# VMAT TBI And CSI Automated Planning Optimization Loop Script Quick Start Guide

E. A. Simiele

(Revised September 22, 2023)

## Contents

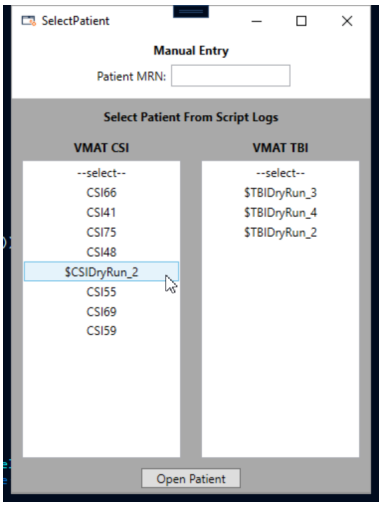


Figure 1: Patient selection UI

# 1 Launching the Script

- Launch from Eclipse over Citrix:
  - Open a patient structure set in external beam planning and run the 'LaunchVMATTBICSIAutoPlan.cs' script (located in the /bin folder of the VMAT-TBI-CSI code)
  - If the launcher script detects a plan as been prepared by the preparation script and is ready for optimization, a third button will appear below VMAT CSI and launches the optimization loop script
  - Select the 'Launch Optimization Loop' button
- Outside of Eclipse on a thick client:
  - Launch the optimization loop executable file in the VMAT-TBI-CSI /bin/VMAT-TBI-CSI folder
  - A simple UI will appear (Figure 1)
  - Select one of the patient MRNs from the CSI list or manually enter the patient MRN and hit Open Patient button

# 2 Basic Information, UI, and Operation

- Review the prepopulated global options in the main UI (see the full user guide for details on what each option does)
- A plan template should automatically be selected

- The UI in this script was designed to be relatively rigid where the user can review and make adjustments to the plan optimization parameters
- **The plan prescriptions cannot be modified in this program!**
- **The normalization volumes cannot be modified in this program!**
- In the global parameters, set the options to the following:
  - Run coverage check –> unchecked
  - Copy and save each optimized plan –> unchecked
  - Max number of optimizations –> 3
  - Run additional optimization to lower hotspots –> checked
  - CSI: PTV V100% –> 95
  - TBI: PTV V100% –> 90
- Review the pre-populated objectives and constraints and make any necessary adjustments
- Once satisfied with the constraints, hit the 'Confirm and Begin Optimization' button to start the optimization loop

# 3 Optimization Loop

- 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 optimization loop progress window. This will tell the program the user wants to stop the optimization loop
  - **NOTE: THE PROGRAM WILL (LIKELY) NOT STOP 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 IN ESAPI**
  - **<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! This will not terminate the optimization loop immediately. A safeguard (basically, a bunch of annoying pop up windows) has been implemented to prevent the user from closing the window before the optimization loop has finished or been safely aborted
- The GUI window can be closed only when the program status is 'Aborted', 'Failed', or 'Finished'
- The current task and progress of the current task is shown by the label and progress bar at the bottom of the UI
- The overall progress of the optimization loop routine is shown in the progress bar in the bottom right of progress window
- 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

## 3.1 Program Flow

## 3.2 Preliminary Checks and Coverage Check

- Once the 'Confirm and Begin Optimization' button is hit, the program reads the configuration settings, planning objectives, and optimization constraints
- 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 not, it will ask the user if they want to continue anyway or stop and insert the couch.
- If all the preliminary checks pass, the optimization loop will commence
- First, the OAR objective priorities are changed to a fraction of the priority specified by the user for each OAR (specifically, 2/3 of the specified priority)
- The progression of the optimization loop varies based on the type of case being optimized: TBI or CSI
  - The case type can be further divided for CSI: initial-only or sequential boost CSI

## 3.3 TBI cases

- The optimization loop is then executed. Each optimization loop iteration consists of:
  - A VMAT optimization with intermediate dose switched off

  - A dose calculation following the optimization
  - Normalization of the plan dose to deliver the requested target coverage to the normalization volume
  - Plan evaluation to determine if the plan goals were met
    * 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 and heater tuning structures are generated after each loop iteration to try and reduce hotspots and improve target coverage, respectively
    * The requested heater and cooler structures can be specified in the plan template .ini files
    * In addition to requesting the dose levels and priorities, the user can also specify the conditions that must be satisified for the script to generate the heater/cooler structure
  - The optimization parameters for the cooler and heater tuning structures and the updated optimization parameters for the 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 CSI 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 (Figure **??**) are performed
- If the user requested an additional optimization to reduce the plan hotspots (Figure **??**), the tuning structure segmenting the high dose regions (e.g., 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 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 or MR4 depending on the TPS configuration
- Once the number of iterations has been reached and the additional optimization performed (if requested), the optimization loop will evaluate if flash structures were added to the structure set with the preparation script. If so, the optimization script will then:

  - Wipe the calculated dose
  - Un-assign the HU to the bolus_flash structure (i.e., treated as air)
  - Recalculate dose using the optimized MLC pattern
  - Normalize the plan to achieve the requested target coverage for the non-flash VMAT target structure

- Once the script finishes, the user is informed the optimization loop is complete and the results have been saved to the Aria database

- You can now open the patient in Eclipse 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

## 3.4 Initial-only CSI cases

- The flow of the initial-only CSI cases identical to the TBI cases except the script will not look for flash structures following completion of the optimization loop

## 3.5 Sequential boost CSI cases

- The flow of sequential boost CSI cases is similar to initial-only cases, however, the script must now optimize the quality of two plans that depend on each other

- All optimizations are performed with intermediate dose switched off to increase efficiency (through testing we found that using intermediate dose for these cases really doesn't improve plan quality)

- Following optimization, dose calculation, and normalization of the initial and boost CSI plans, a plan sum is created

- Since no method exists in v15.6 of ESAPI to create a plan sum, a custom method was developed to create a plan sum

  - The Evaluation Dose method was utilized in Eclipse to "build" a plan sum

- Following creation of the plan sum, the summed dose distribution is evaluated against the planning objectives

- If all of the planning objectives are met, the optimization loop is terminated

- If not, the DVH of each plan is then compared against the requested optimization objectives and adjustments are made to the optimization constraints for each plan

- The next iteration of the optimization loop starts

- Following the maximum number of iterations, an additional optimization is performed for both plans is the user requested one additional optimization to lower hot spots

- If the user requested one additional optimization to lower hot spots, following the additional optimization, the hotspot in the initial CSI plan is evlauted
  - If the hotspot is $> 110\%$, a high-priority cooler structure segmenting the 107% dose regions is added and one additional optimization is performed on the INITIAL CSI plan
  - Following dose calculation, a new plan sum is generated