Optimization solvers: the missing link for a fully open-source energy system modelling ecosystem

Maximilian Parzen[1,*], Julian Hall[2], Jesse Jenkins[3], Tom Brown[4]

[1] Director of PyPSA meets Earth, PhD candidate at the Institute of Energy Systems, University of Edinburgh

[2] Founder of HiGHS, Reader at the School of Mathematics, University of Edinburgh

[3] Founder of GenX, Assistant Professor at the Andlinger Center for Energy and the Enviroment, Princton University

[4] Co-Founder of PyPSA, Professor for Digital Transformation in Energy System, Technical University Berlin

Required funding for open source HiGHS solver developments:
$141k for 12 months

Dear Colleagues,

The International Energy Agency recently estimated that the world would need to triple annual clean energy investment by 2030 to around $4 trillion if it wants to reach net zero emissions by 2050. Ensuring that this energy transition takes place in an orderly way, while maintaining reliability and affordability of supply, is a daunting challenge. Modelling energy systems in software tools can play a key role in planning this transition and helping to avoid misallocation of investment.

Energy system planning requires three things: data, models and solvers. The data is used as input for mathematical models that aim to describe the energy system. These mathematical models are typically formulated as optimization problems, which aim to reduce total system costs given the physical constraints in the system. Once the problem is formulated, it is passed to a solver to find the optimal solution. In the past, energy system modelling was often done behind closed doors, with closed data and proprietary software. This made it very hard to scrutinise the results or build on previous work. Recently, motivated by the need for transparency and a wish to share state-of-the-art tools, there has been a blooming of open energy system data and modelling tools, such as calliope, GenX, oemof, OSeMOSYS, PyPSA, Switch, TEMOA and TIMES to name but a few.

The world of high performance linear optimization solvers is dominated by commercial closed-source software. Licenses for leading commercial solvers typically cost tens of thousands of dollars per year. Many stakeholders outside of academic and corporate settings — including many non-governmental organizations, early-stage companies, and government ministries in lower income countries — simply cannot afford these tools, which is a problem if we aim to tackle climate change globally. Quality open-source solvers exist, but their performance relative to commercial software can make them prohibitively uncompetitive. HiGHS, who is disrupting the open-source solver landscape and leading performance benchmarks wants to change that.

The aim of this document is to motivate potential funders to support the HiGHS open-source solver team. Recent breakthroughs by their team can lead to widespread adoption of efficient and highly detailed energy planning tools that is urgently needed for better decision-making. However, as set out in the following pages by Julian Hall (Reader and leader of the HiGHS Team) significant computational inefficiencies remain, particularly when solving larger instances. To explore and overcome these inefficiencies, ***$141k* funding** is required for the **next 12 months**. Genuinely welcome is also **long-term funding for sustainable open source solver developments**.

Best wishes,

Maximilian Parzen, Julian Hall, Jesse Jenkins and Tom Brown

[1] International Energy Agency (IEA). Net Zero by 2050: A Roadmap for the Global Energy Sector, 2021.

*Corresponding author: max.parzen@ed.ac.uk

Developing HiGHS: the future of high performance open-source linear optimization software

The world of high performance linear optimization solvers is dominated by commercial closed-source software. Licenses for leading commercial solvers are too expensive for many stakeholders outside of academic and corporate settings, including non-governmental organizations, early-stage companies, and government ministries in lower income countries. Without open-source software of comparable performance, these stakeholders are unable to solve the analysis and planning models required for the efficient management of critical resources such as transport, water, energy and health services. Quality open-source solvers exist, but their performance relative to commercial software can make them prohibitively uncompetitive. Commercial solvers have been developed to such a degree that the experience and investment of time required to write an open-source software of comparable performance is enormous, stifling the development and implementation of new solvers.

A recent benchmark of open source solvers on the energy system linear programming (LP) model PyPSA [6, 7] reveals the performance status. An interior point method (rather than the simplex method) is frequently the appropriate solution technique for very large-scale LP problems, including energy transition planning problems. For small and modest sized instances, the performance of HiGHS relative to the popular open-source solvers (GLPK and Cbc), and the leading commercial solver Gurobi, is illustrated in Figure 1. It is seen that GLPK and Cbc are soon wholly uncompetitive with HiGHS, whose performance is comparable with that of Gurobi.

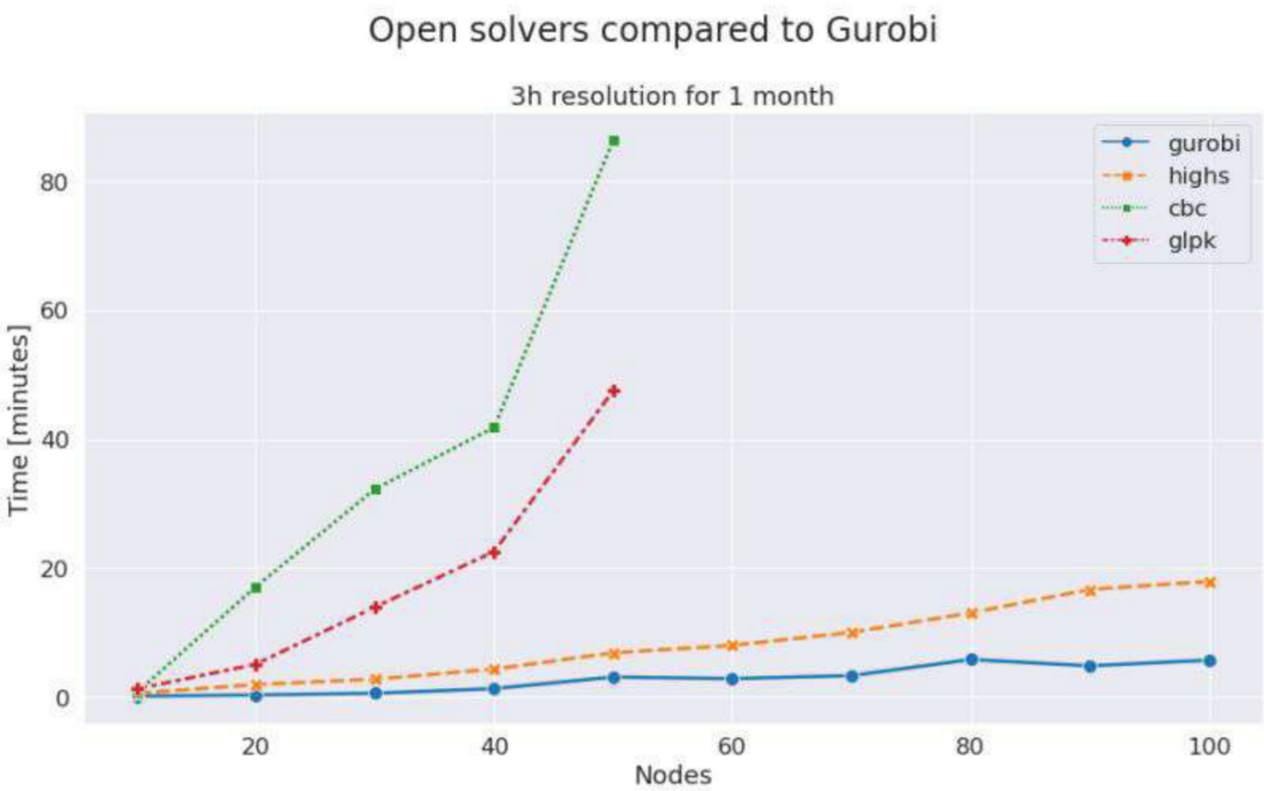

Figure 1: Comparison of open source solvers HiGHS, GLPK and Cbc to commercial solver Gurobi on small and modest sized PyPSA models

While the benchmark shows promising results for HiGHS, problems of practical interest are larger. For these, as illustrated in Table 1, even the current interior point implementation in HiGHS can be unacceptably slow relative to Gurobi (60-100 times slower). Considering that even Gurobi requires a couple of days to solve huge practical problems illustrates that the use of current open

				IPM iterations		Solution time (s)	
Model	Variables	Constraints	Nonzeros	`Gurobi`	`HiGHS`	`Gurobi`	`HiGHS`
pypsa-eur-240t5n	56899	119299	237928	48	49	4	21
pypsa-eur-sec-45t45n	135036	277932	645092	50	76	37	145
pypsa-eur-240t100n	344857	727227	1547449	49	77	521	1148
pypsa-eur-sec-350t45n	1015876	2120132	4916980	100	149	3020	26469
pypsa-eur-8760t5n	1077534	2242566	4393840	143	99	180	11363

Table 1: Comparison of `HiGHS` and `Gurobi` on PyPSA-Eur [6] and PyPSA-Eur-Sec [7] energy system models. “t” refers to timesteps that indicate the temporal resolution and “t” refers to nodes that indicate the spatial resolution.

source solvers is totally impractical. The aim of this proposal is to identify key solver enhancements in the short, medium and long term that will help to bridge the significant gap in performance between `HiGHS` and the major commercial solvers.

What is `HiGHS`?

`HiGHS` is a suite of high performance open-source linear optimization solvers that is currently being developed by Hall, Galabova, Gottwald and Feldmeier in the School of Mathematics at the University of Edinburgh. It has simplex and interior point solvers for linear programming (LP) as well as solvers for mixed-integer programming (MIP) and quadratic programming (QP). Combining exceptionally high quality solvers, written by brilliant young researchers, it is clear from Mittelmann’s industry standard benchmarking results [1, 2, 9] that there is no other remotely comparable open-source software project in the world.
Consultancy work by Hall and Galabova funded much of the development of `HiGHS` and, now that it has attained its prominent position in the open-source world, the School of Mathematics has recently recognised the value of `HiGHS` by employing its two key team members (Galabova and Gottwald). This has been done in the expectation of securing further funding. The scope for future commercial funding, via paid support, advice on using `HiGHS` and bespoke solver development, is likely to be in the area of MIP. Whilst any resulting solver developments will remain open-source, they will favour the MIP solver. The commercial market for funding development of the `HiGHS` LP and QP solvers is limited. So, to achieve the performance gains that this proposal seeks, and before the `HiGHS` team resorts to commercial activities to secure funding, there is a window of opportunity for non-commercial funding of `HiGHS` to drive solver development for the general good.

The 3 phase plan. Enhancing the `HiGHS` performance

`HiGHS` has an interior point solver for LP problems that is seen in the Mittelmann benchmarking results to be significantly better than any other open-source solver. This is accounted for by its novel approach to the numerical linear algebra requirements of implementing an interior point method. However, inefficiencies still remain, and there are also occasions when the traditional numerical linear algebra approach is sure to be preferable. Enhancing the `HiGHS` interior point solver by addressing these two deficiencies offers short and medium term opportunities to enable `HiGHS` to be applied successfully to larger and more challenging LP models.

Interior point solvers

When an interior point method (IPM), rather than the simplex method, is used to solve an LP problem

$$\text{maximize } \boldsymbol{c}^T\boldsymbol{x} \quad \text{subject to } A\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{x} \geq \boldsymbol{0},$$

an IPM solver moves iteratively towards an optimal solution of the LP by computing a sequence of search directions given by the *Newton equations* $G\boldsymbol{\delta} = \boldsymbol{d}$. The matrix of coefficients is given by $G = A\Theta A^T$, where Θ is a diagonal matrix that changes in each iteration, and A is the constraint matrix in the LP. For large practical problems, most of the entries in the matrix A are zero and, as such, it is said to be sparse. The dominant computational cost of IPM solvers is the solution of the Newton equations, and their efficiency depends on exploiting the sparsity of A. Traditional IPM implementations determine a Cholesky decomposition $G = LL^T$ which is then used to solve $G\boldsymbol{\delta} = \boldsymbol{d}$ directly. However, without sophisticated techniques, the sparsity in A is lost in forming L. Indeed, if L is largely nonzero, insufficient memory may be available to compute it. One particular property of A that must be handled carefully is the existence of *dense columns*. For instance, if A has a single column with no zero entries, the matrix G has no zero entries to be exploited when forming the Cholesky decomposition. Popular open-source solvers (but not `HiGHS`) are based on a Cholesky decomposition, and an inability to handle dense columns efficiently is an obvious explanation for their terrible performance when applied to LPs, such as the `PyPSA` model benchmark cases presented above, that have relatively small numbers of dense columns.

The `HiGHS` interior point solver (IPX)

The `HiGHS` interior point solver (IPX) was written in Edinburgh by Lukas Schork under the supervision of Jacek Gondzio. Lukas is now employed by a commercial optimization software company, so is not available to work on IPX. IPX avoids the challenges of solving the Newton equations directly (via a Cholesky decomposition) by solving them iteratively. It uses a standard approach, but a novel and complex preconditioning technique [8] is required. Calculation of the preconditioner by IPX is carried out by performing simplex-like iterations, requiring the solution of many related linear systems of the form $B\boldsymbol{z} = \boldsymbol{r}$. Although these are of the same dimension as the Newton equations, the coefficient matrix B is a square submatrix of A, rather than $A\Theta A^T$, making the systems vastly easier to solve. The solution of linear systems of the form $B\boldsymbol{z} = \boldsymbol{r}$ is also required to apply the preconditioner when using it to solve the Newton equations $G\boldsymbol{\delta} = \boldsymbol{d}$ iteratively. `HiGHS` is comparable with `Gurobi` on 23% of the Mittelmann benchmark LPs, and solves 53% within a factor of ten of the time for `Gurobi`. For the `PyPSA` LPs, `HiGHS` solves many within ten times the time of `Gurobi`. However, for some of the benchmark and `PyPSA` LPs, `HiGHS` is totally uncompetitive with `Gurobi`.

Phase 1: Enhancing the core numerical linear algebra system of IPX

For LPs such as the large-scale energy system models, the solution time when using IPX is dominated by time spent solving simplex-like linear systems $B\boldsymbol{z} = \boldsymbol{r}$. These are notable for the right-hand-side $\boldsymbol{r}$ being a sparse vector. Remarkably, for many LP problems the solution $\boldsymbol{z}$ is sparse. This phenomenon of *hyper-sparsity* was first noted and exploited in the simplex method for LP by Hall and McKinnon [4].

IPX currently solves the systems $B\boldsymbol{z} = \boldsymbol{r}$ by using its own sparse numerical linear algebra software (`basiclu`). This always assumes hyper-sparsity, but the techniques that exploit it are inefficient in cases where $\boldsymbol{z}$ is not sparse. Analysis of IPX when applied to the `PyPSA` energy system models has identified that this is the case for most vectors $\boldsymbol{z}$. This inefficiency is best avoided by using the simplex-specific `HiGHS` numerical linear algebra software (`HFactor`) rather than `basiclu`, since the former also has code for solving standard sparse systems. Indeed, `HFactor` is a fundamentally more sophisticated sparse numerical linear algebra system, and may be universally superior to `basiclu`. It is estimated that using `HFactor` rather than `basiclu` will improve the speed of IPX by a factor of around 2–4.

Changing the numerical linear algebra system at the heart of IPX is no easy task and, since Hall is the only person with an understanding of both IPX and `HFactor`, the most cost-effective way

Name/role	Cost
Developer	$46
Gondzio	$27k
Hall	$27k
Galabova	$14k
Total	$114k

Table 2: Stage 2 costs for developing a Cholesky-based direct solver for IPX

of effecting the change is for him to carry out the work. It will require at least ten days which, at Hall's consultancy rate, will cost $27k. The 70% due to Hall will not be taken as personal income, but provide the HiGHS team with much-needed expenses for disseminating their work at European meetings in the summer of 2022.

Phase 2: A Cholesky-based direct solver for IPX

Although the performance improvement for IPX that will be achieved by using a more efficient core numerical linear algebra system is significant, there will remain many LP problems for which IPX is wholly uncompetitive with major commercial solvers. A medium-term objective for HiGHS is to develop, over two years, a traditional Cholesky-based direct solver for the Newton equations $G\boldsymbol{\delta} = \boldsymbol{d}$. This requires highly efficient sparsity-exploiting code for Cholesky decomposition. However, even if licensing conditions were favourable, an off-the-shelf Cholesky code will not suffice. Due to the wide range of values in the matrix Θ, the system $G\boldsymbol{\delta} = \boldsymbol{d}$ is ill-conditioned, and handling this requires modifications to the Cholesky decomposition. The difference in the accuracy of solutions obtained by direct rather than iterative methods has implications elsewhere in an IPM solver.

Writing a Cholesky-based direct solver for IPX, with techniques to handle dense columns, will take twelve months for a doctoral student with a strong background in numerical linear algebra and well developed skills in large scale C++ programming. Any significant change to the HiGHS codebase has inevitable overhead costs for the team, so will place demands on the time of Hall and Galabova. Jacek Gondzio has decades of experience in the field of computational techniques for IPM and, on a consultancy basis, will provide essential advice. A budget for this work is set out in Table . Note that the cost of the developer is based on 12 months at the University of Edinburgh pre-doctoral level (UE06), plus the overheads taken by the University in any consultancy role. Developing a high performance open-source Cholesky-based IPM solver for LP problems with a performance comparable to the best commercial solvers is valuable in its own right. However, this work also opens the door for the development of interior point based techniques for solving convex quadratic programming (QP) problems and convex cone programming. Enhancement in this direction feeds into the longer term vision of HiGHS set out below.

Phase 3: A longer-term vision for HiGHS

It is a truism in solver development that improvements in performance lead directly to modellers formulating ever more challenging problems. Although there will be incremental improvements in the performance of the existing solvers in HiGHS, a longer-term vision is to develop new solvers within HiGHS that will open the door to solving otherwise intractable problems. However attractive it might be to follow funding aimed at developing an advanced solver for a few restricted classes of problems, as the only general high performance open-source linear optimization software being actively developed in the world, HiGHS has a responsibility to develop in ways that will be of the most general good.

The long-term aim of HiGHS is twofold: to improve the strength and breadth of its solvers and become resilient to any personnel changes. There are three areas for solver development: to enable its mixed-integer programming (MIP) solver to exploit fully the presence of widely-used modelling sub-structures, to enable the solution of linear programming (LP) models via decomposition that would be intractable if solved directly, and to extend the HiGHS quadratic programming (QP) facility to mixed-integer quadratic programming and nonlinear programming (NLP). This would facilitate the opportunity for community development of decomposition-based MIP solvers and mixed-integer nonlinear programming (MINLP) solvers within a single environment.

Open-source projects are notorious for being under-staffed, leading to deficiencies in areas such as software engineering and documentation. COIN-OR and many projects within it are prime examples. Many projects, even hugely valuable ones such as Clp and Cbc, are single-person efforts that become abandoned when that individual changes occupation or interests, or simply grows old. HiGHS is largely the work of young people who must be retained if possible. Expansion of the team cannot just be driven by the development of new solvers. Recruitment to enable the management of a growing project and reinforce its resilience will be increasingly important.

The HiGHS MIP solver has been the game-changer in terms of the project's reputation in the optimization community. Although LP is important, more MIP models are built, including energy systems optimization problems with discrete decisions such as electricity generator unit commitment problems, and it is much harder to write a general fast solver. Although the HiGHS MIP solver is now the best open-source code, the performance gap between it and the best commercial MIP solver (Gurobi) remains huge. This is because there is a great deal more scope for algorithmic development within MIP solution than within continuous optimization. MIP solver development is conducive to teamwork, whereas work on a continuous optimization solver is more of a solo activity. The combinatorial nature of MIP means that large instances in some problem classes are intractable without specialist techniques and the ability of the solver to recognise what spectrum of techniques is best for a particular instance. Only by increasing the range of specialist techniques within the HiGHS MIP solver can it achieve a general performance comparable with the commercial solvers.

Solution of huge-scale LP problems with amenable structure by exploiting decomposition techniques has been around for decades, and there are two common structures inherent in many models. Developing a facility within HiGHS to exploit row-linked block angular (Dantzig-Wolfe) and column-linked block angular (Benders) structure will enable it to solve LPs that would, otherwise, be intractable. Open-source frameworks for solution via decomposition abound, but they assume the use of either commercial or inferior open-source software to solve LP subproblems. The advantage of building decomposition-based solvers within HiGHS is that the subproblem solvers are of high quality and can be tuned and modified to enhance the performance of the top level solver.

Although many practical problems are inherently linear or can be well modelled via linearisations, in some areas the nonlinear behaviour of essential features cannot be ignored. The ability of HiGHS to solve QP problems, either with the active set technique used in its current solver, or with IPM based techniques envisioned above, opens the door to the solution of nonlinear programming (NLP) problems via sequential quadratic programming. As with decomposition, open source NLP solvers exist, but subproblem solutions are found using commercial or inferior open-source software. Hence developing a superior NLP solver is naturally part of the long-term vision for HiGHS.

Resources to support the long-term future of HiGHS are expected to come from various sources. The School of Mathematics sees a value in supporting HiGHS for reputational benefits and the potential for recovering some of its investment via income generation. HiGHS can clearly be monetized by providing support and bespoke solver developments to commercial organizations. It has also been suggested that a premium version could be licensed whilst ensuring that HiGHS continues to offer the world's best open-source optimization software. However, with such funding avenues comes a danger of being drawn into short-term commercial exploitation of what exists, rather than developing stronger and broader solver offerings for the common good. It is clear that, as recognition of HiGHS pre-eminent position in open-source software grows, it will be a magnet for external community activity. Whilst this will bring expert developer resources for free, it adds to the local management requirements and exposes HiGHS to the vulnerability associated with volunteer effort. Major philanthropic investment in HiGHS is a clear route to ensuring that it maintains the position that it has established, delivers on its potential for growth and continues to serve the common good. To realise the complete vision set out above would require an investment of millions of dollars over a decade. With lower levels of investment, the development of HiGHS would have to be more closely targeted. Investment for the common good would also reflect the philosophy of the current members of the HiGHS team. The team is not looking to make personal fortunes, and HiGHS is blessed by the fact that its current team members are happy to work on such an exciting project within the secure and fertile environment of the academic world.

Specific use of donations

The following examples illustrate how different levels of funding could help HiGHS achieve its longer-term vision.

- $25k funds the annual computer hardware investment for development benchmarking
- $50k for four years funds a PhD position in the HiGHS team
- $75k funds a post-doctoral developer position for a year

Donation and sponsorship information

The proposal offers two financial support options for the HiGHS developments. The first is via the Linux Foundation Crowdfunding (LFX) platform. This allows tax-deductible gifts in Canada, UK and the US. The tax deduction on the LFX platform is possible since the fund collector, the University of Edinburgh, is a registered charitable body in these countries [5, 10]. We recommend this path for donors that want to support the development team with unconstrained funding of any scale.

Second option, are sponsorship agreements managed by the University of Edinburgh. These agreements are available for substantial gifts of at least $75k, and can contain contractual conditions and benefits. Depending on the bi-lateral negotiated benefits, these gifts can be either tax-deductible or non tax-deductible [3].

References

[1] Benchmark of Barrier LP solvers. http://plato.asu.edu/ftp/lpbar.html, 2022. Accessed: 15/02/2022.

[2] Benchmark of Simplex LP solvers. http://plato.asu.edu/ftp/lpsimp.html, 2022. Accessed: 15/02/2022.

[3] Gov.UK. Sponsorship guidance (VAT Notice 701/41). https://www.gov.uk/guidance/sponsorship-and-vat-notice-70141. Accessed: 26/03/2022.

[4] J. A. J. Hall and K. I. M. McKinnon. Hyper-sparsity in the revised simplex method and how to exploit it. *Computational Optimization and Applications*, 32(3):259–283, December 2005.

[5] LFX Community Forum. Are LFX donations tax-deductible in case they are collected by a non-profit. https://bit.ly/36PxLA3. Accessed: 26/03/2022.

[6] PyPSA-Eur: An Open Optimisation Model of the European Transmission System. https://github.com/PyPSA/pypsa-eur, 2021. Accessed: 02/02/2022.

[7] PyPSA-Eur-Sec: A Sector-Coupled Open Optimisation Model of the European Energy System. https://github.com/PyPSA/pypsa-eur-sec, 2021. Accessed: 02/02/2022.

[8] L. Schork and J. Gondzio. Implementation of an interior point method with basis preconditioning. *Mathematical Programming Computation*, pages 1–33, 2020.

[9] The MIPLIB2017 Benchmark Instances. http://plato.asu.edu/ftp/milp.html, 2022. Accessed: 15/02/2022.

[10] University of Edinburgh. Donate. https://www.ed.ac.uk/donate. Accessed: 26/03/2022.