HiGHS Newsletter 26.0

This newsletter announces the upcoming third HiGHS workshop, discusses our new interior point solver HiPO, developments with the MIP solver, and introduces a new member of the team

The third HiGHS workshop

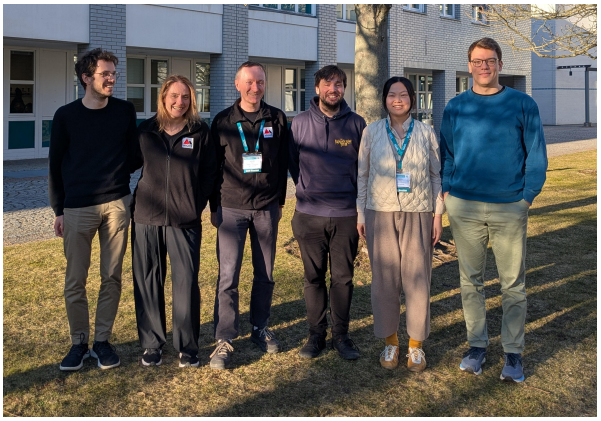
The third HiGHS workshop will take place in Edinburgh on Monday June 1, 2026: the start of the week of SIAM Conference on Optimization (OP26). The workshop will collocate with JuMP-dev 2026, which will be held Sunday, May 31 and Monday June 1, 2026. Some of the industrial participants in the first two workshops will be returning, and registration is via workshop26.highs.dev.

HiPO

Filippo Zanetti has developed a "classical" interior point method (IPM) solver HiPO to complement our existing IPM solver (IPX) for linear programming (LP) problems. IPX is based on iterative solution of the normal equations and, although it can be very fast and accurate, its performance is unpredictable and can be unacceptably slow for very large problems. IPX is also single-threaded, and cannot be extended to solve quadratic programming (QP) or other problems amenable to IPM.

The major task for Filippo in developing HiPO was to write code for multi-threaded factorization of the (quasi-definite) augmented system or (positive definite) normal equations, since there are cases when solving one or the other is strongly preferable. HiPO was first available in HiGHS v1.12 and, from v1.13, was provided with the source code of the underlying Metis, AMD and RCM algorithms required to reorder the matrix of coefficients with the aim of maintaining sparsity and creating the scope for parallelism. A consequence of this is that if HiGHS is built with HiPO, its license is Apache, rather than MIT.

In February 2026, Filippo extended HiPO to solve QP problems, and this facility was made available when HiGHS v1.14 was released in early April.



The MIP solver

Mark Turner has been working hard on multi-threading the primal heuristics and tree search in the mixed-integer programming (MIP) solver. His prototype code offers significant speed-up on problems where these components dominate the solution time. Review and testing of the code has now begun, and it will be made available in HiGHS v1.15.

HiPDLP

Yanyu Zhou has been writing a GPU-accelerated primal-dual hybrid gradient first order solver for LP. Although HiGHS has offered the cuPDLP-C solver for the past two years, research into PDLP algorithms is very active. Yanyu is taking the best of this in creating her solver (HiPDLP), which will run entirely on a GPU. HiPDLP will replace cuPDLP-C in HiGHS v1.15. This work is funded by a contract with the National Energy System Operator (NESO).

Funding

We are pleased to announce that through NumFOCUS, and in collaboration with JuMP, we have received a large grant from the Breakthrough Energy Foundation to improve the performance of JuMP and HiGHS on open energy models.

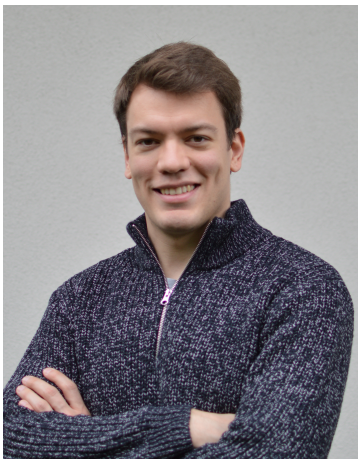
Work in progress

Currently it is not possible to use HiPO or the GPU-accelerated LP solver via the Python interface. However, Ivet Galabova has made a lot of progress on the necessary build system changes, which are also affected by HiPO requiring the Apache license, and we expect HiPO and the GPU-accelerated LP solver to be available via PyPI.

In collaboration with MathWorks, work on improving the irreducible infeasibility system detection facility in HiGHS has begun. Currently the focus is only LP, but a MIP facility is expected to follow.

Thanks to an external contributor, HiGHS will soon be able to handle indicator constraints. Current refactoring will also enable HiGHS to solve problems with special ordered sets. As with indicator constraints, initially this will require the problem to be reformulated as a vanilla MIP, which will limit the instances that can be solved.

Giovanni Ghisalberti



From Bergamo, Italy, Giovanni Ghisalberti has a BSc in Aerospace Engineering from TU Delft, and an MSc in Mathematics from the University of Amsterdam. In January 2026, he began a PhD at the University of Edinburgh under the supervision of Julian Hall. Giovanni is funded by MathWorks, and will focus on solution techniques for large-scale QP problems.