SN Operations Research Forum

MIPLIBing: Seamless benchmarking of mathematical optimization problems and metadata extensions

--Manuscript Draft--

Manuscript Number:	ORFO-D-20-00026R2
Full Title:	MIPLIBing: Seamless benchmarking of mathematical optimization problems and metadata extensions
Article Type:	Short Communication
Funding Information:	
Abstract:	Public libraries of problems such as MIPLIB are fundamental to creating a common benchmark for measuring algorithmic advances across mathematical optimization solvers. They also often provide metadata on problem structure, hardness with respect to state-of-the-art solvers, and solutions with the best objective function value on record. In this short paper, we discuss some ways in which such metadata can be leveraged to create a seamless testing experience. In particular, we present MIPLIBing: a Python library that automatically downloads queried subsets from the current versions of MIPLIB, MINLPLib, and QPLIB, provides a centralized local cache across projects, and tracks the best solution values and bounds on record for each problem. While inspired by similar use cases from other areas, we reflect on the specific needs of mathematical optimization and discuss opportunities to extend benchmark sets to facilitate experimentation with different model structures.
Corresponding Author:	Thiago Serra Bucknell University UNITED STATES
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Bucknell University
Corresponding Author's Secondary Institution:	
First Author:	Thiago Serra
First Author Secondary Information:	
Order of Authors:	Thiago Serra
	Ryan J. O'Neil
Order of Authors Secondary Information:	
Author Comments:	Dear editors of SN Operations Research Forum, We would like to bring to you attention our revised manuscript, which is short communication describing a new tool to manipulate problems libraries such as MINLPLib, MIPLIB, and QPLIB. Sincerely, The authors
Response to Reviewers:	Dear reviewers, We have addressed your final recommendations for publishing the paper. Thank you for your feedback on the manuscript and on how to improve the tool that we developed. Sincerely, The authors

Noname manuscript No.
(will be inserted by the editor)

MIPLIBing: Seamless benchmarking of mathematical optimization problems and metadata extensions

Thiago Serra · Ryan J. O'Neil

Received: date / Accepted: date

Abstract Public libraries of problems such as MIPLIB are fundamental to creating a common benchmark for measuring algorithmic advances across mathematical optimization solvers. They also often provide metadata on problem structure, hardness with respect to state-of-the-art solvers, and solutions with the best objective function value on record. In this short paper, we discuss some ways in which such metadata can be leveraged to create a seamless testing experience. In particular, we present `MIPLIBing`: a Python library that automatically downloads queried subsets from the current versions of MIPLIB, MINLPLib, and QPLIB, provides a centralized local cache across projects, and tracks the best solution values and bounds on record for each problem. While inspired by similar use cases from other areas, we reflect on the specific needs of mathematical optimization and discuss opportunities to extend benchmark sets to facilitate experimentation with different model structures.

Keywords Benchmarking · Mathematical optimization · Problem libraries

1 Introduction

Mixed Integer Programming (MIP) is an enormously successful general-purpose modeling and optimization tool. A vast array of practical problems can be formulated as Mixed Integer Linear Programs (MILPs), one of the most fundamental forms of MIP, which take the form

$$\max \ \{cx \mid Ax \leq b, x \in \mathbb{Z}^p \times \mathbb{R}^q\}.$$

T. Serra
Bucknell University
E-mail: thiago.serra@bucknell.edu

R. J. O'Neil
nextmv
E-mail: ryan@nextmv.io

Wide interest in improving MIP technology has led to significant algorithmic advances [4,14,15,6,17]. In fact, an even broader array of practical problems can be formulated if the objective function and the constraints are nonlinear and able to represent other forms of discontinuity.

In the last decades, the software speedups obtained by such advances have at times outpaced the already impressive hardware speedups of modern computers [4]. Hence, although many practical problems belong to the NP-hard class and are therefore theoretically intractable at scale, modern solvers can frequently solve them at the sizes required by industry.

Part of this success is undoubtedly due to open, standard benchmark instances gathered from real-world problems and shared by the community, which have been used to evaluate solvers in public benchmarks [18]. These tests sets are fundamental to the study and advancement of solver performance. One of the earliest benchmarks was Netlib [10], a collection of Linear Programs (LPs) that could be queried and distributed by email in 1985.

The Mixed Integer Programming Library (MIPLIB) is a collection of MILPs that uses the same MPS file format and file compression to reduce transfer size as Netlib. The first version of MIPLIB appeared in 1992 [2]. Since then, updates in 1996 [3], 2003 [1], 2010 [13], and 2018 [11] included new instances and removed those that are no longer regarded as difficult. The 2010 version of MIPLIB compiled problems through a committee involving participants from both academia and industry for the first time. The latest version, MIPLIB 2017, uses a data-driven approach to problem selection. Its companion website filters and sorts problem instances by their dimensions, current status, and structure.

For the past couple of decades, we have seen the introduction of many libraries generalizing the class of formulated problems to the conic and nonlinear cases. MINLPLib has been introduced in 2001 [5] with features such as an anonymizer of instances from industrial applications, which uses primarily the GMS format. MINLPLib has been recently reorganized with the continuous inclusion of new problems since 2014 [25]. The Conic Benchmark Library (CBLIB) has been introduced in 2012 by aggregating known instances in the literature using the CBF format that was also proposed [9]. Like MINLPLib, CBLIB also follows the paradigm of continuous inclusion of instances. After an initial call for quadratic problems in 2014, QPLIB was released in 2017 as a collection of problems focused on the cases of quadratic terms in the objective function or the constraints [9]. QPLIB adopts a namesake file format, but also distributes instances in a few other formats like MINLPLib.

We propose augmenting the user experience with such problem libraries by providing a software library to simplify access to test instances. We also examine the ways in which other communities share test data, and we suggest strategies to further advancement and collaboration. In particular, we present `MIPLIBing`, a library that creates a seamless testing and benchmarking experience for mathematical optimization solvers.

Our purpose is twofold:

(i) Showcase an extensible tool for testing problems from the current versions of MIPLIB, MINLPLib, and QPLIB; and
(ii) Foster discussion on how we can continue to improve benchmarks with open problem metadata and software.

2 `MIPLIBing`

`MIPLIBing` is a Python library that can be used to browse the MIPLIB, MINLPLib, and QPLIB problem libraries. This software library is currently hosted at `https://github.com /thserra/MIPLIBing` and provides the following features:

(1) A method to search for problems according to their status, minimum or maximum numbers of variables of different types, constraints, non-zeroes, and other characteristics that may be relevant to some or all the libraries;
(2) Downloading and caching of problem files in one of the formats supported by the problem library to a local folder in the machine; and
(3) Local access to the relevant problem metadata, characteristics, and the value of the best-known solution.

The purpose of `MIPLIBing` is to enhance the user experience with such libraries by providing a systematic way to select problems for computational experiments, while simplifying the associated management of files and data. We avoid the dichotomy between downloading the entire set of problems and one problem at a time, and facilitate reproducing experiments at machines in which those problems are not immediately available. By centralizing problem caching, we also avoid unnecessary replication of files across projects. We acknowledge that selectively choosing instances for benchmarking can be controversial [18], but nevertheless relevant if our focus is on a specific type of problem or limited to problems of a given size due to the complexity of the algorithms.

3 Discussion

We can think of `MIPLIBing` as a client-side complement to the server-side files describing problems and website metadata that is currently offered by problem libraries. We do note that MIPLIB also provides code to evaluate its collection of problems on certain solvers and test the feasibility of a new solution. Likewise, MINLPLib and CBLIB offer client-side code to query instances, but without an option for selectively downloading them. CBLIB also facilitates benchmark execution with CPLEX and MOSEK using locally downloaded files. However, to the best of our knowledge, `MIPLIBing` is the first software library that selectively downloads instances from problem libraries. In fact, we argue that there is much to be gained by thinking of problem libraries as centralized providers of metadata, which could potentially be explored in a multitude of ways by the mathematical optimization community.

3.1 Streamlining Access to Problems

There are similar examples in the machine learning community: TensorFlow supports the download of a number of pattern-recognition data sets involving audio, image, and many other media formats [24]. This facilitates accessing such data to simplify benchmarking software libraries and educating users. This successful approach has been implemented by a number of other projects, including JuliaStats [7], statsmodels [22], pandas [16], and scikit-learn [19].

Rather than thinking of each of those mathematical optimization libraries as a single benchmark to measure every method that is designed for that category of problem, we can regard subsets of problems with a certain special structure as the starting point to gain new algorithmic insights [12] or to learn how optimization algorithms work by experimenting with a limited set of problems. That perspective could be strengthened by historically enriching those libraries with structured metadata on when each problem was formulated and how it progressed in time from challenging to hard and then easy to solve.

3.2 Augmenting Problem Data

A more ambitious direction to enrich such libraries would be the inclusion of more problem domain data, possible formulations, or structural information. To a certain extent, libraries for specific problems achieve the former by prioritizing domain knowledge over formulation, such as TSPLIB [20], the DIMACS clique benchmark set [8], and the Grubhub MDRP instances [21]. These allow the modeler to experiment with a variety of formulations and solvers. As mentioned by one of the reviewers, MIPLIB instances are "hidden in plain sight" due to the missing domain information. However, some of these instances have been reformulated for benchmarking purposes, such as by removing a continuous variable serving as a proxy for the objective function to generate additional 0–1 instances [23]. More generally, the identification of structural information has been successfully achieved by the use of tags in MIPLIB, which are used to identify constraints such as knapsack, set covering, cardinality etc.

Incorporating more structured problem data into future problem libraries would allow testing of both general-purpose solvers and alternative formulations of specific problems. This would streamline research and development of new models, while coordinating it with the advancement of MIP solvers.

3.3 Future Work

Based on the feedback received so far, we are currently considering the following directions for extending the scope and functionalities of `MIPLIBing`:

- Offer a command-line version of `MIPLIBing`;
- Maintain files in compressed format in the local machine;

– Limit download to a predefined number of instances when a query returns a large number of instances;
– Download solution files associated with the best solutions on record when they are available; and
– Extend `MIPLIBing` to other problem libraries, such as CBLIB[1].

Acknowledgement: We thank the anonymous reviewers for their feedback and suggestion to improve this manuscript and expand the scope of `MIPLIBing`. We also thank Mihai Banciu for feedback and suggestions on `MIPLIBing`.

Funding: Nothing to declare.

Conflict of Interest: The authors declare that they have no conflict of interest.

References

1. Achterberg, T., Koch, T., Martin, A.: MIPLIB 2003. Operations Research Letters **34**(4) (2006)
2. Bixby, R., Boyd, E., Indovina, R.: MIPLIB: A test set of mixed integer programming problems. SIAM News **25** (1992)
3. Bixby, R., Ceria, S., McZeal, C., Savelsbergh, M.: An updated mixed integer programming library: MIPLIB 3.0. Optima **58** (1998)
4. Bixby, R.E.: Solving real-world linear programs: A decade and more of progress. Operations Research **50**(1) (2002)
5. Bussieck, M., Drud, A., Meeraus, A.: MINLPLib—a collection of test models for mixed-integer nonlinear programming. INFORMS Journal on Computing **15**(1), 114–119 (2003)
6. Conforti, M., Cornuéjols, G., Zambelli, G.: Integer Programming. Springer (2014)
7. contributors, J.: Juliastats. `https://juliastats.org/` (2014)
8. DIMACS: Clique benchmark instances. `https://turing.cs.hbg.psu.edu/txn131/clique.html`
9. Friberg, H.A.: CBLIB 2014: a benchmark library for conic mixed-integer and continuous optimization. Mathematical Programming Computation **8**(2), 191–214 (2016)
10. Gay, D.M.: Electronic mail distribution of linear programming test problems. Mathematical Programming Society Committee on Algorithms Newsletter (COAL) **13**, 10–12 (1985)
11. Gleixner, A., Hendel, G., Gamrath, G., Achterberg, T., Bastubbe, M., Berthold, T., Christophel, P.M., Jarck, K., Koch, T., Linderoth, J., Lübbecke, M., Mittelmann, H.D., Ozyurt, D., Ralphs, T.K., Salvagnin, D., Shinano, Y.: MIPLIB 2017: Data-driven compilation of the 6th mixed-integer programming library (2019)
12. Hooker, J.N.: Testing heuristics: We have it all wrong. Journal of Heuristics **1** (1995)
13. Koch, T., Achterberg, T., Andersen, E., Bastert, O., Berthold, T., Bixby, R.E., Danna, E., Gamrath, G., Gleixner, A.M., Heinz, S., Lodi, A., Mittelmann, H., Ralphs, T., Salvagnin, D., Steffy, D.E., Wolter, K.: MIPLIB 2010. Mathematical Programming Computation **3** (2011)
14. Laundy, R., Perregaard, M., Tavares, G., Tipi, H., Vazacopoulos, A.: Solving hard mixed-integer programming problems with Xpress-MP: A MIPLIB 2003 case study. INFORMS Journal on Computing **21**(2) (2009)

[1] The current release of `MIPLIBing` relies on website data to query instances without downloading them. We would only be able to inlcude CBLIB if the same type of information is aggregated in their website.

15. Lodi, A.: Mixed integer programming computation. In: M. Jünger, T. Liebling, D. Naddef, G. Nemhauser, W. Pulleyblank, G. Reinelt, G. Rinaldi, L. Wolsey (eds.) 50 Years of Integer Programming 1958-2008 (2010)
16. McKinney, W., et al.: pandas: a foundational python library for data analysis and statistics. Python for High Performance and Scientific Computing **14**(9) (2011)
17. Mittelmann, H.: Benchmarks for optimization software (2019). URL `http://plato.asu.edu/bench.html`. Accessed: 2019-12-27
18. Mittelmann, H.D.: Benchmarking optimization software - a (hi)story. SN Operations Research Forum **1** (2020)
19. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011)
20. Reinelt, G.: Tsplib—a traveling salesman problem library. ORSA journal on computing **3**(4), 376–384 (1991)
21. Reyes, D., Erera, A., Savelsbergh, M., Sahasrabudhe, S., O'Neil, R.: The meal delivery routing problem. Optimization Online (2018)
22. Seabold, S., Perktold, J.: Statsmodels: Econometric and statistical modeling with python. In: Proceedings of the 9th Python in Science Conference, vol. 57, p. 61. Scipy (2010)
23. Serra, T., Hooker, J.N.: Compact representation of near-optimal integer programming solutions. Mathematical Programming **182**, 199–232 (2020)
24. TensorFlow: Tensorflow datasets (2020). URL `https://www.tensorflow.org/datasets/catalog/overview`. Accessed: 2020-01-02
25. Vigerske, S.: Towards MINLPLib 2.0 model instance collections. International Symposium on Mathematical Programming (ISMP) (2015)

August 13, 2020

Response Letter for "MIPLIBing: Seamless benchmarking of mathematical optimization problems and metadata extensions"

Thiago Serra and Ryan J. O'Neil

Once more, we would like to thank both reviewers for their diligent work and the associate editor for the coordination effort. We have addressed your final recommendations for publishing the paper. You can find more information about how we handled your suggestions below.

Reviewer 1

> *The authors have correctly addressed the comments I made to the manuscript and after accessing their open repository, I can verify that it works as mentioned in it.*
> *Awaiting for the other reviewer's comments, I only have a minor comment regarding the manuscript. There are several hyphens missing in it, specifically in the words best-known, best-documented, and general-purpose.*
> *I look forward to the publication of this manuscript and the following discussion that will arise around it.*

We have hyphenated some of those terms and replaced "best known" with a different wording that best reflects our intended meaning, such as "best solutions on record" in Section 3.3.

Reviewer 2

> *The manuscript presents an interesting and relevant software contribution to the computational practice of researchers in mathematical programming. The extension of the functionality to two further libraries for MINLP since the initial submission has increased its usefulness further. I have been able to install it easily under the conda environment and a small test worked smoothly.*
> *I wish to thank the authors for this great open-source contribution.*
>
> *RECOMMENDATION*
>
> *I recommend publication of this short communication after incorporating the minor comments below.*
>
> *COMMENTS FOR THE MANUSCRIPT*
>
> *- page 2, line 36: "embracing the CBF format": To the best of my knowledge the CBF format was defined along with the creation of CBLIB, so a different verb may be more accurate.*

We have rephrased that as “The Conic Benchmark Library (CBLIB) has been introduced in 2012 by aggregating known instances in the literature using the CBF format that was also proposed”

> *- page 3, line 11: "manipulate" sounds like the libraries would be modified, but I don't think that's what you mean. Also, why is MINLPLib listed first if MIPLIB was the main focus.*

We have replaced “manipulate” with “browse”, and we are now mentioning MIPLIB first in every sentence that talks about the three problem libraries that are currently supported.

> *- The section on Future Work is very short. Maybe the feature requests below could be included here.*
>
> *COMMENTS FOR THE SOFTWARE*
>
> *- It would be useful if there was an option to download and store instances in compressed form. Some instance files can get huge, so this should maybe even be the default.*
>
> *- It would be useful if solutions could be downloaded, too, whenever available. At least MIPLIB 2017 hosts primal solutions.*
>
> *- It would be useful to have a command line version.*
>
> *- It would be useful if instances could be downloaded in different formats (available, e.g., for QPLIB).*

We changed the text in Section 2 to emphasize that we already support downloading files in different formats: “Downloading and caching of problem files in one of the formats supported by the problem library to a local folder in the machine”. We have also listed your other recommendations in the Future Work section. They all look very relevant.