VMAT TBI autoplanning stand-alone executable quick start guide

E. A. Simiele

(Revised October 30, 2020)

1 Stand-alone executable

1.1 Main program GUI

- Launch the program 'VMATTBI_optLoopMT_vXX.exe', which should be on your desktop
    - Current release version located at:
      Z:\Users\ESimiele\Research\VMAT_TBI\release_versions\executable
- The UI when the stand-alone executable is first launched should look very familiar to the UI of the binary plug-in script
- To begin, open a patient by entering the MRN in the 'Patient MRN:' box and hitting the 'Open Patient' button
    - **NOTE: THE SCRIPT IS LOOKING FOR A PLAN NAMED '_VMAT TBI' CONTAINED IN A COURSE 'VMAT TBI'! IF THE PATIENT DOES NOT HAVE EITHER OF THESE, THE PROGRAM <u>WILL NOT</u> FIND THE PLAN!**
- **You cannot modify the plan prescription in this program!**
- You can choose to run a coverage check before the optimization loop (single optimization, no intermediate dose with all OAR constraint priorities set to 0)
- You can adjust the maximum number of optimizations that will be performed
- You can adjust any of the optimization constraints, which were read from the '_VMAT TBI' plan in the 'VMAT TBI' course

- You can also check whether you want the program to perform an additional optimization to reduce the plan hotspots
  - **This optimization is performed <u>AFTER</u> the main optimization loop**
- You can select whether the script should save each optimized plan. This option copies the resulting plan from each optimization iteration and saves it to the VMAT TBI course with the name ”opt itr <count>” where <count> is the current optimization iteration
  - This is useful if you want to see how the optimization is progressing with each plan. In addition, it also gives you multiple plans to choose from to do final adjustments (I've noticed cases where the plan from the second optimization had better quality than the plan from the third optimization)
- If there were no optimization constraints stored in the '_VMAT TBI' plan, close this script and use the binary plug-in script to set the optimization constraints
- Once you are satisfied with the constraints, hit the 'Confirm Constraints and Begin Optimization' button to start the optimization loop
- **NOTE: YOU DO NOT NEED TO DO ANYTHING SPECIAL TO INCLUDE FLASH IN THE OPTIMIZATION IN THIS SCRIPT. THE PROGRAM AUTOMATICALLY DETECTS IF YOU INTEND TO USE FLASH IN THE OPTIMIZATION LOOP (a warning message will be shown at the beginning of the progress window that indicates you intend to use flash)**

1.2 Optimization loop GUI

- A new UI will pop up and inform the user of the progress of the optimization loop and how the program is adjusting the optimization constraints after each iteration
- The user has the option the abort the optimization loop using the 'Abort' button in the bottom left of the GUI. This will tell the program the user wants to stop the optimization loop
  - **NOTE: THE PROGRAM WILL (LIKELY) NOT CLOSE IMMEDIATELY WHEN YOU HIT THE 'ABORT' BUTTON! THIS IS BECAUSE THE PROGRAM IS TIED UP IN PERFORMING A COMPUTATIONALLY EXPENSIVE TASK IN ECLIPSE (E.G., DOSE CALCULATION) AND THERE IS NO SAFE WAY OF TERMINATING THAT PROCESS**
  - **<u>BE PATIENT!</u> THE PROGRAM IS ROUTINELY CHECKING IF THE USER WANTS TO STOP THE OPTIMIZATION LOOP AND WILL TERMINATE WHEN AN ACCEPTABLE STOPPING POINT HAS BEEN REACHED**

- NOTE: DO NOT try to close the window while the optimization loop is running!
- The GUI window can be closed only when the program status is 'Aborted', 'Failed', or 'Finished'
- At any time, the user can write the displayed text output to a text file at a location of their choosing by hitting the 'Write results to file'. Note, this will overwrite an existing file

1.3 Program flow

- Once the 'Confirm Constraints and Begin Optimization' button is hit, the program reads in the optimization constraint list and stores these parameters
- Preliminary checks are then performed including if the user origin was set, are the isocenter positions set correctly (i.e., all z-positions are rounded-off, x/y positions = 0.0, etc.), etc.
- The program also checks if a couch structure is present. If it is not, it will ask if you want to continue anyway or stop and insert the couch. If the couch and spinning manny couches are inserted, the script checks to see if these structures have contours in the first and last slices of the CT image. If so, the script will ask if you want to remove the contours on those image planes
  - We've found through testing that if these structures are present on the first and last slices of the CT image, the program will give a pop-up message box after each dose calculation saying that beams are transported through a structure where CT image ends. This is just a warning message from Eclipse, but it halts the progression of the program until that warning message is closed (which could be disappointing if you start running the program at night and don't check back until morning)
- If all the preliminary checks pass, a warning message will pop up informing the user that the plan is ready for optimization
- If you elected to run a coverage check, the script will then perform the coverage check by zero-ing the optimization constraints for all OARs and running an optimization with no intermediate dose. Following the optimization, dose is calculated and the plan is normalized to deliver 100% of the Rx dose to 90% of the TS_PTV_VMAT volume
- If the hotspot in the resulting plan is > 140%, a warning message will be printed to the user saying the beam arrangement is likely under-covering the target volume (at this point, you should consider stopping the optimization and adjusting the beam orientations)
  - The plan global hotspot is used as a surrogate measure of the quality of target coverage

- Following the coverage check, the OAR objective priorities are changed from zero to a fraction of the priority specified by the user for each OAR
- The optimization loop is then executed. Each optimization loop iteration consists of:
  - A VMAT optimization with intermediate dose switched on
  - A dose calculation following the optimization (algorithm = AAA_15605, dose grid resolution = $2.5 \times 2.5 \times 2.5$ mm^3)
  - Normalization of the plan dose to deliver 100% of the Rx dose to 90% of either the TS_PTV_VMAT volume or the TS_PTV_FLASH volume (if flash was included in the optimization)
  - Plan evaluation to determine if the plan goals were met (the script automatically selects the appropriate template plan goal list for comparison)
  - If the plan goals were met, the optimization loop is broken. If the goals were not met, the script reviews the relative cost associated with each optimization object and determines how to adjust each optimization parameter
  - Cooler (TS_cooler110) and heater (TS_heater90) tuning structures are also generated after each loop iteration to try and reduce hotspots and improve target coverage, respectively
    - The heater structure TS_heater90 will always be added to the structure set. If the optimization is having trouble covering the target (i.e., if the global hotspot is high or V110% is large), an additional heater structure (TS_heater70) will be added to the structure set to try and improve dose coverage to the regions that are being significantly under-covered
  - The optimization parameters for the cooler and heater tuning structures and the updated optimization parameters for the original target and OAR structures are assigned to the plan
  - If the user elected to save the resulting plan from each optimization, the resulting normalized plan from this iteration is copied and saved to the VMAT TBI course
    - NOTE: If the user did NOT elect to run one additional optimization, the number of "opt itr" plans will be one less than the number of requested optimization iterations. This avoids having two copies of the same plan from the final iteration of the optimization loop
  - The next iteration of the optimization loop is started
- The number of loop iterations requested by the user are performed
- If the user requested an additional optimization to reduce the plan hotspots, the tuning structure segmenting the high dose regions (i.e., dose > 110% Rx dose) from the final optimization iteration is obtained and:
  - The priority for this objective is increased to the maximum priority in the list of optimization objectives

  - The objective dose for this constraint is lowered to 101% Rx dose

- The additional optimization is then performed by continuing the previous optimzation using the previously calculated dose as the intermediate dose. The optimization should start from MR3

- If the user included flash in the optimization, following the optimization loop and one additional optimization to reduce hotspots (if selected), the program will grab the BOLUS_FLASH structure and un-assign its HU, recalculate the dose, and normalize the plan to the TS_PTV_VMAT structure (i.e., the 'true' target volume)

- Once the script finishes, the user is informed the optimization loop is complete and the results are written to the '_VMAT TBI' plan

- You can now open the '_VMAT TBI' plan for the patient and evaluate the quality of the resulting plan(s). If you are completely unhappy, you can reset the dose calculation matrix, delete the MLCs for each field, and try again with different parameters