Parallel Simplex Method in the Radiotherapy Treatment using OpenMP

A. Abdagić, Z. Ohran, A. Pandžić, D. Švraka
Department of computing and informatics
Faculty of Electrical Engineering
Sarajevo, Bosnia and Herzegovina
aabdagic2@etf.unsa.ba, zohran1@etf.unsa.ba, apandzic2@etf.unsa.ba, dsvraka1@etf.unsa.ba

***Abstract*—Efficient radiation therapy optimization is crucial in cancer treatment, necessitating the minimization of radiation doses to critical tissues while maximizing doses to tumor tissues. This paper explores the use of OpenMP to parallelize the simplex code, facilitating the optimization of radiation doses derived from raw CT images. The process involved transforming raw CT images into labeled regions corresponding to tumor and critical tissues. The objective was to achieve a balance between minimizing radiation doses to critical tissues and maximizing doses to tumor tissues. Through the implementation of parallelization techniques, the simplex algorithm is adapted to improve the efficiency of radiation dose optimization. Application of this method is important considering modern trends of multi-core architecture of processors. This paper presents the methodology, challenges faced, and the achieved outcomes in the context of radiation therapy optimization. The implications of the work extend to improving treatment outcomes in cancer patients by precisely modulating radiation doses.**

***Index Terms*—Radiation therapy, optimization, OpenMP, simplex algorithm, parallelization, CT images, tumor tissue, critical tissue**

I. Introduction

Radiotherapy treatment design is the process of choosing how beams of radiation will travel through a cancer patient to treat the disease, and although optimization techniques have been suggested since the 1960s, they are still not widely used. Instead, the vast majority of treatment plans are designed by clinicians through trial-and-error. Modern treatment facilities have the technology to treat patients with extremely complicated plans, and designing plans that take full advantage of the technology is tedious. The increased technology found in modern treatment facilities makes the use of optimization paramount in the design of successful treatment plans.

This research addresses the critical challenge of using the underexplored potential of optimization techniques in the design of the radiotherapy treatment. The focus is specifically on the parallelization of the simplex code using OpenMP, aiming to revolutionize the design of radiation therapy treatment plans. Parallelization enhances computational efficiency, enabling the simultaneous execution of multiple tasks. This accelerates the optimization process, reducing the time required for treatment plan generation. Secondly, the computational demands associated with large datasets can be effectively handled, such as high-resolution CT images. Parallelization brings efficiency, scalability, and enhanced resource utilization to the radiotherapy treatment design problem, offering a pathway to more sophisticated and optimized treatment plans for cancer patients. Using parallelization not only makes the optimization process faster but also helps us explore more possible solutions. This increases the chances of finding the optimal or nearly optimal treatment plans. Also, parallelization makes this optimization system adaptable, so it can handle different levels of complexity in treatment plans and technology.

Planning treatment with radiosurgery involves carefully analyzing computerized tomography (CT) images along with magnetic resonance images to identify critical structures and tumors. The simplex method acts as a mathematical tool to systematically refine the distribution of radiation doses using the data from images, contributing to the creation of treatment plans that are both precise and optimized for each individual case. The plans are designed based on the desired dose levels for specific tumors and critical tissues in the treated area. The primary goal is to maximize the likelihood of curing the disease while minimizing the risk of complications in any affected critical tissue.

II. Related papers

The [1] document explores the application of linear programming and the simplex method in optimizing radiotherapy treatment planning. By formulating a mathematical model, it seeks to deliver an elevated radiation dose to the tumor while minimizing the impact on critical organs. The simplex method serves as a key tool in solving the optimization problem and generating optimal treatment plans. This aligns seamlessly with this work's use of linear programming and the simplex algorithm for optimization, creating a significant synergy in the exploration of similar methodologies to enhance treatment planning precision.

On the other hand, the [2] document introduces a parallel implementation of the simplex method using OpenMP, aiming to enhance the algorithm's performance for large-scale linear programming problems. The parallelization technique involves distributing computation across threads, resulting in notable improvements in acceleration and efficiency with an increased number of threads/cores. This directly corresponds to this paper's objective of accelerating the simplex method through parallel computing. Furthermore, the paper [3] breaks down the simplex method for solving linear programming problems in a detailed way. This helps us understand the process better with step-by-step explanations, examples, and visuals. Citing this paper in this conference paper adds to the basics of the simplex algorithm, making it easier for readers reading this paper to grasp before getting into the details of this parallel computing approach.

In summary, the [1] document's focus on treatment planning and the [2] paper's emphasis on parallel computing techniques both contribute significantly to the overarching goals of this paper, while the [3] paper gives us in detail explanation of the optimization method simplex. These papers have played a main role in shaping and advancing the methodologies employed in this paper, contributing significantly to the overall success of this research in the fields of linear programming, the simplex method, and parallel computing.

III. CT IMAGE PROCESSING TO SET UP THE INITIAL MATRIX FOR SIMPLEX APPLICATION

In the realm of radiotherapy, optimizing dosage distribution is a critical aspect of enhancing treatment efficacy while minimizing adverse effects on surrounding healthy tissues. This study delves into an innovative approach by integrating a parallelized simplex algorithm with the segmentation of CT head images, aiming to precisely define critical tissues, such as the jaw, spinal cord, and brain, as well as cancerous tissues affecting the salivary-parotid glands.

A. Methodology

In this research, the groundwork involved meticulous manual segmentation orchestrated by a medical student at the University of Sarajevo. Specific tissues in CT head images were marked using distinct colors: red for the jaw, blue for the spinal cord, yellow for the brain, and green for cancerous tissues in the salivary-parotid glands. These color-coded markings served as the foundation for subsequent processing in Google Colab, utilizing Python programming. Visual cues from the hues facilitated the creation of masks, with each mask assigning binary values based on tissue marking status. The resulting TXT files, one for each tissue type, encapsulate the spatial distribution of tissues represented by binary values. This dataset, combining color-coded markings with binary representation, underpins the application of the parallelized simplex algorithm for optimizing radiotherapy dosage planning.

B. Data Representation

The culmination of these TXT files into a unified dataset is a crucial step in preparing for the implementation of the parallelized simplex algorithm. This comprehensive dataset encapsulates the spatial distribution of marked tissues, offering a nuanced representation of the treatment landscape. Each tissue type's binary representation facilitates the algorithm's application in the subsequent optimization process.

Central to this study is the formulation of an objective function that seeks to strike a balance between maximizing radiation dosage to cancerous tissues and minimizing exposure to critical structures. The jaw, spinal cord, and brain are identified as critical tissues requiring radiation minimization, while the salivary-parotid gland cancerous tissue demands maximization for effective treatment. The parallelized simplex algorithm is strategically employed to navigate this complex optimization terrain, adhering to the defined constraints associated with each tissue type.

When cancer is diagnosed and there are medical indications for treatment by radiotherapy, various tests are performed on the patient, in order to learn about the location, form and tumor volume, as well as critical tissues in the region to be treated. The use of computerized tomography exams for reliable data collection and in the radiosurgery is required along with a computed tomography and magnetic resonance exam. Based on this data, the dose to be received in the tumor and the volume to be irradiated may be prescribed by the doctor radiotherapist. Together with a physical, the radiation oncologist can then by the analysis of isodose curves define the best type of treatment and the technique to be used. After obtaining the images by computed tomography or magnetic resonance imaging, the minimum dose to be applied to the tumor is prescribed as the maximum doses that critics and healthy tissue can receive. Through these images, it makes the selection of the anatomical structures of interest. On the images obtained, the structures of interest can then be mapped (and critical lesion tissue) through a pixelation. Each pixel is related to an electron density and receives specific coordinates (i,j).

Fig.1. is an image of the appearance of the tissue that will be affected by the radiation rays in the ratio that is derived from the solution obtained through the parallelized simplex algorithm. [1]

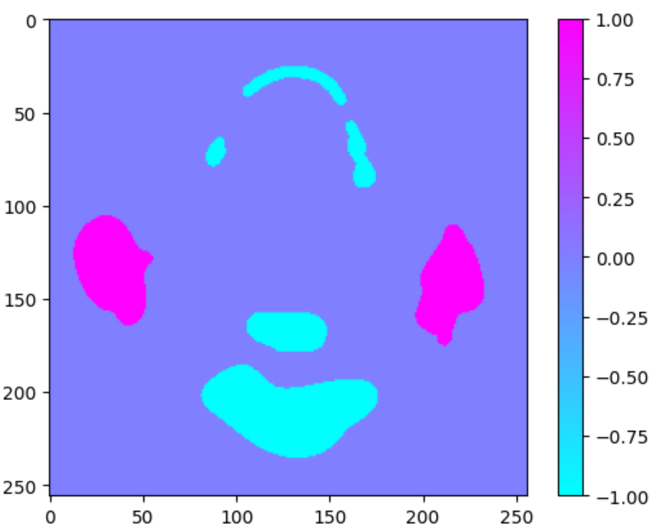

Fig. 1. Visual representation of tumor and critical tissue

IV. OPTIMIZATION METHOD - SIMPLEX

The simplex method is an algorithmic approach for finding the optimal solution to a linear programming model by hand. The first step is to transform the original maximization problem into an equivalent minimization problem. Slack variables are then introduced to convert inequality constraints into equalities. A table is constructed containing the coefficients of the objective function and constraints. The optimality of the current solution is checked by examining the bottom row (zj row) of the table - if any values are negative, the solution is not yet optimal. The pivot variable is identified by finding the smallest negative value in the bottom row and its corresponding column. A new table is created by optimizing the pivot variable to 1 and making other values in the pivot column 0 using row operations, while keeping the rest of the table equivalent. The optimality of the new table is checked again. If still not optimal, new pivot variables are identified and the pivoting process repeats until an optimal solution is reached. [4] The pseudocode for the simplex method is provided below for a concise representation of the algorithmic steps:

Input: $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and a basic feasible solution x with basis B.
Output: The primal-dual optimal pair (x, y) or the information that the given problem is unbounded.

while TRUE
  do Find the non-basis N and matrices A_B, A_N
    Solve $A_B^T y = c_B$.
    Set $u_N = c_N - A_N^T y$.
    if $u_N \geq 0$
      then return (x, y).
    Choose $r \in N$ such that $u_r < 0$.
    Solve $A_B w_B = a_r$, a_r is the r-th column of A.
    if $w_r \leq 0$
      then return "The given LP is unbounded."
    Choose $t := \min_{i\in B} \left\{ \frac{x_i}{w_i} \mid w_{Bi} > 0 \right\}$.
    $x_B := x_B - t w_B$.
    $x_r := t$.
    $B := (B \cup \{r\}) \setminus \{s\}$, where s is the variable leaving the basis.

A. Objective Function and Constraints

In the pursuit of optimizing radiation therapy using the simplex method, the central goal is to maximize the radiation dose to the tumor while adhering to critical dose constraints. The study incorporates data from 126 beams, each with a resolution of 256x256 pixels, which undergoes meticulous processing to form a dosage matrix. By formulating constraints based on minimum tumor dose and maximum critical dose requirements, an objective function is intricately designed to encapsulate the cost of radiation delivery, accounting for both critical and tumor regions. Introducing constraint relaxation for adaptability in both tumor and critical regions, an additional constraint limits the total weight of radiation, ensuring a holistic approach to the optimization problem. The resulting objective function and constraints are expressed in a clear and interpretable manner, with zero constraints replaced by specified values and downsampling applied for computational efficiency. The creation of a matrix representation further readies the model for the application of the simplex method, ensuring a comprehensive and efficient approach to radiation therapy optimization. The linear programming problem can be formulated as follows:

$$\text{Minimize } Z = \text{cost\_critical} - \text{cost\_tumor} \cdot x \tag{1}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{2}$$

$$A_{\text{ub}} \cdot x \geq b_{\text{ub}} \tag{3}$$

$$A_{\text{eq}} \cdot x \leq b_{\text{eq}} \tag{4}$$

$$0 \leq x_i \leq 1 \times 10^{99} \quad \text{for each } i \tag{5}$$

The variable `cost_critical` represents the cost associated with delivering a critical dose of radiation. This cost is likely related to a vital component within the treatment process. The variable `cost_tumor` represents the cost associated with delivering a radiation dose to the tumor. It reflects the expense or impact associated with treating the tumor with radiation. The variable `x` is a vector of decision variables in the optimization problem. Each element x_i in the vector represents the weight assigned to the radiation dose component corresponding to the i-th index. The optimization aims to find the optimal distribution of weights (x_i) that minimizes the overall cost, considering both `cost_critical` and `cost_tumor`.

The upper-bound inequality constraints ($A_{\text{ub}} \cdot x \geq b_{\text{ub}}$) are formulated to guarantee that the weighted sum of radiation doses exceeds a minimum threshold for effective tumor treatment. Here, A_{ub} is a matrix defining the relationship between decision variables x and b_{ub}, which represents the minimum threshold. Conversely, the lower-bound inequality constraints ($A_{\text{eq}} \cdot x \leq b_{\text{eq}}$) are established to ensure that the weighted sum of radiation doses remains within a specified maximum limit to prevent harm to critical structures. In this case, A_{eq} is a matrix defining the relationship between decision variables x and b_{eq}, which represents the maximum limit. Additionally, each decision variable is constrained to be non-negative ($0 \leq x_i$), indicating that the weights assigned to radiation doses can not be negative. Furthermore, the decision variables are allowed to take values up to 1×10^{99}, ensuring sufficient flexibility in the optimization process.

V. Parallelization using OpenMP

The implementation of the simplex algorithm for radiation therapy optimization is enhanced through parallelization using OpenMP. OpenMP (Open Multi-Processing) is an API that supports multi-platform shared memory multiprocessing programming in C, C++, and Fortran. [5] It allows us to write parallel code easily by adding compiler directives to existing code.

In this case, the simplex algorithm was parallelized to take advantage of the multi-core architecture of modern processors. The parallelization primarily focuses on the key operations within the algorithm, such as finding the pivot element and updating the table.

The parallelized version of the simplex algorithm is designed to efficiently distribute the workload across multiple threads, thereby accelerating the optimization process. The parallelization is achieved by adding OpenMP directives to the relevant sections of the code, allowing the algorithm to execute parallel operations concurrently.

Function *func1* (Listing 1) identifies the minimum ratio test within the simplex algorithm by iterating over constraint rows and computing the ratio of the right-hand side value to the pivot column value. The OpenMP pragma on line 2 is used to parallelize the subsequent for loop, which updates the pivot for the current row to find the minimum value and index. The pragma on line 6 indicates a critical section where only one thread can execute at a time. In this case, it ensures that the code inside the critical section, involving the update of a minimum value, is executed atomically to prevent race conditions. The pragma on line 9 specifies an atomic operation, ensuring that counting non-positive coefficients is executed atomically, avoiding data inconsistency due to concurrent threads.

```
1 function func1 <parameters>
2     #pragma omp parallel for reduction
3     for i = 0 to constraintNumb - 1
4         // Update pivot for the current row to
                find the minimum value and index
5
6         #pragma omp critical
7             // Update minimum value
8
9         #pragma omp atomic
10            // Count non-positive coefficients
```

Listing 1. Function to update min and count non-positive coefficients

Function *func2* (Listing 2) is tasked with updating the pivot row in the table by normalizing each element with the chosen simplex pivot. The pragma on line 12, indicates that the subsequent for loop will be parallelized. The iterations of the loop will be distributed among multiple threads, potentially improving performance. Inside the loop (lines 13 and 14), the pivot row is updated, dividing each element by the pivot value.

```
11 function func2 <parameters>
12     #pragma omp parallel for
13     for j = 0 to colN
14         // Update pivot row
```

Listing 2. Function to update the pivot row based on min and pivot

Function *func3* (Listing 3) modifies the table to ensure it adheres to the simplex algorithm by updating non-pivot rows. The pragma on line 16 indicates parallelization of the outer for loop (lines 17-20), which iterates over constraints. Within this loop, line 18 updates the pivot and the inner loop (lines 19-20) iterates over columns, updating each row using the pivot.

```
15 function func3 <parameters>
16     #pragma omp parallel for
17     for i = 0 to constraintNumb - 1
18         // Update pivot
19             for j = 0 to colNumb
20                 // Update each row
```

Listing 3. Function to update other rows in table based on min and pivot

Function *func4* (Listing 4) updates the table to reflect changes after identifying the pivot column. The pragma on line 22 indicates that the subsequent for loop will be parallelized. The reduction clause implies that a reduction operation will be performed during the parallel execution. Reduction is a technique used to combine partial results from different threads into a single result. Inside the loop (lines 23-24), last row in the table is updated. Following the loop, there is a critical section pragma (line 25) indicating that the subsequent code block (line 26) is a critical section, in which the max value is updated. The critical section ensures that only one thread can execute this block at a time, avoiding data inconsistency due to concurrent updates.

```
21 function func4 <parameters>
22     #pragma omp parallel for reduction
23     for j = 0 to colNumb
24         // Update the last row in the table
25         #pragma omp critical
26         // Update the max value
```

Listing 4. Function to update the last row in table based on min, max and pivot

Overall, the parallelization of these functions leverages the capabilities of multi-threading to enhance the computational efficiency of the simplex algorithm, particularly in the critical steps of finding the minimum ratio test and updating the table.

VI. Experimental Results

The computational efficiency of the parallelized simplex algorithm is a crucial aspect of its applicability in radiotherapy dosage optimization. The algorithm's ability to parallelize computations contributes significantly to reducing the overall optimization time. As demonstrated in these experiments, the parallel implementation of the simplex algorithm yields notable improvements in optimization speed compared to traditional sequential methods.

The optimized dosage distribution achieved through the parallelized simplex algorithm showcases its effectiveness in minimizing radiation exposure to critical tissues. The results indicate a substantial reduction in the radiation dosage to the jaw, spinal cord, and brain, ensuring enhanced protection of these vital structures. This outcome is particularly promising in mitigating potential side effects and complications associated with radiation exposure to these critical anatomical regions.

Conversely, for cancerous tissues in the salivary-parotid glands, the parallelized simplex algorithm excels in maximizing radiation dosage. The findings reveal an optimized dosage distribution that effectively targets and delivers therapeutic radiation to the cancerous tissue, enhancing treatment efficacy. This targeted maximization is pivotal in achieving optimal outcomes for cancer treatment while minimizing the impact on surrounding healthy tissues.

Fig.2. is a graphical representation of what the rays look like and how they properly act on certain tissues with a certain optimized dose of radiation. Comparing this picture with the one shown earlier illustrating the appearance of critical tissues, the visual analysis reveals the action of the radiation beam.

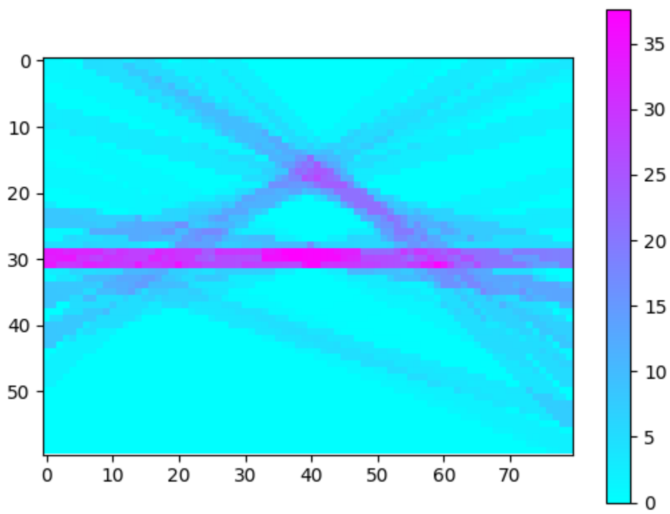

Fig. 2. Visual representation of Parallel Simplex Method in the Radiotherapy Treatment

Experiments were conducted on four types of matrices with varying dimensions, using four threads, to comprehensively assess the capabilities of both serial and parallel simplex algorithms. The matrix dimensions are denoted as mxn, representing the number of constraints and variables, respectively. Both serial and parallel versions of the simplex algorithm were implemented and optimized for the experimental setup. The serial implementation served as the baseline, while the parallel implementation utilized parallel processing capabilities to enhance computational speed.

TABLE I
SERIAL VS. PARALLEL SIMPLEX ALGORITHM SPEEDS

Matrix Dimensions	**Serial Time (ms)**	**Parallel Time (ms)**	**SUF**
132×16384	78.32	45.80	1.71
5000×5000	393.86	180.52	2.18
10000×10000	1528.56	586.51	2.61
10000×15000	1874.47	680.54	2.75

The speed-up factor (SUF) was computed as the ratio of the parallel execution time to the serial execution time for each test case. The values reflect the degree of acceleration achieved through parallelization, providing quantitative evidence of the performance improvements realized in the parallel simplex algorithm.

Graph on Fig.3. illustrates the relationship between the number of threads and the corresponding average execution time for a matrix of dimensions 5000×5000. The x-axis represents the number of threads, while the y-axis denotes the average execution time in seconds.

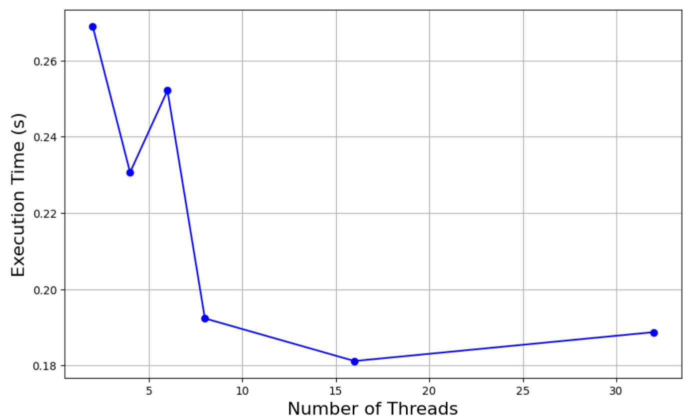

Fig. 3. Average Execution Time vs. Number of Threads

As the number of threads increases, there is a discernible trend in the average execution time. The blue line connecting the data points provides insights into the efficiency gains achieved through parallelization. The data suggests that the execution time decreases as the number of threads increases, showcasing the potential benefits of parallel processing in optimizing the overall computation time.

VII. CONCLUSION

In this study, the parallelization of the simplex algorithm was thoroughly examined and implemented in the context of determining radiotherapy dosage. Experiments with both parallelized and enhanced versions of the algorithm provide a deeper understanding of their performance in optimizing dose distribution for matrices of various dimensions.

The results indicate a significant reduction in execution time through the use of parallelization, showcasing the potential for accelerating the radiotherapy planning process. The enhanced parallel simplex algorithm, in particular, stands out, providing additional benefits in terms of execution time optimization compared to the non-optimized version.

The parallelization of the simplex algorithm proves to be a crucial tool in tailoring radiotherapy to individual patient needs. By introducing parallelization, a balance was achieved between minimizing critical tissue exposure and maximizing cancerous tissue treatment efficacy. This research contributes to the ongoing efforts to enhance the efficiency and effectiveness of radiotherapy planning through parallel computation techniques.

The promising outcomes presented here underscore the potential of parallelized algorithms in advancing personalized and targeted radiotherapy, ultimately leading to improved patient outcomes and more efficient treatment planning processes. Future work in this domain may explore further optimizations, considering specific constraints and characteristics of radiotherapy treatment plans.

REFERENCES

[1] Thais R. Salvador, Silvia M. S. Carvalho, Mayk V. Coelho, “Application of Simplex Method in the Radiotherapy Treatment,” DFQM, Federal University of São Carlos, Sorocaba, São Paulo, Brazil.

[2] Lesia Mochurad, Nataliya Boyko, Nataliia Petryshyn, Myroslava Potokij, Mariia Yatskiv, “Parallelization of the Simplex Method Based on the OpenMP Technology,” Lviv Polytechnic National University, Lviv 79013, Ukraine.

[3] Ajit Singh, “Linear Programming Problem Solving: Simplex Method,” Patna Women’s College, Bihar, India.

[4] Hédi Nabli, ‘An overview on the simplex algorithm,” Faculté des Sciences de Sfax, Département de Mathématiques, Route de Soukra Km 3, BP 1171, Sfax 3000, Tunisia, 15 April 2009, Bihar, India.

[5] OpenMP Architecture Review Board, *OpenMP Application Programming Interface*, November 2021. [Online]. Available: https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5-2.pdf