CERN

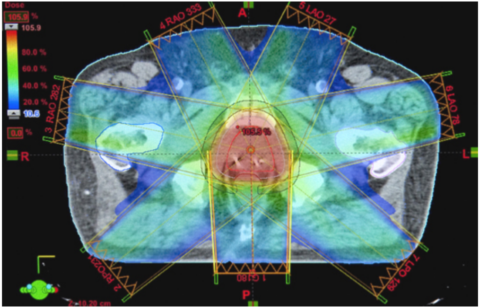
FLUKA pre-optimizer for a Monte Carlo Treatment Planning System (MCTPS)

Guido Arnau
Supervisor: Vasilis Vlachoudis

Treatment Planning System

In **proton therapy**, used to find the treatment plan which:

- minimizes the dose received by healthy organs
- fits the prescribed dose received by the tumour



FLUKA is a general purpose tool for calculations of particle transport and interactions with matter.

FLUKA

- Build an optimizer together with FLUKA to simulate treatment plans and select the best one.

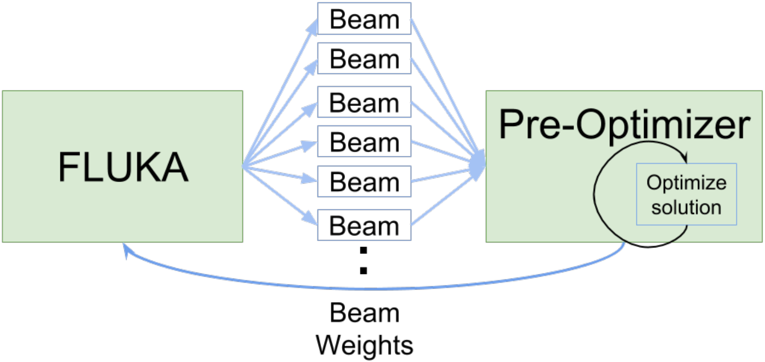
FLUKA

- Build an optimizer together with FLUKA to simulate treatment plans and select the best one.
- Problem: each of the simulations takes 1 hour.

Solution: Pre-Optimizer

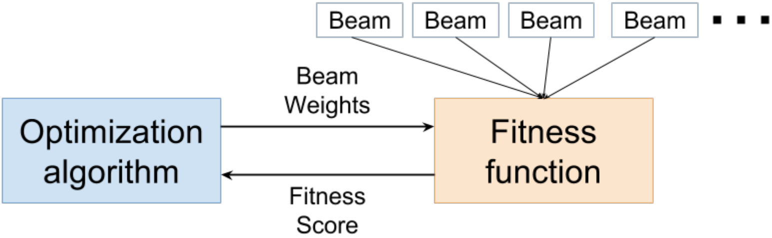
A **multi-threaded** pre-optimizer will be used to approximate a good solution.

MCTPS Process



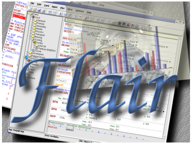
Inside the Pre-Optimizer

Two distinct layers:



Higher Layer: Optimization Algorithm

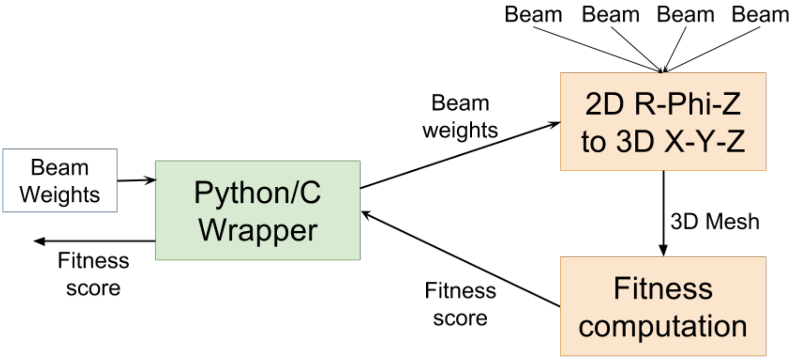
- Written in Python
- Implements:
  - Genetic Algorithm
  - Steepest Descent
- Easily interfaces with Flair



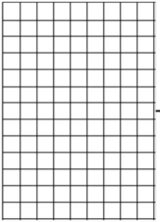
Lower Layer: Computing the Fitness Score

- Implemented in C++
- Interfaces with Python layer using the Python/C API
- Two phases:
  - Merging of the 2D cylindrical projections of the dose per beam into a 3D mesh
  - Computing the fitness score

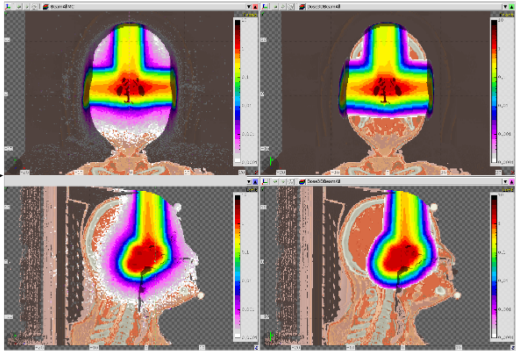
Computing the Fitness Score



2D ρ-ϕ-z to 3D X-Y-Z Transformation



2D Matrix
containing the dose
of each voxel



2D ρ-ϕ-z to 3D X-Y-Z Transformation

- This step is **not** computationally intensive,

2D ρ-ϕ-z to 3D X-Y-Z Transformation

- This step is **not** computationally intensive, but it needs to be applied to **each beam**.

2D ρ-ϕ-z to 3D X-Y-Z Transformation

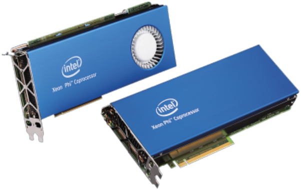
- This step is **not** computationally intensive, but it needs to be applied to **each beam**.
- However, it can be done in **parallel** since there are no dependencies!

Parallelization with the Intel Xeon Phi

Task highly parallelizable $\longrightarrow$
use Intel Xeon Phi coprocessor.

Intel Xeon Phi

Cores	60
Threads	240 (4 HW Threads per core)
Clock Speed	1.053 GHz
Memory	8 GB



OpenMP's Compatibility

- Programming with OpenMP allows us to be able to run the program both on the Xeon Phi and on regular multi-core computers.

Fitness Formula

- Dose received by the tumour as close as the prescribed one as possible.
- Limit the dose received by the healthy organs to the recommended one.

$$\chi^2(\mathbf{N}) = \sum_{j \in PTV} (Prescribed\_Dose - Dose_j)^2 - \sum_{j \in OAR} (Allowed\_Dose - Dose_j)^2$$

Project Objectives

- Test which algorithm is most suited for the pre-optimization.
- Measure the speedup yielded by the Xeon Phi.