Parallelized Simplex Algorithm for LP in Haskell Project Proposal

Bereket Molla (bam2243)

1. **Overview**

Context: Linear programming (LP) is a method to achieve the best outcome in a mathematical model whose requirements are represented by linear relationships. The simplex algorithm is a popular method used to solve LP problems. However, its sequential nature often limits the speed of computation, especially for large-scale problems.

Project Goal: This project aims to develop a parallelized version of the simplex algorithm using Haskell. Haskell's functional nature and robust support for concurrent and parallel programming make it an ideal choice for this project. The goal is to improve the algorithm's efficiency by parallelizing its computational steps, thereby reducing the time complexity for large LP problems. I will focus on adapting the simplex algorithm to utilize Haskell's parallel processing capabilities effectively.

2. Algorithms and Strategies

A Sequential Algorithm: The conventional simplex algorithm operates in a sequential manner, iteratively improving a solution until it reaches an optimal or feasible solution. It involves a series of pivot operations that move the solution from one vertex of the feasible region to an adjacent vertex with a higher objective function value, if possible.

Parallel Opportunities: The potential for parallelization in the simplex algorithm lies in several areas:
- Pivot Operations: Concurrently evaluating multiple pivot options to determine the most favorable move.
- Feasible Region Analysis: Parallel analysis of the feasible region to identify potential next steps or shortcuts.
- Dual Simplex Method: Implementing parallel strategies in the dual simplex method for handling degeneracy and improving efficiency.

My initial strategy will involve identifying the most computationally intensive steps in the simplex algorithm and implementing parallel versions of these steps in Haskell. I will leverage Haskell’s features like lightweight threads and asynchronous operations to distribute these computations effectively across multiple CPU cores.

3. Expected Outcomes and Challenges

Outcomes: The primary outcome will be a Haskell-based implementation of the simplex algorithm that exhibits improved performance on large-scale LP problems due to parallelization. I will measure performance improvements in terms of computation time and resource utilization.

Challenges: Key challenges will include ensuring the correctness and stability of the parallel algorithm, effectively managing data dependencies and synchronization between parallel tasks.

4. Conclusion

By exploiting Haskell's capabilities for parallel computation, I aim to enhance the simplex algorithm's performance, making it a more viable option for complex, real-world LP problems.