Beyond optimization: an algorithmic framework for structured nonmonotone inclusions

A Data Management Plan created using DMPonline.be

Creator: Pieter Pas

Affiliation: KU Leuven (KUL)

Funder: Fonds voor Wetenschappelijk Onderzoek - Research Foundation Flanders (FWO)

Template: FWO DMP (Flemish Standard DMP)

Grant number / URL: 11M9523N

ID: 199079

Start date: 01-11-2022

End date: 01-11-2026

Project abstract:

This project aims to generalize the mathematics of monotone inclusions (generalized equations) to the nonmonotone case.

One type of problem that gives rise to monotone inclusions is the minimization of a convex function, through the first-order optimality conditions. When minimizing nonconvex functions, however, monotonicity of the corresponding inclusion is lost, which makes finding solutions much more challenging. Going beyond minimization problems, many other interesting problems in science and engineering can be posed as the solution of general nonmonotone inclusions. One important example is bilevel optimization, a type of problem that arises in adversarial learning, generative adversarial networks, distributionally robust optimization, risk-averse control, game theory, and so on.

The mathematical foundation for solving nonmonotone inclusions is rather limited, and offers no tractable general algorithms either. This project aims to change that by developing an algorithmic framework for such problems. First, nonconvex optimization is investigated as a special case of the class of nonmonotone inclusions, which is then generalized, implemented as an accessible software package for nonconvex optimization and nonmonotone inclusions, and applied to common problems in control theory, game theory and machine learning.

Last modified: 29-04-2023

Beyond optimization: an algorithmic framework for structured nonmonotone inclusions
FWO DMP (Flemish Standard DMP)

1. Research Data Summary

List and describe all datasets or research materials that you plan to generate/collect or reuse during your research project. For each dataset or data type (observational, experimental etc.), provide a short name & description (sufficient for yourself to know what data it is about), indicate whether the data are newly generated/collected or reused, digital or physical, also indicate the type of the data (the kind of content), its technical format (file extension), and an estimate of the upper limit of the volume of the data.

Dataset Name	**Description**	**New or reused**	**Digital or Physical**	**Digital Data Type**	**Digital Data format**	**Digital data volume (MB/GB/TB)**
		Please choose from the following options: • Generate new data • Reuse existing data	*Please choose from the following options:* • Digital • Physical	*Please choose from the following options:* • Observational • Experimental • Compiled/ aggregated data • Simulation data • Software • Other • NA	*Please choose from the following options:* • .por, .xml, .tab, .cvs,.pdf, .txt, .rtf, .dwg, .gml, … • NA	*Please choose from the following options:* • <100MB • <1GB • <100GB • <1TB • <5TB • <10TB • <50TB • >50TB • NA
Primary software (WP 1 and 3)	Source code for the software developed as part of the PhD project.	Generate new data	Digital	Software	Plain-text C++, Python, Julia, C, Matlab, Fortran source code (.cpp, .hpp, .py, .jl, .c, .m, .f, .f90), including supporting build recipes (.cmake), scripts (.sh) and MarkDown/HTML documentation (.md, .html, .css).	<100MB
Software from previous projects	Source code of the software, numerical solvers and utilities developed by other (past) members of the research group.	Reuse existing data	Digital	Software	Idem.	<100MB
Benchmark problem descriptions and driver code (WP 4)	CasADi and PyTorch code for building the optimization and inclusion problems that will be used to verify and benchmark the numerical solvers.	Generate new data	Digital	Software	Plain-text C++ and Python source code (.cpp, .hpp, .py), including supporting build recipes (.cmake, Makefile), scripts (.sh), and data files (.csv, .npy) with matrices used in the problems.	<100MB
Benchmark problem collections	Collections of even more optimization and inclusion problems to benchmark the numerical solvers.	Reuse existing data	Digital	Software	Plain-text Fortran source code (.f), supporting build recipes (.sh), and problem description files (.sif).	<10GB
Benchmark results (WP 4)	Performance statistics (run time, number of evaluations, suboptimality, constraint violation ...) of the numerical solvers developed as part of the project, as well as the statistics for similar third-party solvers.	Generate new data	Digital	Simulation data	Structured records in YAML or JSON format, as well as comma-separated numerical data (CSV).	<1GB
Benchmark solutions (WP 4)	Numerical solutions produced by the different solvers for later comparison, verification, and visualization.	Generate new data	Digital	Simulation data	Plain-text or binary numerical data files (.csv, .npy, .mat, .dat).	<1GB

If you reuse existing data, please specify the source, preferably by using a persistent identifier (e.g. DOI, Handle, URL etc.) per dataset or data type:

Software developed as part of previous projects:

- https://github.com/kul-optec/QPALM
- https://github.com/kul-optec/QPALM.jl
- https://github.com/kul-optec/superscs

Benchmark problem collections:

- Gould, N.I.M., Orban, D. & Toint, P.L. CUTEst: a Constrained and Unconstrained Testing Environment with safe threads for mathematical optimization. *Comput Optim Appl* **60**, 545–557 (2015). https://doi.org/10.1007/s10589-014-9687-3

Are there any ethical issues concerning the creation and/or use of the data (e.g. experiments on humans or animals, dual use)? Describe these issues in the comment section. Please refer to specific datasets or data types when appropriate.

- No

Will you process personal data? If so, briefly describe the kind of personal data you will use in the comment section. Please refer to specific datasets or data types when appropriate.

- No

Does your work have potential for commercial valorization (e.g. tech transfer, for example spin-offs, commercial exploitation, …)? If so, please comment per dataset or data type

where appropriate.

- No

Do existing 3rd party agreements restrict exploitation or dissemination of the data you (re)use (e.g. Material/Data transfer agreements/ research collaboration agreements)? If so, please explain in the comment section to what data they relate and what restrictions are in place.

- No

Are there any other legal issues, such as intellectual property rights and ownership, to be managed related to the data you (re)use? If so, please explain in the comment section to what data they relate and which restrictions will be asserted.

- No

2. Documentation and Metadata

Clearly describe what approach will be followed to capture the accompanying information necessary to keep data understandable and usable, for yourself and others, now and in the future (e.g., in terms of documentation levels and types required, procedures used, Electronic Lab Notebooks, README.txt files, Codebook.tsv etc. where this information is recorded).

The software is accompanied by extensive documentation, installation instructions and usage examples. The API reference is automatically generated using tools like Doxygen and Sphinx, based on the docstrings and comments present in the source code.
Both the documentation and the API reference are available as HTML files and are hosted (either publicly or internally) for easy access.
All code is managed using the Git version control system, with clear commit messages, and tags are used to be able to easily switch back to a previous version (e.g. a specific release, or a version corresponding to a publication).
The numerical experiments include not only MarkDown or README files describing each experiment and benchmark problem, but also scripts and/or makefiles to easily reproduce the data, as well as scripts to visualize and analyze the existing data.

Will a metadata standard be used to make it easier to find and reuse the data? If so, please specify (where appropriate per dataset or data type) which metadata standard will be used. If not, please specify (where appropriate per dataset or data type) which metadata will be created to make the data easier to find and reuse.

- Yes

Benchmark results will be managed using the dman Data Management package to keep structured data about the simulation together with the results.
For simulations of specific problems of interest (rather than benchmark collections), a system such as EngMeta could be considered.
The software is published on GitHub, with appropriate tags, and made accessible through Zenodo or Papers with Code, which include metadata about the software, and link to the corresponding publications on arXiv. The GitHub repositories also contain a CITATION.cff file to link back to the publications.

3. Data storage & back-up during the research project

Where will the data be stored?

During active use or development, the data will be stored on the researcher's workstation and/or laptop.
Additionally, a time-stamped version of the data will be stored on the STADIUS DATASET Server, which is our research unit's central storage server.
Software will not only be kept locally, but also on the KU Leuven GitLab instance (for private projects) and on GitHub (for public/open-source projects) to allow for efficient collaboration.

How will the data be backed up?

Local copies of the Git repositories are stored on an ESAT workstation which is backed up regularly and automatically.
All data on the STADIUS DATASET Server are backed up daily and replicated to an off-site storage
system housed in the ICTS data center.

Is there currently sufficient storage & backup capacity during the project? If yes, specify concisely.
If no or insufficient storage or backup capacities are available, then explain how this will be taken care of.

- Yes

The volume of data is low, storage and backup capacity are not an issue. The STADIUS DATASET Server has a total capacity of 14.88 TB. The capacity of the dataset server is
monitored daily by the ESAT system admins.

How will you ensure that the data are securely stored and not accessed or modified by unauthorized persons?

The private repositories on GitLab are only accessible to authorized users, verified through the central KU Leuven login system, which requires two-factor authentication.
The public GitHub repositories are managed by members of the research group only, and require two-factor authentication. Releases to software repositories are automated, and the API tokens necessary to upload new releases are accessible only to the continuous integration workflows that can be triggered by authorized maintainers only.

What are the expected costs for data storage and backup during the research project? How will these costs be covered?

Data storage and back up will be performed on the STADIUS DATASET Server and the ESAT/KU Leuven GitLab instance in pre-existing storage facilities of the Department of Electrical Engineering (ESAT). The data volume for this project is small (<14GB), and so are the expected costs. For small volumes like this, the costs are covered by the storage provider.

4. Data preservation after the end of the research project

Which data will be retained for at least five years (or longer, in agreement with other retention policies that are applicable) after the end of the project? In case some data cannot be preserved, clearly state the reasons for this (e.g. legal or contractual restrictions, storage/budget issues, institutional policies...).

All research data (including software and numerical results used in publications) will be preserved for 10 years, according to the KU Leuven RDM policy.

Where will these data be archived (stored and curated for the long-term)?

The data will be stored on the university's central servers (with automatic back-up procedures).
The software itself will be additionally preserved indefinitely in public repositories on GitHub, PyPI and possibly others.

What are the expected costs for data preservation during the expected retention period? How will these costs be covered?

Free of charge for small datasets.

5. Data sharing and reuse

Will the data (or part of the data) be made available for reuse after/during the project? In the comment section please explain per dataset or data type which data will be made available.

- Yes, in an Open Access repository
- Yes, in a restricted access repository (after approval, institutional access only, …)

Software and numerical results used in publications will be made available in public GitHub repositories.
Before publication, development of the software and preliminary experiments will be available in restricted GitLab repositories requiring intitutional access and/or personal authorization.

If access is restricted, please specify who will be able to access the data and under what conditions.

Restricted GitLab repositories will be accessible to the supervisor and other members of the research group, as well as possible collaborators from other research groups.
The access of the data on the STADIUS DATASET Server is regulated by an access control list (ACL)
that grants:
- read-write access to the project owner and the FWO fellow
- read-only access to specific users, such as other members of our research group
The ACL is managed by the project owner (Panagiotis Patrinos).

Are there any factors that restrict or prevent the sharing of (some of) the data (e.g. as defined in an agreement with a 3rd party, legal restrictions)? Please explain in the comment section per dataset or data type where appropriate.

- No

Where will the data be made available? If already known, please provide a repository per dataset or data type.

- https://github.com/kul-optec/alpaqa (Software)
- https://github.com/kul-optec/panoc-gauss-newton-ifac-experiments (Numerical simulations)
- https://github.com/kul-optec/pantr-cdc2023-experiments (Numerical simulations)

When will the data be made available?

Upon publication of research results.

Which data usage licenses are you going to provide? If none, please explain why.

Software will be released under a copyleft license such as the GNU Lesser General Public License. Benchmarks and examples will be made available under a permissive MIT license.

Do you intend to add a PID/DOI/accession number to your dataset(s)? If already available, you have the option to provide it in the comment section.

- Yes

Software will be made available through Zenodo or Papers with Code.

What are the expected costs for data sharing? How will these costs be covered?

GitHub hosting is free for open-source repositories. Sharing code through the KU Leuven GitLab instance is covered by the university.

6. Responsibilities

Who will manage data documentation and metadata during the research project?

The data documentation and metadata will be managed by the FWO fellow, Pieter Pas.

Who will manage data storage and backup during the research project?

The FWO fellow, Pieter Pas will be responsible for data storage. The system administrators and data manager of the research division are responsible for the back up during and after the project.

Who will manage data preservation and sharing?

The FWO Fellow, Pieter Pas, will be responsible for ensuring data preservation and reuse.

Who will update and implement this DMP?

The project owner (Panagiotis Patrinos) bears the end responsibility of updating and implementing this DMP.