Graduate Research Plan

Linear and mixed-integer programming (LP/MIP) are essential methodologies in optimization, forming the backbone for solving complex problems in science, engineering, and many other fields. The software that implements these methods are critical tools, yet widely used solvers often fail due to floating-point errors [1], misreporting a problem as being infeasible or a solution as being optimal. These failures often trace back to their core computational engines, particularly the matrix factorizations used to solve linear systems: Round-off errors in LU factorizations may cause a solver to return incorrect results. While some solvers have begun to incorporate numerically exact subroutines to address this, their high computational cost remains prohibitive, limiting their practical use. I aim to improve this trade-off by designing computationally efficient algorithms and theoretical frameworks to improve the viability of exact linear programming solvers and extend these improvements to mixed-integer optimization.

Intellectual Merit

The goal of my research is to develop efficient and reliable algorithms for exact mixed-integer programming (MIP), a challenging and unsolved area in optimization. Building on my prior work, where I conceived and developed an original algorithm for multi-column LU updates and integrated it into the exact linear programming solver QSopt_ex [2], I propose to pursue three complementary directions:

(1) Enhancing the Efficiency of Exact Optimization Algorithms:

The QSopt_ex solver was developed as an academic prototype to show the feasibility of exact linear programming through precision-boosting, which solves problems in floating point*, verifies the basis in exact arithmetic*, and repairs inconsistencies by re-solving at higher precision. During my REU project, I accelerated the computational bottleneck of *exact verification* by an order of magnitude by incorporating multi-column LU updates into QSopt_ex. This advancement opens two deeper research avenues:

- **Dynamic precision boosting:** Currently, when inconsistencies are detected during the exact basis verification, QSopt_ex increases precision from 64-bit directly to 128-bit and subsequently scales by a factor of 1.5. This leap is often excessive, resulting in costly 128-bit computations, especially when the floating-point solution is nearly correct and requires only a slight precision increase. My research will investigate algorithms to increase the floating-point precision based on properties such as approximate basis conditioning, aiming to provide provable guarantees on the minimum precision required to solve numerically unstable problems.
- **Rational LU updates:** My modification of QSopt_ex uses Saunders's method [3] of LU updates, the same approach employed in its floating-point subroutine. This method prioritizes numerical stability—a concern that is moot when working with exact rational arithmetic. Thus, designing or leveraging existing alternative update methods that prioritize computational efficiency over numerical stability could significantly improve the efficiency of exact multi-column LU updates.

(2) Developing Perturbation-Based Testing Problems for Solver Robustness:

To rigorously evaluate the proposed improvements, I will develop a library of numerically unstable optimization problems that test solver correctness. In prior work testing the performance of exact solvers, I found that existing benchmark instances rarely create the numerical instability needed to trigger multiple exact checks. To address this, I prototyped a perturbation-based method that applies modifications to constraint matrices and right-hand side vectors, generating new and uniquely challenging problems.

I will generalize and refine this approach by exploring how different types and scales of perturbations affect solver behavior. I will also investigate alternative techniques for generating computationally challenging instances, such as constructing ill-conditioned matrices or leveraging pathological examples from numerical linear algebra. These rare instances are valuable stress tests for solver failure modes due to numerical instability, and their systematic study could provide benchmarks to improve solver robustness and guide applications.

(3) Extending Multi-Column LU Updates to Mixed-Integer Programming (MIP)

The advancements in exact LP efficiency and robustness testing culminate in the primary goal of this proposal: extending these methods to exact MIP. In this setting, the core bottleneck is the branch-and-bound framework, which requires solving a vast tree of closely related LP relaxations. Each LP relaxation in the branch-and-bound tree differs only slightly from its parent—typically through a few modified bounds or added cuts—making this structure an ideal candidate for applying multi-column LU updates.

Prior work, such as the exact MIP solver developed by Cook et al. [4], showed the feasibility of exact mixed-integer optimization but explicitly aimed to minimize reliance on the exact LP solver subroutine due to its computational cost. This design choice underscores a critical opportunity: improving the efficiency of exact LP solvers can fundamentally change the design of exact MIP algorithms. By reducing the overhead of exact LP computations, we can enable tighter integration of exact methods within branch-and-bound frameworks, making exact MIP solvers more practical and scalable. This extension raises new and complex theoretical MIP questions: How can we ensure accurate updates with integrality constraints, and how do these updates affect cut generation and node processing to speed up branch-and-bound?

Together, these three research directions form a cohesive plan to advance the reliability and performance of numerical optimization. This research directly confronts the fundamental gap between symbolic correctness and numerical stability. By integrating theoretical insights on adaptive precision with high-performance solver engineering, my work aims to deliver a new class of provably correct optimization tools. The contributions will not only improve the scalability of solvers for both LP and MIP but also provide the broader optimization community with a critical public library for stress-testing any algorithm against the most challenging problems.

Broader Impacts

Reliable optimization is essential because solvers are often regarded as authoritative tools for decision-making [1]. In domains such as structural engineering, energy systems, and aerospace design, even minor numerical errors can propagate into catastrophic failures—compromising safety, efficiency, and cost [1]. This implicit trust makes developing exact and mixed-precision algorithms critical for ensuring accuracy. By making these algorithms open-source and integrating them into existing solvers, this work will enable researchers to obtain verifiably correct solutions to problems that existing solvers could not reliably handle due to numerical instability. To promote reproducibility in numerical optimization research, I plan to release future solver modules with clear documentation. Building on this approach, I recently modified QSopt_ex during my REU project and published it on GitHub with comprehensive user information.

As part of this GRFP project, I plan to mentor undergraduates as a graduate student through workshops and guided research projects. I began this project during my undergraduate studies, and its modular nature—spanning numerical linear algebra, algorithm design, and solver implementation—makes it well-suited for early research experiences. While many areas of mathematical research demand extensive theoretical preparation before meaningful contributions are possible, this project offers an accessible entry point through tangible computational challenges. This research combines theoretical innovation with practical implementation, enabling students to see the immediate impact of their ideas on performance while contributing to advances in numerical optimization. For example, modifying solver behavior and testing numerical stability offers hands-on entry points that build both mathematical intuition and programming skills, arming students with confidence as they tackle deeper underlying mathematical questions. Through these opportunities, I aim to mentor and inspire future researchers, just as I was supported and inspired early in my own journey.

References:

[1] Klotz, E., Newman, A.M, (2013) *'Practical guidelines for solving difficult mixed integer linear programs*.' Surveys in Operations Research and Management Science, Volume 18, Issues 1–2.

[2] Applegate, D., Cook, W., Dash, S., and Espinoza, D. (2007) '*Exact solutions to linear programming problems'*. Operations Research Letters, Volume 35, Issue 6.

[3] Elble, J.M. and Sahinidis, N.V. (2012) '*A review of the LU update in the simplex algorithm*'. Int. J. Mathematics in Operational Research, Volume 4, No. 4.

[4] Cook, W., Koch, T., Steffy, D.E., Wolter, K. (2011). '*An Exact Rational Mixed-Integer Programming Solver'*. IPCO 2011, Lecture Notes in Computer Science, Volume 6655.