eliminate any deviation from 82.8, only the necessary deviation.