Research and Design Rationale for a Hodgkin's Lymphoma Treatment Simulation Program

Date: August 6, 2025
Prepared By: Rakesh V Shetty
Location: Bengaluru, Karnataka, India

1. Objective

The primary objective of this research was to analyze an initial C program designed to simulate Hodgkin's Lymphoma treatment. The analysis aimed to identify discrepancies between the program's logic and established, real-world clinical practices. Based on these findings, the goal was to re-engineer the program in C++, creating a more robust, realistic, and educational simulation that better reflects the complexities of modern oncology care.

2. Analysis of the Initial C Program

The C program served as a functional prototype but contained significant simplifications that diverged from clinical reality.

2.1. Treatment Selection Logic

The program used a simple if/else cascade based on stage, organ health percentages, and a boolean for "bulky tumors." While these factors are important, this logic is a rudimentary representation of a highly complex decision-making process. Real-world regimen selection is multifactorial, involving precise risk stratification (e.g., "early-stage favorable" vs. "early-stage unfavorable") and prognostic scores (like the International Prognostic Score, IPS) which the original code did not account for.

2.2. Treatment Schedule Representation

A core clinical inaccuracy was the representation of a chemotherapy schedule. The program used variables like treatment_days and break_days to define a cycle. This is fundamentally incorrect. Chemotherapy regimens are not administered on a contiguous block of days. They follow precise protocols, such as:

- **ABVD:** Administered intravenously on **Day 1 and Day 15** of a **28-day cycle**.

The model of a simple number of treatment and break days does not capture this essential,

rigid structure.

2.3. Misapplication of Simulated Annealing

The program employed a Simulated Annealing algorithm to "optimize" the treatment schedule by minimizing a cost_function. This represents a misapplication of the optimization technique in this context for two reasons:

1. **Arbitrary Cost Function:** The cost function (treatment_days * 1.0 + break_days * 0.5) has no clinical basis. A real-world "cost" would involve a complex interplay of tumor cell kill dynamics, cumulative organ toxicity, and patient quality of life, which cannot be modeled by a simple linear equation.
2. **Invalid Solution Space:** The algorithm worked by randomly perturbing the number of treatment and break days. As established in 2.2, these schedules are fixed. Oncologists do not randomly alter the number of treatment days in a cycle; they make specific, protocol-driven adjustments like reducing a drug's dosage or delaying the entire next cycle.

2.4. Cycle Adjustment Mechanism

The adjust_next_cycle function correctly identified the need for treatment adaptation based on patient response (e.g., blood counts, tolerance). However, its implementation (data->treatment_days -= 2; data->break_days += 2;) was arbitrary and not reflective of actual clinical interventions.

3. Core Research Findings from Clinical Guidelines

Research based on the provided sources (Cancer.org, Medscape, Cancer Research UK) and general oncology principles yielded the following core concepts that guided the redesign.

3.1. Staging and Risk Stratification

Hodgkin's Lymphoma treatment is critically dependent on a precise classification beyond just stages 1-4. A key distinction is:

- **Early-Stage:** Divided into **Favorable** and **Unfavorable** based on factors like the presence of bulky disease, "B symptoms" (fever, night sweats, weight loss), and the number of lymph node sites involved.
- **Advanced-Stage:** Stages III and IV.

This stratification directly dictates the intensity and type of chemotherapy regimen chosen.

3.2. Common Chemotherapy Regimens & Their Use Cases

- **ABVD:** A long-standing gold standard, especially for **early-stage favorable** disease. Its primary toxicities of concern are pulmonary fibrosis (from **B**leomycin) and cardiotoxicity (from **A**driamycin/Doxorubicin).
- **A+AVD (Brentuximab Vedotin + AVD):** A modern standard for many **advanced-stage** patients and some high-risk early-stage cases. It replaces the lung-toxic Bleomycin with Brentuximab Vedotin, whose main side effect is peripheral neuropathy.
- **Escalated BEACOPP:** A more intensive and toxic regimen reserved for **high-risk, advanced-stage** patients. It has high efficacy but carries significant risks of severe myelosuppression (dangerously low blood counts) and long-term side effects like infertility and secondary cancers.

3.3. Treatment Adaptation Principles

Clinical adjustments are protocol-driven and based on specific events:

- **Myelosuppression:** If White Blood Cell (WBC) or platelet counts fall below critical thresholds, the next cycle is typically **delayed** until recovery. Growth factors (like G-CSF) may be administered.
- **Organ Toxicity:** If tests reveal damage to the heart (e.g., decreased LVEF) or lungs (e.g., decreased DLCO), the offending drug (Doxorubicin or Bleomycin, respectively) may be **discontinued** for all subsequent cycles.
- **Interim PET-CT Scans:** It is standard practice to perform a PET-CT scan after 2 cycles. A very good response (e.g., Deauville score 1-2) may allow for **de-escalation** of therapy (e.g., dropping Bleomycin from the ABVD regimen) to reduce long-term side effects.

4. Design Rationale for the C++ Implementation

The research findings directly informed the new object-oriented design in C++.

4.1. Object-Oriented Modeling

Real-world entities were modeled as C++ classes to encapsulate their data and behavior:

- Patient: Stores static patient characteristics.
- ChemotherapyRegimen: An abstract base class for treatment protocols.

- TreatmentPlan: A controller class that links a Patient to a ChemotherapyRegimen and manages the treatment flow.

4.2. Polymorphism for Regimens

A polymorphic design was used for ChemotherapyRegimen. An abstract base class defines a common interface (getName(), getTotalCycles(), etc.), while derived classes (ABVD, AAVD, BEACOPPescalated) implement the specific, real-world details of each protocol. This makes the system clean, realistic, and easily extensible to include new regimens in the future.

4.3. Replacing Optimization with Realistic Adjustment

The flawed Simulated Annealing algorithm was entirely removed. It was replaced with an adjustment mechanism in the TreatmentPlan::adjustPlanBasedOnFeedback method. This function now takes structured ClinicalFeedback and applies a set of rules derived from the research findings (Section 3.3) to make meaningful, descriptive recommendations (e.g., "Delay cycle," "Consider dose reduction").

4.4. Data Encapsulation and Modularity

The separation of the codebase into header (.h) and source (.cpp) files improves organization. By making class members private and exposing functionality through public methods, the design prevents inadvertent data corruption and makes the code easier to understand and maintain.

5. Conclusion

The initial C program was a valuable conceptual exercise. However, rigorous analysis against clinical guidelines revealed significant inaccuracies in its model of chemotherapy selection, scheduling, and adaptation. By conducting targeted research into real-world Hodgkin's Lymphoma treatment protocols, a new C++ simulation was developed. This improved program leverages object-oriented design principles to create a more accurate, modular, and educationally valuable tool that better reflects the nuanced, protocol-driven nature of cancer treatment.

6. References

- American Cancer Society. (n.d.). *Treating Classic Hodgkin Lymphoma, by Stage*.
- American Cancer Society. (n.d.). *Chemotherapy for Hodgkin Lymphoma*.
- Medscape. (n.d.). *Hodgkin Lymphoma Treatment & Management*.
- Cancer Research UK. (n.d.). *Chemotherapy for Hodgkin lymphoma*.