Email address

mhaberla@calpoly.edu

Name of Submitter

Matt Haberland

Project Submitting this Proposal

SciPy

Is your project Affiliated or Sponsored?

Sponsored

Proposal Title

A Mixed Integer Programming Solver for SciPy

Two Sentence Summary of Proposal

Mixed-integer linear programming is a fundamental problem in scientific computing, but SciPy has no features for solving such problems. To address this, we propose to add the HiGHS mixed-integer programming solver to SciPy.

Description of Proposal (<750 words, < 4500 chars)

In early 2017, I was surprised to find that SciPy did not have any features for mixed-integer linear programming (technically “MILP”, but commonly “MIP”): optimization with a linear objective function, linear constraints, and additional constraints that specified decision variables must have integer values. These problems are ubiquitous across a variety of fields, as many resource allocation and scheduling problems can be expressed in the language of MIP, yet the scientific Python ecosystem lacks a simple, easily accessible, permissively-licensed MIP solver. I tried to use `scipy.optimize.linprog` within another algorithm to solve such problems, but I found bugs in `linprog` that needed to be fixed before that would be possible, so I started contributing to SciPy [1]. Three years later, Nicholas McKibben posted an issue on our GitHub repository suggesting the same capabilities [2]. Now he’s a core developer, too, yet SciPy still doesn’t have a MIP solver!

A recent request and new enthusiasm from other SciPy users [3] have again brought this issue to the forefront, and it’s time for it to be resolved. With the aid of a Small Development Grant, we will fulfill this longstanding need by adding the HiGHS (C++ Library) MIP solver to SciPy. Specifically, we will:

- move HiGHS, which SciPy currently vendors for its (continuous) linear programming capabilities, from the SciPy main repository to a submodule for easier future maintenance;
- upgrade HiGHS to a more recent version;
- wrap the HiGHS MIP solver using Cython;
- create a Python interface for the HiGHS MIP solver in a new function; e.g. `scipy.optimize.intlinprog`;

- add unit tests to verify the new functionality; and
- add benchmarks to track the performance of `intlinprog` over time.

Consideration of Feedback from 2021 Cycle 1 Proposal

“Small concern about the overhead from Cal Poly, you may be able to work as a “consultant” and be paid directly by NumFOCUS instead of through the University.”

Unfortunately, that is not an option for Matt Haberland, but it is for Nicholas McKibben. We ask for Nicholas to be paid directly by NumFOCUS rather than as a Cal Poly contractor to avoid the overhead.

“Other concerns about the timeline, 30 hours may not be enough time to complete this feature.”

Nicholas and I enjoy working on SciPy, and we are happy to volunteer some additional time to these important projects. That said, in this proposal we have adjusted the budget to more closely reflect the actual time needed to complete the work.

“Please make sure to discuss with the relevant communities to make sure there is not overlapping work.”

Pyomo is a Python-based, BSD-licensed modeling language, but solvers must be installed separately. None of the solvers with specialized interfaces are permissively licensed, and as a modeling language, Pyomo does not offer the simple, matrix-centric interface that SciPy seeks to provide.

COIN-OR CBC, GLPK, and lp_solve are all C++ mixed integer programming solvers with Python bindings, but none are easy to use on all major platforms, and none are permissively licensed.

Google OR provides a mixed integer programming solver under the Apache 2.0 license that can be installed on all major platforms, but its only Python interface is through an inconvenient modeling language.

We are not aware of any other closely related projects or the possibility of overlapping work.

References:

[1] "ENH: added "interior-point" method for scipy.optimize.linprog". *scipy/scipy - GitHub*. https://github.com/scipy/scipy/pull/7123. Accessed 18 August 2021.

[2] "intlinprog: integer linear program solver". *scipy/scipy - GitHub*. https://github.com/scipy/scipy/issues/11673. Accessed 18 August 2021.

[3] "Adds Mixed Integer Linear Programming from highs". *scipy/scipy - GitHub*. https://github.com/scipy/scipy/issues/14455. Accessed 18 August 2021.

[4] "ENH: linprog in SciPy 1.2+ discussion". *scipy/scipy - GitHub*. https://github.com/scipy/scipy/issues/9269. Accessed 18 August 2021.

[5] "Detailed SciPy Roadmap". *scipy/scipy - GitHub*. https://docs.scipy.org/doc/scipy/reference/dev/roadmap-detailed.html. Accessed 18 August 2021.

Benefit to Project/Community (<400 words, < 2500 chars)

This project would resolve several open issues [2-4], and it would address the need noted on SciPy's roadmap [5] for `linprog` to support integer constraints. Millions of scientists and engineers worldwide rely on SciPy for solving problems; adding this capability to SciPy would give these users the tool they need to solve MIPs, a very common and important class of problems.

Amount Requested

$4985

Brief Budget Justification - How will the money be spent?

$2,500 is for 50 hours of work by Nicholas McKibben at a rate of $50/hr. Nicholas' primary responsibility will be to write the wrappers for the HiGHS MIP solver, create the public interface scipy.optimize.intlinprog, implement unit tests, and update the documentation. Note: these funds are not to be paid through Cal Poly; rather, they would be paid directly by NumFOCUS to Nicholas McKibben.

$1,794 ($1,643 salary; $151 fringe benefits/payroll taxes) will compensate the proposer for 24 hours of overload work. The proposer's primary responsibilities will be to review the interface written by Nicholas McKibben and to add benchmarks.

$691 is for Cal Poly recovery of indirect costs.

The Cal Poly salary and wage rates are based on the California Polytechnic State University (CPSU) and Cal Poly Corporation (CPC), jointly Cal Poly, established salary and wage rates paid during the 2020-2021 Fiscal year (July 1 – June 30). Benefits for CPSU Faculty summer and overload work include FICA, SUI, and Workers Compensation are calculated at the DHHS pooled rate of 9.2%. Cal Poly's federally negotiated indirect rate is 38.5% of Modified Total Direct Costs, effective July 1, 2020. Modified total direct costs exclude equipment, capital expenditures, charges for patient care, tuition remission, rental costs of off-site facilities, scholarships, and fellowships, participant support costs, and the portion of each subaward in excess of $25,000. The rates in effect at the time the work is performed will be charged to the sponsor.

Timeline of Deliverables

10/15/2021 – decision notification

11/15/2021 – move HiGHS into submodule and upgrade to latest version

12/15/2021 – wrap HiGHS MIP solver using Cython

1/15/2022 – create Python interface scipy.optimize.intlinprog; draft documentation, tests, and benchmarks

2/15/2022 – polish interface, documentation, tests, and benchmarks

3/15/2022 – submit final report

Project Team

Matt Haberland, Nicholas McKibben

I agree to submit a grant report-back if my proposal is selected for funding.

I agree.