

Proceedings of the Stockholm Global Optimization Workshop, STOGO 2025

Edited by
Johanna Skåntorp and Jan Kronqvist,
KTH Royal Institute of Technology



Preface

Forty years ago the first Workshop on Global Optimization was held in Hungary. From its inception, the workshop has brought together researchers from across the world to discuss recent advancements in the field of global optimization. A lot has happened in forty years. In 1985 Karmakar's algorithm for Linear Programs had just been presented, and the computational capabilities of the day imposed fundamentally different limitations. The smartphone in your pocket beats a 1985 supercomputer by a factor of 10^3 in terms of computational speed; Many of the optimization problems that were computationally intractable back then, we easily solve today.

During its first era (1985 – 1995), the workshop was organized – every fifth year – in Hungary. Since then the workshop has been held in: Florence (GO'99, 1999), Hanmer Springs (Let's GO, 2001), Santorini (2003), San José (O'05, 2005), Mykonos (AGO, 2007), Skukuza (SAGO, 2008), Toulouse (TOGO, 2010), Natal (NAGO, 2012), Málaga (MAGO, 2014), Braga (GOW, 2016), Leiden (LEGO, 2018), and in Szeged (HUGO, 2022). And now in Stockholm (STOGO 2025). From the start the workshop has been given as a single track of presentations, and the tradition continues. The format of the workshop is intended to encourage discussions and the exchange of ideas among participants. We strive to have a friendly atmosphere, and also to promote young researchers in the field.

A change this year is the introduction of sponsors. Our sponsors have helped immensely in keeping the registration fee low – enabling more researchers to attend. We have also been lucky enough to attract sponsors that are highly relevant when discussing advancements in the field of optimization. This includes representatives of solvers many of us use in our research, and companies invested in optimization research together with academia. Our sponsors are: Digital Futures, FICO, GAMS, Hexaly, MOSEK, RaySearch Laboratories, the Swedish Operations Research Association, and TRATON.

This year's workshop is hosted by a team in the Optimization and Systems Theory group, in the department of Mathematics at KTH Royal Institute of Technology. The workshop is not held on the KTH campus, as the fall semester starts early in Sweden and classes prevents us from occupying a lecture room for four days. Instead the workshop is held at Piperska Muren, a beautiful venue in the city center, not far from City Hall where the Nobel Banquet is held.

This proceedings book collects the extended abstracts of the presentations and provides an overview of the scientific questions discussed during the workshop. Similar to last year, we will not openly publish the proceedings. Instead we will provide a printed and an electronic copy to participants. This is to avoid any potential issues with later publishing of the ideas presented. There will be special issue in Journal of Global Optimization (and hopefully also one in Mathematical Programming) devoted to manuscripts emerging from the workshop. As usual, the deadline will be a few months out – giving the authors time to process ideas and feedback received at the workshop. The details will be given by the end of the workshop.

This year marks the 40th anniversary of the first GOW. We look forward to an engaging workshop, interesting discussions, and insights into what the next forty years might bring to our field.

Jan Krogqvist
Stockholm, August 21, 2025

Program Committee

- *Sergiy Butenko*
- *Boglárka G.-Tóth*
- *Eligius M.T. Hendrix*
- *Jan Kronqvist (chair)*
- *Andreas Lundell*

Local Organizers

- *Yehor Blokhin*
- *Per Enqvist*
- *Jan Kronqvist*
- *David Liñan Romero*
- *Johanna Skåntorp*
- *Erik Tamm*
- *Shudian Zhao*

Scientific Committee

- *Montaz Ali*, University of the Witwatersrand, South Africa
- *Daniel Aloise*, Polytechnique Montréal, Canada
- *Pietro Belotti*, Politecnico di Milano, Italy
- *Ksenia Bestuzheva*, Zuse Institute Berlin, Germany
- *Merve Bodur*, University of Edinburgh, Scotland
- *Immanuel Bomze*, University of Vienna, Austria
- *Sergiy Butenko*, Texas A&M University, USA
- *Sonia Cafieri*, ENAC Université de Toulouse, France
- *Tibor Csendes*, University of Szeged, Hungary
- *Leandro Farias Maia*, Texas A&M University, USA
- *Boglárka G.-Tóth*, University of Szeged, Hungary
- *Robert Gottlieb*, University of Connecticut, USA
- *Akshay Gupte*, University of Edinburgh, United Kingdom
- *Adrian Göß*, University of Technology Nuremberg, Germany
- *Mohammad Reza Hesamzadeh*, KTH, Sweden
- *Milan Hladík*, Charles University, Czech Republic
- *Eligius Hendrix*, University of Malaga, Spain
- *Jan Kronqvist*, KTH Royal Institute of Technology, Sweden
- *Ksenia Bestuzheva*, GAMS, Germany
- *Andreas Lundell*, Åbo Akademi University, Finland
- *Leo Liberti*, CNRS and École Polytechnique, France
- *Frederic Messine*, ENSEEIHT, Toulouse, France
- *Ruth Misener*, Imperial College London, UK
- *Harsha Nagarajan*, Los Alamos National Laboratory, USA
- *Ivo Nowal*, HAW Hamburg, Germany
- *Alireza Olama*, Åbo Akademi University, Finland
- *Panos Pardalos*, University of Florida, Gainesville, USA
- *Ana Rocha*, University of Minho, Braga, Portugal
- *Jan Rolfes*, Linköping University, Sweden
- *Yaroslav D. Sergeyev*, University of Calabria, Italy
- *Emily Speakman*, University of Colorado Denver, USA
- *Oliver Stein*, Karlsruhe Institute of Technology, Germany
- *Matthew D. Stuber*, University of Connecticut, USA
- *Calvin Tsay*, Imperial College London, UK

digital futures

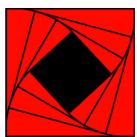
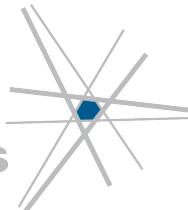
FICO®



G A M S

hexaly
mosek

RaySearch
Laboratories



**Swedish Operations
Research Association**

TRATON



Contents

Program	13
Plenary Talk Aida Khajavirad	23
<i>Aida Khajavirad</i>	
A second-order cone representable class of nonconvex quadratic optimization problems	27
Session 1	29
<i>Kristin Braun, Robert Burlacu, Tobias Kuen, Kristina Rolsing</i>	
Modeling Binary Relations in Piecewise-Linear Approximations	33
<i>Adrian Göß</i>	
Clash of MINLP Relaxations: Piecewise Linear versus Global Parabolic	39
Session 2	43
<i>Sebastian Lämmel, Vladimir Shikhman</i>	
Solving mathematical programs with complementarity constraints by disjunctive relaxations	47
<i>Ho-Ching Chui, Kamil Khan</i>	
Applying Tsoukalas-Mitsos convex relaxations in dynamic global optimization	53
<i>Jana Dienstbier, Frauke Liers, Jan Rolfes</i>	
A Positive Semidefinite Safe Approximation of Multivariate Distributional Robust Constraints Determined by Simple Functions	57
<i>Walid Ben-Ameur, Guillaume Beraud-Sudreau, Hervé Kerivin, Sébastien Martin</i>	
Relaxations of the Delay-Constrained Maximum Concurrent Flow Problem	61
<i>Georgia Stinchfield, Megan Walsh, Yankai Cao, Carl Laird</i>	
SNoGloDe: A Flexible Tool for Decomposition & Global Optimization	65

Session 3	69
<i>Anatolii Kosolap</i>	
New ideas in global optimization	73
<i>Antonio Sasaki, Sophie Demassey, Valentina Sessa</i>	
On Computational Evaluation of Lower Bounds for the Fractional Quadratic Program over the Standard Simplex	77
<i>O. Li, D. Ovalle, M. Bynum, B. Poczos, C. Laird, I.E. Grossmann, J. Peña</i>	
Efficient Convexification of Polynomial Generalized Additive Model Surrogates via Continuous Graham-Scan	81
<i>Sergiy Butenko</i>	
Continuous Cubic Formulations for the Maximum Clique Problem	85
<i>Leo Liberti, Maël Kupperschmitt, Ha Duy Nguyen</i>	
A blueprint for computing with distance geometry	89
Session 4	95
<i>Deborah Hendrych, Mathieu Besançon, Sebastian Pokutta</i>	
Solving Optimal Experiment Design with Mixed-Integer Convex Methods	99
<i>Eva Ley, Maximilian Merkert</i>	
On Solving Partial Inverse Combinatorial Problems with NP-Complete Lower Level	103
<i>G. Mexi, D. Hendrych, S. Designolle, M. Besançon, S. Pokutta</i>	
A Frank-Wolfe-based primal heuristic for quadratic mixed-integer optimization	107
Session 5	111
<i>André Schnabel, Hamdi Burak Usul</i>	
Embedding neural networks into optimization models with GAMSPy	115
<i>Imre Pólik</i>	
Finding long TSP tours in the unit square using the FICO Xpress Global solver	119
<i>Gustaf Ehn</i>	
MOSEK as a system component	121
<i>Julien Darlay</i>	
Computing global optimum in Hexaly	123
Session 6	125
<i>Andreas Lundell, Alireza Olama, Joakim Blomqvist, Johan Ekman, Jan Kronqvist</i>	
Upcoming features in SHOT 2.0	129
<i>Joakim Blomqvist, Andreas Lundell</i>	
Improved Bounding Techniques for Nonconvex MINLP in SHOT	133

<i>Alireza Olama, Andreas Lundell</i>	
A Parallel Framework for Primal Heuristics in SHOT	137
<i>Andrea Ghezzi, Wim Van Roy, Sebastian Sager, Moritz Diehl</i>	
The CAMINO Software Package	141
Session 7	145
<i>Mihály Gencsi, Boglárka G.-Tóth</i>	
Advanced Geometrical test for efficient use of optimality conditions in Interval Branch and Bound methods	149
<i>Robert Gottlieb, Matthew Stuber</i>	
GPU-Parallel Branch-and-Bound with Custom Kernels and specialized PDLP	153
<i>Asimina Marousi, Vassilis M. Charitopoulos</i>	
Integrating Global and Robust Optimisation for Non-Convex Quadratic Programs	159
<i>Sonia Cafieri, Marcel Mongeau, Sébastien Bourguignon, Gwenaël Samain</i>	
Near-optimal global solutions for sparse approximation problems	163
<i>Jon Lee, Emily Speakman, Daphne Skipper</i>	
On the Convex Hull of the Graph of a Simple Monomial	167
Plenary Talk Martin Schmidt	173
<i>Martin Schmidt</i>	
The Burial of Coupling Constraints in Linear Bilevel Optimization	177
Session 8	179
<i>Philip Sosnin, Calvin Tsay</i>	
Exact Verification of Data Poisoning Attacks via Mixed-Integer Programming	183
<i>Tamim Mehr, Ivo Nowak, Jan Kronqvist</i>	
Solving MINLPs using Decomposition, Multiobjective Optimization and Machine Learning	189
<i>Yehor Blokhin, Martin Ryner, Jan Kronqvist</i>	
Utilizing Rotational Transformations for Convex Relaxations of Nonconvex QPs	193
<i>Carl Eggen, Jan Kronqvist, Andreas Lundell, Stefan Volkwein</i>	
Separating Hyperplanes for Mixed-Integer Polynomial Optimization Problems	199
Session 9	203
<i>Marc Teboulle</i>	
Algorithmic Advances in Proximal Gradient Methods for Nonconvex Minimization	207

<i>Ashwin Renganathan, Kade Carlson</i>	
Efficient Batch Multiobjective Bayesian Optimization via Pareto Optimal Thompson Sampling	211
Session 10	221
<i>Seokwoo Kim, Dong Gu Choi</i>	
A Bi-level Service Capacity Planning Model with Interdependent MDP Followers	225
<i>Pascal Börner, Marc E. Pfetsch, Stefan Ulbrich</i>	
Mathematics of Gas Mixtures on Networks – Modeling, Optimization, Existence and Uniqueness	231
<i>Christophe Andrieu, Ettore Fincato, Nicolas Chopin, Mathieu Gerber</i>	
Gradient-free optimization via integration	235
Session 11	239
<i>Eligius M.T. Hendrix, Pablo Guerrero-Garcia, Ana Maria A.C. Rocha</i>	
On optimization over a polyhedral set and Augmented Lagrangians	243
<i>Sebastian Lämmel, Vladimir Shikhman</i>	
Anomalies of the Scholtes regularization for mathematical programs with complementarity constraints	247
<i>Charles Audet, Frédéric Messine</i>	
The Story of Hansen's Octagon	253
<i>Roozbeh Abolpour, Mohammad Reza Hesamzadeh</i>	
Furthest Point Problem: Solution Algorithm and Applications	257
Session 12	263
<i>M. Reza Karimi Garighi, M. Reza Hesamzadeh, J. Kronqvist, G. Dán</i>	
Global Optimization of ACOPF in Unbalanced Distribution Networks	267
<i>Shudian Zhao, Jan Kronqvist, Mohammad Reza Hesamzadeh</i>	
The Strengthened Semidefinite Relaxation for Nordic Day-ahead Energy Market	271
<i>Z. Wang, S. Zhao, J. Kronqvist, M. Reza Hesamzadeh</i>	
A Modified SAVLR Method for Zero Duality Gap in MILPs of Electricity Markets	275

Program





Tuesday September 2nd**12:00-12:30 Registration****12:30-12:40 Welcome****12:40-13:40 Plenary Talk Aida Khajavirad**

<i>Aida Khajavirad:</i> A second-order cone representable class of nonconvex quadratic optimization problems	27
--	----

13:40-14:30 Session 1**Chair: Jan Kronqvist**

<i>Kristin Braun, Robert Burlacu, Tobias Kuen, Kristina Rolsing:</i> Modeling Binary Relations in Piecewise-Linear Approximations	33
---	----

<i>Adrian Göß:</i> Clash of MINLP Relaxations: Piecewise Linear versus Global Parabolic	39
---	----

14:30-15:00 Coffee**15:00-17:00 Session 2****Chair: Jan Rolfes**

<i>Sebastian Lämmel, Vladimir Shikhman:</i> Solving mathematical programs with complementarity constraints by disjunctive relaxations	47
---	----

<i>Ho-Ching Chui, Kamil Khan:</i> Applying Tsoukalas-Mitsos convex relaxations in dynamic global optimization	53
---	----

<i>Jana Dienstbier, Frauke Liers, Jan Rolfes:</i> A Positive Semidefinite Safe Approximation of Multivariate Distributional Robust Constraints Determined by Simple Functions	57
---	----

<i>Walid Ben-Ameur, Guillaume Beraud-Sudreau, Hervé Kerivin, Sébastien Martin:</i> Relaxations of the Delay-Constrained Maximum Concurrent Flow Problem	61
---	----

<i>Georgia Stinchfield, Megan Walsh, Yankai Cao, Carl Laird:</i> SNoGloDe: A Flexible Tool for Decomposition & Global Optimization	65
--	----

17:00-17:20 Industry Session

Wednesday September 3rd**08:30-10:30 Session 3****Chair: Sergiy Butenko**

<i>Anatolii Kosolap:</i> New ideas in global optimization	73
<i>Antonio Sasaki, Sophie Demassey, Valentina Sessa:</i> On Computational Evaluation of Lower Bounds for the Fractional Quadratic Program over the Standard Simplex	77
<i>O. Li, D. Ovalle, M. Bynum, B. Poczos, C. Laird, I.E. Grossmann, J. Peña:</i> Efficient Convexification of Polynomial Generalized Additive Model Surrogates via Continuous Graham-Scan	81
<i>Sergiy Butenko:</i> Continuous Cubic Formulations for the Maximum Clique Problem	85
<i>Leo Liberti, Maël Kupperschmitt, Ha Duy Nguyen:</i> A blueprint for computing with distance geometry	89

10:30-11:00 Coffee**11:00-12:15 Session 4****Chair: Leo Liberti**

<i>Deborah Hendrych, Mathieu Besançon, Sebastian Pokutta:</i> Solving Optimal Experiment Design with Mixed-Integer Convex Methods	99
<i>Eva Ley, Maximilian Merkert:</i> On Solving Partial Inverse Combinatorial Problems with NP-Complete Lower Level	103
<i>G. Mexi, D. Hendrych, S. Desagnolle, M. Besançon, S. Pokutta:</i> A Frank-Wolfe-based primal heuristic for quadratic mixed-integer optimization	107

12:15-13:15 Lunch**13:15-14:45 Session 5****Chair: Mathieu Besançon**

<i>André Schnabel, Hamdi Burak Usul:</i> Embedding neural networks into optimization models with GAMSPy	115
<i>Imre Pólik:</i> Finding long TSP tours in the unit square using the FICO Xpress Global solver	119
<i>Gustaf Ehrensson:</i> MOSEK as a system component	121
<i>Julien Darlay:</i> Computing global optimum in Hexaly	123

14:45-15:15 Coffee

15:15-16:45 Session 6**Chair: Boglárka G.-Tóth**

<i>Andreas Lundell, Alireza Olama, Joakim Blomqvist, Johan Ekman, Jan Kronqvist:</i>	
Upcoming features in SHOT 2.0	129
<i>Joakim Blomqvist, Andreas Lundell:</i> Improved Bounding Techniques for Nonconvex MINLP in SHOT	133
<i>Alireza Olama, Andreas Lundell:</i> A Parallel Framework for Primal Heuristics in SHOT	137
<i>Andrea Ghezzi, Wim Van Roy, Sebastian Sager, Moritz Diehl:</i> The CAMINO Software Package	141

19:30**Dinner**

Thursday September 4th

08:30-10:30 Session 7

Chair: Emily Speakman

<i>Mihály Gencsi, Boglárka G.-Tóth:</i> Advanced Geometrical test for efficient use of optimality conditions in Interval Branch and Bound methods	149
<i>Robert Gottlieb, Matthew Stuber:</i> GPU-Parallel Branch-and-Bound with Custom Kernels and specialized PDLP	153
<i>Asimina Marousi, Vassilis M. Charitopoulos:</i> Integrating Global and Robust Optimisation for Non-Convex Quadratic Programs	159
<i>Sonia Cafieri, Marcel Mongeau, Sébastien Bourguignon, Gwenaël Samain:</i> Near-optimal global solutions for sparse approximation problems	163
<i>Jon Lee, Emily Speakman, Daphne Skipper:</i> On the Convex Hull of the Graph of a Simple Monomial	167

10:30-11:00 Coffee

11:00-12:00 Plenary Talk Martin Schmidt

<i>Martin Schmidt:</i> The Burial of Coupling Constraints in Linear Bilevel Optimization	177
--	-----

12:00-13:00 Lunch

13:00-14:30 Session 8

Chair: Matthew Stuber

<i>Philip Sosnin, Calvin Tsay:</i> Exact Verification of Data Poisoning Attacks via Mixed-Integer Programming	183
<i>Tamim Mehr, Ivo Nowak, Jan Kronqvist:</i> Solving MINLPs using Decomposition, Multiobjective Optimization and Machine Learning	189
<i>Yehor Blokhin, Martin Ryner, Jan Kronqvist:</i> Utilizing Rotational Transformations for Convex Relaxations of Nonconvex QPs	193
<i>Carl Eggen, Jan Kronqvist, Andreas Lundell, Stefan Volkwein:</i> Separating Hyperplanes for Mixed-Integer Polynomial Optimization Problems	199

14:30-15:00 Coffee

15:00-15:30 Industry Session

15:30-16:20 Session 9

Chair: Calvin Tsay

<i>Marc Teboulle:</i> Algorithmic Advances in Proximal Gradient Methods for Nonconvex Minimization	207
<i>Ashwin Renganathan, Kade Carlson:</i> Efficient Batch Multiobjective Bayesian Optimization via Pareto Optimal Thompson Sampling	211

17:30 Social

Friday September 5th

09:00-10:15 Session 10

Chair: Martin Schmidt

- Seokwoo Kim, Dong Gu Choi:* A Bi-level Service Capacity Planning Model with Interdependent MDP Followers 225

- Pascal Börner, Marc E. Pfetsch, Stefan Ulbrich:* Mathematics of Gas Mixtures on Networks – Modeling, Optimization, Existence and Uniqueness 231

- Christophe Andrieu, Ettore Fincato, Nicolas Chopin, Mathieu Gerber:* Gradient-free optimization via integration 235

10:15-10:45 Coffee

10:45-12:15 Session 11

Chair: Eligius Hendrix

- Eligius M.T. Hendrix, Pablo Guerrero-Garcia, Ana Maria A.C. Rocha:* On optimization over a polyhedral set and Augmented Lagrangians 243

- Sebastian Lämmel, Vladimir Shikhman:* Anomalies of the Scholtes regularization for mathematical programs with complementarity constraints 247

- Charles Audet, Frédéric Messine:* The Story of Hansen's Octagon 253

- Roozbeh Abolpour, Mohammad Reza Hesamzadeh:* Furthest Point Problem: Solution Algorithm and Applications 257

12:15-13:15 Lunch

13:15-14:30 Session 12

Chair: Andreas Lundell

- M. Reza Karimi Garighi, M. Reza Hesamzadeh, J. Kronqvist, G. Dán:* Global Optimization of ACOPF in Unbalanced Distribution Networks 267

- Shudian Zhao, Jan Kronqvist, Mohammad Reza Hesamzadeh:* The Strengthened Semidefinite Relaxation for Nordic Day-ahead Energy Market 271

- Z. Wang, S. Zhao, J. Kronqvist, M. Reza Hesamzadeh:* A Modified SAVLR Method for Zero Duality Gap in MILPs of Electricity Markets 275

14:30-15:00 Final Session

15:00-16:00 Coffee

Plenary Talk
Aida Khajavirad





A second-order cone representable class of nonconvex quadratic optimization problems

Aida Khajavirad,

Lehigh University, Department of Industrial and Systems Engineering, PA, USA
aik220@lehigh.edu

Abstract We consider the problem of minimizing a sparse nonconvex quadratic function over the unit hypercube. By developing an extension of the Reformulation Linearization Technique (RLT) to continuous quadratic sets, we propose a novel second-order cone (SOC) representable relaxation for this problem. By exploiting the sparsity of the quadratic function, we establish a sufficient condition under which the convex hull of the feasible region of the linearized problem is SOC-representable. While the proposed formulation may be of exponential size in general, we identify additional structural conditions that guarantee the existence of a polynomial-size SOC-representable formulation, which can be constructed in polynomial time. Under these conditions, the optimal value of the nonconvex quadratic program coincides with that of a polynomial-size second-order cone program. Our results serve as a starting point for bridging the gap between the Boolean quadric polytope of sparse problems and its continuous counterpart.

Session 1





Modeling Binary Relations in Piecewise-Linear Approximations*

Kristin Braun¹, Robert Burlacu², Tobias Kuen¹, Kristina Rolsing²

¹*Fraunhofer IIS, Nuremberg, Germany, {kristin.braun, tobias.kuen}@iis.fraunhofer.de*

²*University of Technology, Nuremberg, Germany, {robert.burlacu, kristina.rolsing}@utn.de*

Abstract We investigate the integration of cutting-plane techniques within the binary variable space arising from piecewise-linear approximations in nonlinear programming. We identify the emergence of the bipartite implication polytope as a substructure in various formulations of piecewise-linear relaxations. Extending the theoretical framework of the bipartite implication polytope to a multipartite context, we enhance the applicability of these results. Our analysis reveals that discretized variables and their interrelations are prevalent not only in top-level equations but also within the expression trees utilized for one-dimensional reformulations. By examining these common relationships, we develop tailored cutting planes aimed at tightening the linear programming relaxation of the resultant mixed-integer programs.

Keywords: Piecewise-Linear Approximations, Mixed-Integer Nonlinear Programming, Cutting Planes

1. Introduction

Mixed-integer nonlinear programs (MINLPs) are pivotal in modeling complex systems across various disciplines, encompassing both discrete and continuous variables alongside nonlinear relationships. A prevalent strat-

*This work was part of the project EKSSE, funded by the Federal Ministry of Education and Research (BMBF)

egy to address the challenges posed by MINLPs is employing piecewise-linear relaxations to approximate nonlinear functions inside a disjunctive programming approach, thereby transforming the original problem into a mixed-integer linear program (MIP) that is more tractable for existing state-of-the-art solvers. This approach leverages the maturity and efficiency of MIP solvers to find solutions to problems that would otherwise be computationally prohibitive. Effectively solving a MIP hinges on the strength of its continuous relaxation; weak relaxations can render MIPs computationally intractable.

Recent advancements have focused on tightening the linear relaxations through innovative formulation techniques and enhancing the efficacy of the solving methods [3], [6], [9], [10]. The most widely used cuts such as McCormick envelopes [8] primarily act on continuous variables to tighten convex relaxations of nonlinear terms. However, the introduction of binary variables in piecewise-linear relaxations creates a unique opportunity to leverage the interdependencies among these variables.

A notable structure that encapsulates such interdependencies is the bipartite implication polytope (BIP) [4], which models conditional relations across three sets of binary variables. This polytope has been studied for its ability to capture the essence of these conditional relationships, providing a foundation for deriving valid inequalities that can tighten the linear relaxation. In this work, we extend the theory to relations across arbitrary many sets of binary variables and term this polytope Multipartite Implication Polytope (MPIP).

2. Multipartite Implication Polytope

We consider an instance involving $n \in \mathbb{N}$ implying-sets of binary variables, where, without loss of generality, each set has cardinality $m \in \mathbb{N}$. From each implying set, a single element must be chosen. For the i -th implying set, the choice of the element is denoted by an index $j_i \in [m]$, where $[m]$ denotes $\{1, \dots, m\}$.

The choices across all implying sets are represented by an index vector $\mathbf{j} = (j_1, \dots, j_n) \in [m]^n$. The corresponding implied set has cardinality $k \in \mathbb{N}$. The choices in the implying sets imply a corresponding element $l \in [k]$ from the implied set.

The implication relationship between a potential choice in each implying set and the result in the implied set is defined through a mapping

$$\varphi : [m]^n \rightarrow [k], \mathbf{j} = (j_1, \dots, j_n) \mapsto l,$$

where $\varphi(\mathbf{j}) = l$ determines the index l of the corresponding element in the implied set. We speak of a multipartite implication instance, if parameters m, n, k and mapping φ are given. The requirement to select exactly one element from the i -th implying set is modeled as a multiple choice constraint $\sum_{j \in [m]} x_j^i = 1$. The binary points fulfilling the multiple choice constraints for all implying sets and the implied set are captured by the set

$$M := \left\{ (\mathbf{x}^1, \dots, \mathbf{x}^n, \mathbf{y}) \in \{0, 1\}^{mn+k} \mid \sum_{j \in [m]} x_j^i = 1, \forall i \in [n], \sum_{l \in [k]} y_l = 1 \right\}.$$

Using the mapping φ , we restrict the set MC to ensure only feasible relations by making use of a constraint involving a product. This is formalized in the set

$$S(\varphi) := \left\{ (\mathbf{x}^1, \dots, \mathbf{x}^n, \mathbf{y}) \in M \mid \prod_{i \in [n]} x_{j_i}^i \leq y_{\varphi(\mathbf{j})}, \forall \mathbf{j} \in [m]^n \right\}.$$

The multipartite implication polytope is defined as its convex hull, i.e.,

$$MPIP(\varphi) := conv(S(\varphi)).$$

3. MPIP in Piecewise-Linear Relaxations

The multiple-choice method [1] for piecewise linear approximations divides the domain of each variable into intervals and introduces binary variables to indicate which interval is active. When applying this method to relations between multiple continuous variables in the model, a multipartite implication structure naturally emerges, though with a slight modification. The key difference from the already existing theory for the bipartite implication polytope is that selecting specific intervals for the implying variables does not necessarily determine a unique interval for the implied. Instead, it restricts it to a subset of possible intervals, depending on the actual values chosen within the intervals of the implying variables. This means the relation mapping ϕ would need to map each input index vector to a set of possible index values rather than a single value.

This appearance of the MPIP in the multiple-choice method serves as a starting point to analyze further piecewise-linear reformulation techniques like the more commonly applied incremental method.

4. Variable Relations in Expression Trees

Beyond the explicit model equations, additional structure arises from expression trees, i. e., graph-based representations of how a complex non-linear expression is built from simpler subexpressions using basic operators. When using expression trees [5] to reformulate the problem such that it contains only one-dimensional non-linear functions, relations between both original variables and artificially created variables emerge naturally.

Example 1. Consider the constraint $\sin(xy) = 0$. We introduce an auxiliary variable z and reformulate the constraint as $z = xy$ and $\sin(z) = 0$. Assume bounded domains $x, y \in [0, 4]$, $z \in [0, 16]$. Discretize each domain into 4 intervals:

$$\begin{aligned} I_i &= [i - 1, i) && \text{for } x, \\ J_j &= [j - 1, j) && \text{for } y, \\ K_k &= [4(k - 1), 4k) && \text{for } z, \end{aligned}$$

where $i, j, k \in [4]$. Binary variables χ_i, ξ_j, ζ_k indicate membership in I_i, J_j, K_k , respectively. The relation mapping $\phi: [4]^2 \rightarrow \mathcal{P}([4])$ is defined such that $\phi(i, j)$ contains k if $\exists x \in I_i, y \in J_j$ with $xy \in K_k$. This mapping is computed as:

$$\phi(i, j) = \{k : [(i - 1)(j - 1), ij) \cap [4(k - 1), 4k) \neq \emptyset\}.$$

Figure 1 visualizes ϕ , where cell (i, j) lists the set $\phi(i, j)$. For instance:

- $\phi(1, 1) = \{1\}$ since $x \in [0, 1], y \in [0, 1] \implies z \in [0, 1] \subseteq [0, 4]$.
- $\phi(2, 4) = \{1, 2\}$ since $x \in [1, 2], y \in [3, 4] \implies z \in [3, 8] = [3, 4] \cup [4, 8]$.

One inequality that can be obtained using MPIP theory is

$$\chi_3 + 2\chi_4 + \xi_3 + 2\xi_4 \leq \zeta_2 + 2\zeta_3 + 2\zeta_4 + 2.$$

It cuts off the non-integer point

$$s^\chi = (0, 0, \frac{2}{3}, \frac{1}{3}), \quad s^\xi = (0, 0, \frac{2}{3}, \frac{1}{3}), \quad s^\zeta = (0, 1, 0, 0).$$

	j			
ϕ	1	2	3	4
1	{1}	{1}	{1}	{1}
2	{1}	{1}	{1,2}	{1,2}
3	{1}	{1,2}	{2,3}	{2,3}
4	{1}	{1,2}	{2,3}	{3,4}

Figure 1: Relation mapping ϕ showing possible k for each (i, j) pair in Example 1.

Its validity can be easily checked by checking all 24 points in the set of feasible points

$$S^{SET}(\varphi) := \left\{ (\chi, \xi, \zeta) \in M \mid \chi_i \xi_j \leq \sum_{k \in \varphi(j)} \zeta_k, \forall (i, j) \in [4]^2 \right\}. \quad (1)$$

5. Summary

We introduce the multipartite implication polytope (MPIP) as an extension of the bipartite implication polytope to model interdependencies among binary variables. We study the emergence of MPIP in piecewise-linear relaxations and expression tree reformulations, proposing tailored cutting planes acting on the binary variables to strengthen linear relaxations.

References

- [1] Balakrishnan, A. et. al. (1989). A composite algorithm for a concave-cost network flow problem. *Networks*. 19. 175–202.
- [2] Braun, K. et. al. (2023). A Computational Study for Piecewise Linear Relaxations of Mixed-Integer Nonlinear Programs. *Optimization Online*.
- [3] Burlacu, R. et. al. (2020). Solving mixed-integer nonlinear programmes using adaptively refined mixed-integer linear programmes. *Optimization Methods and Software*. 35. 37–64.
- [4] Burlacu, R. et. al. (2024). The Bipartite Implication Polytope: Conditional Relations over Multiple Sets of Binary Variables. *Optimization Online*.
- [5] Kim, J. at. al.. Learning symbolic expressions: Mixed-integer formulations, cuts, and heuristics. *arXiv preprint arXiv:2102.08351 [math.OC]*, 2021.
- [6] Lyu, B. et. al. (2024). Modeling combinatorial disjunctive constraints via junction trees. *Mathematical Programming*. 204. 385–413.
- [7] Markowitz, H. et. al. (1957). On the Solution of Discrete Programming Problems. *Econometrica*. 25. 84–110.
- [8] McCormick, G. P. (1976). *Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems*. *Mathematical Programming*, 10(1), 147–175.
- [9] Nagarajan, H. et. al. (2016). Tightening McCormick Relaxations for Nonlinear Programs via Dynamic Multivariate Partitioning. *Principles and Practice of Constraint Programming*. Cham: Springer International Publishing. 369–387.
- [10] Nowak, I. (2005). *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming*. Basel: Birkhäuser.

Clash of MINLP Relaxations: Piecewise Linear versus Global Parabolic

Adrian Göß¹

¹*University of Technology Nuremberg (UTN), Nuremberg, Germany, adrian.goess@utn.de*

Abstract Relaxation techniques play a pivotal part in solving MINLP problems by providing primal bounds and guiding the solution process. However, as more relaxation techniques become available, it remains unclear when to favor one method over another. This work seeks to address this question by comparing two approaches: piecewise linear (PWL) and global parabolic (PAR) relaxations. While efficient implementations exist for PWL, PAR has only recently been introduced and thus lacks tractable procedures for large scales. Therefore, a novel algorithm for global parabolic approximations of one-dimensional functions is introduced. Its computational results are leveraged to perform a systematic, efficacy-based comparison of PWL and PAR relaxations on MINLPLib instances. Although final results are forthcoming, the article outlines a conceptual framework for identifying characteristics that favor one relaxation over the other.

Keywords: Mixed-Integer Nonlinear Programming, Relaxations, Piecewise Linear Approximation, Parabolic Approximation

1. Introduction

This work considers mathematical programming problems of the form

$$\begin{aligned}
 & \min c(\mathbf{x}) \\
 & \text{s.t. } f_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, m, \\
 & \mathbf{x} \in [\underline{\mathbf{x}}, \bar{\mathbf{x}}], \\
 & \mathbf{x} \in \mathbb{Z}^d \times \mathbb{R}^{n-d},
 \end{aligned} \tag{1}$$

where $c, f_j : \mathbb{R}^n \rightarrow \mathbb{R}$ are continuously differentiable, $j = 1, \dots, m$, and $d, n, m \in \mathbb{N}_0$. The bounds \underline{x} and \bar{x} are considered finite. In addition, the functions f_j are assumed to be *factorable* which is the case for most functions in the MINLPLib [3]. That is, by introducing additional variables the problem can be transformed such that the constraint functions are linear combinations of constants, variables, bilinear terms, and one-dimensional functions. This is equivalent to the ability of formulating the problem as an *expression tree*, see [6] for the introduction of the similar *expression graph*. We refer to (1) as a Mixed-Integer Nonlinear Programming (MINLP) problem or simply Mixed-Integer Nonlinear Program (MINLP).

When trying to solve a MINLP, one classical approach involves simplifying the problem, thus creating a relaxed version, and solving this relaxation. The returned objective value serves as a lower bound for the original problem, and the corresponding solution can support the remaining solution process. For example, it can be used to compute a cutting plane, refine the relaxation on infeasible areas or fix its integral components in the MINLP to end up with a Nonlinear Program (NLP).

In the literature, there exist numerous relaxation techniques. Recently, the *piecewise linear* (PWL) relaxation received new attention (see, for instance, [2]) and the *parabolic approximation and relaxation* (PAR) approach was introduced [5]. PWL provides linear constraint functions as approximations on a partition of the original domain and introduces one additional variable for each subinterval of the partition to determine the active one. When applied to each one-dimensional nonlinear constraint function in a factorable MINLP, this transformation essentially yields a Mixed-Integer Linear Problem (MIP). In contrast, PAR constructs quadratic functions without bilinear terms (*paraboloids*), which individually underestimate a nonlinear constraint function globally and as a whole serve as an approximation. This replaces the original functions with quadratic, potentially non-convex constraints, but does not necessitate the addition of variables. The resulting relaxation is a Mixed-Integer Quadratically-Constrained Program (MIQCP) with univariate quadratic terms only. An illustration of both approaches is provided in Figure 1.

This work presents a systematic comparison between PWL and PAR across different approximation accuracies and tries to recommend the favorable approach in terms of efficacy based on the properties of the underlying problem setting. In contrast to the efficient implementations for PWL [1], for PAR, the original technique is suited for Lipschitz functions on arbitrary dimension, but suffers from scalability issues when it comes to large function domains and/or small approximation accuracy. Hence, a novel algorithm for parabolic approximation of one-dimensional func-

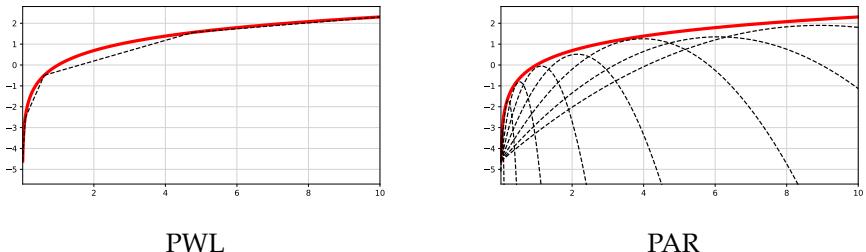


Figure 1: Both approximation techniques for \ln on $[0.01, 10]$ with $\varepsilon = 0.5$.

tions is introduced in section 2 to ensure comparability on a large test set of problems. The conceptual framework for comparing its efficacy to PWL is described in section 3. For ease of explanation, the relaxations are discussed as underestimating ones in the remainder.

2. Computing Parabolic Approximations

The algorithm introduced for parabolic approximation of one-dimensional, continuously differentiable functions is based on two parts: a main and a subroutine. Given the domain $\mathcal{D} = [a, b] \subset \mathbb{R}$ of a constraint function f , the main routine starts by considering a local sub-domain $\mathcal{D}' = [a', b'] \subset \mathcal{D}$ with $a' = a$. After computing an approximating parabola on \mathcal{D}' by the subroutine, it proceeds with an interval $[a'', b'']$, where $a'' = b'$, and iterates until \mathcal{D} is covered. The size of \mathcal{D}' and the following sub-domains is adjusted during the computation of the approximating parabola.

Based on an approximation accuracy $\varepsilon > 0$ the subroutine aims to compute a parabola $p(x) = \alpha x^2 + \beta x + \gamma$, which

1. passes through $f - \varepsilon$ at the bounds of \mathcal{D}' ,
2. deviates at most ε on \mathcal{D}' from f , and
3. is a global underestimator of f on \mathcal{D} .

Given α , condition 1 allows computing the remaining coefficients β and γ . However, when evaluated at specific points, conditions 2 and 3 impose bounds on α . By computing the maximum of $f - p$ and $p - f$ on \mathcal{D} and \mathcal{D}' such points can be identified, leading to a successive narrowing of α . This process is repeated until a suitable α is found or the necessity for a smaller \mathcal{D}' is returned. Tailoring the parabolic approximation to one-dimensional functions enables the benchmarking against PWL discussed next.

3. Performance Comparison

For comparing the relaxation techniques, the MINLPLib is considered, which comprises various MINLP instances. Nearly all such are available in OSiL format [4] which resembles the expression tree structure. This facilitates filtering instances, where the variable bounds of the one-dimensional functions are finite and numerically tractable for either relaxation technique.

Since the individual domains for a particular type of one-dimensional function may differ in each occurrence, a clustering is applied such that only a moderate number of bound combinations must be approximated for each function. This allows computation of the piecewise linear and global parabolic approximations for all bound-function combinations and multiple accuracies. With these at hand, it becomes straightforward to construct the expression tree and to substitute the one-dimensional function with the linear or parabolic relaxation on the respective clustered interval. In particular, for each instance INST , a linear INST-lin and a parabolic relaxation INST-par are created. These are solved using state-of-the-art solvers, e.g., Gurobi or SCIP. The evaluation of the results is performed by means of performance profiles and shifted geometric means (SGMs) of the running times, categorized by each accuracy. These measures in combination with statistics about the characteristics of each instance – e.g., number of function types, number of variables, or average function domain size – can establish an instance-based relaxation selection inside solvers.

References

- [1] Braun, K., & Burlacu, R. (2025). Benchmarking Piecewise Linear Reformulations for MINLPs: a Computational Study Based on the Open-Source Framework PWL-T-Rex. Preprint Optimization Online.
- [2] Burlacu, R., Geißler, B., & Schewe, L. (2020). Solving mixed-integer nonlinear programmes using adaptively refined mixed-integer linear programmes. Optimization Methods and Software. 35(1). 37-64.
- [3] Bussieck, M. R., Drud, A. S., & Meeraus, A. (2003). MINLPLib—a collection of test models for mixed-integer nonlinear programming. INFORMS Journal on Computing. 15(1). 114-119.
- [4] Fourer, R., Ma, J., & Martin, K. (2010). OSiL: An instance language for optimization. Computational optimization and applications. 45(1). 181-203.
- [5] Göß, A., Burlacu, R., & Martin, A. (2025). Parabolic Approximation & Relaxation for MINLP. Preprint arXiv:2407.06143.
- [6] Schichl, H., & Neumaier, A. (2005). Interval analysis on directed acyclic graphs for global optimization. Journal of Global Optimization, 33, 541-562.

Session 2





Solving mathematical programs with complementarity constraints by disjunctive relaxations

Sebastian Lämmel¹, Vladimir Shikhman¹

¹*Chemnitz University of Technology, Faculty of Mathematics, Reichenhainer Str. 41, 09126 Chemnitz, Germany, sebastian.laemmel@mathematik.tu-chemnitz.de*

Abstract We propose a new disjunctive relaxation for mathematical programs with complementarity constraints (MPCC) based on the geometrical shape of that introduced in [4]. By doing so, we overcome problems with the linear independence constraint qualification and improve convergence results for C-, M-, and S-stationary points. Moreover, we expect the numerical stability of the proposed scheme to be improved in comparison to [5], due to the avoidance of cancellation. Additionally, we reveal the following surprising phenomenon: Nondegenerate minimizers of the relaxation may converge to saddle points of MPCC. We show that under generic assumptions this disadvantageous behavior can be prevented. As a consequence, the topological type of a sequence of C-stationary points of the disjunctive relaxation can be traced up to the quadratic and biactive indices.

Keywords: mathematical programs with complementarity constraints, disjunctive optimization, nondegeneracy, index shift, topological type, S-stationarity

1. Introduction

We consider mathematical programs with complementarity constraints:

$$\text{MPCC : } \min_x f(x) \quad \text{s.t.} \quad x \in M$$

with

$$M = \{x \in \mathbb{R}^n \mid F_{1,j}(x) \cdot F_{2,j}(x) = 0, F_{1,j}(x) \geq 0, F_{2,j}(x) \geq 0, j = 1, \dots, \kappa\},$$

where the defining functions $f \in C^2(\mathbb{R}^n, \mathbb{R})$, $F_1, F_2 \in C^2(\mathbb{R}^n, \mathbb{R}^\kappa)$ are twice continuously differentiable. MPCCs have been widely studied in the literature for several decades, see e.g. [6]. They naturally appear if dealing with generalized Nash equilibrium problems, bilevel optimization, and semi-infinite programming to name a few applications.

The main difficulty of dealing with MPCCs is that their feasible set has kink-like structure. This nonsmoothness of the MPCC feasible set motivates the introduction of regularization methods. We provide a new regularization method, based on the structure of the feasible set of the relaxation from [4]. This new relaxation is given as the following disjunctive programm

$$\mathcal{D}(t) : \min_x f(x) \quad \text{s.t.} \quad x \in M^{\mathcal{D}(t)}$$

with

$$M^{\mathcal{D}(t)} = \left\{ x \in \mathbb{R}^n \mid \begin{array}{l} \max\{t - F_{1,j}(x), t - F_{2,j}(x)\} \geq 0, \\ F_{1,j}(x) \geq 0, F_{2,j}(x) \geq 0, j = 1, \dots, \kappa \end{array} \right\}.$$

where the parameter $t > 0$ is positive.

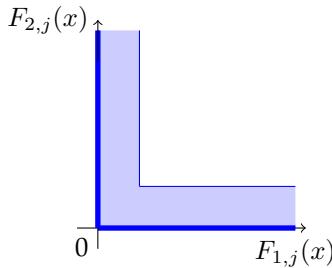


Figure 1: Feasible set $M^{\mathcal{D}(t)}$ for $\kappa = 1$.

For this relaxation, we provide convergence and well-posedness results. Furthermore, we reveal surprising effects when dealing with degenerate sequences.

2. Main results

For our main results, we consider different types of stationarity for MPCC. Needless to say that they are all necessary for minimizers of MPCC under the tailored linear independence constraint qualification, called MPCC-LICQ.

Definition 1 (Stationarity for MPCC). *A feasible point $\bar{x} \in M$ is called W-stationary for MPCC if there exist multipliers*

$$\bar{\sigma}_{1,j}, j \in a_{01}(\bar{x}), \quad \bar{\sigma}_{2,j}, j \in a_{10}(\bar{x}), \quad \bar{\varrho}_{1,j}, \bar{\varrho}_{2,j}, j \in a_{00}(\bar{x}),$$

such that

$$\begin{aligned} \nabla f(\bar{x}) = & \sum_{j \in a_{01}(\bar{x})} \bar{\sigma}_{1,j} \nabla F_{1,j}(\bar{x}) + \sum_{j \in a_{10}(\bar{x})} \bar{\sigma}_{2,j} \nabla F_{2,j}(\bar{x}) \\ & + \sum_{j \in a_{00}(\bar{x})} (\bar{\varrho}_{1,j} \nabla F_{1,j}(\bar{x}) + \bar{\varrho}_{2,j} \nabla F_{2,j}(\bar{x})). \end{aligned} \quad (1)$$

Moreover, the W-stationary point \bar{x} is called:

- C-stationary if

$$\bar{\varrho}_{1,j} \cdot \bar{\varrho}_{2,j} \geq 0 \text{ for all } j \in a_{00}(\bar{x}); \quad (2)$$

- M-stationary if

$$\bar{\varrho}_{1,j} > 0, \bar{\varrho}_{2,j} > 0 \text{ or } \bar{\varrho}_{1,j} \cdot \bar{\varrho}_{2,j} = 0 \text{ for all } j \in a_{00}(\bar{x}); \quad (3)$$

- S-stationary if

$$\bar{\varrho}_{1,j} \geq 0, \bar{\varrho}_{2,j} \geq 0 \text{ for all } j \in a_{00}(\bar{x}). \quad (4)$$

We are able to ensure C- and M-stationarity (for MPCC) of the limiting point under MPCC-LICQ.

Theorem 2 (Convergence for C- or M-stationarity). *Suppose a sequence of C- or M-stationary points $x^t \in M^{\mathcal{D}(t)}$ of $\mathcal{D}(t)$ with multipliers (ζ^t, η^t, ν^t) converges to \bar{x} for $t \rightarrow 0$. Let MPCC-LICQ be fulfilled at $\bar{x} \in M$. Then, \bar{x} is a C- or M-stationary point of MPCC, respectively.*

In order to ensure S-stationarity (for MPCC) of the limiting point, an additional assumption is required. One such assumption is MPCC-ND2 which is generic and based only on the limiting point. Precisely, it requires that the multipliers associated with biactive constraints do not vanish.

Proposition 3 (Convergence for S-stationarity). *Suppose a sequence of S-stationary points $x^t \in M^{\mathcal{D}(t)}$ of $\mathcal{D}(t)$ with multipliers (ζ^t, η^t, ν^t) converges to \bar{x} for $t \rightarrow 0$. Let MPCC-LICQ and MPCC-ND2 be fulfilled at $\bar{x} \in M$. Then, \bar{x} is an S-stationary point of MPCC.*

As part of our convergence analysis, we also reveal the following surprising phenomenon in connection to Proposition 3. In absence of MPCC-ND2, even a sequence of minimizers may converge to a saddle point.

With next Theorem 4 we provide conditions to prevent the change of topological type while converging. For this, an additional condition, MPCC-ND4, is needed. It requires that the multipliers associated with non-biactive constraints do not vanish. It should be mentioned that not only the topological type of nondegenerate points, i.e. the respective C-indices, are tracked but – as a novelty – different indices are tracked separately. In particular, MPCC-QI and DISJ-QI are the numbers of negative eigenvalues of the restricted Hessian matrix of the respective Lagrange function, see e.g. [2]. The biactive indices MPCC-BI and DISJ-BI count pairs of negative biactive multipliers and pairs of biactive relaxed constraints, respectively. Recall that nondegeneracy refers to the validity of tailored constraint qualifications, non-vanishing of certain multipliers, and second-order regularity, cf. [3] for MPCC and [1] for disjunctive optimization problems.

Theorem 4 (Convergence: QI and BI). *Suppose a sequence of nondegenerate C-stationary points $x^t \in M^{\mathcal{D}(t)}$ of $\mathcal{D}(t)$ with quadratic and biactive indices DISJ-QI and DISJ-BI, respectively, converges to \bar{x} for $t \rightarrow 0$. If $\bar{x} \in M$ is a nondegenerate C-stationary point of MPCC with multipliers $(\bar{\sigma}, \bar{\varrho})$, then we have for its quadratic and biactive indices MPCC-QI and MPCC-BI, respectively:*

$$\max \{ \text{DISJ-QI} - |a_{01}^0(\bar{x})| - |a_{10}^0(\bar{x})|, 0 \} \leq \text{MPCC-QI} \leq \text{DISJ-QI}$$

and

$$\text{MPCC-BI} = \text{DISJ-BI}.$$

Here, $|a_{01}^0(\bar{x})|$ gives the number of vanishing multipliers associated with a vanishing $F_{1,j}(\bar{x})$ and a non-vanishing $F_{2,j}(\bar{x})$. Likewise, $|a_{10}^0(\bar{x})|$ gives the number of vanishing multipliers associated with a non-vanishing $F_{1,j}(\bar{x})$ and a vanishing $F_{2,j}(\bar{x})$. If additionally MPCC-ND4 holds at \bar{x} , then $\text{MPCC-QI} = \text{DISJ-QI}$.

For its proof, we show that the Hessian of the Lagrange function of the relaxation approaches the Hessian of the Lagrange function of MPCC. A careful analysis of the restricting tangent spaces yields the result.

The well-posedness of the disjunctive relaxation is provided by the following result.

Theorem 5 (Well-posedness: *QI* and *BI*). *Let $\bar{x} \in M$ be a nondegenerate C-stationary point of MPCC with quadratic and biactive indices MPCC-*QI* and MPCC-*BI*, respectively. Assume that MPCC-ND4 holds at \bar{x} . Then, for all sufficiently small t there exists a nondegenerate C-stationary point $x^t \in M^{\mathcal{D}(t)}$ of $\mathcal{D}(t)$ within a neighborhood of \bar{x} , which has the same respective quadratic and biactive indices*

$$\text{DISJ-QI} = \text{MPCC-QI}, \quad \text{DISJ-BI} = \text{MPCC-BI}.$$

Moreover, for any fixed t sufficiently small, such x^t is the unique C-stationary point of $\mathcal{D}(t)$ in a sufficiently small neighborhood of \bar{x} .

References

- [1] Jongen, H. Th., Rückmann, Jan-J., and Stein, O. (1997). *Disjunctive Optimization: Critical Point Theory*. SIAM Journal on Optimization, 93: 321–336.
- [2] Jongen H. Th., Jonker P., and Twilt F. (2000). *Nonlinear Optimization in Finite Dimensions*. Kluwer Academic Publishers, Dordrecht.
- [3] Jongen H. Th., Shikhman V., and Rückmann Jan-J. (2009). MPCC: Critical Point Theory. SIAM Journal on Optimization, 20: 473–484.
- [4] Kanzow, C. and Schwartz, A. (2013). *A New Regularization Method for Mathematical Programs with Complementarity Constraints with Strong Convergence Properties*. SIAM Journal on Optimization, 23: 770–798.
- [5] Kanzow, C. and Schwartz, A. (2015). *The Price of Inexactness: Convergence Properties of Relaxation Methods for Mathematical Programs with Complementarity Constraints Revisited*. Mathematics of Operations Research, 40: 253–275.
- [6] Facchinei, F. and Pang, J.-S. (2003). *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer New York, NY.

Applying Tsoukalas-Mitsos convex relaxations in dynamic global optimization

Ho-Ching Chui,¹ Kamil A. Khan¹

¹*McMaster University, Hamilton, Canada, kamilkhan@mcmaster.ca*

Abstract We introduce an approach for constructing lower bounds of nonconvex nonlinear programs with ODE solutions embedded, for use in global optimization methods, and employing a recent convex relaxation rule by Tsoukalas and Mitsos in a new way. Our lower bounds are always at least as tight as those of previous approaches, and are significantly tighter in several numerical examples; one example is briefly presented in this extended abstract.

Keywords: Convex relaxations, Ordinary differential equations, Lower bounds

1. Problem formulation

Throughout this document, inequalities involving vectors apply to each component simultaneously. Let $P \subset \mathbb{R}^{n_p}$ be an interval of the form $P = \{\mathbf{p} \in \mathbb{R}^{n_p} : \mathbf{p}^L \leq \mathbf{p} \leq \mathbf{p}^U\}$. Overall, we consider a similar setup to [1]. We consider the dynamic optimization problem:

$$\min_{\mathbf{p} \in P} J(\mathbf{p}), \quad (1)$$

where the objective function J is defined in terms of a known cost function $g : \mathbb{R}^{n_p} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}$:

$$J(\mathbf{p}) := g(\mathbf{p}, \mathbf{x}(t_f, \mathbf{p})),$$

where $t \mapsto \mathbf{x}(t, \mathbf{p})$ denotes the solution on $I := [0, t_f] \rightarrow \mathbb{R}^{n_x}$ of the following parametric ordinary differential equation (ODE):

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{p}, \mathbf{x}), \quad \mathbf{x}(0, \mathbf{p}) = \mathbf{x}_0(\mathbf{p}). \quad (2)$$

Suppose that the functions g , \mathbf{f} , and \mathbf{x}_0 are continuously differentiable, and that the parametric solution $\mathbf{x}(t, \mathbf{p})$ of (2) exists on $I \times P$. Standard ODE theory implies that this solution is unique. The objective function J is thus continuously differentiable as well, but is not assumed to be convex.

We suppose that we seek to solve the problem (1) to global optimality by a branch-and-bound-based approach. Thus, as a critical subproblem in this approach, we seek to evaluate a useful lower bound J^L for which $J^L \leq J(\mathbf{p})$ for each $\mathbf{p} \in P$. Nontrivial constraints are compatible with our new approaches, but we neglect them here for simplicity.

To proceed, suppose that we have access to continuous convex relaxations of g on any interval subdomain, along with *state bounds* $(\mathbf{x}^L, \mathbf{x}^U)$ and *state relaxations* $(\mathbf{x}^{cv}, \mathbf{x}^{cc})$ (in the sense of [3]) for the ODE (2).

Two established concepts will be relevant here. First, Tsoukalas and Mitsos developed a general rule to construct convex relaxations of composite functions $h(\mathbf{p}) = \psi(\phi(\mathbf{p}))$ [6]. Roughly, given convex/concave relaxations ψ^{cv}/ψ^{cc} of ψ , and similarly for ϕ , a convex relaxation of h on P is given by:

$$h^{cv}(\mathbf{p}) := \min_{\zeta \in Z} \psi^{cv}(\zeta) \quad \text{subject to: } \phi^{cv}(\mathbf{p}) \leq \zeta \leq \phi^{cc}(\mathbf{p}). \quad (3)$$

Tsoukalas and Mitsos applied this rule in cases where ψ is particularly simple (e.g. a single operation from a scientific calculator), and the optimization problem (3) always has a closed-form solution. Nevertheless, their rule remains valid even when there is no closed-form solution.

Next, the primary established way to convert state relaxations of the ODE (2) into a lower bound on the NLP (1) is by minimizing a convex relaxation constructed using *generalized McCormick relaxations* [4, 7].

2. New approach

Given the setup of Section 1, we suggest computing a lower bound for (1) as follows. Let g^{cv} denote a continuous convex relaxation of the cost function g on the domain $P \times [\mathbf{x}^L(t_f), \mathbf{x}^U(t_f)]$. Then, compute:

$$J^L := \min_{\substack{\mathbf{p} \in P \\ \boldsymbol{\xi} \in \mathbb{R}^n}} g^{cv}(\mathbf{p}, \boldsymbol{\xi}) \quad \text{subject to: } \mathbf{x}^{cv}(t_f, \mathbf{p}) \leq \boldsymbol{\xi} \leq \mathbf{x}^{cc}(t_f, \mathbf{p}), \quad (4)$$

noting that this optimization problem is a convex NLP. We posit that this yields a valid, useful lower bound for (1).

2.1 Interpretation and validity

Consider the following variant of a Tsoukalas-Mitsos relaxation J^{cv} of J , viewed as a composition of g and \mathbf{x} :

$$J^{\text{cv}}(\mathbf{p}) := \min_{\boldsymbol{\xi} \in \mathbb{R}^n} g^{\text{cv}}(\mathbf{p}, \boldsymbol{\xi}) \quad \text{subject to: } \mathbf{x}^{\text{cv}}(t_f, \mathbf{p}) \leq \boldsymbol{\xi} \leq \mathbf{x}^{\text{cc}}(t_f, \mathbf{p}).$$

With J^L defined by (4), observe that $J^L = \min_{\mathbf{p} \in P} J^{\text{cv}}(\mathbf{p}) \leq \min_{\mathbf{p} \in P} J(\mathbf{p})$, and so (4) does indeed provide a correct lower bound of (1).

Hence, (4) may be viewed as a direct but unconventional application of the Tsoukalas-Mitsos relaxations [6]. The lower bound (4) may also be viewed as a partial application of the Auxiliary Variable Method [5], with the state relaxations of the ODE (2) left unaffected. To our knowledge the bounding approach (4) has not been previously documented, though it is an application of established methods.

2.2 Versatility and tightness

One of the benefits of our approach is that (4) does not require McCormick-style relaxations of the cost function g . A second benefit is as follows, with proof omitted in this extended abstract. If the convex relaxation g^{cv} in (4) is chosen to be a McCormick relaxation of g [2], then the lower bound J^L in (4) is always no less than the corresponding generalized-McCormick-based lower bound from [3], and may be significantly greater. We expect that both methods will incur similar computational costs for evaluating a single lower bound.

3. Numerical example

We implemented our new approach in the language Julia, using the package `McCormick.jl` [8] to construct McCormick-style convex relaxations, `DifferentialEquations.jl` to solve ODEs numerically, and `IPOPT` in `JuMP` to solve NLPs to local optimality. This implementation was applied to several instances of (1), and in each case, our new approach using McCormick relaxations of g significantly outperformed the established approach using generalized McCormick relaxations. One such result is shown in Figure 1.

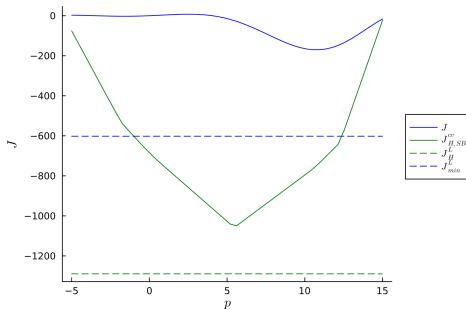


Figure 1: The objective function J (solid blue) in a small instance of (1), a convex relaxation by an established approach based on generalized McCormick relaxations (solid green), our new lower bound J^L (dashed blue), and a lower bound based on interval arithmetic (dashed green).

References

- [1] Ho-Ching Chui and Kamil A. Khan. Applying sampling-based convex relaxations to dynamic process models. In *IFAC-PapersOnLine*, volume 58, pages 337–341, 2024.
- [2] Alexander Mitsos, Benoit Chachuat, and Paul I Barton. McCormick-based relaxations of algorithms. *SIAM Journal on Optimization*, 20(2):573–601, 2009.
- [3] Joseph K Scott and Paul I Barton. Improved relaxations for the parametric solutions of ODEs using differential inequalities. *Journal of Global Optimization*, 57(1):143–176, 2013.
- [4] Joseph K Scott, Matthew D Stuber, and Paul I Barton. Generalized McCormick relaxations. *Journal of Global Optimization*, 51(4):569–606, 2011.
- [5] Mohit Tawarmalani and Nikolaos V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Springer, Dordrecht, 2002.
- [6] Angelos Tsoukalas and Alexander Mitsos. Multivariate McCormick relaxations. *Journal of Global Optimization*, 59(2-3):633–662, 2014.
- [7] Achim Wechsung, Joseph K. Scott, Harry A. J. Watson, and Paul I. Barton. Reverse propagation of McCormick relaxations. *Journal of Global Optimization*, 63(1):1–36, September 2015.
- [8] M. E. Wilhelm and M. D. Stuber. EAGO.jl: Easy Advanced Global Optimization in Julia. *Optimization Methods and Software*, 37(2):425–450, 2022.

A Positive Semidefinite Safe Approximation of Multivariate Distributional Robust Constraints Determined by Simple Functions

Jan Dienstbier,¹ Frauke Liers,¹ Jan Rolfs^{2,3}

¹*Friedrich-Alexander University Erlangen-Nuremberg, Erlangen, Germany*

²*Linköping University, Linköping, Sweden, jan.rolfs@liu.se*

³*RISE - Research Institutes of Sweden*

Abstract Single-level reformulations of (non-convex) distributionally robust optimization (DRO) problems are often intractable, as they contain semiinfinite dual constraints. Based on such a semiinfinite reformulation, we present a safe approximation, that allows for the computation of feasible solutions for DROs that depend on nonconvex multivariate simple functions. Moreover, the approximation allows to address ambiguity sets that can incorporate information on moments as well as confidence sets. The typical strong assumptions on the structure of the underlying constraints, such as convexity in the decisions or concavity in the uncertainty found in the literature can at least in part be overcome. In order to achieve algorithmic tractability, the presented safe approximation is then realized by a discretized counterpart for the semiinfinite dual constraints. The approximation leads to a computationally tractable mixed-integer positive semidefinite problem for which state-of-the-art software implementations are readily available. The tractable safe approximation provides sufficient conditions for distributional robustness of the original problem, i.e., obtained solutions are provably robust.

Keywords: Distributionally Robust Optimization, Mixed-Integer Programming, Semi-infinite Programming

1. Introduction

In this talk, we consider distributionally robust optimization problems (DRO), i.e., we aim to optimize *decision variables* $x \in \mathbb{R}^k$ contained in a polytope \mathcal{X} with respect to random parameters distributed to an unknown probability distribution \mathbb{P} . That is, we aim to solve a DRO of the following form:

$$\max c^\top x : \tag{1a}$$

$$x \in \mathcal{X}, \tag{1b}$$

$$b \leq \min_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}}(v(x, t)), \tag{1c}$$

where (1c) is called a DRO constraint. In particular, we denote the *ambiguity set* of probability measures by \mathcal{P} . Please note that $\mathcal{P} = \{\delta_t : t \in T\}$ with Dirac point measures δ_t turns (1) into a standard robust optimization problem and $\mathcal{P} = \{\mathbb{P}\} \Rightarrow$ turns (1) into a stochastic optimization problem. In this talk, we further restrict the considered random variable $t \in T$ to compact support $T \subseteq \mathbb{R}^k$, where $k \geq 2$. For $k = 1$, we refer to [3]. It is worth noting, that $k \geq 2$ is a significantly more challenging case as it does not only increase the dimension of the (1), but also allows to incorporate correlated random variables. As we will see, computationally, this increase of modeling capability has to be paid for with semidefinite constraints compared to linear ones considered in [3].

Two major challenges in DRO originate from the exact specification of the ambiguity set \mathcal{P} , where a tradeoff between realism and tractability is to be found and the potential nonconvexity of the function v that connects decision and uncertainty. For the former, there are mainly two types of ambiguity sets considered in the literature, namely discrepancy based ambiguity sets, that are based on the distance to an estimate of the true probability measure \mathbb{P} and *moment-based* ambiguity sets. Moment based approaches merely rely on information on the moments, i.e., on estimates μ, Σ of the expected value and covariance matrix, see e.g. [1], [2] for further details.

As we aim to approach general nonconvex functions v , we consider *indicator functions*

$$v(x^-, x^+, t) = a \mathbb{1}_{[x^-, x^+]}(t) = \begin{cases} a & \text{if } t \in [x^-, x^+] \\ 0 & \text{otherwise} \end{cases}$$

with a constant $a > 0$, which can be seen as one of the easiest type of nonconvex functions.

2. Main Result

Let us denote the cone of nonnegative continuous functions supported on T by $\mathcal{C}(T)_{\geq 0}$. Then, we demonstrate that for certain moment-based ambiguity sets it is possible to reformulate RHS in constraint (1c) as a semiinfinite conic program (SIP) with respect to $\mathcal{C}(T)_{\geq 0}$:

$$\max_{\substack{Y_1 \in \mathcal{S}_{\geq 0}^{k+1}, Y_2 \in \mathcal{S}_{\geq 0}^k, \\ z \in \mathbb{R}_{\geq 0}^I}} \langle D, Y_2 \rangle + \sum_{i \in I} d_i z_i \quad (2a)$$

$$\text{s.t. } a \mathbb{1}_{[x^-, x^+]^c}(t) - \langle A_1, Y_1 \rangle + \langle A_2, Y_2 \rangle - \sum_{i \in I} a'_i \mathbb{1}_{T_i}^c(t) z_i \in \mathcal{C}(T)_{\geq 0}, \quad (2b)$$

where $D \in \mathcal{S}^k$ symbolizes a bound on the covariance matrix of t , d_i bounds on confidence sets T_i on t . Moreover, the coefficients A_1, A_2 as well as a'_i are related to the expectation, variance and confidence sets respectively and the function $\mathbb{1}^c$ denotes a continuous approximation of the indicator function $\mathbb{1}$, which exists due to Urysohn's lemma. With slightly modified proofs from the literature, it is now possible to show:

Theorem 1 (Dienstbier, Liers, R.). *The duality gap of the minimization problem in (1c) and (2) is zero.*

Similarly, to the classical dualization trick in robust optimization, Theorem 1 allows us to incorporate (2) into the original DRO (1). However, this again results in an SIP, which is generally intractable. Consequently, the main result of the presented talk is to present a safe approximation scheme, which is based on Mixed-Integer Semidefinite Programming (MISDP).

Theorem 2 (Dienstbier, Liers, R.). *Let T_N be a discretization of T and $L > 0$ be the Lipschitz constant of the polynomial part in (2b). Suppose $x^-, x^+ \in \mathcal{X}$ implies $x^- \leq x^+$ and set $\tilde{b}_{\bar{t}} = 0$ for every $\bar{t} \notin T_N$. Then, a solution to the following MISDP is feasible for (1).*

$$\max_{x^-, x^+, \tilde{b}, \Delta^-, \Delta^+, Y_1, Y_2, z} c^\top (x^-, x^+)^\top \quad (3a)$$

$$\text{s.t. } \langle D, Y_2 \rangle + \sum_{i \in I} d_i z_i \geq b \quad (3b)$$

$$\begin{aligned} & \tilde{a}_{\bar{t}} - \langle A_1, Y_1 \rangle + \langle A_2, Y_2 \rangle \\ & - \sum_{i \in I} a'_i \mathbb{1}_{T_i}^c(\bar{t}) z_i - L \delta_N \sqrt{k} \geq 0 \end{aligned} \quad \forall \bar{t} \in T_N, \quad (3c)$$

$$x^-, x^+ \in \mathcal{X}, \tilde{b}, \Delta^+, \Delta^- \in \{0, 1\}^{T_N} \cap \mathcal{X}', \quad (3d)$$

$$Y_1 \in \mathcal{S}_{\geq 0}^{k+1}, Y_2 \in \mathcal{S}_{\geq 0}^k, z \in \mathbb{R}_{\geq 0}^I, \quad (3e)$$

where $\mathcal{X}, \mathcal{X}'$ denote polytopes.

For a better overview, let us elaborate on the variables in (3). Recall that the decision variables from the first stage are denoted by x^-, x^+ . The auxiliary variables $\Delta^-, \Delta^+ \in \{0, 1\}^{T_N} \cap \mathcal{X}'$ allow to connect the binaries $\tilde{b}_{\bar{t}}$ with the continuous variables x^-, x^+ in a way that $\tilde{b}_{\bar{t}}$ serves as a discrete approximation of $\mathbb{1}_{[x^-, x^+]}^c(t)$. The above program is then complemented by the dual variables of the DRO constraint (1c). For further details we refer the interested reader to [4].

Computing the MISDP in (3) is challenging for current MISDP solvers, but an academic example on how to capture the probability mass of $\mathbb{P} \in \mathcal{P}$ within a box $[x^-, x^+]$ is presented in [4]. However, as running times strongly scale with the discretization width of T_N , future research will aim to downsizing the model without incurring loss in obtained quality.

Acknowledgments

The authors wish to thank Dr. Shudian Zhao and Johanna Skåntorp for their support.

References

- [1] Delage, E. and Ye, Y (2010). Distributionally Robust Optimization Under Moment Uncertainty with Application to Data-Driven Problems. *Oper. Res.* 58. 595-612.
- [2] Wiesemann, W., Kuhn, D. and Sim, M. (2014). Distributionally Robust Convex Optimization. *Oper. Res.* 62. 1358-1376
- [3] Dienstbier, J., Liers, F. and Rolfs, J. (2023) A Safe Approximation Based on Mixed-Integer Optimization for Non-Convex Distributional Robustness Governed by Univariate Indicator Functions. <https://arxiv.org/abs/2301.11185>
- [4] Dienstbier, J., Liers, F. and Rolfs, J. (2023) A Positive Semidefinite Safe Approximation of Multivariate Distributionally Robust Constraints Determined by Simple Functions. <https://arxiv.org/abs/2310.05612>

Relaxations of the Delay-Constrained Maximum Concurrent Flow Problem

W. Ben-Ameur,¹ G. Beraud-Sudreau,² H. Kerivin,³ S. Martin,²

¹*SAMOVAR, TélécomSudParis, Institut Polytechnique de Paris, Palaiseau, France*

²*Huawei Technologies France, Boulogne-Billancourt, France*

³*Limos, University Clermont Auvergne, Clermont-Ferrand, France*

Abstract This paper investigates the Maximum Concurrent Flow problem under delay constraints. Delay constraints are only imposed on paths actually used, thus introducing a conditional constraint complicating the optimization process and making the problem NP-Hard. A new convex relaxation based on convex envelopes is proposed and compared to previously known relaxations both theoretically and numerically. We also prove that the problem is strongly NP-hard even in the single-source case.

Keywords: Multi-commodity flows, On/off constraints, Convex envelope

1. Problem presentation

1.1 The delay-constrained maximum concurrent flow problem

Let $D = (V, A)$ be a directed graph where V is the set of vertices and A is the set of arcs. Each arc $a \in A$ has a capacity c_a . A set of commodities K is considered. A commodity $k \in K$ has a source s^k , a sink (or destination) t^k and a size (or volume) b^k . We also use P^k to denote a given nonempty subset of paths from s^k to t^k . For $p \in P^k$, x_p^k is the proportion of the commodity $k \in K$ carried through the path p . A given value d^k represents the maximum delay related to commodity k that should not be exceeded. The

delay through an arc a , denoted by d_a , is given by the Kleinrock formula $d_a = 1/(c_a - x_a)$ where x_a denotes the flow (or load) of a . Hence, if a path $p \in P^k$ is actually used (i.e., if $x_p^k > 0$), the total delay through p should be less than d^k . The Delay-Constrained Maximum Concurrent Flow problem (noted DCMCF) can then be expressed as follows:

$$\begin{aligned} \text{DCMCF:} \quad & \max \gamma \\ & \sum_{p \in P^k} x_p^k \geq \gamma \quad \forall k \in K \quad (1) \\ & x_a \geq \sum_{k \in K, p \in P^k | a \in p} x_p^k b^k \quad \forall a \in A \quad (2) \\ & x_a \leq c_a \quad \forall a \in A \quad (3) \\ & \sum_{a \in p} \frac{1}{c_a - x_a} \leq d^k \quad \text{if } x_p^k > 0 \quad \forall p \in P^k \quad (4) \\ & x_p^k \geq 0 \quad \forall k \in K, p \in P^k. \quad (5) \end{aligned}$$

Observe that γ represents here the maximum proportion of each commodity that can be carried through the network. Constraints (1) make the connection between γ and the x_p^k proportions, while (2) allows to express the flow on an arc a as the sum of flows on paths containing a . Constraints (3) introduce capacity limitations. The delay constraints are stated in (4). Observe that (4) is active if and only if the corresponding path is used (i.e., if $x_p^k > 0$). After multiplication by x_p^k , (4) can be reformulated as:

$$\sum_{a \in p} \frac{x_p^k}{c_a - x_a} \leq x_p^k d^k \quad \forall k \in K, p \in P^k. \quad (6)$$

1.2 Some related work

The Delay-Constrained Multi-Commodity flow problem has been introduced, with a cost minimization objective, in [1]. NP-hardness has been shown there and several relaxations have been proposed such as relaxing (4) into $\sum_{a \in p} \frac{(x_p^k)^2}{c_a - x_a} \leq x_p^k d^k$ (x_p^k is less than 1 in the version studied in [1]).

Introducing a new set of binary variables z_p^p representing the usage of path $p \in P^k$ for commodity $k \in K$, a big-M relaxation is proposed in [2] where constraints $\sum_{a \in p} \frac{1}{c_a - x_a} \leq d^k z_p^k + d_p^{k \max} (1 - z_p^k)$ are considered and $d_p^{k \max}$ is an upper-bound of the delay on p . Another relaxation based on disjunctive programming is proposed in [2] replacing (4) by constraints

$$\sum_{a \in p} \frac{(z_p^k)^2}{z_p^k c_a - x_a + (1-z_p^k) u_a} \leq z_p^k d^k, \text{ where } u_a \text{ is an upper-bound of variable } x_a \text{ for every arc } a \in A.$$

2. Main contributions

2.1 Convex relaxation of the delay constraints

In order to properly relax constraints (4), we studied the function f defined on $\{(x_p^k, x_a) \in [0, u_p^k] \times [l_a, u_a] | x_a \geq b^k x_p^k\}$ by $f(x_p^k, x_a) = \frac{x_p^k}{c_a - x_a}$ where l_a is a lower bound of x_a and u_p^k is an upper bound of x_p^k . We show that the convex envelope can be exactly expressed as the pointwise maximum of 3 convex functions $\check{f}(x_p^k, x_a) = \sup(r(x_p^k, x_a), s(x_p^k, x_a), t(x_p^k, x_a))$, with :

$$r(x_p^k, x_a) = \frac{x_p^k}{c_a - \frac{b^k x_p^k}{1 - \frac{x_a - b^k x_p^k}{u_a}}}, \quad s(x_p^k, x_a) = \frac{(x_p^k)^2}{x_p^k c_a - u_p^k x_a + (u_p^k - x_p^k) u_a} \text{ and } t(x_p^k, x_a) =$$

$\frac{x_p^k}{c_a - l_a}$. A relaxation is obtained by writing that $\sum_{a \in p} \check{f}(x_p^k, x_a) \leq x_p^k d^k$ for each commodity $k \in K$ and path $p \in P^k$. Note that one can easily prove that the set of constraints derived from disjunctive programming in [2] are dominated by constraints $\sum_{a \in p} s(x_p^k, x_a) \leq x_p^k d^k$, implying that the new relaxation is tighter than the one described above.

Notice that the tightness of the relaxations presented above depends on the upper and lower bounds used for x_p^k and x_a and the paths delays. Closed formulas were presented in [1] to compute values of u_a and u_p^k . It is also possible to compute these bounds by iteratively solving convex problems using the relaxation above. While requiring significant processing efforts, computing these bounds leads to tighter relaxations.

2.2 Computational complexity

DCMF has been proved to be NP-hard in [1]. We prove that it is strongly NP-hard even in the single-source case. The proof is based on a reduction from the 3-partition problem.

3. Preliminary numerical experiments

We tested the relaxations described above on instances of the SND-Library. For each instance, the 10 shortest paths were computed using Yen's algo-

rithm and delays d^k were set 10% above the longest delay, along all paths, if the load of all arcs in the network was 0.

Figure 1 shows a performance chart of the relative gaps (between upper-bounds derived from the relaxations and a lower-bounds derived from a heuristic) for the different relaxations described above. These results confirm that the relaxation suggested in [2] ("Disjunct.") is indeed tighter than the naive big-M formulation ("Big-M"), and that the full computation of the convex envelope ("Convex-Envelope") leads to further tightening of the relaxation. It also shows that gaps are reduced when the variable bounds are computed more finely ("Opt") compared to bounds computed with the closed formula introduced in [1] ("CF").

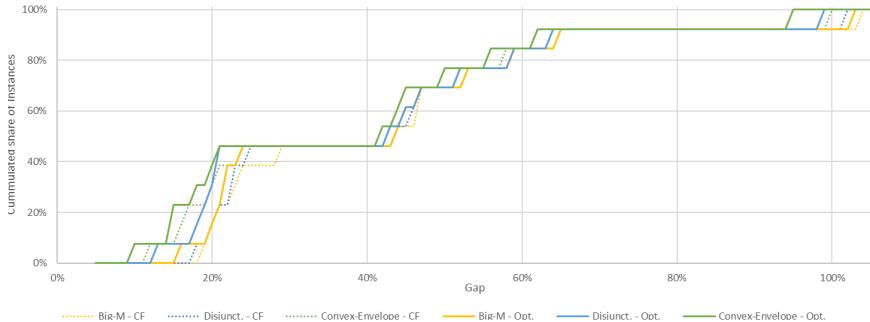


Figure 1: Relaxation gap performance profile - with our without optimization of the variables bounds

4. Conclusion

We proposed a new relaxation of the Delay-constrained multi-commodity flow problem, and showed both theoretically and through numerical experiments that it dominates the already known relaxations of this problem. Further work should provide theoretical guarantees for these relaxations as well as more extensive numerical results.

References

- [1] W. Ben-Ameur and A. Ouorou. Mathematical models of the delay constrained routing problem. *Algorithmic Operations Research*, 1:94–103, 2006.
- [2] H. Hijazi, P. Bonami, G. Cornuéjols, and A. Ouorou. Mixed-integer nonlinear programs featuring "on/off" constraints. *Computational Optimization and Applications*, 52(2):537–558, 2012.

SNoGloDe: A Flexible Tool for Decomposition & Global Optimization

Georgia Stinchfield¹, Megan Walsh¹, Yankai Cao², Carl Laird ¹

¹*Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA*
claird@andrew.cmu.edu

²*Chemical and Biological Engineering, University of British Columbia, B.C., Canada*
yankai.cao@ubc.ca

Abstract Tailored decomposition and algorithmic approaches quickly become necessary to efficiently solve large-scale nonlinear optimization problems to global optimality. We present a flexible software package, SNoGloDe: Structured Nonlinear Global Decomposition, implemented in Python using the algebraic modeling language Pyomo [1]. SNoGloDe exploits the block angular structure of an optimization problem, decomposing and employing a tailored spatial branch-and-bound (sBnB) tree with a plug-and-play software framework to exploit problem-specific information while solving subproblems in parallel.

Keywords: Parallel Decomposition, Nonlinear Stochastic Programming, Software

1. Introduction

The need to efficiently solve large-scale nonlinear optimization problems to global optimality arises across various fields. Although current global optimization solvers handle small problem instances well, we target large-scale cases where decomposition is essential for tractability, enabling parallel solution of smaller subproblems using HPC resources.

While many decomposition methods (e.g., Dantzig-Wolfe, Benders) are well established, few extend to the global optimization of nonconvex problems. Li and Grossmann (L&G) [4] introduced a Benders-based branch-

and-cut method for two-stage stochastic programs with nonconvex constraints, using a priority-driven sBnB strategy similar to that proposed by Cao and Zavala (C&Z) [2]. Both approaches branch on first-stage variables and solve subproblems globally (branching on the second-stage decisions implicitly). They differ by their relaxation techniques: C&Z use the disaggregated subproblems, whereas L&G use Benders and Lagrangian cuts. In this work, we will focus on the approach proposed by C&Z. While the sBnB element in both C&Z and L&G suffers from the clustering problem (with one exception in the case of L&G) [6], these priority-branching based decomposition approaches provide practical performance on many large-scale optimization problems.

2. Methodology & Framework

We have developed a Python-based tool, SNoGloDe (Structured Nonlinear Global Decomposition), based on the approach of C&Z. SNoGloDe is intended to solve block-angular optimization problems. The algorithm prioritizes branching on the complicating variables within a sBnB framework. The lower bounding problem is formed by dropping coupling constraints. By obtaining globally optimal solutions to the lower bounding problem, we implicitly branch on the subproblem-specific variables, thereby maintaining convergence guarantees to the global optimum. As many global optimization solvers can reliably tackle small nonconvex problems, we do not require additional relaxations. However, SNoGloDe offers a plug-in option that allows users to refine relaxations further, enabling easy integration of the L&G approach or the development of custom relaxations.

Generating upper bounds within a sBnB tree is highly customizable, which is why SNoGloDe offers a custom plug-in. We generate a candidate solution for the complicating variables within each node of the sBnB tree, fixing the candidate within each subproblem (implicitly satisfying the complicating constraints) and solving for the subproblem-specific variables (in parallel) to determine an upper bound. Additionally, efficiently traversing a sBnB tree is a challenge. Selecting which complicating variable to branch on, where to split the selected variable domain, and which node to solve next significantly influences computational time. In SNoGloDe, we offer various branching and queuing methods while also providing plug-in functions to define problem-specific strategies. We offer parallel capabilities, using a Message-Passing Interface (MPI) configuration to solve n -subproblems across k -ranks ($n \leq k$) for the lower (dropping complicating constraints) and upper bound (fixing complicating variables).

3. Case Study: Stochastic PID Tuner

We test our framework on a two-stage stochastic PID tuner case study from C&Z to tune the controller to handle a range of uncertainty realizations. We consider changes in the setpoint, model uncertainty, and disturbance fluctuations as uncertain parameters. The first-stage decisions are the controller gains, while the second-stage decisions are the controller state trajectories over time for each uncertain scenario.

$$\min_{x_s(t), K_p, K_i, K_d} \sum_{s \in S} \int_0^T e_s(t)^2 dt \quad (1a)$$

$$\text{s.t. } \frac{dx_s(t)}{dt} = -\tau_s^x x_s(t)^2 + \tau_s^u u_s(t) + \tau_s^d d_s \quad \forall s \in S \quad (1b)$$

$$u_s(t) = K_p e_s(t) + K_i \int_0^T e_s(T) dT + K_d \frac{de_s(t)}{dt} \quad \forall s \in S \quad (1c)$$

$$e_s(t) = x_s(t) - \bar{x}_s \quad \forall s \in S \quad (1d)$$

The objective (1a) aims to minimize the error between the state, x_s , and the setpoint, \bar{x}_s , over all uncertain scenarios $s \in S$. Constraint (1b) tracks changes over time for the state in each scenario $s \in S$, x_s , while Constraint (1c) tracks the control response over time. We use the Pyomo DAE extension [5] to discretize the differential equations and integrals.

Addressing (1) with a small number of scenarios (i.e., $|S|$) can be solved in a reasonable time using global solvers. However, as the number of scenarios increases, global solvers (without decomposition) struggle to close the optimality gap. For example, consider $|S| = 500$. Given an hour, Gurobi [3] (with nonconvex settings) reaches a gap of 14.9% while SNoGloDe (run serially in the same time) reaches a gap of 3.7%. Given 10 parallel processors, SNoGloDe improves this gap to 1% in the same 1 hour time window.

4. Conclusions & Future Work

Decomposition coupled with priority branching strategies within a sBnB tree, as proposed by C&Z [2], provides an efficient means for solving large-scale block-angular optimization problems to ϵ -global optimality. Not only does decomposition provide tractability for some large-scale optimization problems, but the decomposition strategy also allows for the parallel solution of subproblems. While this method will suffer from the clustering problem [6], in many cases, we are satisfied with the ϵ -optimality gap pro-

vided by these methods, especially when off-the-shelf solvers cannot reach a reasonable optimality gap alone. In future work, we aim to strengthen the lower bound by updating the Lagrange multipliers associated with the dualized non-anticipativity constraints. Currently, C&Z sets all multipliers to zero, not updating them as the algorithm progresses.

Acknowledgments

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This work used Bridges-2 at Pittsburgh Supercomputing Center through allocation MTH240062 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

- [1] Bynum, M. L., Hackebeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Siirola, J. D., Watson, J., & Woodruff, D. L. (2021). *Pyomo—optimization modeling in python*, 3rd ed. Springer Science & Business Media.
- [2] Cao, Y., & Zavala, V. M. (2019). A scalable global optimization algorithm for stochastic nonlinear programs. *Journal of Global Optimization*, 75(2), 393-416.
- [3] Gurobi Optimization, LLC (2024). Gurobi Optimizer Reference Manual. <https://www.gurobi.com>
- [4] Li, C., & Grossmann, I. E. (2019). A generalized Benders decomposition-based branch and cut algorithm for two-stage stochastic programs with nonconvex constraints and mixed-binary first and second stage variables. *Journal of Global Optimization*, 75, 247-272.
- [5] Nicholson, B., Siirola, J. D., Watson, J., Zavala, V. M., & Biegler, L. T. (2018). pyomo.dae: a modeling and automatic discretization framework for optimization with differential and algebraic equations. *Mathematical Programming Computation*, 10(2), 187-223.
- [6] Robertson, D., Cheng, P., & Scott, J. K. (2025). On the convergence order of value function relaxations used in decomposition-based global optimization of nonconvex stochastic programs. *Journal of Global Optimization*, 91(4), 701-742.

Session 3





New ideas in global optimization

Anatolii Kosolap¹

¹*Oles Honchar Dnipro National University, Dnipro, Ukraine, anivkos@ua.fm*

Abstract We present our method of exact quadratic regularization (EQR), which transforms global optimization problems into a problem of maximizing the vector norm on a convex set in $(n + 1)$ -dimensional space. The point of the global minimum becomes the closest to the origin and serves as an attraction point for the local solver. This simplifies the search for a global minimum. Computational experiments confirm the effectiveness of EQR.

Keywords: Global optimization, Exact quadratic regularization, Multimodal problems, Large-scale multimodal problems, Computational experiments.

1. Introduction

Most optimization models of applied problems are multimodal. Only a small number of such problems are linear or convex and unimodal. Multimodal problems are usually NP-hard. This does not mean that we cannot develop effective methods for solving multimodal problems. The example of the non-polynomial simplex method gives confidence in the development of practically effective methods for solving applied multimodal problems.

Existing deterministic and stochastic methods are not efficient [1]. This is confirmed by computational experiments for difficult test problems with unknown solutions. A convex relaxation of multimodal problems sometimes allows us to obtain optimal solutions, but often such solutions are far from optimal. The best transformation for global optimization problems is exact quadratic regularization. Extensive computational experiments confirm its effectiveness.

2. Exact quadratic regularization

The first idea is to find a transformation of the multimodal problem that simplifies the search for the global minimum.

Consider the following global optimization problem:

$$\min\{f_0(x) | f_i(x) \leq 0, i = 1, \dots, m, x \in E^n\}, \quad (1)$$

where all functions $f_i(x)$ are twice continuously differentiable, and x is a vector in n -dimensional Euclidean space E^n . Let the solution of the problem (1) exist, and let x^* be a global minimum of the problem (1).

The problem (1) is transformed into the following equivalent problem

$$\min\{\|x\|^2 | f_0(x) + s \leq \|x\|^2, f_i(x) \leq 0, i = 1, \dots, m\}, \quad (2)$$

where $\|x\|^2 = x_1^2 + \dots + x_n^2 + x_{n+1}^2$ and $s \geq \|x^*\|^2 - f_0(x^*)$.

After such a transformation, the point of the global minimum of the problem (2) will be closest to the origin. Obviously, from the constraints, $f_0(x^*) + s = \|x^*\|^2$ and $f_0(x^0) + s \leq \|x^0\|^2$, it follows that $\|x^*\|^2 \leq \|x^0\|^2$, where x^0 is the point of local minimum. In addition, transformation (2) allows the use of exact quadratic regularization to convert problem (2) into the following:

$$\begin{aligned} \min\{\|x\|^2 | f_0(x) + s + (r-1)\|x\|^2 \leq d, f_i(x) + r\|x\|^2 \leq d, i = 1, \dots, m, \\ r\|x\|^2 = d\}, \end{aligned} \quad (3)$$

where the parameter $r > 0$ is chosen so that the feasible region

$$\{x | f_0(x) + s + (r-1)\|x\|^2 \leq d, f_i(x) + r\|x\|^2 \leq d, i = 1, \dots, m\}$$

becomes convex.

The problem (3) is multimodal and very difficult, primarily due to the non-convex constraint $r\|x\|^2 = d$. Therefore, the second idea is to transform the problem (3) into the following:

$$\max\{\|x\|^2 | f_0(x) + s + (r-1)\|x\|^2 \leq d, f_i(x) + r\|x\|^2 \leq d, i = 1, \dots, m\}, \quad (4)$$

and the condition $r\|x\|^2 = d$ separately.

Theorem 1. Let (x^*, d^*) be the solution of problem (4) for the minimum $d = d^*$, and suppose the condition $r\|x^*\|^2 = d^*$ is satisfied. Then x^* is the global minimum point of the problem (1).

A minimum of d^* implies a minimum of $\|x^*\|^2$; thus, x^* is the global minimum point of the problem (1).

We find the minimum value of d over several iterations. To start the iterations, it is necessary to find the minimum allowable value of d . This value d_0 can be easily found by solving the following convex problem:

$$\min\{d|f_0(x) + s + (r - 1)\|x\|^2 \leq d, f_i(x) + r\|x\|^2 \leq d, i = 1, \dots, m\}. \quad (5)$$

We find the solution to the problem (1) using an algorithm:

Step 1: Choose s, r, x^0 , and solve the problem (5).

Step k: Solve problem (4) for $d = d_k$ ($d_k = d_{k-1} + \Delta d_k$, $\Delta d_k > 0$, $k = 1, \dots$). If the condition $r\|x^k\|^2 = d_k$ is satisfied, then the stop; x^k is the solution to the problem (1).

The proposed exact quadratic regularization has several advantages.

1. The starting point of the problem (4) is always feasible.
2. The transformed multimodal problem (4) can become unimodal (the solution to the problem (4) coincides with the solution to the convex problem (5)).
3. EQR can be used for a wide class of multimodal problems.
4. EQR allows us to solve large-scale multimodal problems.
5. EQR is easy to implement in software.
6. EQR has a simple geometric interpretation.

If $r \rightarrow \infty$, then the feasible region of the problem (4) approaches a sphere, and the problem (4) becomes unimodal.

3. Computational experiments

For a computer implementation of the EQR method, it is sufficient to have a local solver. We use Open Solver for Excel. Table 1 shows the results of solving the test unconstrained optimization problems with unknown solutions for $n = 100$. These test functions have been used in the past by many authors [2], [3] and hence they constitute a convenient platform for comparing computational results. The solutions obtained using the EQR method are compared with the results obtained by the Python solver, which uses evolutionary search. We see that the EQR method allowed us to obtain significantly better solutions. The solution to the problem (4) may depend on the choice of the values of r, x_{n+1} , and the step of changing d . The choice of d depends on the value of $|r\|x\|^2 - d|$, which decreases monotonically when the d increases. The iterations of the algorithm are shown in Table 2.

Table 1: The results of solving the test unconstrained optimization problems with unknown solutions.

<i>Problem</i>	<i>EQR best min</i>	<i>Best known min</i>
Egg holder	-89948.532	-89938(web)
Rana	-50865.166	-47332(web)
Adjman	-30.37418	-23.30464(py)
Bird	-5230.32938	-4097.8487 (py)
Liang's	-85466.348	-33162.8595(py)
Mishra 5	-0.164278	11.919746(py)
Mishra 6	-197.53142	-154.07665(py)
Ackley 3	-21766.8655	-19494.03687(py)
Siam	-175.64016	-51.496279(py)
Trefethen	-154.182453	17.02436(py)

Table 2: Algorithm iterations for the Bird function ($s = 12000, r = 1500$).

d_k	10857	20000	80000	120000	≥ 180000
$f_0(x^k)$	-1191.4	-2199.4	-4267.8	-5028.2	-5230.33

4. Summary

We used exact quadratic regularization to transform global optimization problems into vector norm maximization on a convex set. An efficient procedure for solving the transformed problem is proposed. For its implementation, it is sufficient to have a local solver. This allows us to solve large-scale multimodal problems with thousands of variables and constraints. The computational experiments confirm the effectiveness of EQR.

References

- [1] Locatelli, M., Schoen, F. (2021). (Global) Optimization: Historical notes and recent developments. *EURO Journal on Computational Optimization*. 9. 1-15.
- [2] Jamil, M., Yang, XS. (2013). A literature survey of benchmark functions for global optimization problems. *Int. J. Math. Model. Numer. Optim.* 4(2). 150-194.
- [3] Voglis, C., Lagaris, I.E. (2009). Towards “Ideal Multistart”. A stochastic approach for locating the minima of a continuous function inside a bounded domain. *Appl. Math. Comput.* 123. 216-229.

On Computational Evaluation of Lower Bounds for the Fractional Quadratic Program over the Standard Simplex

Antonio Sasaki¹ Sophie Demassey¹ Valentina Sessa¹

¹*Mines Paris, Université Paris Sciences et Lettres, France,
 {antonio.sasaki, sophie.demassey, valentina.sessa} @minesparis.psl.eu*

Abstract

This work evaluates techniques for computing a lower bound for the Standard Fractional Quadratic Program (StFQP). These results may be useful to be integrated into algorithms to globally solve this type of problem. Moreover, we implement computational experiments to assess the trade-off between cost and tightness of the computed lower bounds.

Keywords: Fractional quadratic programming, global optimization.

1. Introduction

Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ be real symmetric matrices, with \mathbf{B} positive-definite (PD), and let the *standard simplex* be denoted by $\Delta := \{\mathbf{x} \in \mathbb{R}_+^n \mid \mathbf{e}^\top \mathbf{x} = 1\}$, where $\mathbf{e} = (1, \dots, 1)^\top \in \mathbb{R}^n$. With this notation in place, we define the *Standard Fractional Quadratic Program* (StFQP) as follows

$$\min_{\mathbf{x} \in \Delta} \lambda(\mathbf{x}) := \frac{\mathbf{x}^\top \mathbf{A} \mathbf{x}}{\mathbf{x}^\top \mathbf{B} \mathbf{x}}. \quad (1)$$

Since \mathbf{B} is PD, the function λ is well-defined over the feasible set Δ . Problem (1) can model a wide range of applications in economics, finance, communication, and engineering [9]. In this case, the aim is to optimize the

performance of a given system expressed as the ratio of two functions to accurately represent the balance between two aspects of the system, such as production cost versus production time, return versus risk of an investment.

Contrary to the case where the numerator is a linear function and the denominator is concave, a stationary point (SP) of StFQP (1) is not necessarily a global minimum because the objective function is not quasi-convex. In this case, computing a global solution is known to be NP-hard [1].

1.1 Finding stationary points

The computation of a stationary point for StFQP can be implemented by exploiting its relation with the class of the symmetric *Eigenvalue Complementarity Problem* (EiCP) [9]. In fact, by applying the Karush–Kuhn–Tucker (KKT) conditions to (1), one shows that any SP \bar{x} with associated value $\lambda = \lambda(\bar{x})$ must satisfy the following system

$$\mathbf{x} \geq \mathbf{0}, \quad \mathbf{e}^\top \mathbf{x} = 1, \quad \mathbf{A}\mathbf{x} - \lambda \mathbf{B}\mathbf{x} \geq \mathbf{0}, \quad \mathbf{x}^\top (\mathbf{A}\mathbf{x} - \lambda \mathbf{B}\mathbf{x}) = 0, \quad (2)$$

so that $(\bar{\lambda}, \bar{x})$ is a solution of EiCP(\mathbf{A}, \mathbf{B}) defined in (2). This problem was introduced by Seeger [13] in the special case where \mathbf{B} is the identity matrix, and later extended to any PD matrix \mathbf{B} by Queiroz et al. [12]. Modern state-of-the-art algorithms for symmetric EiCPs (including ADMM [8], SPL [6], DC-programming [10], spectral BAS [2], etc.) can compute these complementarity eigenpairs from (2) and generate SPs of (1). By (2), the global minimizer of StFQP is the eigenvector associated with the *smallest complementary eigenvalue* λ_{\min} of the pair (\mathbf{A}, \mathbf{B}) . Then, thanks to the combinatorial nature of the EiCP, a procedure able to compute all complementarity eigenvalues could be applied and λ_{\min} can be obtained after sorting these values. However, it is important to mention that the number of complementary eigenvalues grows exponentially with problem dimension, and enumerating all is only possible for small instances [5].

Other effective methods to calculate SPs of (1) include a sequential algorithm by Júdice et al. [9], which applies a efficient implementation of the Dinkelbach's method to the linear quadratic fractional problem obtained linearizing the numerator in (1). Moreover, Boț et al. [4] propose an extrapolated proximal subgradient algorithm, which computes SPs of more general nonconvex and nonsmooth fractional programs.

2. On lower bounds for StFQP

In this work, we are interested in evaluating lower bounds for StFQP, that is, we search for ℓ which satisfies the following

$$\ell \leq \lambda(\mathbf{x}^*), \quad (3)$$

where $\lambda(\mathbf{x}^*)$ is the global optimal value of (1).

Computing a tight lower bound is essential in many solvers using branch-and-bound techniques, for pruning the search tree and speeding convergence. However, excessively costly computations for obtaining a good ℓ can prevent these benefits, so the art lies in balancing bound-tightening effort against enumeration work. Inspired by the study of Bomze et al. [3] regarding lower bounding techniques for the *Standard Quadratic Program*, and motivated by the need for an efficient global solver for the fractional case, this work presents a systematic comparison of both emerging and classical bounding strategies, with the dual aim of deepening theoretical insights and guiding an improvement of optimization algorithms for (1).

Early on, Preisig [11] derived lower (and upper) bounds for (1) by optimizing the quadratic forms in the numerator and denominator independently, using their extreme eigenvalues to localize the quotient.

Through copositive optimization and its tight semidefinite programming (SDP) relaxations, Amaral et al. [1] show that the StFQP can be reformulated as a linear optimization problem over the *completely positive cone* \mathcal{C}_n^* . From that, they derive a hierarchy of SDP relaxations by replacing \mathcal{C}_n^* with the *doubly nonnegative cone* $\mathcal{P}_n \cap \mathcal{N}_n \supset \mathcal{C}_n^*$. These bounds strictly dominate the usual convex-envelope approaches (RLT/LP), although they are computationally expensive for large instances.

For the case where \mathbf{B} is the identity, Fernandes et al. [5] introduced algebraic procedures for finding bounds based on the localization set of the classical eigenvalues associated with the matrix \mathbf{A} . Recently, He et al. [7] extended this result, proposing two Gershgorin-type localization sets for EiCP with \mathbf{B} being the identity. In this work, we generalize these bounds to the case of an arbitrary PD matrix \mathbf{B} .

3. Concluding remarks

We carry out a comparison of the lower bound obtained by the methods discussed above from both theoretical and computational aspects. This analysis aims to determine the effectiveness of each method in providing

accurate lower bounds while considering the computational resources they require. These results will be presented at the workshop.

References

- [1] Amaral, P., Bomze, I. M., and Júdice, J. (2014). Copositivity and constrained fractional quadratic problems. *Math. Program.*, 146, 325–350.
- [2] Brás, C., Fischer, A., Júdice, J. J., Schönenfeld, K., Seifert, S. (2017). A block active set algorithm with spectral choice line search for the symmetric eigenvalue complementarity problem. *Appl. Math. Comput.*, 294, 36–48.
- [3] Bomze, I. M., Locatelli, M., and Tardella, F. (2008). New and old bounds for standard quadratic optimization: dominance, equivalence, and incomparability. *Math. Program.*, 115, 31–64.
- [4] Bot, R. I., Dao, M. N., and Li, G. (2022). Extrapolated proximal subgradient algorithms for nonconvex and nonsmooth fractional programs. *Math. Oper. Res.*, 47(3), 2415–2443.
- [5] Fernandes, L. M., Júdice, J. J., Sherali, H. D., and Forjaz, M. A. (2014). On an enumerative algorithm for solving eigenvalue complementarity problems. *Comput. Optim. Appl.*, 59, 113–134.
- [6] Fukushima, M., Júdice, J., Oliveira, W., and Sessa, V. (2020). A sequential partial linearization algorithm for the symmetric eigenvalue complementarity problem. *Comput. Optim. Appl.*, 77, 711–728.
- [7] He, J., Liu, Y., and Shen, X. (2023). Localization sets for Pareto eigenvalues with applications. *Appl. Math. Lett.*, 144, 108711.
- [8] Júdice, J., Fukushima, M., Iusem, A. N., and Sessa, V. (2021). An alternating direction method of multipliers for the eigenvalue complementarity problem. *Optim. Methods Softw.*, 36, 337–370.
- [9] Júdice, J., Sessa, V., and Fukushima, M. (2022). Solution of fractional quadratic programs on the simplex and application to the eigenvalue complementarity problem. *J. Optim. Theory Appl.*, 193, 545–573.
- [10] Niu, S., Chen, X., and Yang, F. (2019). DC-based approaches for the eigenvalue complementarity problem. *Comput. Optim. Appl.*, 73, 151–181.
- [11] Preisig, J. C. (1996). Copositivity and the minimization of quadratic functions with nonnegativity and quadratic equality constraints. *SIAM J. Control Optim.*, 34(4), 1135–1150.
- [12] Queiroz, M., Júdice, J., and Humes, C. (2004). The symmetric eigenvalue complementarity problem. *Math. Comput.*, 73(248), 1849–1863.
- [13] Seeger, A. (1999). Eigenvalue analysis of equilibrium processes defined by linear complementarity conditions. *Linear Algebra Appl.*, 292(1–3), 1–14.

Efficient Convexification of Polynomial Generalized Additive Model Surrogates via Continuous Graham-Scan

Owen Li¹, Daniel Ovalle¹, Michael L. Bynum², Barnabas Poczos¹, Carl D. Laird¹, Ignacio E. Grossmann¹, Javier Peña¹

¹*Carnegie Mellon University, Pittsburgh, US, {tianwei2, dovallev, jfp}@andrew.cmu.edu*

²*Sandia National Lab, Albuquerque, US, mlbynum@sandia.gov*

Abstract This work addresses the challenge of efficiently updating convex envelopes for univariate polynomials within global optimization frameworks, with a focus on surrogate modeling and Generalized Additive Models (GAMs). We propose a novel algorithm that combines a bisection method for computing bitangents with a continuous Graham-Scan approach to dynamically maintain tight convex relaxations as variable domains change. This approach directly constructs convex envelopes for each univariate functional form, facilitating more efficient branch pruning in spatial branch-and-bound algorithms. The results, both theoretical and experimental, show that our approach provides tighter and more efficient relaxations compared to existing methods.

Keywords: Convexification, Surrogate Model, Polynomials, Convex Envelope

1. Introduction

The global optimization of surrogate models constructed from data arises in a wide range of engineering applications, motivating the need for efficient optimization techniques. One class of surrogate models that is used in practice is the Generalized Additive Model (GAM), which represents complex functions via sums of univariate functions. Formally, a GAM is defined as:

$$M(x) = \Phi \left(\sum_{i=1}^n \phi_i(x_i) \right) \quad (1)$$

where Φ is a linking monotonic function and each ϕ_i is a univariate functional form [1]. In this work, we focus on the case where each ϕ_i is a univariate polynomial, such as spline-based representations.

Deterministic methods like Spatial Branch-and-Bound (sBB) optimize GAMs by partitioning the feasible space, constructing convex relaxations, and pruning suboptimal regions. The efficiency of sBB relies on the tightness of these relaxations, but classical techniques like McCormick relaxations often provide loose bounds for nested univariate terms, limiting their practical use [2]. Tight convex envelopes for univariate polynomials can improve sBB performance, but they typically require recomputation with each variable bounds update, introducing computational overhead. This work introduces an algorithm for dynamically updating convex envelopes of univariate polynomials, using a bisection algorithm to compute tangent lines and a modified Graham Scan to maintain the convex envelope as domain bounds change. By exploiting the structure of univariate functions, the method computes bitangent points efficiently and supports rapid domain updates. Applying this method to GAM surrogate models yields tight convex relaxations. Under suitable conditions, we can formally prove that using these dynamic envelopes preserves tightness at the global optimum, which we omit here for conciseness of presentation.

2. Methodology

2.1 Bi-tangent Computation

The first step of our approach is to compute the convex intervals of a polynomial. Formally, for any degree- n univariate polynomial p defined over a domain $\mathcal{I} := [x_L, x_U] \subset \mathbb{R}$, we define the associated set of *convex intervals* $CI_{\mathcal{I}}(p)$ as the collection of disjoint intervals $\{I_1, I_2, \dots\}$ such that each I_j is maximal with respect to inclusion and $\frac{d^2}{dx^2}p(x) \geq 0$ for all $x \in I_j$. Identifying these intervals allows us to localize the regions where the polynomial is convex, which is critical for constructing tight convex relaxations.

We first approximate all roots of $\frac{d^2}{dx^2}p(x)$ to a relative error 2^{-b} (i.e., machine precision), which can be done in $O(n^3 + n \log^2 n \log b)$ time following [3]. We take the union of the set of variable bounds and the set of roots, we sort it and, for each adjacent pair (r_i, r_{i+1}) , we evaluate the second derivative at the midpoint; if positive, (r_i, r_{i+1}) defines a convex interval.

With the convex intervals identified, next we define the *bitangents*. Given a degree- n univariate polynomial p and two convex intervals $I_1, I_2 \in CI(p)$, we define the *bitangent* $\text{bitan}_p(I_1, I_2)$ as the unique affine function $g(x)$ such that $g(x) \leq p(x)$ for all $x \in I_1 \cup I_2$, and $p(x) - g(x)$ has exactly one

root in each of I_1 and I_2 . These bitangents can be computed numerically using a bisection algorithm. Specifically, to compute the bitangent between two disjoint convex intervals I_x and I_y , we solve the system $D_p(x, y) = D_p(y, x) = 0$ under the constraint $(x, y) \in I_x \times I_y$, where $D_p(\cdot, \cdot)$ denotes the Bregman divergence. We can formally prove that this computation requires $O(b^2)$ time and achieves arithmetic precision for $\text{bitan}_p(I_x, I_y)$.

2.2 Continuous Graham Scan

To construct the convex envelope, we propose an algorithm that computes and updates the bitangents between convex intervals in an iterative manner. The algorithm traverses the sorted convex intervals, computing bitangents between consecutive intervals. Since the convex envelope must be composed of bitangents with monotonically increasing slopes, if a newly computed bitangent violates this condition, the left convex interval is removed from the candidate set, and a new bitangent is recomputed. This iterative correction mirrors the behavior of the classical Graham Scan algorithm for constructing convex hulls. We can formally prove that the initial construction of the convex envelope takes $O(n^3 + n \log^2 n \log b + nb^2)$ time, while subsequent updates as the domain changes require only $O(nb^2)$ time, since root recomputation is avoided.

Algorithm 1 Continuous Graham Scan

```

Input:  $p \in \text{poly}_n$ ,  $\mathcal{I} = [x_L, x_U] \subset \mathbb{R}$ 
Initialize empty stack  $S$ 
for  $c_1, c_2, \dots$  in  $CI_{\mathcal{I}}(p)$  do
     $l \leftarrow \text{bitan}_p(c_i, c_{i+1})$ 
    while  $S$  not empty and  $\text{top}(S)$  has slope greater than that of  $l$  do
         $l_{bad} := S.\text{pop}()$ 
         $c_j \leftarrow$  the left convex interval of  $l_{bad}$ 
         $l \leftarrow \text{bitan}_p(c_j, c_{i+1})$ 
    end while
     $S.\text{push}(l)$ 
end for
return  $S$ 

```

3. Results and Conclusions

For three different linking functions Φ , we generated 50 instances of problems by summing 2 randomly generated GAMs, each with 5 univariate components of degree $n = 6$ and variable bounds $x_i \in [-1.5, 1.5]$ for all $i \in \{1, \dots, 5\}$. The global optima of these models were computed using SCIP v.9.0.1. We then compared the average relative gap between the global optimum and the relaxations provided by our method (which applies our proposed relaxation to each GAM and aggregates them using McCormick's procedure), factorable programming using McCormick re-

laxations [2] (which we refer to as McCormick), and AlphaBB [4]. All experiments were implemented using Pyomo and Coramin [5].

Table 1: Average Gap from Different Methods

$\Phi(x)$	Average Gap [%] Standard Deviation		
	McCormick [2]	AlphaBB [4]	Proposed
x	11.66 28.73	305.69 807.73	0.00 0.00
e^x	18.55 8.18	4710.20 2307.84	1.70 2.14
sigmoid(x)	27.31 8.41	1237332 ^a 2041218	1.53 1.86

^aAlphaBB produced unbounded relaxations in 5 instances; the statistics are computed in the remaining 45 instances.

Table 1 shows that our method produces the tightest relaxation for sums of GAMs, which are structurally similar to Kolmogorov–Arnold Network models. This is expected, as each individual GAM relaxation is tight at its minimum, and any slackness arises solely from their sum. This explains the zero gap observed in the case of sums of polynomials, as the sum of polynomials is itself a polynomial, that can be relaxed tightly. For future work, we aim to embed our relaxation into a spatial branch-and-bound framework to assess computational performance and to investigate more efficient methods for solving the bitangent conditions.

Disclaimer: Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

References

- [1] Hastie, T.J. (2017). Generalized additive models. *Statistical models in S*, 249-307.
- [2] McCormick, G.P. (1976). Computability of global solutions to factorable non-convex programs: Part I—Convex underestimating problems. *Mathematical Programming*, 10(1), 147-175.
- [3] Pan, V.Y. and Chen, Z.Q. (1999). The complexity of the matrix eigenproblem. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, 507-516.
- [4] Androulakis, I.P., Maranas, C.D., and Floudas, C.A. (1995). α BB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7, 337-363.
- [5] Bynum, M.L., Castillo, A., Ceccon, F., Knueven, B., Laird, C.D., and Siirola, J.D. (2020). Coramin: An Open-Source Toolset for Developing MINLP Algorithms. Sandia National Lab. (SNL-NM), Albuquerque, NM (United States).

Continuous Cubic Formulations for the Maximum Clique Problem*

Sergiy Butenko

Texas A&M University, College Station, TX, USA butenko@tamu.edu

Abstract We study several continuous cubic formulations of the maximum clique problem that explicitly capture triple interactions among vertices/edges, in contrast to the classical quadratic Motzkin–Straus formulation. Specifically, we revisit three exact cubic models, defined over the standard simplex, the edge-based unit sphere, and a mixed vertex–edge variable space, and analyze their structural properties and optimality conditions. We propose rounding procedures that convert feasible solutions of these formulations into cliques. Computational experiments on benchmark graphs demonstrate that the cubic formulations, when paired with the proposed rounding strategies, yield high-quality cliques and can outperform quadratic approaches in practice.

Keywords: Maximum Clique Problem, Cubic Optimization

1. Introduction

We consider a simple, undirected graph $G = (V, E)$ with the vertex set V and the edge set $E \subseteq \binom{V}{2}$. A subset of vertices $C \subseteq V$ is a *clique* if the induced subgraph $G[C]$ is complete. A clique is *maximal* if it is not a part of a larger clique and, *maximum* if there is no larger clique in G . The *maximum clique problem* is to find a clique of largest cardinality in G , referred to as the *clique number* and denoted by ω .

*This work was partially supported by Air Force Office of Scientific Research under grant FA9550-23-1-0300.

Inspired by the Motzkin-Straus standard quadratic program [3], cubic formulations of the maximum clique problem arise naturally when modeling the presence of triangles in a graph. While most continuous approaches to the problem are based on quadratic formulations, typically involving the adjacency matrix, these capture only pairwise interactions between vertices. In contrast, cubic formulations incorporate triple interactions, thereby directly encoding the fundamental building block of cliques beyond size two.

Several continuous cubic formulations closely related to the clique number have been proposed in the literature. Butenko et al. [1] introduced a homogeneous cubic maximization problem over the standard simplex in $\mathbb{R}^{|V|}$, whose optimal value admits a closed-form expression in terms of ω . Sós and Straus [5] considered a cubic formulation over the unit sphere in $\mathbb{R}^{|E|}$, establishing an explicit connection between its optimal value and the clique number. Nesterov [4] proposed a related cubic problem in an augmented variable space in $\mathbb{R}^{|V|+|E|}$, also yielding an exact expression for the optimal value. In the following subsections, we restate these results and briefly outline our contributions.

2. The formulations

Let \mathcal{C}_3 denote the set of all 3-vertex cliques in G . We assume that $\mathcal{C}_3 \neq \emptyset$.

Proposition 1 (Butenko et al. [1]). *Consider the problem*

$$\begin{aligned} & \text{maximize} && \sum_{\{i,j,k\} \in \mathcal{C}_3} x_i x_j x_k \\ & \text{subject to} && \sum_{i \in V} x_i = 1 \\ & && x \geq 0. \end{aligned} \tag{1}$$

The optimal objective function value f^* of (1) satisfies

$$f^* = \binom{\omega}{3} \cdot \frac{1}{\omega^3} = \frac{(\omega-1)(\omega-2)}{6\omega^2} \tag{2}$$

and is attained at the characteristic vector

$$x^{C^*} := \begin{cases} 1/|C^*|, & \text{if } i \in C^*, \\ 0, & \text{otherwise,} \end{cases}$$

of a maximum clique C^* of G .

Proposition 2 (Sós and Straus [5]). *The clique number ω of G satisfies*

$$\binom{\omega}{3} \cdot \binom{\omega}{2}^{-3/2} = \max_{x \in X} \sum_{\{i,j,k\} \in \mathcal{C}_3} x_{ij} x_{ik} x_{jk}, \quad (3)$$

where $X = \left\{ x \in \mathbb{R}^{|E|} \mid \sum_{\{i,j\} \in E} x_{ij}^2 = 1, x \geq 0 \right\}$. The maximum above is attained by setting

$$x_{ij} = \begin{cases} \binom{\omega}{2}^{-1/2}, & i, j \in C^* \\ 0, & \text{otherwise} \end{cases}$$

for a maximum clique C^* of G .

Note that the non-negativity constraint can be dropped without affecting optimality, since changing all negative entries of the solution vector to their absolute values neither decreases the objective function value nor violates the feasibility. Hence, we will study the following cubic formulation over the unit sphere:

$$\begin{aligned} & \text{maximize} && \sum_{\{i,j,k\} \in \mathcal{C}_3} x_{ij} x_{ik} x_{jk} \\ & \text{subject to} && \sum_{\{i,j\} \in E} x_{ij}^2 = 1. \end{aligned} \quad (4)$$

The following formulation, based on Nesterov's work [4], was used by Hillar and Lim [2] to establish the NP-hardness of the symmetric tensor eigenvalue problem.

Proposition 3 (Nesterov [4]). *Consider the problem*

$$\begin{aligned} & \text{maximize} && \sum_{\{i,j\} \in E} x_i x_j y_{ij} \\ & \text{subject to} && \sum_{i \in V} x_i^2 + \sum_{\{i,j\} \in E} y_{ij}^2 = 1. \end{aligned} \quad (5)$$

The optimal objective function value f^* of (1) satisfies

$$f^* = \left(\frac{2(\omega - 1)}{27\omega} \right)^{1/2}. \quad (6)$$

This maximum is attained by setting, for a maximum clique C^* of G :

$$x_i = \begin{cases} \sqrt{\frac{2}{3\omega}}, & \text{if } i \in C^*, \\ 0, & \text{otherwise,} \end{cases} \quad y_{ij} = \begin{cases} \frac{1}{\sqrt{3(\omega)}}, & \text{if } \{i, j\} \subseteq C^*, \\ 0, & \text{otherwise.} \end{cases}$$

3. Contributions

The continuous formulations introduced above yield real-valued solutions that must be converted into feasible combinatorial solutions to extract actual cliques. We present several practical rounding strategies that allow to obtain a clique based on a feasible solution of a continuous formulation.

We perform a systematic comparison of the formulations, assessing both their direct continuous performance and the quality of cliques obtained via rounding. Results indicate that cubic formulations can produce superior local optima and offer competitive or better clique sizes compared to standard quadratic models.

Our findings highlight the potential of higher-order continuous models for combinatorial optimization, suggesting promising avenues for algorithmic development beyond quadratic approaches. Moreover, the study of cubic formulations complements the extensive literature on quadratic and polynomial optimization, potentially leading to new rounding strategies and improved computational performance in clique-finding algorithms.

References

- [1] Butenko, S. and Makovenko, M. and Pardalos, M.. (2022). A hierarchy of standard polynomial programming formulations for the maximum clique problem. *SIAM Journal on Optimization*. 32. 2102–2128.
- [2] Hillar, C. J. and Lim, L.-H.. (2013). Most Tensor Problems Are NP-Hard. *Journal of the ACM*. 60. 1–39.
- [3] Motzkin, T. S. and Straus, E. G.. (1965). Maxima for graphs and a new proof of a theorem of Turán. *Canad. J. Math.*. 17. 533–540.
- [4] Nesterov, Y. E.. (2003). Random walk in a simplex and quadratic optimization over convex polytopes. CORE Discussion Paper 2003/71.
- [5] Sós, V. T. and Straus, E. G.. (1982). Extremals of functions on graphs with applications to graphs and hypergraphs. *Journal of Combinatorial Theory, Series B*. 32. 246–257.

A blueprint for computing with distance geometry

Leo Liberti¹, Maël Kupperschmitt¹, Ha Duy Nguyen¹

¹*LIX CNRS, École Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau, France,
 {leo.liberti,mael.kupperschmitt–le-fao,ha-duy.nguyen}@polytechnique.edu*

Abstract The equivalence between **NP**-hard problems given by polynomial reductions can be exploited to solve a problem in terms of another problem. Usually “other problems” are those in which humans model naturally, such as SAT. We propose instead to use the fundamental problem of distance geometry, i.e. drawing a graph in a K -dimensional Euclidean space by points and segments as long as the edge weights. The geometrical nature of this problem affords novel solution opportunities in continuous space (local search, relaxations) when solving combinatorial problems.

Keywords: Distance Geometry Problem, Complexity, Computation

1. Introduction

The DISTANCE GEOMETRY PROBLEM (DGP) is as follows: given a positive integer K and a simple edge-weighted undirected graph $G = (V, E, d)$, decide if there exists a *realization* $x : V \rightarrow \mathbb{R}^K$ such that

$$\forall \{u, v\} \in E \quad \|x_u - x_v\|_2^2 = d_{uv}^2, \tag{1}$$

where d_{uv} is the edge weight. If K is fixed we denote the problem as DGP_K .

We exploit the equivalence properties of the complexity class **NP** (decision problems for which every YES instance can be certified in polynomial time) in order to solve hard problems for **NP** in terms of the DGP. We call this short paper a “blueprint” in the sense that many ideas herein are sketched and computationally untested.

2. NP-hardness primer

A problem P is *hard* for the class **NP** if every other problem in **NP** can be *polynomially reduced* to P . A polynomial reduction from a problem $Q \in \text{NP}$ to P is a polytime algorithm that transforms every YES instance of Q in a YES instance of P and every NO instance of Q in a NO instance of P . Then P can be no easier than Q (up to a polynomial factor) since otherwise one could solve P instead of Q and obtain the same answer, which would make Q easier than itself. From this it follows that every problem in **NP** is at most as hard as any **NP-hard** problem and that all **NP-hard** problems have the same hardness.

Polynomial reduction algorithms from Q to P transform problem instances of Q into decision-invariant instances of P , a process often called *modelling*. The prototypical **NP-hard** problem is **SAT**¹. In [2], the trace of the polytime certification algorithm for YES instances of **NP** was modelled by an infinite family of **SAT** instances that are satisfiable iff the certification algorithm succeeds. By [2], modelling **SAT** in terms of new problems P is sufficient to prove the **NP-hardness** of P .

2.1 Categorizations of NP-hard problems

NP-hard problems that also belong to **NP** are called **NP-complete**. This provides a first categorization.

A second one is given by *weakly* and *strongly* **NP-hard** problems. If an **NP-hard** problem Q has a partly numerical input (e.g. integers), and the polytime reduction to P represents some of these integers in unary form (e.g. $n \in \mathbb{N}$ in Q is transformed to a set of n vertices in P), then P is weakly **NP-hard**. If all integers are represented in binary form (e.g. n is represented as a weight on an edge), then P is strongly **NP-hard**. Although we apply this categorization to the problems, it really applies to reductions.

The third categorization is the one between *general* and *specific* problems: humans naturally use the former modelling, such as **SAT**, **MATHEMATICAL PROGRAMMING** (**MP**), **CONSTRAINT PROGRAMMING** (**CP**). General problems usually involve sentences with variables. Problems that are perceived to lack these properties are called specific. Variables, however, are not crucial to mathematics [7]. The description length is more important: modelling a specific problem by a general (e.g. k -**CLIQUE** in **MP**) leads to reasonably short descriptions of the resulting **MP** instance, whereas encoding a **BI-**

¹The **SAT** problem asks the question: is a given conjunction of clauses each of which is a disjunction of literals satisfiable?

NARY LINEAR PROGRAMMING (BLP)² instance by DGP_1 leads to a longer description w.r.t. the original instance size.

2.2 The DGP is NP-hard

In [5] we find several reductions to the DGP, both weak and hard. In particular, the DGP_1 is weakly NP-hard by reduction from PARTITION³ and strongly NP-hard by reduction from 3SAT⁴.

The DGP_1 is known to be NP-complete while this is not known for the DGP_K for $K > 1$ [1]. In $K > 1$ dimensions an instance with integer edge weights might lead to a realization that necessarily involves real algebraic numbers, such as e.g. a triangle graph with unit weights. We do not know a polytime algorithm for checking whether Eq. (1) holds if x_u, x_v have real algebraic components. The DGP is considered a specific problem.

3. Motivations

The reason why we believe that there could be advantages in modelling by DGP is the geometric nature of the realization x : (i) although we do not know how to use x to certify a YES instance precisely, it can be used approximately (e.g. using floating-point operations) to verify Eq. (1) to any desired accuracy; (ii) the fact that x ranges over \mathbb{R}^K reduces any combinatorial problem $Q \in \text{NP}$ to one in continuous space, paving the way for the use of solution algorithms involving gradients and Hessians; (iii) obtaining a relaxation of a DGP_K instance may be as simple as solving the instance in a higher-dimensional space.

4. Contributions

4.1 Saxe's reduction from 3SAT to DGP_1

Our first contribution is a proof that the strongly polynomial reduction from 3SAT to DGP_1 (with edge weights restricted to $\{1, \dots, 4\}$) in [5] is valid. The original construction is based on three weighted graph gadgets, the most complicated of which represents the j -th clause $L_{j1} \vee L_{j2} \vee L_{j3}$ of the given 3SAT. Saxe writes:

²A subclass of MP with linear objective and constraints and binary variables only.

³Given a list (a_1, \dots, a_n) is there $I \subseteq [n] = \{1, \dots, n\}$ such that $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$?

⁴Like SAT, but every clause has exactly 3 literals.

careful study of the graph [...] will reveal that it is impossible to embed it in the lin in such a way that [...] all three of the L_{jk} are sent to -1 (FALSE), but if one or more of the L_{jk} are to be sent to 1 (TRUE) then [...] exactly one such embedding is possible.

In fact, the clause gadget is wrong: L_{j2} is placed at the wrong vertex, and one of the edges is missing its weight. The technical report [6] corrects the gadget but still lacks a proof, which is nontrivial. We were able to write a satisfactory proof based on showing that realized cycles in the graphs fail to close correctly⁵ unless at least one literal L_{jk} is placed at $x_{L_{jk}} = 1$.

4.2 Strong reduction from BLP to DGP_1

Our second contribution is a reduction from BLP to DGP_1 based on § 4.1. First, we reduce BLP to a SAT: this involves turning all negative integers to positive and replace corresponding variables to negative literals, transform conditional additions to adder circuits, and introduce an integer comparison circuit. Then the circuit sequence is turned into a SAT instance [3], which is then transformed to a 3SAT by successive “linearization” [4]. Finally, the 3SAT is reduced to DGP_1 by § 4.1.

4.3 Weak relaxed reduction from BLP to DGP_1

Our third contribution is a “relaxed reduction” from BLP to DGP_1 . Each of the m inequalities $A_i y \leq b_i$ of the BLP is transformed to an equation via a binary encoding of a slack variable s_i as $\sum_{h < \log_2 U_i} 2^h \beta_h$ where $U_i \geq \sum_j |A_{ij}|$

and each $\beta \in \{0, 1\}$. The equations are penalized by squaring them and then summed together. Each squared variable ν^2 is replaced by ν since all variables are binary. Cross products $\nu \mu$ are linearized to a single variable ξ . In this form, the instance is equivalent to a SUBSET-SUM⁶, which can be easily reduced to PARTITION. We then apply the weak reduction to DGP_1 in [5]⁷. This BLP→ DGP_1 reduction is relaxed because, upon solving the DGP_1 , the vertex positions corresponding to the ξ variables may take values different from the corresponding product $\nu \mu$. We address this issue via a solution algorithm for the DGP_1 , called Branch-and-Prune in 1D (BP_1), which can correct such errors at runtime⁸.

⁵I.e. the first vertex position x_α on the cycle path is different from the last vertex position x_ω .

⁶Given a list (a_1, \dots, a_n) of integers and a value b , is there an $I \subseteq [n]$ such that $\sum_{i \in I} a_i = b$?

⁷In the original paper as well as in [6] the proof is only sketched: we wrote a full proof.

⁸ BP_1 places the next vertex left or right of the previous one (vertex orders allowing this always exist when $K = 1$ and the graph is connected). When two out of the three variables ν, μ, ξ are fixed to certain values in a branch, the third value is fixed: BP_1 prunes the other branch.

4.4 Geometric relaxations

We apply the strong reduction to an infeasible BLP instance to obtain an infeasible DGP₁. We consider the error formulation⁹ $\min\{\sum_{\{u,v\} \in E}(s_{uv}^+ + s_{uv}^-) \mid \|x_u - x_v\|_2^2 = d_{uv}^2\}$. We show that globally solving this instance with $K = 2$ in increasingly thin slabs around the first coordinate leads to increasingly tightened relaxations tending towards the exact optimal value of the ℓ_1 error. We define some convex relaxations of MINERR-DGP.

5. Conclusion

Much work remains to be done: (1) streamlining our reductions; (2) improving our BP₁ implementation; (3) tight and efficient geometric relaxations; (4) mapping relaxation bound values from the DGP back to the BLP. Lastly, we are also looking at possible reductions based on computability rather than complexity.

References

- [1] N. Beeker, S. Gaubert, C. Glusa, and L. Liberti. Is the distance geometry problem in NP? In A. Mucherino, C. Lavor, L. Liberti, and N. Maculan, editors, *Distance Geometry: Theory, Methods, and Applications*, pages 85–94. Springer, New York, 2013.
- [2] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Symposium on the Theory of Computing*, STOC, pages 151–158, New York, 1971. ACM.
- [3] N. Eén and N. Sörensson. Translating pseudo-boolean constraints into SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 2:1–25, 2006.
- [4] R. Karp. Reducibility among combinatorial problems. In R. Miller and W. Thatcher, editors, *Complexity of Computer Computations*, volume 5 of *IBM Research Symposia*, pages 85–104, New York, 1972. Plenum.
- [5] J. Saxe. Embeddability of weighted graphs in k -space is strongly NP-hard. *Proceedings of 17th Allerton Conference in Communications, Control and Computing*, pages 480–489, 1979.
- [6] J. Saxe. Two papers on graph embedding problems. Technical Report CMU-CS-80-102, Dept. of Computing, Carnegie-Mellon University, 1980.
- [7] A. Tarski and S. Givant. *A formalization of set theory without variables*. Number 41 in American Mathematical Society Colloquium Publications. AMS, Providence, RI, 1987.

⁹Called MINERR-DGP.

Session 4





Solving Optimal Experiment Design with Mixed-Integer Convex Methods

Deborah Hendrych,^{1,3} Mathieu Besançon,² Sebastian Pokutta^{1,3}

¹*Zuse Institute Berlin*

²*Université Grenoble Alpes, Inria, LIG, CNRS*

³*Technische Universität Berlin*

Abstract We tackle the Optimal Experiment Design Problem, which consists of choosing experiments to run or observations to select from a finite set to estimate the parameters of a system. The objective is to maximize some measure of information gained about the system from the observations, leading to a convex integer optimization problem. We leverage Boscia.jl, a recent algorithmic framework [5], which is based on a nonlinear branch-and-bound algorithm with node relaxations solved to approximate optimality using Frank-Wolfe algorithms. One particular advantage of the method is its efficient utilization of the polytope formed by the original constraints which is preserved by the method, unlike alternative methods relying on epigraph-based formulations. We assess our method against both generic and specialized convex mixed-integer approaches. Computational results highlight the performance of our proposed method, especially on large and challenging instances. This is a summary of [4].

Keywords: optimal design, mixed-integer optimization, convex optimization, first-order methods

1. Optimal Design of Experiment

Suppose we want to fit a regression model:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^m} \|A\boldsymbol{\theta} - \mathbf{y}\|, \quad (1)$$

where $A \in \mathbb{R}^{m \times n}$ is known and encodes m experiments, $\mathbf{y} \in \mathbb{R}^m$ is *unknown* and encodes the responses of the experiments and $\boldsymbol{\theta} \in \mathbb{R}^n$ are the parameters to be estimated. We assume that A has full column rank and $n << m$.

Running all experiments, potentially even multiple times to account for errors, is often not realistic because of time and cost constraints. Thus, the *Optimal Experiment Design Problem (OEDP)* aims to find the subset of size N of the experiments providing the “most information” about the experiment space [6]. In general, the *number of allowed experiments* $N \in [n, m]$ is required in order to allow a solution to the regression model in Equation (1).

To quantify the information, we introduce the so-called *information matrix* X defined as

$$X(\mathbf{x}) = A^\top \text{diag}(\mathbf{x}) A \quad (2)$$

where $\mathbf{x} \in \mathbb{N}_0^m$ is a design.

Next, we require an information function ϕ on \mathbb{S}_{++}^n that is positively homogeneous, concave, nonnegative, non-constant, upper semi-continuous and respects the Loewner ordering. Different choices of ϕ lead to different so-called criteria. We focus on two popular criteria: the A-optimality criterion and the D-optimality criterion leading us to the following Mixed-Integer Non-Linear Programming (MINLP) formulations:

$\min_{\mathbf{x}} \text{Tr}((X(\mathbf{x}))^{-1})$	$\min_{\mathbf{x}} -\log(\det(X(\mathbf{x})))$
s.t. $\sum_{i=1}^m x_i = N$	s.t. $\sum_{i=1}^m x_i = N$
(A-Opt)	(D-Opt)
$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$	$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$
$\mathbf{x} \in \mathbb{Z}^m$	$\mathbf{x} \in \mathbb{Z}^m$

Note that the feasible region of both problems is a fairly simple polytope, namely a scaled and truncated probability simplex, so the difficulty stems from the objectives.

2. Mixed-Integer Convex Optimization

In [5], a new approach for solving convex MINLP was proposed based on a nonlinear branch-and-bound algorithm with node relaxations solved to approximate optimality using Frank-Wolfe algorithms [3, 2]. This framework was implemented in the Julia package `Boscia.jl` and is the main solver used in this work. We leverage this framework to solve the OEDP and test its performance against both generic and specialized convex mixed-integer approaches.

Most specialized approaches for OEDP are applicable to only one criterion. The most general approach known to us was introduced in [1]. It is also a branch-and-bound approach with a coordinate-exchange type of algorithm as a node solver. We refer to it as `Co-BnB`.

For generic approaches, we consider Outer Approximation (OA) approaches. We represent the nonlinear objectives using cones, once in a direct formulation utilizing the Julia package `Hypatia.jl` and a second-order conic formulation introduced by [7].

A selection of the results is shown in Table 1 and Figure 1. For the computational experiments, we generated two random data sets for each problem, one with independent data and the other with correlated data. In terms of size, we consider problems with the number of experiments $m \in [50, 120]$ and the number of parameters $n \in \{m/10, m/4\}$. In total, there are 50 instances for each problem and data set. The new method `Boscia.jl` shows a superior performance over the other solvers. Additionally, `Boscia.jl` is more flexible than its strongest competitor, `Co-BnB`.

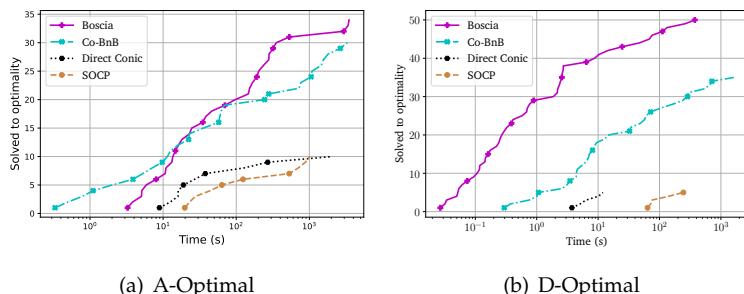


Figure 1: Number of solved instances over time for the A-Optimal and D-Optimal Experiment Design Problems with correlated data.

Table 1: Performance comparison between the solvers. We generate two data sets for each problem, one with independent data and the other with correlated data. Shown are the percentage of solved instances and the geometric mean of the solve time (shifted by 1 second) over all instances. Note that there are 50 instances for each problem and data set.

Type	Corr.	Solver	# solved	% solved	Time (s)	Rel. gap
AO	no	Boscia	26	52%	374.74	0.0851
		Co-BnB	24	48%	401.67	51.4694
		Direct Conic	4	8%	2382.52	2.4436
		SOCP	10	20%	1578.03	Inf
AO	yes	Boscia	34	68%	227.69	0.0336
		Co-BnB	30	60%	301.8	2058.4159
		Direct Conic	10	20%	1515.23	30.3669
		SOCP	10	20%	1861.35	Inf
DO	no	Boscia	34	68%	136.92	0.0208
		Co-BnB	29	58%	216.31	2.7678
		Direct Conic	10	20%	1241.37	0.676
		SOCP	10	20%	2169.4	Inf
DO	yes	Boscia	50	100%	2.56	0.0094
		Co-BnB	35	70%	101.89	0.0809
		Direct Conic	5	10%	1398.01	0.0206
		SOCP	5	10%	2526.74	Inf

References

- [1] Selin Damla Ahipaşaoğlu. A branch-and-bound algorithm for the exact optimal experimental design problem. *Statistics and Computing*, 31(5):65, 2021.
- [2] Mathieu Besançon, Alejandro Carderera, and Sebastian Pokutta. FrankWolfe.jl: A high-performance and flexible toolbox for Frank–Wolfe algorithms and conditional gradients. *INFORMS Journal on Computing*, 2022.
- [3] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.
- [4] Deborah Hendrych, Mathieu Besançon, and Sebastian Pokutta. Solving the optimal experiment design problem with mixed-integer convex methods. *arXiv preprint arXiv:2312.11200*, 2023.
- [5] Deborah Hendrych, Hannah Troppens, Mathieu Besançon, and Sebastian Pokutta. Convex mixed-integer optimization with Frank–Wolfe methods, 2023.
- [6] Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.
- [7] Guillaume Sagnol and Radoslav Harman. Computing exact D-optimal designs by mixed integer second-order cone programming. 2015.

On Solving Partial Inverse Combinatorial Problems with NP-Complete Lower Level

Eva Ley¹, Maximilian Merkert¹

¹TU Braunschweig, Germany {eva.ley, m.merkert}@tu-braunschweig.de

Abstract Partial inverse combinatorial optimization problems (PICPs) are bilevel problems in which the leader modifies the costs of a combinatorial problem such that an optimal follower's solution contains a set of required elements. PICPs can be solved via a decomposition scheme with bilinear cuts for feasible follower solutions. We strengthen the problem formulation with linear cuts based on local exchanges. In this talk, we focus on the Travelling Salesman Problem (TSP) as follower problem. First computational results show the strength of our cutting planes.

Keywords: Bilevel Optimization, Inverse Problems, Mixed-Integer Optimization

1. Introduction

Partial inverse combinatorial problems (PICPs) can be formulated as (optimistic) bilevel problems with coupling constraints in the following way

$$\begin{aligned} \min_{w \in \mathbb{R}^n, y} \quad & \|w\| \\ \text{s.t.} \quad & y_e = 1 \quad \forall e \in R \\ & y \in \arg \min_{y' \in Y} (d + w)^\top y'. \end{aligned}$$

The follower has a combinatorial problem with feasible set $Y \subseteq \{0, 1\}^n$, for which the cost coefficients in the objective function can be modified by the leader. The leader aims for minimal-weight modifications such that there

is an optimal follower solution that includes certain required elements R . Due to the bilinear follower objective, general-purpose methods for bilevel mixed-integer linear problems are unsuitable, see e.g. [5], or survey [7].

Complete inverse combinatorial problems (CICPs) are a special case in which R specifies a unique follower solution. For polynomial-time solvable follower problems, CICPs are solvable in polynomial time [1], while PICPs can be NP-hard, see [8] and references therein. For several NP-hard problems including Hamiltonian cycle and vertex cover, determining feasibility for corresponding PICPs is Σ_2^P -complete [6]. Interestingly, primal bound verification for the complete inverse mixed-integer linear problem is coNP-complete [3]. If the follower uses a certain fixed polynomial-time heuristic for TSP instead of solving the problem to optimality, the corresponding complete inverse problem can be solved in polynomial time [4].

2. Solution Method

PICP can be equivalently reformulated using one bilinear constraint for every feasible follower solution:

$$\begin{aligned} \min_{w \in \mathbb{R}^n, y \in Y} \quad & \|w\| \\ \text{s.t.} \quad & y_e = 1 \quad \forall e \in R \\ & (d + w)^\top y \leq (d + w)^\top z \quad \forall z \in Y. \end{aligned}$$

To solve a PICP, it suffices to only add those constraints that would otherwise be violated. The separation problem is exactly the follower problem. For solving the complete inverse mixed-integer linear problem, only a subset of these cuts suffices [2].

We strengthen the above formulation by adding *linear* cuts based on local exchanges. A first type of cuts arises from complete inverse subproblems: If there is a path of required edges in the partial inverse TSP (PITSP), this path must have minimal weight among all Hamiltonian paths on these vertices. Thus, the separation problem of these *Sub-TSP cuts* for PITSP is a (smaller) TSP. Furthermore, we use more general local exchanges where less structure in the set of required elements R is needed. Relocating a node in a tour, i.e. removing it from between two nodes and reinserting it elsewhere, replaces three edges with three new ones, together forming two triangles, see Figure 1. Whenever the three removed edges lie in R and the exchange shortens the tour, it violates a *double-triangle (DT)* cut. Relocating a required subpath instead of a single node generalises these cuts. Exact separation for



Figure 1: Cuts for PITSP: A required path $\{AB, BC, CD\} \subseteq R$ must not have larger weight than the alternative path $\{AC, BC, BD\}$ (left). For required edges $\{AB, CD, DE\} \subseteq R$, we derive a so-called *double-triangle (DT) cut* comparing their weights to edges AD, BD, CE (right).

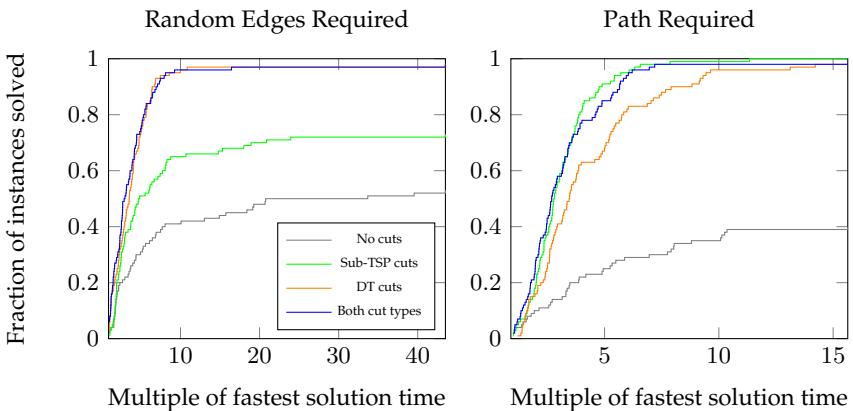


Figure 2: Performance of our methods for PITSP.

DT cuts is done via an integer program which is justifiable since the original problem can be expected to be Σ_2^p -hard.

3. Computational Experiments for PITSP

We compare the standard decomposition approach with versions where we add our linear cuts after separation. Our basic implementation is in Python with Gurobi to solve the master problem, the follower problem and the cut separation problems. We use the Miller-Tucker-Zemlin formulation for TSPs. All 100 PITSP instances are randomly generated on a complete graph with 14 vertices with weights of the edges uniformly chosen from $\{1, \dots, 100\}$. The set R consists of seven random edges and contains no cycle. In the second test, these edges are required to form a path. We minimize the ℓ_1 -norm of the weight modifications with a time limit of 600 seconds.

Required	Method	Solved	Mean	Std	Min	25%	50%	75%	Max solved
Random	No cuts	52	378.2	238.0	25.7	112.8	545.0	600.0	581.7
	Sub-TSP cuts	72	279.4	227.5	18.3	85.6	188.4	600.0	569.8
	DT cuts	97	144.1	116.1	18.2	75.4	108.9	168.8	428.1
	Both cut types	97	132.8	105.7	14.7	71.2	109.6	157.4	365.1
Path	No cuts	39	416.4	237.9	9.2	136.2	600.0	600.0	369.2
	Sub-TSP cuts	100	106.2	71.0	17.1	63.6	92.2	128.4	474.1
	DT cuts	98	144.3	114.3	20.8	76.7	115.2	174.4	564.3
	Both cut types	98	109.7	88.3	7.3	66.7	93.7	124.0	308.1

Table 1: Statistics on running times for PITSP in seconds. For the calculation of mean, standard deviation, and quartiles, the time limit of 600 seconds has been used for instances that have not been solved.

The strength of the different cut types for PITSP depends on the structure of the set of required elements R , see Figure 2 and Table 1. Note that weights of edges can only be included in a Sub-TSP cut if these edges belong to a required path of at least three edges. In contrast, DT cuts can include weights of isolated required edges. In both cases, adding our linear cuts improves the running times significantly.

References

- [1] Ahuja, R.K., Orlin, J.B (2001). Inverse optimization. *Oper. Res.* 49(5), 771–783.
- [2] Bodur, M., Chan, T.C.Y., Zhu, I.Y. (2022). Inverse mixed integer optimization: Polyhedral insights and trust region methods. *INFORMS Journal on Computing* 34(3), 1471–1488.
- [3] Bulut, A., Ralphs, T.K. (2021). On the complexity of inverse mixed integer linear optimization. *SIAM Journal on Optimization* 31(4), 3014–3043.
- [4] Chung, C., Demange M. (2012). On inverse traveling salesman problems. *4OR* 10(2), 193–209.
- [5] Fischetti, M., Ljubic, I., Monaci, M., Sinnl, M. (2017). A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research* 65(6), 1615–1637.
- [6] Johannes, B. (2011). New Classes of Complete Problems for the Second Level of the Polynomial Hierarchy. PhD thesis, Technischen Universität Berlin.
- [7] Kleinert, T., Labb  , M., Ljubic, I., Schmidt, M. (2021). A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization* 9, 100007.
- [8] Ley, E., Merkert, M. (2024). Solution methods for partial inverse combinatorial optimization problems in which weights can only be increased. *Optimization Online*.

A Frank-Wolfe-based primal heuristic for quadratic mixed-integer optimization

Gioni Mexi,¹ Deborah Hendrych,¹ Sébastien Designolle^{1,4}, Mathieu Besançon², Sebastian Pokutta^{1,3}

¹*Zuse Institute Berlin*

²*Université Grenoble Alpes, Inria, LIG, CNRS*

³*Technische Universität Berlin*

⁴*ENS Lyon, Inria, LIP*

Abstract We propose a primal heuristic for generic quadratic mixed-integer optimization. Our method extends the Boscia framework – originally a mixed-integer convex solver leveraging a Frank-Wolfe-based branch-and-bound approach – to address non-convex quadratic objective and constraints. We reformulate nonlinear constraints, introduce pre-processing steps, and a suite of heuristics including rounding strategies, gradient-guided selection, and large neighborhood search techniques that exploit integer-feasible vertices generated during the Frank-Wolfe iterations. Computational results demonstrate the effectiveness of our method in solving challenging MIQCQPs.

Keywords: mixed-integer optimization, quadratic optimization first-order methods

1. Mixed-Integer Optimization from First-Order Methods

Boscia [3] is a mixed-integer convex solver that operates a branch-and-bound scheme, utilizing Frank-Wolfe (FW) methods [2, 1] to solve continuous subproblems and leverages a mixed-integer (linear) programming

(MIP) solver to implement the linear minimization oracle (LMO) required by FW. We present a sketch of the overall solution process in Figure 1.

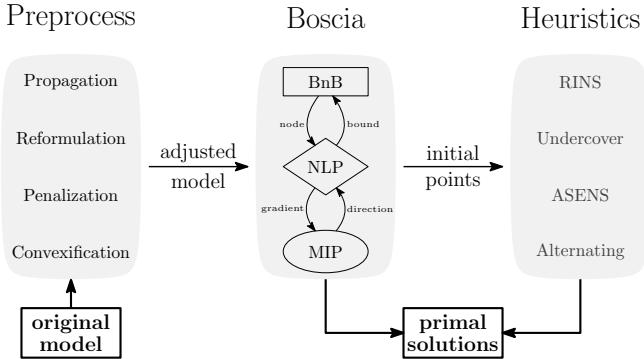


Figure 1: Overview of our approach. The model first undergoes a preprocessing whose main goal is to transfer nonlinearities into the objective function. Our workhorse is indeed the Frank-Wolfe-based solver *Boscia* which handles nonlinear problems (NLP) by suitably combining calls to a mixed-integer programming (MIP) solver in a branch-and-bound (BnB) framework. The points collected at each node in this process are then fed to various heuristics to reach better solutions.

In the standard Frank-Wolfe procedure [2], the Linear Minimization Oracle (LMO) typically solves a Linear Programming (LP) subproblem. Boscia solves a Mixed-Integer Programming (MIP) problem when FW requires an LMO call, using the gradient of the current iterate as the objective function. By doing so, Boscia optimizes convex relaxations over the convex hull of mixed-integer feasible points instead of the continuous relaxation of the constraint, resulting in a much smaller branch-and-bound search and directly leveraging the MIP solver machinery, e.g., cutting planes, conflicts analysis, heuristics. Importantly, optimizing over the convex hull of integer-feasible solutions is performed by Boscia without an explicit algebraic description of this feasible region.

Generic mixed-integer quadratically-constrained quadratic problems (MIQCQPs) did not fit the framework of convex mixed-integer optimization that Boscia was designed for, requiring several key changes to the algorithm. We employed a power penalty relaxation introduced in [4], ensuring that the solver could process these constraints while maintaining feasibility.

Another key challenge was Boscia's assumption of convexity, which does not always hold for the given MIQCQP instances. To address this, we implemented several modifications: 1) disabling node pruning in the branch-

and-bound process, 2) ignoring the lower bound from the Frank-Wolfe gap, and 3) adjusting Boscia’s solution storage mechanism to retain all solutions. Even though the focus of the approach is on the primal side, the framework is also very suited to derive high-quality relaxation bounds, since the constraint penalization and many transformations are exact reformulations or produce relaxations of the original problem.

2. Heuristics

We highlight three families of heuristics implemented within Boscia’s search process for MIQCQPs.

Rounding. One of the simplest techniques is rounding, where fractional values are rounded to the nearest integer. Additionally, Boscia employs *probabilistic rounding* for binary variables, where variables are randomly fixed to 0 or 1 based on probability distributions, and Frank-Wolfe is then used to solve for the remaining continuous variables. Another specialized rounding method is designed for *0/1 polytopes*, ensuring that the rounded solution remains within the feasible region.

Gradient-based heuristics. We utilize *gradient-based heuristics*, which explore a set of promising vertices. These heuristics are particularly suitable for Boscia since the algorithmic components they require (gradient and linear minimization oracles) are precisely those that are available. The method assumes an LMO-compatible feasible region, quadratic constraints have thus been reformulated as penalties. From an initial extreme point, the heuristic will compute the gradient, and call the LMO to compute the next extreme point. The algorithm iterates for a certain budget of maximum iterations or until it cycles to a vertex already encountered.

Large Neighborhood Search. We combined our approach with different large neighborhood search heuristics, namely **ASENS** which we introduced and is based on the *active set*, i.e., the set of vertices that is maintained by FW to form the current iterate, **undercover** which fixes variables to remove nonlinearities, and **RINS**.

Specialized heuristics for QUBOs. For QUBOs whose objective matrix has a *bipartite* underlying graph, an initial solution can be improved on by alternatively optimizing over each component of the graph.

Table 1 summarizes our results on the test set of 95 instances provided by the MIP 2025 competition¹, which we break down in four categories: QUBO, MIQP, MIQCP, and MIQCQP. All experiments use a five-minute time limit. Finally, we ran our algorithm on instances from the QPLIB benchmark set. In total, our submitted version of the heuristic found feasible solutions for 248 out of the 319 QPLIB instances with discrete variables. The average time to the first feasible solution was 7.84 seconds, with an average optimality gap of 11.38% and a primal integral of 42.79. We also found eight new best-known solutions for QPLIB instances.

Table 1: Performance summary by problem category

Category	Found	Time 1 st	Gap	Primal integral
QUBO	14/14	0.95	0.16	1.78
MIQP	39/39	1.13	1.73	9.39
MICQP	32/38	33.32	10.94	101.57
MICQCP	3/4	1.83	9.79	104.15
Total	88/95	12.83	5.10	29.18

References

- [1] Mathieu Besançon, Sébastien Designolle, Jannis Halbey, Deborah Hendrych, Dominik Kuzinowicz, Sebastian Pokutta, Hannah Troppens, Daniel Viladrich Herrmannsdoerfer, and Elias Wirth. Improved algorithms and novel applications of the FrankWolfe.jl library, 2025.
- [2] Gábor Braun, Alejandro Carderera, Cyrille W Combettes, Hamed Hassani, Amin Karbasi, Aryan Mokhtari, and Sebastian Pokutta. Conditional gradient methods. *arXiv preprint arXiv:2211.14103*, 2022.
- [3] Deborah Hendrych, Hannah Troppens, Mathieu Besançon, and Sebastian Pokutta. Convex integer optimization with Frank-Wolfe methods. 2022.
- [4] Kartikey Sharma, Deborah Hendrych, Mathieu Besançon, and Sebastian Pokutta. Network design for the traffic assignment problem with mixed-integer Frank-Wolfe. In *INFORMS Optimization Society Annual Conference Proceedings*. INFORMS, 2024.

¹ Available at: <https://www.mixedinteger.org/2025/competition/>

Session 5





Embedding neural networks into optimization models with GAMSPy

André Schnabel,¹ Hamdi Burak Usul,²

¹GAMS Software GmbH, Germany, aschnabel@gams.com

²GAMS Software GmbH, Germany, busul@gams.com

Abstract GAMSPy provides Python’s flexibility with GAMS’s modeling power. It bridges the gap between machine learning (ML) and traditional mathematical modeling by offering auxiliary classes for common neural network (NN) layers and activation functions, automatically converting network architectures into GAMSPy model expressions. In this article, we demonstrate how GAMSPy enables embedding a pre-trained NN into an optimization model.

Keywords: Modeling Languages, Machine Learning, Neural Networks

1. Introduction

The General Algebraic Modeling System (GAMS) started as the first domain specific language for mathematical modeling in the 1970s at the World Bank. Over the years, the algebraic modeling language evolved and the GAMS distribution grew with many supporting software products. The Python package GAMSPy¹ is a more recent addition, and allows specifying models directly in Python in a way that closely resembles the design philosophy of the GAMS language.

While using ML models to generate or pre-process input parameters might be sufficient for some problems, many problems require a tighter in-

¹<https://gamspy.readthedocs.io>

tegration of machine learning and optimization models by embedding the ML model. To support this, GAMSPy was extended with the `formulations` package² package offering various automatic reformulations.

The solver Gurobi offers a similar library `gurobi-machinelearning`³, which has more limited support for embedding NN. The package only supports linear layers and ReLU as single activation function. Furthermore it restricts the user to Gurobi as solver.

Another alternative package to `gamsPy.formulations` is OMLT⁴. OMLT targets the open-source Pyomo optimization package. While generally also a capable solution, OMLT formulations tend to be more verbose and model generation with the GAMS execution system utilized by GAMSPy often outperforms Pyomo.⁵

2. Surrogate model example

The following model is an adaption of the section 4.1 of Bhosekar and Ierapetritou [1]. It is a simple artificial problem that exhibits basic properties of real-world problems found in chemical engineering. A stream of input material $A \left[\frac{\text{mol}}{\text{h}} \right]$ is filled into a reactor r and then converted into two output products B and E by a separator module s . The demands of $B \left[\frac{\text{mol}}{\text{h}} \right]$ and $E \left[\frac{\text{mol}}{\text{h}} \right]$ are known and the task is to choose reactor and separator modules that minimize the overall costs. The reactors differ by volume $V_r \left[\text{m}^3 \right]$ and price $c_r \left[\text{k\$} \right]$. The separators are assumed to be ideal, and are characterized by bounds lb_s and ub_s on the amount of throughput they can process per hour and price c_s . Table 1 shows the specifications of the available reactor and separator modules.

Table 1: Design options for reactor and separator.

Options	Reactor (m^3)	c_r (k\$)	Separator (F_A mol/h)	c_s (k\$)
Option 1	5	400	30–50	300
Option 2	20	850	40–70	720
Option 3	35	1200	60–100	980
Option 4	50	1650	90–140	1210

The solution methodology in [1] involves obtaining the feasibility constraints via a simulation implemented as nonlinear program. When one

²<https://gamsPy.readthedocs.io/en/latest/reference/gamsPy.formulations.html>

³<https://gurobi-machinelearning.readthedocs.io/>

⁴<https://pypi.org/project/omlt/>

⁵As shown in "Performance in Optimization Models" at <https://tinyurl.com/gamsperf>

instead uses as surrogate model (a NN trained on the simulation) to determine the probability of feasibility for a given reactor volume V , input flow rate F_a , and demanded output flow rates F_B and F_E , the remaining model can be expressed as the following mixed-integer linear program (MILP):

$$\text{minimize} \quad \sum_r c_r \cdot y_r + \sum_s c_s \cdot y_s + c_a \cdot F_a \quad (1)$$

$$\text{s.t.} \quad \sum_r y_r = 1, \quad \sum_s y_s = 1 \quad (2)$$

$$V = \sum_r V_r \cdot y_r \quad (3)$$

$$NN(V, F_a, F_B, F_E) \geq 0.99 \quad (4)$$

$$F_a \leq ub_s + (1 - y_s) \cdot M \quad \forall s \quad (5)$$

$$F_a + (1 - y_s) \cdot M \geq lb_s \quad \forall s \quad (6)$$

$$y_r \in \{0, 1\}, y_s \in \{0, 1\}, F_a \in \mathbb{R}_+, V \in \mathbb{R}_+ \quad (7)$$

The objective function (1) computes the total cost incurred by the choice of reactor r and selector s plus the cost for the input material flow rate F_a . Constraints (2) ensure exactly one reactor and exactly one separator is chosen. Equation (3) links the auxiliary variable V for the available volume with the volume of the chosen reactor. Constraint (4) uses the embedded NN to predict a feasibility probability for a given tuple of reactor volume V , input flow rate F_a and output material flow rate demands F_B and F_E . The feasibility prediction acquired through NN inference must be at least 99%. Equations (5) and (6) make sure the input material flow rate is inside the operational bounds of the chosen selector using a bigM-formulation. The decision variable domains follow in (7), declaring the input material flow rate and available volume auxiliary variable as continuous, and the reactor- y_r and selector-choice y_s indicator variables as binary.

To solve the model with a MILP solver, the parameterized NN term NN in (5) must be expanded into a set of additional constraints and variables, such that the left-hand side of (5) evaluates to the value acquired by doing a forward propagation of input vector (V, F_a, F_B, F_E) through the network layers. The NN is sequential and consists of 4 linear layers connected with ReLU and sigmoid as activation functions for the inner layers and output layer respectively. Listing 1.1 shows relevant Python code for including the trained NN inside the model. The full source code is available on GitHub⁶.

⁶<https://github.com/GAMS-dev/surrogate-model>

```

model = nn.Sequential(nn.Linear(4, 10), nn.ReLU(), nn.Linear(10, 10),
    nn.ReLU(), nn.Linear(10, 15), nn.ReLU(), nn.Linear(15, 1),
    nn.Sigmoid())
drop_sigmoid_model = nn.Sequential(*list(model.children())[:-1])
# train network
...
m = gp.Container()
# input to neural network: a0=[V,Fa,Fb,Fe]
a0 = gp.Variable(m, name="a0", domain=gp.math.dim([4]))
a1 = gp.Variable(m, name="a1", domain=gp.math.dim([4])) # a0 normalized
normalize_input[...] = a1 == (a0 - x_mean_par) / x_std_par
...
seq_formulation = gp.formulations.TorchSequential(m, drop_sigmoid_model)
z5, _ = seq_formulation(a1)
check_feasibility[...] = z5[0] >= 4.59511985013459 # 0.99 probability

```

Listing 1.1: Code excerpt for linking the trained NN into the MILP

3. Numerical results

The solver Baron solves the instance from Table 1 to optimality within 0.06 seconds given a white-box monolithic MINLP whereas CPLEX solves the surrogate model MIP in only 0.03 seconds. Solution quality is identical, as the loss of precision from the NN approximation in (4) does not cut the optimal solution from the feasible region. Besides computational gains, embedding NN has more advantages. Simulations are not always available in closed form expression, i.e., when using a proprietary simulator software. Using NN also works when only historical data from the physical plant is available and no simulation model exists.

4. Summary

This article showed how to solve a basic problem by formulating it as a MILP where a NN was integrated as the left-hand side of a constraint. The NN was implemented in PyTorch and the optimization model in GAMSPy. The GAMSPy package formulations was used to replicate the structure of the NN in the MILP. Integrating NN and optimization models can be used for many other applications like adversarial input generation, model verification, and customized training of NN.

References

- [1] Atharv Bhosekar and Marianthi Ierapetritou. Modular design optimization using machine learning-based flexibility analysis. *Journal of Process Control*, 90:18–34, 2020.

Finding long TSP tours in the unit square using the FICO Xpress Global solver

Imre Pólik,

FICO

imre.polik@gmail.com

Abstract Finding a set of points in the unit square such that the length of the optimal TSP tour is longest possible was first mentioned in the 1940s. The asymptotics were established in the 1980s, but the actual solution of the problem for a given number of points is still an open question. In this talk we will give a theoretical background of the problem, show some bounding techniques and establish new point configurations, some of which are provably optimal. We will formulate several conjectures and pose some open questions. Computational experiments with the FICO Xpress Global optimization solver are also presented.

MOSEK as a system component

Gustaf Ehn,

MOSEK

gustaf.ehn@mosek.com

Abstract Outside of academia it is hard to find interest in solving an optimization problem just to have solved it. The solution of the optimization should be useful in some sense, it should allocate personnel to shift, control a robot etc. That means for a solver to be useful it needs to be integrable into a larger tech stack. The ability to integrate MOSEK into solutions is why we can exist as a company. In this talk we will dig into the details of what is required of a solver in order for it to be a reliable system component.

Computing global optimum in Hexaly

Julien Darlay,

Hexaly, *Co-Founder & Head of Science*

jdarlay@hexaly.com

Abstract

Hexaly is a model-and-run solver that combines heuristics with exact methods. In addition to supporting the classical MINLP formalism, it introduces a set-based modeling approach that simplifies the formulation of some combinatorial problems, such as vehicle routing, production scheduling and packing. For problems involving scalar operators only, Hexaly's exact solver employs a nonlinear branch-and-bound algorithm. For set-based models, automatic reformulations are applied to derive either compact models that can be solved with standard branch & bound or extended formulations, which are then solved using branch price and cut or column elimination techniques. This talk will present the key techniques used by Hexaly to compute optimal solutions to nonlinear problems.

Session 6





Upcoming features in SHOT 2.0

Andreas Lundell¹, Alireza Olama¹, Joakim Blomqvist², Johan Ekman¹ and Jan Kronqvist³

¹*Department of Information Technology, Åbo Akademi University, Vaasa, Finland
firstname.lastname@abo.fi*

²*Department of Mathematics, Åbo Akademi University, Turku, Finland
joakim.blomqvist@abo.fi*

³*Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden
jankr@kth.se*

Abstract The Supporting Hyperplane Optimization Toolkit (SHOT) is an open-source polyhedral approximation-based solver for mixed-integer nonlinear programming (MINLP). In this extended abstract, we present some of the new features of the upcoming SHOT 2.0 release. These include expanded callback support, offering users greater control over the solver, as well as a Python interface facilitating rapid prototyping and easier integration into advanced workflows. Furthermore, support for HiGHS as an mixed-integer programming (MIP) subsolver in SHOT has been added, providing a fast, actively maintained open-source alternative for solving MINLP problems.

Keywords: Mixed-integer nonlinear programming (MINLP), Supporting Hyperplane Optimization Toolkit (SHOT)

1. Introduction

SHOT is a global solver for convex MINLP [1]. For general nonconvex MINLP, SHOT is a heuristic, *i.e.*, local, solver, but due to its reformulation capabilities and tight integration with the underlying MIP solver, it can also solve some classes of nonconvex problems to global optimality [2].

The solution process in SHOT consists of a dual strategy providing a lower bound to the global objective value, and a primal strategy providing integer-feasible solutions, the objective value of the lowest of which provides a primal bound. The dual strategy in SHOT is based on polyhedral (outer) approximation of the feasible set formed by the nonlinear constraints in the problem considered, and utilizes two different types of methods for creating linearizations of the nonlinear feasible set: the extended supporting hyperplane (ESH) [3] and extended cutting plane (ECP) [4] algorithms. These algorithms solve sequences of mixed-integer linear programming (MILP), mixed-integer quadratic programming (MIQP), or mixed-integer quadratically constrained quadratic programming (MIQCQP) problems. The solution points to the subproblems are used to generate cuts in the form of supporting hyperplanes or cutting planes. The primal strategy in SHOT is based on heuristics, *e.g.*, solving NLP problems or performing root searches.

The upcoming SHOT 2.0 release provides several improvements, mainly for advanced users, these includes additional callback support in the C++ interface that enables more fine-control of the solution process, as well as a Python interface. The new version also adds support for HiGHS [5] as a subsolver in the dual strategy.

2. Additional callbacks

In previous versions, SHOT’s C++ interface offered only basic support for user control of and information about the solution process through callbacks. Specifically, users could define (*i*) a termination callback to prematurely stop the solver based on custom criteria, such as a desired number of solutions found or a specific solution quality, and (*ii*) a callback triggered whenever a new primal solution was discovered by SHOT.

With SHOT 2.0, this functionality will be significantly extended by introducing several new callbacks, providing users with much greater flexibility to customize and influence the solver’s behavior:

Rootsearch Point Callback: Allows users to specify interior end points for the rootsearch in the ESH algorithm. This can improve performance by providing an initial search direction for the polyhedral approximation.

Cut Generation Callback: Enables users to iteratively add custom cuts in the dual strategy, either replacing or complementing the ECP or ESH method. These cuts are passed on to the MIP solver, also as lazy constraints if supported by the subsolver. This callback can also be used

to pass an initial polyhedral approximation for warm-starting the solver, facilitating experimentation with novel cut generation strategies.

Constraint Violation Callback: Provides external feasibility checking, *i.e.*, rejection, of candidate primal solutions.

Primal Solution Callback: Gives users the ability to inject primal solutions into the solver, for example, solutions obtained from external computations such as heuristics.

Dual Bound Callback: Allows users to provide their own dual bounds (used for termination in SHOT) during the solution process, potentially improving convergence behavior by enabling custom bounding strategies in SHOT’s dual strategy.

These new callbacks open up SHOT as a highly flexible platform for algorithm development, prototyping, and advanced research in global optimization. Specifically, when combining the cut generation and constraint violation callbacks, this allows SHOT to handle external constraints that are not originally included in the MINLP problem formulation through a mechanism similar to lazy constraints in MIP solvers.

3. Python interface

In earlier versions, it was already possible to use SHOT within Python environments through Pyomo’s AMPL interface. In this setup, optimization problems are formulated through Pyomo, and when solving the problem, it is exported to an AMPL NL-file passed to the SHOT binary. SHOT then solves the problem externally, and the solution is written to a file read by Pyomo. Although functional, this approach introduced certain limitations — most notably, a reliance on Pyomo, which is a large and complex dependency, and restricted interaction with the solver beyond basic input/output. Starting with SHOT 2.0, we introduce a dedicated Python interface directly linked to SHOT’s C++ API. This integration eliminates the need for external problem formats and intermediate file handling, providing a faster and more seamless user experience. Importantly, it reduces dependency overhead and makes SHOT significantly easier to integrate into lightweight Python projects.

Beyond these practical advantages, the new interface greatly expands what is possible for users and developers. It exposes SHOT’s internal callback mechanisms to Python, allowing users to hook into the solver’s execution flow, customize behavior, and experiment with new heuristics or

polyhedral approximation strategies. This feature opens the door for SHOT to serve not only as a solver but also as a flexible platform for developing and testing novel optimization techniques.

Additionally, the Python interface provides access to SHOT’s internal functionalities: users can load model files, configure solver options programmatically, inspect detailed solution information, and retrieve the complete solution pool with ease. Thus, the interface enables new ways of utilizing SHOT in advanced workflows and applications.

4. Support for HiGHS as MIP solver

Until now, SHOT has supported Gurobi, CPLEX, and Cbc as the underlying MIP solvers in its dual strategy. In the upcoming release, support for HiGHS, a fast and rapidly developing open-source MIP solver, is added. This addition strengthens SHOT’s competitiveness and provides users with an efficient, fully free alternative to commercial solvers.

Moreover, since HiGHS currently does not natively support quadratic objective functions or constraints like many of the commercial MIP offerings, SHOT naturally extends HiGHS’ capabilities, enabling the combined solver to solve convex MIQCQP problems to global optimality.

References

- [1] A. Lundell, J. Kronqvist and T. Westerlund. The supporting hyperplane optimization toolkit for convex MINLP. *Journal of Global Optimization*, 84, 2022, pp. 1–41.
- [2] A. Lundell and J. Kronqvist. Polyhedral approximation strategies for nonconvex mixed-integer nonlinear programming in SHOT. *Journal of Global Optimization*, 82(4), 2022, pp. 863–896.
- [3] J. Kronqvist, A. Lundell and T. Westerlund. The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization*, 64, 2016, pp. 249–272.
- [4] T. Westerlund and F. Pettersson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 19, 1995, pp. 131–136.
- [5] Q. Huangfu and J. A. J. Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1), 2018, pp. 119–142.

Improved Bounding Techniques for Nonconvex MINLP in SHOT

Joakim Blomqvist,¹ Andreas Lundell²

¹*Department of Mathematics, Åbo Akademi University, Turku, Finland*
joakim.blomqvist@abo.fi

²*Department of Information Technology, Åbo Akademi University, Vaasa, Finland*
andreas.lundell@abo.fi

Abstract The Supporting Hyperplane Optimization Toolkit (SHOT) is an optimization solver based on polyhedral outer approximation, primarily designed for convex mixed-integer nonlinear programming (MINLP) problems. Nonconvexities are handled through reformulations and heuristic techniques, and thus, the solver is positioned somewhere between a global and heuristic solver for nonconvex problems. This paper introduces two heuristic enhancements to improve SHOT's nonconvex capabilities, primarily by reducing the optimality gap. The first technique involves solving a convex bounding problem to improve the dual bound, and the second updates an existing primal objective cut heuristic within SHOT to improve the primal bound.

Keywords: Mixed-integer nonlinear programming (MINLP), Nonconvex MINLP, Supporting Hyperplane Optimization Toolkit (SHOT)

1. Introduction

Due to the computational complexity of nonconvex MINLP problems, obtaining a guaranteed optimal solution or tight global bounds within a limited time may not always be feasible. Therefore, users often accept finding a good feasible solution rather than a globally optimal one. Heuristics are thus often used, as they allow solvers like SHOT to explore the search space more efficiently and find good feasible solutions within a reasonable time.

While heuristics do not guarantee global optimality, they can significantly improve solution quality and help guide the overall process, making them a valuable complement to strategies specifically designed for global optimization in nonconvex MINLP. Techniques for handling nonconvexities are essential in this context, as treating these as convex often results in infeasible problems or solutions that are far from optimal.

2. Background

In this paper, we consider general nonconvex MINLP problems of the following form:

$$\begin{aligned}
 & \text{minimize} && \mathbf{c}^T \mathbf{x} \\
 & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{a}, \quad \mathbf{B}\mathbf{x} = \mathbf{b}, \\
 & && g_k(\mathbf{x}) \leq 0, \quad k \in K_g, \\
 & && h_k(\mathbf{x}) \leq 0, \quad k \in K_h, \\
 & && q_k(\mathbf{x}) = 0, \quad k \in K_q, \\
 & && \underline{x}_i \leq x_i \leq \bar{x}_i \quad \forall i \in I = \{1, 2, \dots, n\}, \\
 & && x_i \in \mathbb{R}, x_j \in \mathbb{Z} \quad \forall i \in I \setminus I_Z, \forall j \in I_Z.
 \end{aligned} \tag{1}$$

In this formulation, I_Z denotes the index set corresponding to integer variables. Here, $g(\mathbf{x}) \leq 0$ represents the nonlinear convex constraints, while $h(\mathbf{x}) \leq 0$ and $q(\mathbf{x}) = 0$ represent the nonlinear nonconvex constraints. Additionally, all nonlinear constraints are assumed to be differentiable. Although the assumption of a linear objective simplifies the model, it does not limit generality, since a nonlinear objective can be reformulated using an epigraph reformulation. It is worth noting, however, that SHOT internally treats nonlinear objectives separately.

In SHOT [1], the dual bound (DB) of Problem (1) is obtained by the extended cutting plane (ECP) and extended supporting hyperplane (ESH) algorithms, referred to as dual strategies, while the primal bound (PB) is obtained from feasible, or so-called primal, solutions to the problem. The DB represents a valid lower bound on the optimal objective value obtained from solving a polyhedral outer approximation of Problem (1), while PB refers to the objective value of the best-known feasible solution. The dual strategy aims to improve the DB by constructing and refining a polyhedral outer approximation of the feasible region using cuts generated by the ECP and ESH strategies. Meanwhile, the primal strategy, based on primal heuristics, helps the solver quickly find good feasible solutions, thereby

improving the PB. SHOT combines these strategies to solve MINLP problems, strengthen both bounds and improve the overall solution process.

Detailed information on SHOT's functionality for solving convex MINLP can be found in [1], and for solving nonconvex MINLP in [2]. The improved bounding techniques introduced in this paper can be found in [3].

3. New Convex Bounding Strategy

In practice, SHOT prefers generating cuts for convex constraints, and only when this is no longer possible, the solver will add cuts for the nonconvex constraints. In other words, after N iterations, SHOT will have generated a set of cuts, which, for simplicity, we denote by $\text{CUT}_C^N(\mathbf{x})$, for the convex constraints and obtained a valid lower bound LB_N . After an additional M iterations SHOT will have generated the cuts in the set $\text{CUT}_C^{N+m_1}(\mathbf{x})$ and $\text{CUT}_{NC}^{m_2}(\mathbf{x})$ for the convex and nonconvex constraints respectively. Since the cuts for the nonconvex constraints may exclude feasible solutions, the resulting LB obtained may no longer be valid. To regain a valid LB, the following convex bounding problem can be solved:

$$\begin{aligned} & \text{minimize} && \mathbf{c}^T \mathbf{x} \\ & \text{subject to} && \mathbf{A}\mathbf{x} \leq \mathbf{a}, \quad \mathbf{B}\mathbf{x} = \mathbf{b}, \\ & && \text{CUT}_C^{N+m_1}(\mathbf{x}) \leq 0, \\ & && \underline{x}_i \leq x_i \leq \bar{x}_i \quad \forall i \in I = \{1, 2, \dots, n\}, \\ & && x_i \in \mathbb{R}, x_j \in \mathbb{Z} \quad \forall i \in I \setminus I_Z, \forall j \in I_Z. \end{aligned} \tag{2}$$

If this problem results in a new valid lower bound, LB_{N+M} , then a better LB for Problem (1) has been obtained. Note that if the chosen subsolver supports mixed-integer quadratically constrained quadratic programming (MIQCQP) problems, then convex quadratic constraints or objective functions can be included directly into Problem (2).

4. Improved Primal Objective Cuts

To force SHOT to keep searching for better primal solutions to nonconvex problems, a primal objective cut can be utilized. This cut is given by:

$$\mathbf{c}^T \mathbf{x} \leq \gamma \cdot \text{PB}. \tag{3}$$

The default setting in SHOT is $\gamma = 0.999$, which ensures that the algorithm searches for a solution with an objective value strictly smaller than the current PB. This process can be repeated until a user-defined number of iterations is reached, a satisfactory solution is found, or the optimality gap becomes sufficiently small. However, this introduces an iterative process, as each solution must improve marginally upon the previous one. To improve efficiency, adjusting γ or finding a quicker method to lower the objective function value could help.

One such method is a golden section-based line search between the primal and dual bounds. This approach generates alternative cuts to more efficiently explore subproblems in search of better primal solutions. Each iteration solves a new subproblem with either an objective cut given by

$$\mathbf{c}^T \mathbf{x} \leq \text{PB} - \gamma \cdot (\text{PB} - \text{DB}) = \text{PB}'_{(1,\mathcal{P})} \quad (4)$$

or

$$\mathbf{c}^T \mathbf{x} \leq \text{DB} + \gamma \cdot (\text{PB} - \text{DB}) = \text{PB}'_{(2,\mathcal{P})}, \quad (5)$$

where $\gamma = 0.618$. The set \mathcal{L} , initialized as (DB, PB) , stores bounds used to compute new PB' values. It is updated and sorted after each iteration based on each new PB' , and SHOT continues the search between each pair in \mathcal{L} until a primal solution is found or a termination criterion is met.

5. Summary

This paper discussed enhancements to SHOT for solving nonconvex MINLP problems. The convex bounding strategy improves the dual bound, while the updated primal cut procedure improves the primal bound.

References

- [1] A. Lundell, J. Kronqvist and T. Westerlund. The supporting hyperplane optimization toolkit for convex MINLP. *Journal of Global Optimization*, 84, 2022, pp. 1–41.
- [2] A. Lundell and J. Kronqvist. Polyhedral approximation strategies for nonconvex mixed-integer nonlinear programming in SHOT. *Journal of Global Optimization*, 82(4), 2022, pp. 863–896.
- [3] J. Blomqvist. *Enhancements for Nonconvex MINLP in SHOT*, Master’s thesis, Åbo Akademi University, 2025. URL: <https://urn.fi/URN:NBN:fi-fe2025050637502>

A Parallel Framework for Primal Heuristics in SHOT

Alireza Olama,¹ Andreas Lundell¹

¹*Information Technology, Åbo Akademi University, Vaasa, Finland* firstname.lastname@abo.fi

Abstract The Supporting Hyperplane Optimization Toolkit (SHOT) is a solver designed for Mixed-Integer Nonlinear Programming (MINLP) problems. In its current form, SHOT handles primal heuristics sequentially on a single thread, which blocks the solver's execution until the heuristic process completes. In this work, we utilize SHOT's modular architecture to develop a new approach that enables primal heuristics to run in parallel with the main optimization process, without blocking its progress. This allows for improved efficiency and scalability of the solver. The proposed framework will be evaluated on a representative set of benchmark problems from the MINLPLib library to assess its performance and effectiveness.

Keywords: Mixed-Integer Nonlinear Programming (MINLP), Parallel Primal Heuristics, Supporting Hyperplane Optimization Toolkit (SHOT)

1. Introduction

The Supporting Hyperplane Optimization Toolkit (SHOT) is an open source global solver designed to solve Mixed Integer Nonlinear Programming (MINLP) problems, with a primary focus on convex MINLP [1, 2]. SHOT employs a primal-dual strategy to obtain a sequence of lower and upper bounds on the objective function value for the optimization problem. These bounds are then utilized to terminate the algorithm and to determine the quality of the solution. The dual strategy relies on polyhedral outer approximation (POA) techniques, such as the Extended Supporting Hyperplane (ESH) [3] and Extended Cutting Plane (ECP) [4] methods, to create

linear approximations of the feasible region defined by the nonlinear constraints. Meanwhile, the primal strategy focuses on finding integer-feasible solutions using heuristic algorithms. This can involve solving Nonlinear Programming (NLP) subproblems obtained by fixing the integer variables in the original MINLP problem, or utilizing the solution pool generated by a Mixed-Integer Programming (MIP) solver (such as Gurobi or CPLEX) to solve multiple NLP instances simultaneously.

One limitation of SHOT's current architecture is that its primal heuristics execute sequentially on a single thread and block the solver's overall progress until the heuristic process completes. This creates a bottleneck especially in large-scale problems where finding feasible solutions can take considerable time and often requires tasks such as solving NLP subproblems. The sequential approach also fails to leverage modern multi-core systems that can handle multiple tasks simultaneously. To address this limitation we propose a parallel implementation of the primal heuristics allowing them to run concurrently with the main solver as it continues its dual strategy. By parallelizing the heuristics we aim to reduce solution times, find multiple feasible solutions by solving different NLP subproblems in parallel using the solution pool provided by the MIP solver, and improve the quality of upper bounds.

2. SHOT Architecture

SHOT is implemented in C++17 with a single-threaded programming model at its core but leverages multithreading within the MIP solver layer if the chosen solver supports parallel execution. The software architecture of SHOT follows a task-based programming model, as illustrated in Fig. 1. The solver progresses through a sequence of tasks including *Create MIP*, *Add cuts*, *Solve MIP*, *Check gap*, and *Goto*. These tasks are executed repeatedly until the primal-dual gap falls below a predefined threshold. All tasks interact with a central *Environment* data structure which manages the solver's core components including user settings, the original and reformulated problem definitions, the modeling system, the task handler, and results storage. In SHOT, solution strategies are constructed by creating a task queue where tasks are arranged in a predefined order and executed sequentially on a single thread. This task-based design enables better algorithm management, greater flexibility for extensions, and easier integration of various user-defined settings. By structuring the solver around discrete tasks, SHOT maintains a clean and modular architecture that simplifies the addition of new features and the customization of solution flows. Further-

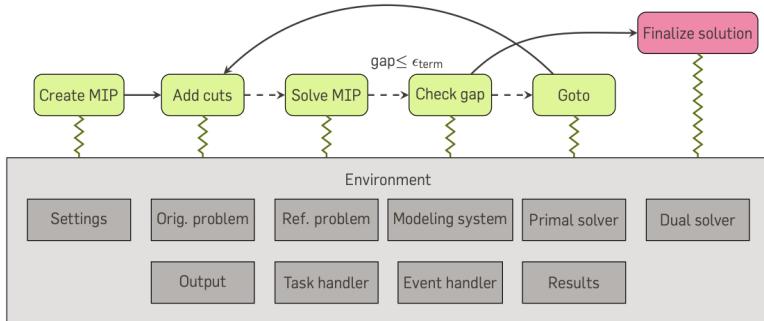


Figure 1: SHOT's Software Architecture

more, this task-based model naturally aligns with parallel programming paradigms, specifically task parallelism, where independent or loosely coupled tasks can be executed concurrently. This structure provides a foundation for introducing parallel primal heuristics, which is discussed in detail in the next section.

3. Parallel Primal Heuristics

To enable parallel primal heuristics functionality in SHOT, we extend SHOT's existing task-based architecture to enable true concurrent execution of both primal and dual strategies. A new software component called `TaskParallel` is introduced, which allows the solver to execute the dual strategy on the main thread while spawning multiple worker threads to run primal heuristics independently. For example, several NLP solvers such as IPOPT and CONOPT can be launched in parallel, where each solver waits for feasible integer points generated by the MIP solver and immediately starts solving fixed NLP subproblems once a new solution is available. Other primal heuristics can also be integrated into this framework, running in the background to continuously explore new feasible solutions without interrupting the dual optimization process. This parallel design improves solver responsiveness and better utilizes modern multi-core hardware. A custom thread pool is implemented in SHOT to manage the concurrent execution of primal heuristics. The pool is built using standard C++ constructs such as `std::thread`, `std::mutex`, and `std::condition_variable` to ensure safe and efficient multithreaded operation. Instead of relying on busy-waiting, a condition-variable-based signaling mechanism is used to wake

idle threads whenever new tasks are added to the queue which improves scalability and optimizes CPU resource usage. The TaskParallel component handles task dispatch by immediately assigning new primal heuristic tasks to available worker threads as soon as the dual strategy produces a candidate solution. This design enables primal heuristics to run continuously in parallel with the dual strategy without blocking or disrupting its execution flow.

Inter-thread communication and data sharing are managed through the Environment data structure, which provides threads with controlled access to problem data, solver settings, and solution results. To ensure thread safety and avoid data races, the Environment uses fine-grained synchronization through mutexes, allowing threads to lock only the specific parts of the data they modify. When a worker thread finds a new primal solution, it performs an atomic update to the shared results, ensuring that the data remains consistent without requiring full locks. The main thread monitors these atomic updates asynchronously and integrates new solutions into the solver's workflow.

References

- [1] A. Lundell, J. Kronqvist and T. Westerlund. The supporting hyperplane optimization toolkit for convex MINLP. *Journal of Global Optimization*, 84, 2022, pp. 1–41.
- [2] A. Lundell and J. Kronqvist. Polyhedral approximation strategies for nonconvex mixed-integer nonlinear programming in SHOT. *Journal of Global Optimization*, 82(4), 2022, pp. 863–896.
- [3] J. Kronqvist, A. Lundell and T. Westerlund. The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization*, 64, 2016, pp. 249–272.
- [4] T. Westerlund and F. Pettersson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 19, 1995, pp. 131–136.

The CAMINO Software Package

Andrea Ghezzi¹, Wim Van Roy², Sebastian Sager³, Moritz Diehl^{1,4}

¹*Department of Microsystems Engineering (IMTEK), University of Freiburg, Germany*
andrea.ghezzi@imtek.uni-freiburg.de, mortiz.diehl@imtek.uni-freiburg.de

²*Atlas Copco Airpower NV* wim.van.roy@atlascopco.com

³*Institute of Mathematical Optimization, Otto-von-Guericke Universität Magdeburg, Germany* sager@ovgu.de

⁴*Department of Mathematics, University of Freiburg*

Abstract This work presents CAMINO (a Collection of Algorithms for Mixed-Integer Nonlinear Optimization), an open-source Python package. CAMINO leverages CasADi, a symbolic framework for automatic differentiation with interfaces to many solvers, and can process mixed-integer nonlinear programs (MINLPs) defined using CasADi or as AMPL “.n1” files. Thanks to its friendly code structure and tight integration with CasADi, CAMINO is highly adaptable. Over the past year, it has been successfully applied in several domains including control of district heating networks, building climatization, and space-craft trajectory planning. A promising algorithm within CAMINO is the recent Sequential Benders-based Mixed-Integer Quadratic Programming (S-B-MIQP) algorithm [6], which provides guaranteed convergence for convex MINLPs while offering effective heuristic solutions for nonconvex ones. Comparative evaluations against state-of-the-art open-source solvers on a large subset of MINLPLib instances demonstrate that S-B-MIQP achieves comparable performance for convex MINLPs while outperforming existing methods for nonconvex ones. CAMINO is accessible at <https://github.com/minlp-toolbox/CAMINO>.

1. A brief introduction to CAMINO

CAMINO is an open-source framework designed to tackle MINLPs with both flexibility and performance in mind. Built with high-level object-oriented Python, CAMINO enables researchers and practitioners to rapidly prototype and implement novel algorithmic approaches while maintaining computational efficiency. The framework leverages CasADi [1], a powerful symbolic framework for automatic differentiation with extensive solver interfaces, allowing CAMINO to handle complex optimization problems with minimal overhead. At its core, CAMINO addresses MINLPs formulated as:

$$\begin{aligned} & \underset{x \in X \subset \mathbb{R}^{n_x}}{\text{minimize}} && f(x, p) \\ \text{MINLP :} \quad & \text{subject to} && x_{\text{lb}} \leq x \leq x_{\text{ub}}, \\ & && g_{\text{lb}} \leq g(x, p) \leq g_{\text{ub}}, \\ & && x_i \in \mathbb{Z}, i \in I \subseteq \{1, \dots, n_x\}, \end{aligned} \quad (1)$$

where X is a bounded polyhedral set, functions $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_g}$, $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_h}$ are twice continuously differentiable, and $p \in \mathbb{R}^{n_p}$ represents a vector of parameters.

The symbolic expressions x, p, f, g are CasADi expressions which can be created either using the Python interface of CasADi or using the CasADi builder for reading AMPL “.nl” files. The symbolic expressions for x, p, f, g and the set I , expressed as a list of boolean with length n_x , are used to instantiate an object of the class `MinlpProblem`, while an object of the class `MinlpData` collects the values $x_{\text{lb}}, x_{\text{ub}}, g_{\text{lb}}, g_{\text{ub}}$, numerical values for the parameter vector p , and an initial guess for x . Solver settings are collected in the class `Settings` and runtime statistics in the class `Statistics`. Finally, the user selects a MINLP algorithm and can instantiate a solver from the class `MinlpSolver`. Table 1 lists some of the available MINLP solvers.

CAMINO offers both direct integration with established MINLP solvers and its own implementation of algorithms. For instance, for Bonmin [2] which is already accessible through CasADi, CAMINO provides a wrapper with some additional functionalities beyond the CasADi `nlpso1` interface. However, CAMINO’s true strength lies in its comprehensive implementation of advanced algorithms, including outer approximation, feasibility pump, and S-B-MIQP, where CasADi serves to formulate subproblems and connect to appropriate underlying solvers. The framework’s flexibility allows users to select from any solver interfaced with CasADi based on their specific requirements.

Table 1: Subset of available solvers in CAMINO.

Solver	Description
bonmin-bb	Nonlinear branch-and-bound in Bonmin [2]
fp	Feasibility pump [4]
oa	Outer approximation [5, 3]
oa-qp	Quadratic outer approximation [5]
s-b-miqp	Sequential Benders-based MIQP [6]

Complete list of solvers available at <https://github.com/minlp-toolbox/CAMINO>.

CAMINO allows to chain different solvers such that the solution of a solver can be used as an initial guess for the successive solver. For instance, the user can chose the feasibility pump to quickly obtain a feasible point and then switch to an outer approximation algorithm to compute an optimal point simply by declaring `fp+oa` as a solver.

Benchmark results. We compare S-B-MIQP with SHOT v1.1 [7] and Bonmin v1.8 [2], on a subset of instances from MINLPLib that possess an AMPL “.nl” representation. In total, we have selected 116 convex MINLPs and 156 nonconvex MINLPs. Computations run on a single thread of a Intel(R) Xeon(R) W-2225 CPU @ 4.10GHz with 16GB of memory, with a time limit of 300 seconds. To the best of authors’ knowledge, all algorithms share same tolerances for termination, same solvers for the subproblems with same settings. A detailed comparisons with all the relevant data will be soon published in the CAMINO repository. Fig. 1 and Fig. 2 show the obtained results in form of performance profiles for the objective value and wall time.

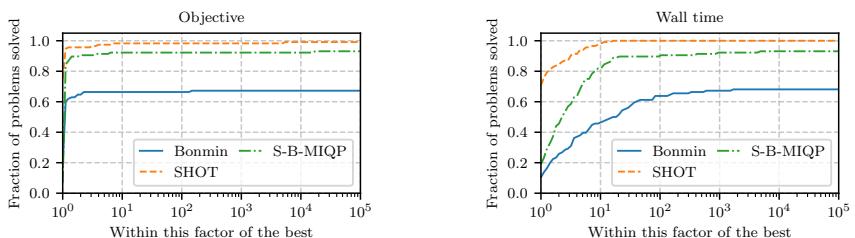


Figure 1: Performance profile for the subset of convex MINLPs.

Outlook. We want to expand the CasADi interface to MIP solvers in order to support callbacks. This enhancement will enable “single-tree” implementations of outer approximation approaches, as done in SHOT [7],

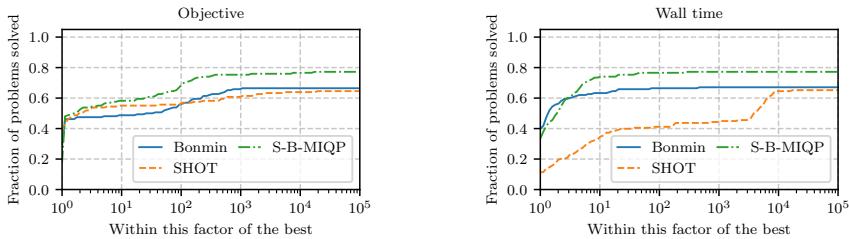


Figure 2: Performance profile for the subset of nonconvex MINLPs.

offering potential improvements in both solution time and algorithm robustness. Finally, CAMINO serves as an algorithmic prototyping platform where promising approaches can be identified, refined, and ultimately implemented in high-performance C++ code. These optimized implementations can then be natively distributed with CasADi, making advanced MINLP solvers accessible to CasADi’s community.

References

- [1] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi – a software framework for nonlinear optimization and optimal control. *Math. Program. Comput.*, 11(1):1–36, 2019.
- [2] P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An Algorithmic Framework For Convex Mixed Integer Nonlinear Programs. Technical report, IBM T. J. Watson Research Center, 2005.
- [3] M. Duran and I.E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Program.*, 36(3):307–339, 1986.
- [4] Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104:91–104, 2005.
- [5] R. Fletcher and S. Leyffer. Solving Mixed Integer Nonlinear Programs by Outer Approximation. *Math. Program.*, 66:327–349, 1994.
- [6] Andrea Ghezzi, Wim Van Roy, Sebastian Sager, and Moritz Diehl. A sequential benders-based mixed-integer quadratic programming algorithm. *arXiv preprint arXiv:2404.11786*, 2024.
- [7] Andreas Lundell, Jan Kronqvist, and Tapio Westerlund. The supporting hyperplane optimization toolkit for convex minlp. *Journal of Global Optimization*, 84(1):1–41, 2022.

Session 7





Advanced Geometrical test for efficient use of optimality conditions in Interval Branch and Bound methods

Mihály Gencsi, Boglárka G.-Tóth

University of Szeged, {gencsi,boglarka}@inf.szte.hu

Abstract

The Interval Branch and Bound (IBB) method is a widely used approach for solving nonlinear programming problems where a rigorous solution is required. In the literature, a wide range of variations of IBB exists. However, few IBB implementations use the Karush-Kuhn-Tucker (KKT) or the Fritz-John (FJ) optimality conditions to eliminate non-optimal boxes. We extend the Interval Branch and Bound (IBB) method with an advanced geometrical test, deciding when the FJ optimality conditions cannot hold or whether it is convenient to use the FJ optimality conditions or not. The efficiency of the advanced geometrical test was considered through computational experiments on a generated benchmark. The best variant can save more than 40% of computational time in average using the designed geometrical test.

Keywords: Interval Arithmetic Branch and Bound method Fritz-John optimality conditions Geometrical test

1. Introduction

There are many real applications where it is not enough to find a near-optimal solution, but a guaranteed solution is necessary. This is the case, for instance in physics or chemistry, when we look for the stability point of a substance. We focus on solving inequality-constrained global optimization problems, namely, the following nonconvex nonlinear problem:

$$\begin{aligned}
& \underset{x \in \mathbf{y} \subseteq \mathbb{R}^n}{\text{minimize}} && f(x) \\
& \text{subject to} && g_i(x) \leq 0, \quad i \in M_c,
\end{aligned} \tag{1}$$

where $f, g_i : \mathbb{R}^n \rightarrow \mathbb{R}, i \in M_c$ are continuously differentiable nonlinear functions, and the interval box $\mathbf{y} = [\underline{y}, \bar{y}]$ gives the general bound constraints. A bound constraint for $x_i \in [\underline{y}_i, \bar{y}_i]$ can be formulated also as $p_{i_u}(x) = x_i - \bar{y}_i$ and $p_{i_l}(x) = \underline{y}_i - x_i$. Thus, we can use $p_j(x) \leq 0, j \in M_b$ for the bound constraints, where $M_b = \{i_u, i_l \mid i = 1, \dots, n\}$.

In this study, we investigate the efficient use of optimality conditions to improve the IBB method. We focus on an advanced geometrical test that eliminates non-optimal boxes and avoids using the FJ conditions when they are ineffective, increasing their success rate and reducing computation time.

1.1 The Normalized Interval Fritz-John Optimality Condition System

The normalized interval version of the Fritz-John optimality condition system (NIFJ-CS), following [1], for a given box \mathbf{x} is a system of interval linear equations in matrix form

$$\phi(\mathbf{t}) = \begin{bmatrix} \mu_0 + \sum_{i \in B \cup C} \mu_i - 1 \\ \mu_0 \cdot \nabla \mathbf{f}(\mathbf{x}) + \sum_{i \in B} \mu_i \cdot \nabla \mathbf{p}_i(\mathbf{x}) + \sum_{j \in C} \mu_j \cdot \nabla \mathbf{g}_j(\mathbf{x}) \\ \mu_i \cdot \mathbf{p}_i(\mathbf{x}) & i \in B \\ \mu_j \cdot \mathbf{g}_j(\mathbf{x}) & j \in C \end{bmatrix} = 0, \tag{2}$$

where $\mathbf{f}(\mathbf{x}), \mathbf{p}_i(\mathbf{x}), \mathbf{g}_j(\mathbf{x})$ are the inclusion functions, $\nabla \mathbf{f}(\mathbf{x}), \nabla \mathbf{p}_i(\mathbf{x}), \nabla \mathbf{g}_j(\mathbf{x})$ are the inclusions of the gradients for $f(x), p_i(x), g_j(x)$, respectively. The system is formalized only in active constraints, that is, for the constraints where the inclusion $\mathbf{p}_i(\mathbf{x}), \mathbf{g}_j(\mathbf{x})$ contains zero ($B = \{i \in M_B \mid 0 \in \mathbf{p}_i(\mathbf{x})\}$, $C = \{j \in M_C \mid 0 \in \mathbf{g}_j(\mathbf{x})\}$). The vector of variables is $\mathbf{t} = [\mathbf{x}, \boldsymbol{\mu}]^T$, which is $N = n + |B| + |C| + 1$ dimensional. The normalization in the first line of (2) is important for the FJ conditions. If the system of equations (2) has a solution with $\boldsymbol{\mu} \geq 0$, then a box \mathbf{x} can contain an optimizer.

In the literature, there are several approaches to solving Fritz-John optimality conditions. Studies [2, 3] describe three different methods: calculating bounds on the Lagrange multipliers only, enclosing both the Lagrange multipliers and the location of the optimal points, and using the Taylor expansion of the Fritz-John conditions.

2. Advanced geometrical optimality test

The KKT necessary optimality conditions are satisfied when we can formulate the negative gradient of the objective function as a linear combination of the constraints' gradients with non-negative real scalars. The FJ conditions allow the option that the linear combination of the constraints' gradients gives zero (with non-negative non-all-zero real scalars). In this case, the point x^* can be a local or global optimizer point. Graphically, the FJ necessary conditions require that there is a direction in the negative cone of the objective's gradient,

$$\mathbf{CF} = \{d \in \mathbb{R}^n \mid d \in -\mu_0 \nabla f(\mathbf{x}), \mu_0 \geq 0\},$$

which lies in the conic hull of the gradient enclosures. I.e. $\mathbf{CF} \cap \mathbf{CH} \neq \emptyset$, where the conic hull in the interval setting is defined as

$$\mathbf{CH} = \left\{ d \in \mathbb{R}^n \mid d \in \sum_{i \in B} \mu_i \nabla p_i(\mathbf{x}) + \sum_{i \in C} \mu_i \nabla g_i(\mathbf{x}), \mu \geq 0, \mu \neq 0 \right\}.$$

Depending on the quality of the enclosures, the overestimation of the conic hulls can range from being very sharp to being very tight. However, what really matters is that the conic hulls will always enclose the exact gradients, i.e. $CH \subseteq \mathbf{CH}$. That is, we may not realize that an interval cannot contain an optimizer, but we never reject a box that might contain one.

Instead of directly using the Lagrange estimator method or the Lagrange estimator + NIFJ-CS method, we build a preliminary method that avoids unnecessary computations, namely the Advanced Geometrical Test.

The test involves performing necessary computations step by step, saving computational time, and returning as soon as possible by examining the conic hull \mathbf{CH} and the cone \mathbf{CF} . First we check if the gradient $\nabla f(\mathbf{x})$ contains zero, or if one of the gradients of the active constraints contains zero in its interior. If either is true, the box cannot be reduced, so we stop. Otherwise, for each dimension, we determine the sign(s) of the gradient enclosures (could be +, -, 0+, 0-, or +-). Having these signs, we check whether they are strictly different, hence the optimality conditions cannot hold. Therefore, we stop the procedure, and the box can be discarded.

Next, we check whether every orthant contains gradient enclosures of the active constraints, which means the conic hull \mathbf{CH} is full, so we stop.

In the one active constraint case, we calculate the possible intervals of the only Lagrange multiplier for each dimension, and if the intersection of these intervals is empty, we discard the box and stop the procedure.

A similar check can be done for more active constraints: for the interval hull of the gradient enclosures of the active constraints, one can calculate for each dimension the possible intervals for an overall Lagrange multiplier. Similarly, if the intersection of these intervals is empty, we discard the box and stop the procedure.

Finally, we pair two dimensions, where for at least one, all gradient enclosures of the active constraints are either non-negative or non-positive. For each such pair we calculate the slopes for CH and CF stored in M_g and M_f , as on Figure 1. If $M_f \cap M_g = \emptyset$ for a pair, then the box can be rejected. Although many pairings might exist, orderings in pairings allow us to avoid exploring all possible pairs.

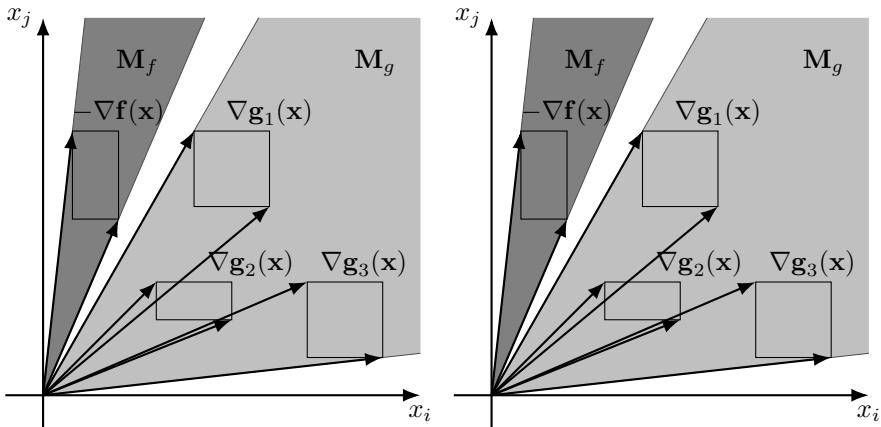


Figure 1: A discardable and a non-discardable case by the optimality conditions

References

- [1] Mihály Gencsi and Boglárka G.-Tóth. The Fritz-John condition system in interval branch and bound method. *Annales Mathematicae et Informaticae*, 2023.
- [2] Eldon Hansen and G. William Walster. *Global Optimization Using Interval Analysis: Revised And Expanded*. CRC Press, 2003.
- [3] E.R. Hansen and G.W. Walster. Bounds for Lagrange multipliers and optimal points. *Computers & Mathematics with Applications*, 25(10):59–69, 1993.

GPU-Parallel Branch-and-Bound with Custom Kernels and Specialized PDLP

Robert X. Gottlieb¹ and Matthew D. Stuber¹

¹*Chemical & Biomolecular Engineering, University of Connecticut, Storrs, CT, USA*

Abstract Two novel developments are detailed that are critical to hardware acceleration of deterministic global optimization: automatic relaxation subgradient calculation on a GPU for LP relaxations, and a native GPU-parallel LP solver. To calculate relaxation subgradients, we rely on a novel method to automatically generate problem- and expression-specific GPU-compatible functions, with preliminary results indicating a 10–100x speedup over a CPU implementation of McCormick-based rules. To address LPs, we developed a novel GPU-accelerated variant of the PDLP method. Preliminary results show increased performance of up to 100x relative to commercial solvers, specifically in the case of solving large numbers of small-sized LP instances. Subsequently, we combine these two developments to form a GPU-accelerated global optimizer. The performance of these developments are demonstrated on a range of problems, across different hardware capabilities, and against existing commercial solvers.

Keywords: SIMD Parallelization, Global Optimization, Branch-and-Bound

1. Introduction

Modern tools for deterministic global optimization (DGO) use a variety of advanced techniques to solve problems that were once considered intractable. Such techniques include reformulations to more easily solvable problem forms [1], specialized domain reduction techniques [2], and the development of improved relaxations or envelopes of critical functional forms [3, 4], among others. However, these techniques alone are not suffi-

cient to render all DGO problems tractable, and for many problems of practical interest, obtaining a solution still requires processing thousands or millions of branch-and-bound (B&B) nodes. For these problems, processing power and calculation throughput are critical factors that determine how quickly a global solution can be found. In other fields that faced a similar computational bottleneck, such as in machine learning model training [5], the dominant strategy has become the use of parallel processing hardware such as graphics processing units (GPUs). While new tools and techniques continue to make many DGO problems solvable, a move toward parallelization using GPU hardware would serve to increase the solution speed for a wide variety of the most challenging problems via faster calculation throughput.

To the best of our knowledge, all existing DGO solvers—besides our solver [6]—are designed to run exclusively on CPU hardware. For a GPU-based global solver, one of the challenges is aligning the B&B algorithm with the strengths of the hardware. Particularly, GPU-based local solvers typically become more effective on large-scale instances, such as the NLP solver MadNLP.jl [8], which realizes a speedup over a CPU implementation on problem instances with more than 20,000 variables. DGO problems are typically several orders-of-magnitude smaller than this, which implies that individual bounding subproblems are unlikely to benefit from GPU acceleration. Instead, in this work, we focus on simultaneously processing large numbers of B&B nodes, aligning the size of the B&B node stack with the parallelization capacity of GPUs.

The state-of-the-art approach for obtaining B&B node lower bounds is to construct linear programs (LPs) from subtangent hyperplanes of relaxations of the objective and constraints. Consequently, and as discussed in [7], a GPU-based DGO solver would need two key features to be competitive with existing solvers: a method of calculating relaxation subgradients, and a method of solving LPs. In this paper, we detail our development of GPU-parallel implementations of these two features, each with the goal of addressing many B&B nodes simultaneously.

2. Implementations and Results

GPU-Parallel Relaxation Subgradients. In this work, we utilize a source-code generation approach to compute McCormick-based relaxations of functions on a GPU—conceptually similar to our previous work [6]—in an entirely novel way. This approach contrasts the operator overloading-like scheme used by McCormick.jl [9]. Given an input expression, such as a symbolically defined objective function or constraint, we begin by per-

forming a factorization of the expression to generate a primal trace. We then apply transformations as desired, such as automatically identifying mathematical forms for which specialized relaxation rules exist and substituting those expressions. From this modified primal trace, we are prepared to construct numerical functions for evaluation.

Finally, the primal trace is parsed and a GPU-compatible function (a “kernel”) is automatically generated that sequentially applies generalized McCormick rules to its inputs (variables and their domains) to calculate relaxations (and their subgradients) of the original expression on the given variable domains. The generation of such kernels, valid on any subdomain, is possible because the objective and constraints of a problem are identical across B&B nodes. This also allows us to exploit problem sparsity, both here and in our LP solver, to speed up calculations. These custom, problem-specific kernels apply GPU parallelization to return relaxations for many B&B nodes simultaneously, each of which necessarily has a unique domain. From preliminary testing, this approach outperforms McCormick.jl [9] over a range of expression complexities by roughly 10–100x.

GPU-Parallel LP Solver. Standard approaches for solving LPs are mostly variants of the simplex and interior point methods, although in recent years, first-order methods have become more popular due to their potential for easier GPU acceleration. A notable recent development is the PDLP algorithm [10], and its open-source GPU implementation cuPDLP.jl [11]. In this work, we develop a cuPDLP.jl-based solver that scales with the number of LPs solved as opposed to instance size, with the primary focus being the smaller LPs encountered in B&B lower-bounding procedures. Additionally, since the goal is to incorporate this solver into a B&B scheme, and we generate relaxation subgradients on GPUs, we eliminate the expensive step of transferring LP instances between CPU and GPU memory. Effectively, the global optimizer would handle node organization, domain reduction, and the upper-bounding problems using the CPU, and would offload the entirety of the lower-bounding work, which is generally the most time-intensive step, to the GPU.

Similar to the novel relaxation approach presented in this work, the performance of this version of PDLP scales with the number of LPs being solved. Preliminary benchmark results shown in Table 1 indicate speed improvements of up to 100x relative to commercial LP solvers on small problems. We attribute this to the lack of CPU-GPU memory transfer, the specialized use case of handling multiple LPs, and the focus on comparatively small LP instances.

Table 1: Average solve times over 20480 LP instances generated from affine relaxations of partitioned examples from the MINLPLib database. Parallelized PDLP solves LPs in parallel in a GV100 GPU, and GLPK solves LPs in serial on an Intel Xeon W-2195 CPU. This version of GPU-accelerated PDLP is most dominant on extremely small LPs and at weaker tolerance.

Example	Variables	Constraints	Avg. Solution Time (μ s)			
			Parallelized PDLP		GLPK	
			1E-8 Tol.	1E-4 Tol.	1E-8 Tol.	1E-4 Tol.
ex4_1_1	1	0	1.186	0.665	75.238	73.944
rbrock	2	0	0.661	0.323	78.462	75.092
ex3_1_1	8	6	5.676	5.538	104.361	103.909
chem	11	4	23.655	13.132	137.179	143.991
ramsey	33	22	18.066	18.102	204.618	213.732
lakes	90	78	749.429	749.268	1144.600	1155.039

3. Conclusion

In this work, we discuss our latest GPU-accelerated relaxation subgradient technique and describe the implementation of a GPU-parallelized LP solver based on PDLP. We aim to demonstrate the strengths of these approaches across numerical experiments and a range of hardware capabilities. We also aim to demonstrate how these approaches can be paired together into a GPU-accelerated DGO solver, and compare its performance on benchmark test sets against commercial solvers.

References

- [1] Amarger, R.J. et al. (1992). An automated modelling and reformulation system for design optimization. *Comput. Chem. Eng.* 16(7): 623–636.
- [2] Zhang, Y. et al. (2020). Optimality-based domain reduction for inequality-constrained NLP and MINLP problems. *J. Global Optim.* 77(3): 425–454.
- [3] Wilhelm, M.E. and Stuber, M.D. (2023). Improved convex and concave relaxations of composite bilinear forms. *J. Optim. Theory and Appl.* 197: 174–204.
- [4] Wilhelm, M.E. et al. (2022). Convex and concave envelopes of artificial neural network activation functions for deterministic global optimization. *J. Global Optim.* 85: 569–594.
- [5] Abadi, M. et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. doi: 10.48550/ARXIV.1603.04467

- [6] Gottlieb, R.X. et al. (2024). Automatic source code generation for deterministic global optimization with parallel architectures. *Optim. Methods Software*. 1–39.
- [7] Gottlieb, R.X. and Stuber, M.D. (2024). GPU-accelerated deterministic global optimization. In: ISMP 2024, Montréal, Canada.
- [8] Shin, S. et al. (2024). Accelerating optimal power flow with GPUs: SIMD abstraction of nonlinear programs and condensed-space interior-point methods. *Electr. Power Syst. Res.* 236: 110651.
- [9] Wilhelm, M.E. et al. (2020). PSORLab/McCormick.jl. URL: <https://github.com/PSORLab/McCormick.jl>
- [10] Applegate, D., et al. (2021). Practical large-scale linear programming using primal-dual hybrid gradient. doi: 10.48550/ARXIV.2106.04756.
- [11] Lu, H. and Yang, J. (2024). cuPDLPjl: A GPU implementation of restarted primal-dual hybrid gradient for linear programming in Julia. doi: 10.48550/ARXIV.2311.12180.

Integrating Global and Robust Optimisation for Non-Convex Quadratic Programs

Asimina Marousi and Vassilis M. Charitopoulos *

Department of Chemical Engineering, The Sargent Centre for Process Systems Engineering, University College London, London, UK

* v.charitopoulos@ucl.ac.uk

Abstract In this work, we introduce a novel spatial Branch-and-Bound algorithm integrated with robust cutting-planes for solving non-convex robust optimisation problems. The algorithm systematically incorporates global and robust optimisation techniques, leveraging a McCormick relaxation. The proposed algorithm is evaluated on benchmark pooling problems with uncertain feed quality, demonstrating algorithm stability and solution robustness.

Keywords: Robust optimisation, Global optimisation, Spatial Branch and Bound, Pooling problems, Cutting planes

1. Introduction

The main challenge of robust optimisation lies in reformulating the uncertain semi-infinite problem into a deterministic and tractable form. Convex analysis methods such as robust reformulations and robust cutting set methods can effectively address convex robust optimisation problems under convex uncertainty sets. However, extending these techniques to non-convex problems remains largely unexplored. To bypass the need for convexity assumption for the upper-level problem, a global optimisation method was proposed by iteratively relaxing the non-convex constraints while solely relying on continuity assumptions [1]. The non-convex pooling problem under uncertainty was examined both via dual reformulation

and robust cutting sets, while adaptively adjusting the tolerance of the selected global solver[2]. A generalised cutting set method was proposed in [3] for robust nonlinear process systems engineering problems. General robust cutting plane frameworks rely on generating uncertainty cuts for feasibility violations while the first-level optimisation problem is handled by a suitable solver. Recently, an interval Branch-and-Bound algorithm was developed to address benchmark unconstrained non-convex optimisation problems of low dimensionality under robust uncertainty [4].

Contributions of this work: We develop a novel spatial-Branch-and-Bound algorithm for non-convex continuous robust optimisation problems, aiming to explore the computational benefits of a concurrent optimality and robustness search.

2. Robust-spatial-Branch-and-Bound for the pooling problems

The key idea of the Robust spatial Branch-and-Bound (RsBB) algorithm is that while exploring the branch-and-bound tree, we assess the robustness of nodes entailing the best-found solutions. Hence, the main difference with existing robust cutting plane algorithms lies in the fact that the robustness search in the second-level problem is augmenting an optimality search in the first-level problem. In the proposed algorithm, we aim to integrate the robust cutting planes within the spatial Branch-and-Bound (sBB) algorithm. Relying on a local NLP solver, the proposed algorithm is evaluating the robustness of the solutions in a branching tree. The outline of the proposed algorithm is presented in Fig.1. The performance of the proposed approach is assessed via benchmark QCQP pooling problems under feed quality uncertainty for box, ellipsoidal and polyhedral uncertainty sets of varying size.

Let Π denote the original non-convex pooling problem entailing bi-linear terms both in the objective function and in a number of constraints. The corresponding convex relaxation of Π_R is derived using McCormick envelopes . For the initial variable bounds at the root node, problem (Π) is solved using a local solver. If the computed solution ϕ_n^Π at a node n is as good as the best-found so far ϕ_{bf} , an infeasibility test is performed to evaluate the robustness of the obtained solution. The infeasibility test entails solving linear problems π that aim to maximise the uncertain constraint violation. If such a violation occurs, then the corresponding robust cuts are added both to Π and Π_R . The cutting plane algorithm continues at the root node until no more robustness violations are detected. Until this step,

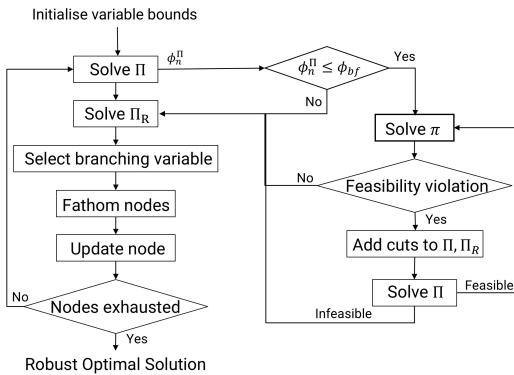


Figure 1: Schematic representation of the Robust spatial Branch-and-Bound algorithm

the proposed algorithm follows the typical robust cutting plane algorithm. The following steps of the algorithm correspond the Branch-and-Bound algorithm. Nodes for which the Π_R solution is greater than the best-found solution of Π are fathomed. The next node is updated as the waiting node with the lowest Π_R solution. If the waiting nodes are exhausted, the robust optimal solution of the problem is obtained. The detailed description of the algorithm as well as the assumptions used can be found in [5].

Focusing on foulds2 problem [6] and the box uncertainty set, Fig. 2 shows that the number of tree nodes explored by the RsBB algorithm decreases with the uncertainty set size increase (Ψ), so does the optimality gap at the root node. To elucidate this observation, in Fig. 3 we examine the number for robust cut rounds of the same problem at the root node. It can be observed that that for $\Psi = 0.15$, 11 robust cut iterations lead to the robust optimal solution, while $\Psi = 0.1$ results in fewer cuts and a higher gap. Overall, increasing the conservativism of the robust problem improves the RsBB performance since the feasible region is restrained by robust cuts.

3. Summary

The proposed RsBB algorithm combines the principles of sBB and robust cutting planes to solve continuous non-convex problems under convex uncertainty sets. The proposed approach is applied to pooling problems utilising a McCormick envelop to obtain convex lower bounds. We observed

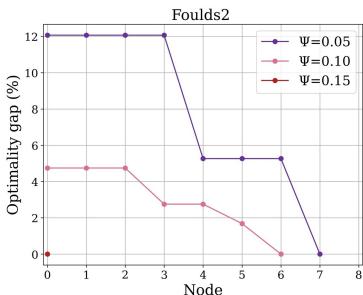


Figure 2: Optimality gap closure for number of nodes explored in foulds2 problem under different sizes of box uncertainty sets.

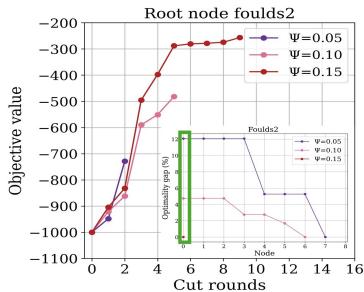


Figure 3: Objective value for number of robust cut iterations at the root node of foulds2 problem.

that using robust cutting planes in the sBB algorithm serves as an effective feasibility bounds tightening method, while facilitating the optimality search.

Acknowledgments

Financial support under the EPSRC grants: EP/W003317/1, EP/V034723/1 and EP/V051008/1 is gratefully acknowledged by the authors.

References

- [1] Mitsos, A., Tsoukalas, A. (2015). Global optimization of generalized semi-infinite programs via restriction of the right hand side. *J. Glob. Opt.* 61, 1–17.
- [2] Wiebe, J., Cecílio, I., & Misener, R. (2019). Robust Optimization for the Pooling Problem. *Ind. Eng. Chem. Res.*, 58(28), 12712–12722.
- [3] Isenberg, N. M., Akula, P., Eslick, J. C., Bhattacharyya, D., Miller, D. C., & Gounaris, C. E. (2021). A generalized cutting-set approach for nonlinear robust optimization in process systems engineering. *AICHE J.*, 67(5), e17175.
- [4] Carrizosa, E., Messine, F. (2021). An interval branch and bound method for global robust optimization. *J. Glob. Opt.* 80, 507–522.
- [5] Marousi A., Charitopoulos V. M. (2025). Global and Robust Optimisation for Non-Convex Quadratic Programs. arXiv preprint. arXiv:2503.07310.
- [6] Foulds, L.R., Haugland, D., Jörnsten, K. (1992). A bilinear approach to the pooling problem. *Optimization* 24, 165–180.

Near-optimal global solutions for sparse approximation problems

Sonia Cafieri,¹ Marcel Mongeau,¹ Sébastien Bourguignon,² Gwenaël Samain³

¹*ENAC, Université de Toulouse, sonia.cafieri@enac.fr, marcel.mongeau@enac.fr*

²*Nantes Université, École Centrale Nantes, CNRS, LS2N,
sebastien.bourguignon@ec-nantes.fr*

³*independent researcher, samain.gwen@laposte.net*

Abstract We address sparse approximation problems. We propose an original continuous-optimization formulation and build on such a formulation to develop a method to obtain near-optimal global solutions.

Keywords: Continuous reformulation, Sparse approximation, Near-optimal global solutions, Branch-and-bound

1. Introduction

Sparse approximation is of great interest to the research community, in particular in signal processing and statistics [1]. Given a linear model $y \simeq Ax$, with $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$, the sparse approximation problem consists in finding a sparse solution vector x that has a small number of non-zero components (that is, a small ℓ_0 “norm”) such that Ax approximates a data vector y . The problem can be formulated as the minimization of the quadratic approximation error penalized by the number of non-zero components of x :

$$\min_x \frac{1}{2} \|y - Ax\|_2^2 + \mu \|x\|_0, \quad (1)$$

where $\mu > 0$ is a penalty parameter. This problem is NP-hard, and is often addressed in signal processing applications either by local techniques, or by the ℓ_1 -norm relaxation of the ℓ_0 term [1]. It has been shown in [4] that the problem can be reformulated as a mixed-integer problem, thereby global solutions can be computed using branch-and-bound methods. This allows the computation of global solutions for instances of moderate size.

We propose an original reformulation of the above mixed-integer problem, that yields a nonconvex continuous optimization problem. Such a problem is used to build a solution method giving near-optimal global solutions.

2. Continuous-optimization reformulation

Let us consider the logical disjunctive constraint $f_1(x) \leq 0 \vee f_2(x) \geq 0$, for some real-valued functions f_1 and f_2 of $x \in \mathbb{R}^n$. A continuous optimization model not relying on adding extra binary variables, as usually done in the literature, has been introduced in [5] for such a constraint. It is based on a continuous piecewise-quadratic penalty function, and referred to as the continuous quadratic penalty (CQP) formulation. For an optimization problem whose only combinatorial aspect is represented by the above logical condition, the CQP formulation allows one to obtain a fully continuous formulation of the optimization problem (see [6] for an example of successful application).

For problem (1), a mixed-integer formulation was obtained by [4] introducing a binary decision-variable vector z , that allows one to count the non-zero components of x (through the logical constraint $z_i = 0 \implies x_i = 0$, $i = 1, \dots, n$, that also reads $z_i \neq 0 \vee x_i = 0$, $i = 1, \dots, n$, under the disjunctive form):

$$\min_{x,z} \quad \frac{1}{2} \|y - Ax\|_2^2 + \mu \sum_i^n z_i \quad (2)$$

$$\text{s.t.} \quad z_i \neq 0 \vee x_i = 0, \quad i = 1, \dots, n \quad (3)$$

$$z_i \in \{0, 1\}, \quad i = 1, \dots, n. \quad (4)$$

Let us consider a continuous variable \tilde{z} such that $0 \leq \tilde{z} \leq 1$, and the disjunctive constraint (3) rewritten as $\tilde{z}_i \geq 1 \vee x_i = 0$, $i = 1, \dots, n$. In this paper, we apply the CQP formulation to the latter disjunctive constraint expressed as follows, so as to fit into the form addressed in [5]:

$$(1 - \tilde{z}_i \leq 0 \vee -x_i \geq 0) \quad \text{and} \quad (1 - \tilde{z}_i \leq 0 \vee x_i \geq 0), \quad i = 1, \dots, n. \quad (5)$$

The CQP formulation yields the introduction, for each $i = 1, \dots, n$, of two piecewise-quadratic functions, one for each disjunction of (5), that are added as penalties in the objective function. The form of these piecewise-quadratic functions is presented in [5]. The resulting overall problem, whose decision variables are x and \tilde{z} , is a fully (smooth) continuous nonconvex optimization problem.

3. Near-optimal global solutions

The continuous optimization problem obtained through the CQP formulation can be solved using state-of-the-art optimization solvers. We use IPOPT [7], to obtain locally optimal solutions or a proof of infeasibility.

We then build on the solution of such a problem to obtain a two-phase algorithm. The first phase is the local solution of the continuous CQP formulation, and the second phase addresses a mixed-integer problem. Let Z be the set $Z = \{i : \tilde{x}_i = 0\}$, where \tilde{x} is the vector of locally-optimal continuous values of x . The second phase of computation is a branch-and-bound method applied to the mixed-integer formulation obtained from (2)-(3)-(4), imposing moreover $z_i = 0, i \in Z$. This implies $x_i = 0, i \in Z$. This may lead to move to another solution with additional zero-valued components of x .

Preliminary numerical tests, on instances generated as in [3], show that near-optimal or even global optimal solutions can be computed with the proposed procedure, although the global optimality is not guaranteed. The computing time is significantly reduced compared to the branch-and-bound solution of the mixed-integer formulation (2)-(3)-(4), where the logical constraint (3) is addressed through the usual big-M formulation.

As an illustration, for a randomly generated synthetic instance of a variable selection problem in statistics with $n = 30$, $m = 20$, we obtain the computing times presented in Table 1, where *baseline* refers to the original mixed-integer problem (2)-(3)-(4) solved by COUENNE [2].

Table 1: Comparison of computing times (seconds) for solving a variable selection problem.

<i>Dimension</i>	<i>baseline</i>	<i>1st phase</i>	<i>2nd phase</i>
$n = 30, m = 20$	7.1	0.31	2.8

4. Summary

We propose an original continuous optimization formulation for a sparse approximation problem, and derive a branch-and-bound-based two-phase method to obtain efficiently near-optimal global solutions. The approach will be furtherly developed to obtain an efficient method for guaranteed global optimal solutions.

References

- [1] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, January 2012.
- [2] Pietro Belotti, Jon Lee, Leo Liberti, Francois Margot, and Andreas Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods & Software*, 24(4-5):597–634, 2009.
- [3] Ramzi Ben Mhenni, Sébastien Bourguignon, and Jordan Ninin. Global optimization for sparse solution of least squares problems. *Optimization Methods and Software*, 37(5):1740–1769, 2022.
- [4] Sébastien Bourguignon, Jordan Ninin, Hervé Carfantan, and Marcel Mongeau. Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance. *IEEE Transactions on Signal Processing*, 64(6):1405–1419, 2016.
- [5] Sonia Cafieri, Andrew R. Conn, and Marcel Mongeau. The continuous quadrant penalty formulation of logical constraints. *Open Journal of Mathematical Optimization*, 4:1–12, 2023.
- [6] Sonia Cafieri, Andrew R. Conn, and Marcel Mongeau. Mixed-integer nonlinear and continuous optimization formulations for aircraft conflict avoidance via heading and speed deviations. *European Journal of Operational Research*, 310(2):670–679, 2023.
- [7] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

On the Convex Hull of the Graph of a Simple Monomial

Jon Lee,¹ Daphne Skipper,² Emily Speakman³

¹*University of Michigan, Ann Arbor, MI, USA* jonxlee@umich.edu

²*United States Naval Academy, Annapolis, MD, USA* skipper@usna.edu

³*University of Colorado Denver, Denver, CO, USA* emily.speakman@ucdenver.edu

Abstract Algorithms for solving difficult mixed-integer non-linear optimization problems rely on obtaining good convex relaxations of substructures in the formulation. Given that there are multiple ways in which one can obtain a valid convex outer approximation, it is natural to consider how alternatives may be reasonably compared. By definition, the convex hull provides the tightest convex outer approximation. Thus, by computing its $(n + 1)$ -dimensional volume, we obtain a measure that can be used to evaluate the tightness of other approximations by comparison. In this work, we give a formula for the volume of the graph of a monomial on a non-negative box domain, where at most one of the lower bounds defining the box is positive.

Keywords: Polytope, Volume, Global Optimization, Mixed-integer Non-linear Optimization, Monomial, Spatial Branch-and-Bound

1. Introduction

Global optimization of a general factorable mixed-integer non-linear problem is typically achieved using the spatial branch-and-bound (sBB) framework. This framework starts by decomposing the functions of the given formulation into low-depth compositions of library functions, namely affine functions and functions of just a few variables (usually ≤ 3). Following this, bound propagation and domain partitioning is used to systematically

exclude unpromising parts of the feasible region. It is critical that we have tight convex relaxations of the graphs of the library functions because the practical efficiency of sBB fundamentally depends on the strength of the relaxations used.

To quantify the tightness of different convex relaxations in n -dimensional space, volume is a natural metric. Originally proposed in [1], the approach has been applied in evaluating and comparing polyhedral relaxations; see [2] and the many references therein.

Multilinear functions are important building blocks in many non-convex optimization problems, and as such, have motivated a substantial body of research aimed at deriving high-quality convex relaxations. There has been some previous work on volume computations for low-dimensional multilinear terms; see [2],[5],[6],[7],[8]. In what follows, we establish a volume formula for general dimension in a special case.

1.1 Definitions and Notation

We assume some familiarity with polytopes. Briefly, $\text{conv}(\cdot)$ denotes convex hull, and $\text{vol}_d(B)$ denotes the d -dimensional volume (i.e., Lebesgue measure) of a convex body $B \subset \mathbb{R}^d$. For $n \geq 2$ and $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, with $0 \leq a_i < b_i, i = 1, 2, \dots, n$, let

$$C_n(\mathbf{a}, \mathbf{b}) := \text{conv} \left(\left\{ \begin{pmatrix} \mathbf{x} \\ y \end{pmatrix} \in \mathbb{R}^{n+1} : \mathbf{a} \leq \mathbf{x} \leq \mathbf{b}, y := \prod_{i=1}^n x_i \right\} \right).$$

It is well known that this fundamental object, the convex hull of the graph of $\prod_{i=1}^n x_i$ on the box domain $[\mathbf{a}, \mathbf{b}]$, is a polytope having 2^n extreme points, each of which has the form $(\mathbf{x}^\top, y)^\top$, where each of the n components of \mathbf{x} is set to either its lower or upper bound, and $y := x_1 x_2 \dots x_n$; see [4].

For conciseness, we let C_n^k denote $C_n(\mathbf{a}, \mathbf{b})$ in the case of $a_i := 0$, for $i = 1, 2, \dots, n-k$. That is, we allow at most k variables (arbitrarily x_{n-k}, \dots, x_n) to have a non-zero lower bound. In general, C_n^k and various relaxations of it, are very relevant in the study of sBB algorithms for global optimization.

1.2 Background

C_2^2 is just a tetrahedron — with volume $\frac{1}{2}(b_1 - a_1)^2(b_2 - a_2)^2$. C_3^3 has already been the subject of intense investigation. In particular, [7] established the following volume formula using triangulation, with the result subsequently re-proved in [6] using a completely different method.

Theorem 1 ([7, Theorem 4.1]). *For $n = k = 3$, assuming (without loss of generality) that $a_1/b_1 = \min\{a_i/b_i : 1 \leq i \leq 3\}$, we have*

$$\begin{aligned} \text{vol}_4(C_3^3) &= (b_1 - a_1)(b_2 - a_2)(b_3 - a_3) \times \\ &(b_1(5b_2b_3 - a_2b_3 - b_2a_3 - 3a_2a_3) + a_1(5a_2a_3 - b_2a_3 - a_2b_3 - 3b_2b_3)) / 24. \end{aligned}$$

[7] also presented volume formulae associated with the natural, simpler, and often-used (“double McCormick”) relaxations. In doing so, they could evaluate which such relaxations are the tightest, and when such relaxations are adequate, compared to the true convex hull. Moreover, they were able to characterize how the results depend on the specific values of a and b .

Moving away from work in low dimensions, the following result handles the general case of all-zero lower bounds. We note that the proof is considerably simpler than that of Theorem 1.

Theorem 2 ([2, Theorem 23]). *For $k = 0$ and arbitrary $n \geq 2$, and with all $a_i = 0$, we have*

$$\text{vol}_{n+1}(C_n^0) = \frac{n!-1}{(n+1)!} \prod_{i=1}^n b_i^2.$$

An elusive goal of ours is to obtain a general formula for $\text{vol}_{n+1}(C_n^k)$, for all $k = n \geq 4$.

2. Result

Next, we present a new volume formula, for general n and $k = 1$. This directly generalizes Theorem 2, however, its proof is significantly more complicated. We provide a proof sketch, omitting many technical details.

Theorem 3. *For all $n \geq 2$,*

$$\text{vol}_{n+1}(C_n^1) = \frac{(b_n - a_n) \prod_{i=1}^{n-1} b_i^2}{(n+1)!} \left((n! - 1)b_n + ((n-1)! - n)a_n \right).$$

Note that it is easy to check that setting $a_n = 0$ in Theorem 3, we obtain Theorem 2. Further, it is easy to check that plugging $n = 3$ into Theorem 3 gives the same formula as plugging $a_1 = a_2 = 0$ into Theorem 1.

Proof (sketch): Let E denote the set of extreme points of C_n^1 . Recall that these are the 2^n points in \mathbb{R}^{n+1} such that the n components of x are set to either their upper or lower bound, with y equal to the product of the x -components. Moreover, in the case of C_n^1 , all but one x -variable has a lower bound of zero. Therefore, in all but two of the extreme points, we have $y = 0$. We exploit this structure to prove our volume result.

Now, define $v_1 := (b_1, \dots, b_{n-1}, a_n, b_1 \dots b_{n-1} a_n)^\top$ and $v_2 := (b_1, \dots, b_{n-1}, b_n, b_1 \dots b_{n-1} b_n)^\top$ to be the two extreme points of C_n^1 that do not lie on the hyperplane $y = 0$, and consider the pyramid $P_{n+1} := \text{conv}(E \setminus \{v_2\})$. P_{n+1} has its base in the hyperplane $y = 0$, and has the point v_1 as its apex, its volume is calculated to be $\frac{(n-1)!-1}{(n+1)(n-1)!} \cdot a_n(b_n - a_n) \cdot \prod_{i=1}^{n-1} b_i^2$.

Clearly, $C_n^1 = \text{conv}(P_{n+1} \cup \{v_2\})$, and therefore, to find its volume, we can sum the volume of P_{n+1} with any additional volume created by adding in the extreme point v_2 . To do this, we establish which facets of P_{n+1} are violated by v_2 , and compute the volume of the convex hull of v_2 with each facet that separates v_2 from P_{n+1} .

After computing the facets of P_{n+1} , we observe that v_2 is separated from P_{n+1} by exactly n of the facets. Moreover, the additional volume generated from $n-1$ of these n facets is the same, and given by the expression $(b_n - a_n)^2 \cdot \prod_{j=1}^{n-1} b_j^2 / n(n+1)$. Finally, the remaining violated facet generates an additional volume of $\frac{(n-1)!-1}{n!(n+1)} \cdot b_n(b_n - a_n) \cdot \prod_{j=1}^{n-1} b_j^2$. This is enough to obtain our result. \square

3. Outlook

As we have mentioned, we would like to obtain a general formula for $\text{vol}_{n+1}(C_n^k)$, for all $k = n \geq 4$. A seemingly more manageable goal is the case of $k = 2$ and arbitrary $n \geq 4$, or even $k = 3$, which would then generalize Theorem 1.

Another question that we are pursuing is obtaining facet descriptions of the C_n^k . For $n = k = 3$, this is known, see [3], and was instrumental in getting to the associated volume formula in [7]. Finally, as in [7], we would like to compare any derived volume formulae with formulae for the volumes of natural and simpler relaxations. Such comparisons have the potential to yield both theoretical insight and practical guidance for modelers and algorithm developers.

Acknowledgments

This work was begun at the Workshop on Circuit Diameters and Augmentation: Recent Advances in Linear and Integer Optimization; sponsored under NSF grant 2006183 (PI: S. Borgwardt). J. Lee was supported in part by U.S.A. Office of Naval Research grant N00014-24-1-2694.

References

- [1] Jon Lee and Walter D. Morris, Jr. Geometric comparison of combinatorial polytopes. *Disc. Appl. Math.*, 55:163–182, 1994. [https://doi.org/10.1016/0166-218X\(94\)90006-X](https://doi.org/10.1016/0166-218X(94)90006-X).
- [2] Jon Lee, Daphne Skipper, and Emily Speakman. Algorithmic and modeling insights via volumetric comparison of polyhedral relaxations. *Math. Prog.*, 170:121–140, 2018. <https://doi.org/10.1007/s10107-018-1272-6>.
- [3] Clifford Meyer and Christodoulos Floudas. Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes. *J. of Global Opt.*, 29:125–155, 2004. <https://doi.org/10.1023/B:JOGO.0000042112.72379.e6>.
- [4] Anatoliy D. Rikun. A convex envelope formula for multilinear functions. *J. of Global Opt.*, 10:425–437, 1997. <https://doi.org/10.1023/A:1008217604285>.
- [5] Emily Speakman. *Volumetric Guidance for Handling Triple Products in Spatial Branch-and-Bound*. Ph.D. Dissertation, University of Michigan, April 2017.
- [6] Emily Speakman and Gennadiy Averkov. Computing the volume of the convex hull of the graph of a trilinear monomial using mixed volumes. *Disc. Appl. Math.*, 308:36–45, 2022. <https://doi.org/10.1016/j.dam.2019.09.007>.
- [7] Emily Speakman and Jon Lee. Quantifying double McCormick. *Math. of Oper. Res.*, 42(4):1230–1253, 2017. <https://doi.org/10.1287/moor.2017.0846>.
- [8] Emily Speakman, Han Yu, and Jon Lee. Experimental validation of volume-based comparison for double-McCormick relaxations. In D. Salvagnin and M. Lombardi, editors, *CPAIOR 2017*, pages 229–243. Springer, 2017. https://doi.org/10.1007/978-3-319-59776-8_19.

Plenary Talk
Martin Schmidt





The Burial of Coupling Constraints in Linear Bilevel Optimization

Martin Schmidt,

Trier University, Department of Mathematics, Trier, Germany

martin.schmidt@uni-trier.de

Abstract It has been common sense in (linear) bilevel optimization that problems with coupling constraints are more difficult to tackle than those without such constraints. While the modeling capabilities in terms of the feasible sets are indeed richer because coupling constraints allow to model disconnected feasible sets, complexity theory did not see any difference between problems with or without coupling constraints. In this talk, we show that there is no difference at all when one considers optimal solutions instead of just feasible points. For the case of optimistic bilevel problems, we proved this in 2024. The same result for pessimistic bilevel is from this year. In total, our results show that - on the level of optimal solutions - there is no difference between optimistic as well as pessimistic bilevel optimization with or without coupling constraints. As a "corollary" and rather surprisingly, all theoretical results and algorithms for pessimistic bilevel optimization without coupling constraints also apply to pessimistic bilevel optimization with coupling constraints.

Session 8





Exact Verification of Data Poisoning Attacks via Mixed-Integer Programming

Philip Sosnin¹ & Calvin Tsay²

¹*Department of Computing, Imperial College London, UK* p.sosnin23@imperial.ac.uk

²*Department of Computing, Imperial College London, UK* c.tsay@imperial.ac.uk

Abstract This work introduces the first sound and complete verification method for data poisoning attacks on neural networks. We model adversarial manipulations, model training, and test-time deployment as a single mixed-integer program, allowing us to compute provably optimal poisoning attacks and derive exact robustness guarantees against them. Our approach exactly models various attack scenarios and solves to optimality via mixed-integer quadratic programming, establishing precise guarantees on the effectiveness of any poisoning attack.

Keywords: Mixed-Integer Programming, Neural Network Verification.

Extended Abstract

Data poisoning remains a significant security risk for machine learning applications in safety-critical domains [7]. While modern datasets have grown prohibitively large for comprehensive quality checks, recent research demonstrates that corrupting even a small fraction (<1%) of training data can introduce serious vulnerabilities, including degraded performance or hidden backdoor patterns [9].

Unlike inference-time attacks that affect single model evaluations, data poisoning attacks contaminate the training process itself, potentially impacting all downstream model users. These attacks typically fall into three categories: untargeted attacks that disrupt training convergence, targeted

attacks that degrade performance for specific inputs, and backdoor attacks that implant hidden trigger patterns causing abnormal behaviour.

Previous works have developed a range of defences and certification methods to guarantee robustness against data poisoning. Early work focused on statistical outlier detection [6], offering only limited guarantees. Later approaches, such as randomized smoothing for linear and kernel models [4] and data partitioning strategies [1, 3, 8], provide stronger guarantees for limited training settings. Recent certification methods like Abstract Gradient Training [5] and FullCert [2] use relaxations of training dynamics to bound poisoning effects, but suffer from loose guarantees due to over-approximation. Overall, existing methods remain fundamentally incomplete, often yielding vacuous bounds under practical training and attack scenarios.

An Exact Formulation of Data-Poisoning Attacks

Our work presents the first sound and complete method for verification of poisoning attacks on neural networks trained via gradient-based optimization. We provide an exact representation of adversary capabilities, the entire training pipeline, and test-time deployment, allowing us to exactly compute the solution to the following constrained optimization problem:

$$\begin{aligned} \min_{\tilde{\mathcal{D}}} \quad & \frac{1}{N} \sum_{i=1}^N \mathcal{J}[f(x_i, \theta), y_i] && \text{(Poisoning Attack Goal)} \\ \text{subject to} \quad & \tilde{\mathcal{D}} \in \mathcal{T}(\mathcal{D}) && \text{(Data Poisoning)} \\ & \theta = M(f, \tilde{\mathcal{D}}) && \text{(Model Training)} \end{aligned}$$

where f is a neural network with parameters θ trained by the gradient-based training algorithm M on the poisoned dataset $\tilde{\mathcal{D}}$. The poisoning adversary's capabilities are represented by the set \mathcal{T} . The attack objective is modelled by the objective function \mathcal{J} , which is evaluated over a test dataset. This adversarial objective is versatile and can encompass any adversarial model behaviour on the test set. A straightforward example is the test-set accuracy, which forms the goal of untargeted poisoning attacks. More complex attack goals, such as targetted misclassification of only certain test examples, can also be encoded via \mathcal{J} .

We define the poisoning adversary's capabilities as the set \mathcal{T} . Their objective is an attack goal \mathcal{J} , which is measured against a test dataset. This goal \mathcal{J} is versatile and can encompass any adversarial model behavior desired on that test set. A straightforward example is an untargeted poisoning at-

tack designed to lower the model’s accuracy. However, J can also encode more intricate goals, such as deliberately misclassifying only particular examples within the test set.

Our mixed-integer programming (MIP) formulation encodes the entire network training and evaluation of the model. For each training iteration, we encode the forward pass and backward pass of the model using binary, linear, and quadratic constraints. We capture the weight update step by modelling the optimizer’s behaviour (e.g., SGD with momentum) as additional constraints linking parameters across training iterations.

We develop formulations that model common poisoning adversary capabilities, including label-flipping, bounded, and unbounded attack scenarios. By introducing binary decision variables to represent poisoned samples and enforcing constraints on their permissible manipulations, we precisely capture the relationship between poisoning decisions and the final trained model.

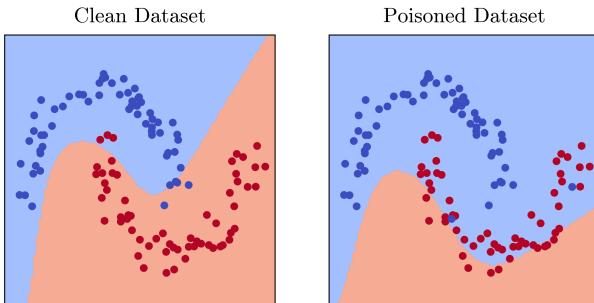


Figure 1: Optimal label flipping attack on a simple Half-moons dataset computed using our MIP formulation for a linear classifier model. With only 2 flipped labels, an optimal adversary can reduce the accuracy of the model from 96% to 54%.

Solving this MIP formulation to optimality yields both the worst-case poisoning attack and a certificate ensuring that no more effective attack exists within the given threat model. Our experiments on small scale benchmarks show that our method can verify robustness properties for adversary strengths that are orders of magnitude greater than those handled by existing approaches, indicating that current methods are overly conservative in their guarantees. Furthermore, we demonstrate that our provably optimal attacks significantly outperform existing heuristic poisoning attacks, revealing critical gaps in current defence and attack strategies.

This work delivers two significant contributions: (1) exact guarantees on poisoning robustness where previous methods provide only vacuous bounds, and (2) the first provably optimal poisoning attacks for given adversarial capabilities and objectives. While computational constraints currently limit application to small datasets, short training pipelines, or compact models, our approach establishes a foundation for developing tight relaxations and more efficient verification techniques. Additionally, our provably optimal attack strategies can enhance understanding of data poisoning vulnerabilities and inform more effective defences.

Acknowledgments

We would like to thank The Turing Institute for their support and resources during this research.

References

- [1] Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defense against general poisoning attacks. *arXiv preprint arXiv:2006.14768*, 2020.
- [2] Tobias Lorenz, Marta Kwiatkowska, and Mario Fritz. Fullcert: Deterministic end-to-end certification for training and inference of neural networks, 2024.
- [3] Keivan Rezaei, Kiarash Banihashem, Atoosa Chegini, and Soheil Feizi. Run-off election: Improved provable defense against data poisoning attacks. In *International Conference on Machine Learning*, pages 29030–29050. PMLR, 2023.
- [4] Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR, 2020.
- [5] Philip Sosnin, Mark N Müller, Maximilian Baader, Calvin Tsay, and Matthew Wicker. Certified robustness to data poisoning in gradient-based training. *arXiv preprint arXiv:2406.05670*, 2024.
- [6] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30, 2017.
- [7] Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys*, 55(8):1–35, 2022.

- [8] Wenzhao Wang, Alexander J Levine, and Soheil Feizi. Improved certified defenses against data poisoning with (deterministic) finite aggregation. In *International Conference on Machine Learning*, pages 22769–22783. PMLR, 2022.
- [9] Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International conference on machine learning*, pages 7614–7623. PMLR, 2019.

Solving MINLPs using Decomposition, Multiobjective Optimization and Machine Learning

Tamim Mehr¹, Ivo Nowak¹, Jan Kronqvist²

¹*Hamburg University of Applied Sciences, Berliner Tor 21, 20099 Hamburg, Germany,*
ivo.nowak@haw-hamburg.de

²*KTH Stockholm jankr@kth.se*

Abstract Motivated by column generation for solving huge optimization problems, we propose a new decomposition method for solving general MINLP problems by iteratively improving neural network approximations of a multiobjective MINLP reformulation using piecewise linear approximation. Numerical results with models of the MINLPlib will be presented.

Keywords: Mixed Integer Nonlinear Programming, Multiobjective Programming, Column Generation, Machine Learning

1. Introduction

Consider a sparse complex optimization problem

$$\min_{x \in \Omega} f(x), \quad (1)$$

where the objective function f and the constraint functions are sparse nonlinear or linear functions. Since (1) is sparse, it can be reformulated as a block-separable problem, [3]:

$$\min c^T x \quad \text{s. t. } Ax \leq b, \quad x_{B_k} \in X_k, \quad \forall k \in K. \quad (2)$$

where c is a linear objective function, $X_k \subset \mathbb{R}^{n_k}$ is defined by mixed-integer nonlinear local constraints, $A \in \mathbb{R}^{(m,n)}$, $b \in \mathbb{R}^m$ define the *coupling constraints* and $B_k \subset [n]$ is a *variable block*.

Motivated by inner approximation, i.e. column generation, algorithms for solving huge crew scheduling problems, we developed Decogo [5] for solving MINLP [3] and PDE-based engineering design problems [2], which can be formulated as (2).

2. Multiobjective Reformulation and Approximation using ML

The multiobjective (MO) view on (2) changes the focus from the complete feasible set to the relevant set of Pareto optimal points. A similar reformulation is presented in [1] for MIP and in [4] for MINLP. Consider the following MO sub-problem of block k , where we aim to minimize m_k resources simultaneously

$$\min U_k x_k \quad \text{s. t. } x_k \in X_k, \quad (3)$$

where $U_k \in \mathbb{R}^{(m_k, |B_k|)}$ is a coefficient matrix for block $k \in K$ defined as follows:

$$U_k := \begin{pmatrix} c_{B_k}^T \\ A_{j, B_k} \end{pmatrix}.$$

A feasible solution $x_k \in X_k$ with $w_k = U_k x_k$ dominates another solution $y_k \in X_k$ with $v_k = U_k y_k$ if $w_{ki} \leq v_{ki}$ for all $i \in [m_k]$ and $w_{ki} < v_{ki}$ for at least one index $i \in [m_k]$. A feasible solution $x_k \in X_k$ is efficient (or Pareto optimal) if there does not exist another solution that dominates it. An image of a Pareto optimal solution is also known as a nondominated point (NDP). The set of efficient points of problem (3), called efficient set, is defined by

$$\overline{X}_k := \{x \in X_k : U_k x \text{ is a NDP of (3)}\} \quad (4)$$

and the set of nondominated points of problem (3), called Pareto front, is defined by

$$\overline{W}_k := U_k \overline{X}_k. \quad (5)$$

Proposition 1. Let x^* be a solution of problem (2) and $w_k^* := U_k x_{B_k}^*$. Then w^* is a solution of the following problem

$$\min E_1 w \quad \text{s. t. } E_2 w \leq b, \quad w_k \in \overline{W}_k, \quad k \in K \quad (6)$$

with

$$E_1 w := \sum_{k \in K} w_{k1}, \quad E_2 w := \sum_{k \in K} \sum_{j \in [2:m_k]} w_{kj}.$$

Let $\underline{w}_k := \min\{U_k x : x \in X_k\}$ and $\bar{w}_k := \max\{U_k x : x \in X_k\}$ be the ideal and Nadir point, respectively, defining extreme Pareto points $\tilde{w}_{kj}^i := \bar{w}_{ki}$ for $j = i$ and $\tilde{w}_{kj}^i := \underline{w}_{ki}$ for $j \neq i$. Let $S_k := \text{conv}(\{\tilde{w}_k^i\}_{i \in [m_k]})$ and η_k be an orthogonal vector to S_k . The Pareto set (5) can be represented by

$$\bar{W}_k = \{s_k + \tau_k(s_k) \cdot \eta_k : s_k \in S_k\},$$

where $\tau_k(s_k)$ is the distance of $s_k \in S_k$ to \bar{W}_k in direction η_k . An approximation of \bar{W}_k is defined by

$$\bar{W}_{k,\theta_k} := \{s_k + \tau_{k,\theta_k}(s_k) \cdot \eta_k : s_k \in S_k\}, \quad (7)$$

where $\tau_{k,\theta_k}(s_k)$ is a neural network (NN) model of $\tau_k(s_k)$. The training points $(s_k, \tau_k(s_k))$ for the NN-models $\tau_{k,\theta_k}(s_k)$ are computed from inner points $\hat{X}_k \subset X_k$, which are generated by the new MINLP algorithm, described in the next section.

3. Solving the MINLP using fast CG and Piecewise Linear Approximation

We combine Column Generation (CG) and Piecewise Linear Approximation (PLA) for iteratively computing solution candidates of (2). In the beginning an initial PLA is computed by solving the following convex relaxation of (6):

$$\min E_1 w \quad \text{s. t. } E_2 w \leq b, \quad w_k \in \text{conv}(\bar{W}_k), \quad \forall k \in K \quad (8)$$

using **fast CG over the NN-models** \bar{W}_{k,θ_k} . In each iteration the PLA is refined by partitioning the image space by patches D_{kp} , as shown in Figure 1 and solving the following nonconvex relaxation:

$$\min E_1 w \quad \text{s. t. } E_2 w \leq b, \quad w_k \in \bigcup_{p \in P_k} \text{conv}(\bar{W}_k \cap D_{kp}), \quad k \in K. \quad (9)$$

A new MINLP algorithm based on these ideas is implemented in Decogo [5]. Numerical results with instances of the MINLPLib will be reported.

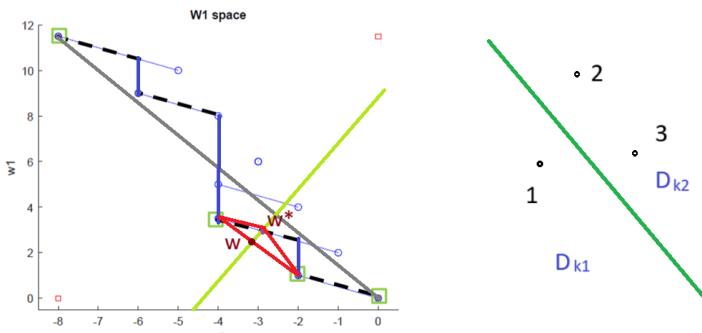


Figure 1: Left: The PLA of the Pareto set is refined by removing the red triangle in image spaces W_1 . The grey line shows S_k and the green line illustrates the coupling constraints $E_2 w = b$.

Right: Subdivision into two pieces D_{k1} and D_{k2} separating the best approximation point \hat{v}_k^1 from the other points \hat{v}_k^2 and \hat{v}_k^3

References

- [1] M. Bodur and S. Ahmed and N. Boland and G. L. Nemhauser, Decomposition of loosely coupled integer programs: A multiobjective perspective Optimization Online, 2016, http://www.optimization-online.org/DB_FILE/2016/08/5599.pdf
- [2] Sebastian Mau. Topology Optimization using Column Generation Methods. Master Thesis, HAW Hamburg, 2022.
- [3] Pavlo Muts, Stefan Bruche, Ivo Nowak, Ouyang Wu, Eligius MT Hendrix, and George Tsatsaronis. A column generation algorithm for solving energy system planning problems. *Optimization and Engineering*, pages 1–35, 2021.
- [4] Pavlo Muts, Ivo Nowak, and Eligius MT Hendrix. On decomposition and multiobjective-based column and disjunctive cut generation for MINLP. *Optimization and Engineering*, 22(3):1389–1418, 2021.
- [5] Tamim Mehr, Pavlo Muts, Ivo Nowak and Ouyang Wu Decogo - decomposition-based global optimizer. <https://github.com/ouyang-w-19/decogo>, 2020.

Utilizing Rotational Transformations for Convex Relaxations of Nonconvex QPs

Yehor Blokhin¹, Martin Ryner¹, Jan Kronqvist¹

¹*Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden
jankr@kth.se*

Abstract We present a framework that utilizes rotational transformations to enhance convex relaxations of nonconvex quadratic problems. By transforming the variable space we are able to better capture the feasible set defined by linear constraints, and obtain a stronger convex relaxation within the feasible set. Especially when the feasible set is poorly captured by the variable bounds, the rotation can result in stronger bounds when used together with convexification techniques like α BB and McCormick envelopes.

Keywords:

Convexification, Nonconvex QP, Rotational transformation

1. Introduction

Here, we focus on problems of the form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x + \frac{1}{2} x^\top Q x \\ \text{s.t. } & Ax \leq b, \quad lb \leq x \leq ub, \end{aligned} \tag{1}$$

where Q is a symmetric, but not positive definite, matrix. We assume each variable x_i has a finite lower bound lb_i and upper bound ub_i . We define the feasible set and the box given by the variable bounds as

$$\mathcal{F} = \{x \in \mathbb{R}^n : Ax \leq b, lb \leq x \leq ub\}, \quad \mathcal{B} = \{x \in \mathbb{R}^n : lb \leq x \leq ub\}. \tag{2}$$

Several approaches have been proposed over the years for convexifying problem (1), such as different SDP relaxations [4], α BB [1], and relaxations based on McCormick envelopes [5]. Here, we present a framework based on a rotational transformation that can be used in conjunction with an α BB type relaxation or relaxing the problem through McCormick envelopes. The main idea is to transform the problem to better capture the constraints $Ax \leq b$ and obtain a stronger relaxation.

For the proposed framework to be effective, the constraints $Ax \leq b$ need to have a clear role in the optimization problem, and we will formalize this in the following assumption.

Assumption 1. *We assume the feasible set \mathcal{F} is significantly smaller than the box \mathcal{B} , i.e., a large portion of \mathcal{B} is not within \mathcal{F} .*

If $\mathcal{F} = \mathcal{B}$, it is unlikely that the proposed framework can result in a stronger convex relaxation.

2. Rotational transformations

The rotational transformation is based on mapping the problem into a (rotated) space by

$$y = Bx, \quad (3)$$

where $y \in \mathbb{R}^n$, and B is a given invertible matrix. Thus, the rotational transformation is defined by the matrix B .

We can then represent problem (1) in the y -space as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top B^{-1}y + \frac{1}{2}y^\top (B^{-1})^\top Q B^{-1}y \\ \text{s.t.} \quad & AB^{-1}y \leq b, \\ & lb \leq B^{-1}y \leq ub. \end{aligned} \quad (4)$$

If B is orthonormal, then $y = Bx$ is simply a rotation of the space.

An example of a rotation that is frequently used in this nonconvex QP setting is given by choosing $B = V^\top$, where V is obtained from an eigenvalue decomposition of Q , i.e., $VDV^\top = Q$ with D being a diagonal matrix with the eigenvalues and V formed by the eigenvectors. This “Eigenvector-based” rotation has the advantages that the quadratic part of the objective is given by $\sum_{i=1}^n \frac{\lambda_i}{2} y_i^2$, and convexifying this expression is easy as we have a sum of univariate convex or concave functions over a box domain. Such an approach is, e.g., used by [3] in a branch-and-bound framework.

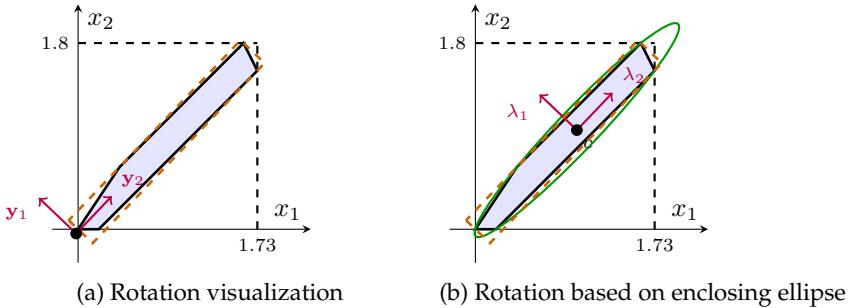


Figure 1: The shaded area represents the feasible set \mathcal{F} . The dashed black lines represent the smallest box defined by the bounds on the x -variables. Similarly, the dashed orange lines represent the smallest box given by the bounds on the rotated y -variables. The green ellipse is the smallest ellipse containing the feasible set \mathcal{F} , and the arrows in the right figure show the eigenvectors of the ellipse.

Our goal is to exploit the rotation to form a tighter convex underestimator of the objective function over the feasible set \mathcal{F} . *Our work is based on the simple observation that the variable bounds play a crucial role in several convexification techniques.* For example, a convexification based on McCormick envelopes will be tight along the boundaries of the box \mathcal{B}_y defined by the upper and lower bounds of the y -variables. Similarly, an α BB relaxation will be tight at the corners of the box. We will exploit this property, and we aim to rotate the space such that we can better capture the feasible set \mathcal{F} . The idea to rotate the space to better capture the feasible set in the variable bounds is illustrated in Figure 1(a).

The key is that the rotation allows us to make the convex relaxation tight in different parts of the search space. We can also combine different rotations to form convex underestimators that are tight in different locations of the search space by transforming the convexified function back into the x -space and repeating the procedure. The main challenges are: I) how to determine a good rotation, and II) how to efficiently determine strong variable bounds after the transformation. We are investigating several methods for determining a good rotation, but due to space limitations we only present the method based on the enclosing ellipse here.

3. Rotation based on enclosing ellipsoid

To capture the shape of the feasible set \mathcal{F} , we propose to base the rotation on the smallest enclosing ellipsoid [2]. Once we have computed the ellipse,

we can choose the rows of the transformation matrix B as the eigenvectors of the ellipse. The ellipse also provides bounds on the y -variables, although not necessarily tight bounds. The rotation based on the enclosing ellipsoid does not necessarily result in the smallest box in the y -space, but can provide a good approximation. The concept is illustrated in Figure 1(b).

4. Illustrative example

To illustrate the main ideas, consider the following problem

$$\begin{aligned} \min_x \quad & -x_1^2 - 0.5x_1x_2 - 2x_2^2 + 5x_1 - 5x_2 \\ \text{s.t. } & -x_1 + x_2 \leq 0.2, \quad x_1 - x_2 \leq 0.2, \\ & -1.5x_1 + x_2 \leq 0, \quad 2x_1 + x_2 \leq 5, \\ & 0 \leq x_1, x_2 \leq 2. \end{aligned} \tag{ex1}$$

The smallest enclosing ellipsoid is illustrated in Figure 1(a). The ellipse is given by $E = \{x : (x - c)^T P(x - c) \leq 1\}$. We obtain the rotation basis by selecting the normalized eigenvectors of P as the rows of rotation matrix B .

Now, let's analyze the impact of the rotation. The optimal function value for this problem $f^*(x) = -8.60$, while the αBB relaxation in the x -variable space results in a lower bound of $f_{\alpha BB}^*(x) = -10.66$. Here the rotation allows us to better capture the feasible set \mathcal{F} and the rotated αBB relaxation results in the stronger lower bound $f_{R\alpha BB}^*(x) = -8.62$.

5. Summary

Here we have presented the basic idea of utilizing a rotational transformation in conjunction with convexification technique to obtain a stronger relaxation by better exploiting the linear constraints.

Acknowledgments

The authors gratefully acknowledge financial support from Vergstiftelsen and Göran Gustafssons stiftelse

References

- [1] Claire S Adjiman, Stefan Dallwig, Christodoulos A Floudas, and Arnold Neumaier. A global optimization method, α BB, for general twice-differentiable constrained NLPs—I. Theoretical advances. *Computers & Chemical Engineering*, 22(9):1137–1158, 1998.
- [2] Nathaniel Bowman and Michael T Heath. Computing minimum-volume enclosing ellipsoids. *Mathematical Programming Computation*, 15(4):621–650, 2023.
- [3] Marcia Fampa, Jon Lee, and Wendel Melo. On global optimization with indefinite quadratics. *EURO Journal on Computational Optimization*, 5:309–337, 2017.
- [4] Christodoulos A Floudas and Viswanathan Visweswaran. Quadratic optimization. *Handbook of global optimization*, pages 217–269, 1995.
- [5] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

Separating Hyperplanes for Mixed-Integer Polynomial Optimization Problems

Carl Eggen,¹ Jan Kronqvist,² Andreas Lundell³ and Stefan Volkwein¹

¹*Department of Mathematics and Statistics, University of Konstanz, Universitätsstraße 10, 78464 Konstanz, Germany, {carl.eggen, stefan.volwein}@uni-konstanz.de*

²*Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden, jankr@kth.se*

³*Information Technology, Åbo Akademi University, Vaasa, Finland, andreas.lundell@abo.fi*

Abstract Polyhedral outer approximation is an established method in mixed-integer nonlinear programming. However, computing such approximations for nonconvex problems remains challenging. When the optimization problem is polynomial, the moment-/sum-of-squares-hierarchy from polynomial optimization can be used to strengthen an initial relaxation of the feasible set by iteratively adding linear inequalities and separating infeasible points.

Keywords: Mixed-integer nonlinear programming, Polyhedral outer approximation, Polynomial optimization, Semidefinite programming

1. Introduction

Mixed-integer nonlinear optimization problems (MINLP) are optimization problems that involve both continuous and discrete variables as well as linear and nonlinear constraints. One fundamental concept for solving such problems is the relaxation (i.e., enlargement) of the feasible set to obtain lower bounds on the optimal value. These relaxations should be much easier to solve than the original problem and can be obtained by a polyhedral

outer approximation of the feasible set. Algorithms based on polyhedral outer approximation usually strengthen a relaxation by iteratively adding linear inequalities (e.g. [2]). As long as the solution of the relaxation is infeasible for the original problem, an attempt is made to compute a new linear inequality (also referred to as a hyperplane) that excludes the infeasible solution. This attempt may fail if the feasible set is nonconvex. Nevertheless, it is still possible to obtain tight lower bounds on the optimal value of the original MINLP, if the objective is linear, which can be always ensured by an epigraph reformulation. This is shown by the following proposition.

Proposition 1. *Let $M \subseteq \mathbb{R}^n$. Then the optimal value of a linear function f over M is equal to the optimal value of f over the convex hull $\text{conv}(M)$ of M .*

Therefore, our aim is to approximate the convex hull of the feasible set using hyperplanes. If M is already a convex set, those hyperplanes can be calculated by using the first order Taylor series expansion of the constraints as in [2, 4], but the same method cannot be directly transferred to the non-convex case. However, if the objective and the constraints are all polynomials, which we then call a mixed-integer polynomial optimization problem (MIPOP), the moment-/sum-of-squares-hierarchy from polynomial optimization can be applied. An MIPOP is formally defined as follows:

$$\begin{aligned} f_{\min} = \min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad g_j(x) \geq 0, j = 1, \dots, k \\ x^l \leq x \leq x^u, x_i \in \mathbb{Z} \text{ for } i \in I. \end{aligned} \quad (\text{MIPOP})$$

In (MIPOP) the objective f and the constraints g_j are polynomials in $\mathbb{R}[x]$. Without loss of generality, we assume that f is a linear polynomial with $f(0) = 0$. All variables are bounded by $x^l, x^u \in \mathbb{R}^n$ and some of them are integer ($I \subseteq \{1, \dots, n\}$). Let M denote the feasible set and $S \supseteq M$ the relaxed feasible set, when the integer constraints are dropped. Since the convex hulls of these two sets can differ, we first concentrate on finding hyperplanes approximating the convex hull of the larger set S .

A similar idea of computing linear inequalities for nonconvex feasible sets was already established in [1], but was not integrated into an algorithmic framework to iteratively tighten the lower bound of the optimal value of an MINLP.

2. Separating Hyperplanes for MIPOPs

Let \bar{x} be a solution of a relaxation, but infeasible for (MIPOP). Further we assume that $\bar{x} \notin \text{conv}(S)$. Since S is compact, also $\text{conv}(S)$ is compact and

there exists a hyperplane that strongly separates \bar{x} and $\text{conv}(S)$, that means there exists a linear polynomial $h \in \mathbb{R}[x]_1$ such that $h(\bar{x}) \geq \varepsilon$ for some $\varepsilon > 0$ and $h \leq 0$ on $\text{conv}(S)$. Finding such a linear polynomial can be formulated as the infinite-dimensional linear optimization problem

$$\sup_{h \in \mathbb{R}[x]_1} h(\bar{x}) = w_h^\top \bar{x} - b_h \quad \text{subject to} \quad h(x) \leq 0 \text{ for all } x \in S, \quad (\text{HYP})$$

$$\|w_h\|_\infty \leq 1.$$

Since $\bar{x} \notin \text{conv}(S)$, \bar{x} and $\text{conv}(S)$ can be strongly separated and (HYP) has always a feasible linear polynomial with positive optimal value. Otherwise, if the optimal value of (HYP) is 0 (i.e., $h = 0$ is an optimal solution), then \bar{x} cannot be strongly separated from $\text{conv}(S)$ and that means $\bar{x} \in \text{conv}(S)$. Thus, the optimal value of (HYP) also certifies if separation is possible.

However, solving (HYP) is not easy in general because of the semi-infinite constraint. Substituting the box constraint $x^l \leq x \leq x^u$ by the quadratic conditions $\varphi_i(x) \geq 0$ with

$$\varphi_i(x) := (x_i - x_i^l)(x_i^u - x_i) \in \mathbb{R}[x],$$

for all $i \in \{1, \dots, n\}$, we gain a polynomial description of the feasible set S . To find a separating hyperplane, we can now invoke the moment-/sum-of-squares-hierarchy by replacing the semi-infinite nonnegativity constraint with a degree-bounded sum-of-squares constraint [3]:

$$\begin{aligned} \sup h(\bar{x}) \quad &\text{subject to} \quad -h = \sigma_0 + \sum_{j=1}^k \sigma_j g_j + \sum_{i=1}^n \sigma_{k+i} \varphi_i, \quad (\text{SOS}_d^{\text{HYP}}) \\ &h \in \mathbb{R}[x]_1, \|w_h\|_\infty \leq 1, \\ &\sigma_0, \dots, \sigma_{k+n} \text{ are SOS,} \\ &\deg(\sigma_0), \deg(\sigma_1 g_1), \dots, \deg(\sigma_{k+n} \varphi_n) \leq 2d. \end{aligned}$$

As the name suggests, a polynomial is a sum-of-squares (SOS), if it can be written as a sum of squared polynomials. Recognizing whether a polynomial is an SOS can be done by semidefinite programming. Thus, $(\text{SOS}_d^{\text{HYP}})$ can be solved with an SDP solver for each $d \in \mathbb{N}$.

If h is a solution of $(\text{SOS}_d^{\text{HYP}})$ with $h(\bar{x}) > 0$ for some $d \in \mathbb{N}$, then h is a separating hyperplane and the linear inequality $h(x) \leq 0$ can be used to strengthen the relaxation separating the relaxed solution \bar{x} . This leads to the following separating hyperplane algorithm for MIPOPs:

Algorithm 1 Ideal separating hyperplane algorithm for MIPOPs

Require: MIPOP, $d \in \mathbb{N}$, tolerance $\varepsilon > 0$.

- 1: Define initial relaxed set Ω_0 and obtain initial solution x_0^{relax} .
 - 2: **while** violation of nonlinear constraints $\geq \varepsilon$ **do**
 - 3: Solve $(\text{SOS}_d^{\text{HYP}})$ and add hyperplane to Ω_k separating x_k^{relax} .
 - 4: Solve relaxation with new relaxed set Ω_{k+1} and obtain solution x_{k+1}^{relax} .
 - 5: $k \leftarrow k + 1$.
 - 6: **end while**
 - 7: **return** x_k^{relax} and terminate the algorithm.
-

Up to now, Algorithm 1 is only an ideal version of how the moment-/SOS-hierarchy can be embedded into the solution process of an MIPOP. We plan to investigate the choice of parameters, conditions for termination and convergence of the algorithm, as well as its practical implementation and scalability. Furthermore, we will explore how integrality can be exploited. If the numerical tests are convincing, we will integrate our method into the solver SHOT [4].

3. Summary

In this abstract, we propose a polyhedral outer approximation algorithm for mixed-integer polynomial optimization problems, even if some constraints are nonconvex. Adapting the moment-/SOS-hierarchy from polynomial optimization, we strengthen an initial relaxation of the feasible set by iteratively adding linear inequalities and separating infeasible points.

References

- [1] Behrends, S. and Schöbel, A. (2020). Generating valid linear inequalities for nonlinear programs via sums of squares. *Journal of Optimization Theory and Applications*. 186. 911–935.
- [2] Kronqvist, J., Lundell, A. and Westerlund, T. (2016). The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization*. 64. 249–272.
- [3] Lasserre, J. B. (2001). Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*. 11. 796–817.
- [4] Lundell, A., Kronqvist, J. and Westerlund, T. (2022). The supporting hyperplane optimization toolkit for convex MINLP. *Journal of Global Optimization*. 84. 1–41.

Session 9





Algorithmic Advances in Proximal Gradient Methods for Nonconvex Minimization *

Marc Teboulle

School of Mathematical Sciences, Tel Aviv University teboulle@tauex.tau.ac.il

Abstract Proximal based methods have emerged as a powerful approach for efficiently solving well-structured optimization problems. We review recent advances of these algorithms in the nonconvex and nonsmooth setting, emphasizing convergence analysis and highlighting simple schemes tailored to modern applications like quadratic inverse problems and constrained nonnegative matrix factorization.

Keywords: nonsmooth nonconvex minimization, non Euclidean/Bregman proximal methods, global convergence, semi-algebraic functions

1. Introduction

Nonconvex and nonsmooth optimization models abound in modern applications such as machine learning and signal/image processing, often offering more faithful representations than their convex relaxations. However, even for simple cases, finding global minima is NP-hard! As a result, when designing/analyzing optimization algorithms, the most realistic goal we can hope is often to ensure a method that behaves well e.g., global convergence of the generated sequence to a critical point of the objective function. Beyond theoretical challenges, another central issue is on the computational front. Indeed, problems are typically very large scale, demanding efficient and tractable iterative methods. Here, we focus on a broad class of nonconvex and nonsmooth optimization problems under data assumptions that strike a balance between generality and structure. On one hand, the class is rich enough to encompass a wide range of inter-

*Partially supported by the Israel Science Foundation ISF grant No. 2619-20 and the German Research Foundation DFG grant No. 800240.

esting applications; on the other, by exploiting specific problem structures and data characteristics, allows for a global convergence analysis of simple and scalable algorithms relevant in many applications.

2. A Bregman Proximal Gradient Scheme

Consider the basic nonsmooth nonconvex minimization problem:

$$(CM) \quad \min \{ \Psi(x) \equiv f(x) + g(x) : x \in \mathbb{R}^n \},$$

with $f : \mathbb{R}^n \rightarrow (-\infty, \infty]$ nonsmooth extended valued (allowing constraints) and $g : \mathbb{R}^n \rightarrow \mathbb{R}$ is C^1 . This "simple" model (M) captures various classes of smooth, nonsmooth, convex and nonconvex problems. In addition, a central and usual assumption is that g admits a global L -Lipschitz continuous gradient ($g \in C_L^{1,1}$). It has been extensively studied in the convex setting, namely with both f, g convex [7]. Here, we completely depart from both the convex assumption on (f, g) , and the smoothness assumption on g , i.e., g does not admit a global Lipschitz gradient.

How to handle $g \notin C_L^{1,1}$? This question was addressed in the convex case [2, 1] and extended to nonconvex setting in [4]. Here, for ease of description, we adopt a simplified version from [4]. The underlying idea is based on better capturing the geometry of the problem at hand by using a suitable proximity measure which adapts to a given objective to produce a more accurate approximation.

Lemma 1. (Fundamental Approximation) Let $g, h : \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuously differentiable functions, and assume that h is convex. Then, (g, h) are L -smooth adaptable (L -SMAD) for some $L > 0$ if and only if

$$g(x) \leq g(y) + \langle \nabla g(y), x - y \rangle + LD_h(x, y) \quad \forall x, y \in \mathbb{R}^n.$$

Here, D_h stands for the Bregman Distance associated to h :

$$D_h(x, y) := h(x) - [h(y) + \langle \nabla h(y), x - y \rangle].$$

With $h(\cdot) = \|\cdot\|^2/2$, we recover the classical descent Lemma for $g \in C_L^{1,1}$; for more examples and the general setting see [1, 4].

Let $x \in \mathbb{R}^n$ and $\lambda > 0$. The Bregman Proximal Gradient (BPG) map is:

$$T_\lambda(x) := \operatorname{argmin}_u \left\{ f(u) + \langle \nabla g(x), u - x \rangle + \frac{1}{\lambda} D_h(u, x) \right\}.$$

Note that since f is *nonconvex*, the mapping T_λ needs not be single-valued. Under adequate qualification conditions [4], we can prove well-posedness. A basic iteration step of BPG for (CM) reads: for any $x \in \mathbb{R}^n : x^+ \in T_\lambda(x)$. We consider more structure to allow deriving explicit schemes in many applications, and focus on the nonconvex Block Composite Model [8]:

$$(B\text{-CM}) \quad \inf\{\Psi(x) \equiv g(x_1, x_2, \dots, x_d) + \sum_{j=1}^d f_j(x_j) : x_j \in \mathbb{R}^{n_j}, j = 1, \dots, d\}.$$

The objective can be decomposed into d separable blocks. If we fix $d - 1$ blocks equal to some point \bar{x} and only keep the j -th block free, then we obtain the following composite model. For $\bar{x} \in \mathbb{R}^n$ and $j = 1, 2, \dots, d$ let: $g^{(\bar{x}, j)}(x_j) \equiv g(\bar{x}_1, \dots, \bar{x}_{j-1}, x_j, \bar{x}_{j+1}, \dots, \bar{x}_d)$, and consider:

$$\inf \left\{ g^{(\bar{x}, j)}(x_j) + f_j(x_j) : x_j \in \mathbb{R}^{n_j} \right\}.$$

Our method is *block coordinate type*, operates sequentially on the d blocks.

Assumption A For any $\bar{x} \in \mathbb{R}^n$ and $j = 1, 2, \dots, d$:

- (i) There exists a kernel $h^{(j)}$ such that $(h^{(j)}, g^{(\bar{x}, j)})$ are $L^{(\bar{x}, j)}$ -SMAD.
- (ii) There exists some $\bar{L}^{(j)}, \underline{L}^{(j)} \in \mathbb{R}_{++}$ such that $\underline{L}^{(j)} \leq L^{(\bar{x}, j)} \leq \bar{L}^{(j)}$.
- (iii) The function $h^{(j)}$ is strongly convex with parameter σ_j .
- (iv) $\nabla h^{(j)}$ and ∇g are Lipschitz continuous on any bounded subset of $\mathbb{R}^{n_j}, \mathbb{R}^n$.

Note that assumption (iii) is easy to enforce (since g is nonconvex!) and (iv) is harmless.

Cyclic Block Bregman Proximal Gradient - CBBPG

Initialization. Set $\rho \in (0, 1)$ and let $x^0 \in \mathbb{R}^n$.

General Step. For $k = 0, 1, \dots$, for $j = 1, \dots, d$, choose $\lambda^{(k, j)} = \rho(L^{(x^{(k, j-1)}, j)})^{-1}$ and compute $\gamma^{(k, j)} := \nabla g^{(x^{(k, j-1)}, j)}(x_j^k)$ and

$$x_j^{k+1} \in \operatorname{argmin} \left\{ f_j(x) + \langle \gamma^{(k, j)}, x - x_j^k \rangle + \left(\lambda^{(k, j)} \right)^{-1} D_{h^{(j)}}(x, x_j^k) : x \in \mathbb{R}^{n_j} \right\}.$$

Theorem 2. (Global Convergence) Consider problem (B-CM) under Assumption A. Let $\{x^k\}_{k \in \mathbb{N}}$ be a bounded sequence generated by the CBBPG method.

(i) Subsequential convergence. Any limit point x^∞ of $\{x^k\}_{k \in \mathbb{N}}$ is a critical point of Ψ , i.e., $0 \in \partial\Psi(x^\infty)$.

(ii) Global convergence. If Ψ is semi-algebraic, then the sequence $\{x^k\}_{k \in \mathbb{N}}$ has finite length and converges to a critical point x^* of Ψ .

The main tool to prove this result relies on the general methodology of [3] based on the uniformization of the so-called Kurdyka-Łojasiewicz (KL) property. Also note that semi-algebraic functions abound in applications.

3. Applications to two Prototype Models

Many fundamental modern applications can be cast through the composite model (B-CM). Two prototypes are quadratic inverse and constrained nonnegative matrix factorization problems. We will present the resulting simple explicit schemes for these two prototype problems.

4. Alternative to L-SMAD

Back to the nonconvex model (CM), finding an appropriate h such that g is L-SMAD can be a challenging task. A natural question arises: is it still possible to develop a convergent proximal scheme for (CM) when g has only a *locally* Lipschitz gradient? We answer this affirmatively. Building on our recent work [5] on Lagrangian methods for nonlinear constrained problems, here we exploit the simpler structure of (CM) to derive a specific convergent *dynamic* proximal gradient scheme [6].

References

- [1] H. H. Bauschke, J. Bolte, and M. Teboulle. (2017). A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42. 330-348.
- [2] B. Birnbaum, N. R. Devanur, and L. Xiao. (2011) Distributed algorithms via gradient descent for fisher markets. In *Proceedings of the 12th ACM conference on Electronic commerce*. 127–136.
- [3] J. Bolte, S. Sabach, and M. Teboulle. (2014). Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146. 459-494.
- [4] J. Bolte, S. Sabach, M. Teboulle, and Y. Vaisbourd. (2018). First order methods beyond convexity and lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM J. Optimization*, 28. 2131-2151.
- [5] E. Cohen, N. Hallak, and M. Teboulle. (2022). A dynamic alternating direction of multipliers for nonconvex minimization with nonlinear functional equality constraints. *Journal of Optimization Theory and Applications*, 193. 324–353.
- [6] E. Cohen and M. Teboulle. (2024). The Proximal Gradient Method with Locally Lipschitz Gradient. Preprint.
- [7] M. Teboulle. (2018). A simplified view of first order methods for optimization. *Mathematical Programming*, 170. 67-96.
- [8] M. Teboulle and Y. Vaisbourd. (2020). Novel Proximal Gradient Methods for Nonnegative Matrix Factorization with Sparsity Constraints. *SIAM J. Imaging Sciences*, 13. 381-421.

Efficient Batch Multiobjective Bayesian Optimization via Pareto Optimal Thompson Sampling

Ashwin Renganathan,¹ Kade Carlson,¹

¹*Penn State, University Park, PA 16802, USA*

Abstract Classical evolutionary approaches for multiobjective optimization are quite accurate but incur a lot of queries to the objectives; this can be prohibitive when objectives are expensive to evaluate. A sample-efficient approach to solving multiobjective optimization, in the absence of derivative information, is via Gaussian process (GP) surrogates and Bayesian optimization (BO). However, current multiobjective Bayesian optimization (MOBO) methods are limited in accuracy – we attribute this to a difficult “inner” optimization of an acquisition function which could be nonconvex, nondifferentiable, and/or stochastic. To overcome these challenges, we propose a Thompson sampling (TS) based approach (q POTS). Whereas TS chooses candidates according to the probability that they are optimal, q POTS chooses candidates according to the probability that they are Pareto optimal. q POTS hybridizes BO and evolutionary algorithms and hence gets the best of both worlds. q POTS is endowed with theoretical guarantees and demonstrates superior empirical performance.

Keywords: Multiobjective optimization, Bayesian optimization, Thompson sampling

1. Introduction

Mathematically, we consider the simultaneous constrained optimization of K objectives

$$\begin{aligned} & \max_{\mathbf{x} \in \mathcal{X}} \{f_1(\mathbf{x}), \dots, f_K(\mathbf{x})\} \\ \text{s.t. } & c_i(\mathbf{x}) = 0, \quad i \in \mathcal{E} \\ & c_i(\mathbf{x}) \geq 0, \quad i \in \mathcal{I} \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ is the design variable, $f_k : \mathcal{X} \rightarrow \mathbb{R}$, $\forall k = 1, \dots, K$, are expensive zeroth-order oracles (i.e., no derivative information). $c_i : \mathcal{X} \rightarrow \mathbb{R}$ are nonlinear constraints, also expensive zeroth-order oracles, where \mathcal{E} and \mathcal{I} are the sets of indices of equality and inequality constraints, respectively. \mathcal{X} is the domain of f and c . We are interested in identifying a *Pareto* optimal set of solutions, which “Pareto dominates” all other points. Let $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_K(\mathbf{x})]^\top$; when a solution $\mathbf{f}(\mathbf{x})$ Pareto dominates another solution $\mathbf{f}(\mathbf{x}')$, then $f_k(\mathbf{x}) \geq f_k(\mathbf{x}')$, $\forall k = 1, \dots, K$ and $\exists k \in [K]$ such that $f_k(\mathbf{x}) > f_k(\mathbf{x}')$. We write Pareto dominance as $\mathbf{f}(\mathbf{x}) \succ \mathbf{f}(\mathbf{x}')$. The set $\mathcal{Y}^* = \{\mathbf{f}(\mathbf{x}) : \nexists \mathbf{x}' \in \mathcal{X} : \mathbf{f}(\mathbf{x}') \succ \mathbf{f}(\mathbf{x})\}$ is called the *Pareto frontier* and the set $\mathcal{X}^* = \{\mathbf{x} \in \mathcal{X} : \mathbf{f}(\mathbf{x}) \in \mathcal{Y}^*\}$ is called the *Pareto set*.

We are interested in sample-efficient (that is, minimal queries to the oracles) approaches to solving our problem. The de facto approach is to use evolutionary algorithms (EA), e.g., NSGA (8; 9). While they are accurate they are not sample efficient. Instead, we propose to use multiobjective Bayesian optimization (MOBO) with Gaussian process (GP) surrogate models for each objective. MOBO, on the other hand, offers the promise of sample efficiency, but existing methods suffer from several limitations.

1.1 Related work

One of the most common approaches in MOBO are the ones based on scalarizations. Examples include ParEGO (13), MOEA/D-EGO (21), and qNParEGO (6). Scalarizations have the drawback that they are not sample efficient and often fail to capture disconnected and non-convex/concave Pareto frontiers. Another line of work uses the hypervolume indicator; but this usually results in a difficult nonconvex acquisition function. Examples include EHVI and its batch variants (3; 10; 4; 6); gradient based extensions of the EHVI also exist (20; 5), and SUR (16) and DGEMO (14). A third major category of work in MOBO is the use of entropy-based acquisition functions. These primarily include PESMO (11), MESMO (1), PFES (18), and JESMO (12; 19). However these acquisition functions are still not available

in closed form. The use of Thompson sampling (TS) in MOBO is relatively underexplored to the best of our knowledge. Existing methods inherit the aforementioned limitations; e.g., TSEMO (2), TS-TCH, and MORBO (7).

Existing limitations and our contributions.. Existing approaches suffer from one or more of the following limitations. (i) Difficult acquisition function optimization. (ii)Inability to handle constraints, and (iii) Inability or difficulty handling batch acquisitions. We overcome these limitations in this work.

2. METHODOLOGY

We first fit K independent posterior GP models: $\mathcal{M}_1, \dots, \mathcal{M}_K$ for the K objectives, with observations $\mathcal{D}_n^{1:K} \triangleq \{\mathcal{D}_n^1, \dots, \mathcal{D}_n^K\}$. In the multiobjective setting, acquisition functions are typically defined in terms of a hypervolume (HV) indicator. HV of a Pareto frontier \mathcal{Y}^* is the K -dimensional Lebesgue measure λ of the region dominated by \mathcal{Y}^* and bounded from below by a reference point $\mathbf{r} \in \mathbb{R}^K$. That is, $HV(\mathcal{Y}^* | \mathbf{r}) = \lambda_K(\bigcup_{\nu \in \mathcal{Y}^*} [\mathbf{r}, \nu])$, where $[\mathbf{r}, \nu]$ denotes the hyper-rectangle bounded by vertices \mathbf{r} and ν . EHVI (3; 10) computes the expectation (with respect to the posterior GP) of the improvement in HV due to a candidate point. Our approach, however, does not involve the computation of HV but is a direct extension of TS to the multiobjective setting.

2.1 qPOTS: Batch Pareto optimal Thompson sampling for MOBO

Inspired from TS, qPOTS chooses points according to the probability that they are Pareto optimal. That is, we choose \mathbf{x} according to the probability

$$p_{\mathcal{X}^*}(\mathbf{x}) = \int \delta \left(\mathbf{x} - \arg \max_{\mathbf{x} \in \mathcal{X}} \{Y_1(\mathbf{x}), \dots, Y_K(\mathbf{x})\} \right) p(Y_1 | \mathcal{D}_n^1) \dots p(Y_K | \mathcal{D}_n^K) dY_1 \dots dY_K, \quad (2)$$

where the integral is over the support of the joint distribution of all posterior GPs. In practice, this is equivalent to drawing sample paths from the K posterior GPs $Y_i(\cdot, \omega)$, $i = 1, \dots, K$, and choosing the next point(s) from its Pareto set $\mathbf{x}_{n+1} \in \arg, \max_{\mathbf{x} \in \mathcal{X}} \{Y_1(\mathbf{x}, \omega), \dots, Y_K(\mathbf{x}, \omega)\}$. We use the “reparametrization trick” to sample from the posterior (again we exclude notations for individual objectives): $Y(\cdot, \omega) = \mu_n(\cdot) + \Sigma_n^{1/2}(\cdot) \times Z(\omega)$, where Z is a standard multivariate normal random variable. Then, we solve the

following cheap multiobjective optimization problem

$$X^* = \arg \max_{\mathbf{x} \in \mathcal{X}} \{Y_1(\mathbf{x}, \omega), \dots, Y_K(\mathbf{x}, \omega)\}, \quad (3)$$

that can be solved using any method; we choose evolutionary approaches, e.g., the NSGA-II (9), so that we keep the best of both worlds: accuracy of evolutionary approaches and sample efficiency of MOBO. The next batch of q points is then chosen as $\mathbf{x}_{n+1:q} \in X^*$.

2.2 Extension to constrained problems

Let there be a total of C constraints $\{c_1, \dots, c_C\}$, and $\mathcal{I} \cup \mathcal{E} = [C]$. Further, let $Y_{k+i \in \mathcal{I}}$ and $Y_{k+i \in \mathcal{E}}$ denote the posterior GP of the i th inequality constraint and equality constraint functions, respectively. This way, we fit a total of $K + C$ posterior GPs $Y_1, \dots, Y_K, Y_{K+1}, \dots, Y_{K+C}$. For all $i \in [C]$, we define the constrained qPOTS to choose points according to

$$p_{\mathcal{X}^*}(\mathbf{x}) = \int \delta \left(\mathbf{x} - \arg \max_{\mathbf{x} \in \mathcal{X}} \{Y_1(\mathbf{x}), \dots, Y_K(\mathbf{x})\} \times \right. \\ \left. \mathbb{1}_{\{\bigcap_i Y_{k+i \in \mathcal{E}}(\mathbf{x})=0\}} \times \mathbb{1}_{\{\bigcap_i Y_{k+i \in \mathcal{I}}(\mathbf{x}) \geq 0\}} \right] p(Y_1 | \mathcal{D}_n) \dots p(Y_K | \mathcal{D}_n) dY_1 \dots dY_K, \quad (4)$$

where $\mathbb{1}_{\bigcap_i Y_{k+i \in \mathcal{I}}(\mathbf{x}) \geq 0}$ and $\mathbb{1}_{\bigcap_i Y_{k+i \in \mathcal{E}}(\mathbf{x})=0}$ are the indicator functions for joint feasibility with inequality and equality constraints, respectively, that take a value of 1 when feasible and 0 otherwise. The equivalent of (3) is then given by

$$X^* = \arg \max_{\mathbf{x} \in \mathcal{X}} \{Y_1(\mathbf{x}, \omega), \dots, Y_K(\mathbf{x}, \omega)\} \times \\ \mathbb{1}_{\{\bigcap_i Y_{k+i \in \mathcal{E}}(\mathbf{x})=0\}} \times \mathbb{1}_{\{\bigcap_i Y_{k+i \in \mathcal{I}}(\mathbf{x}) \geq 0\}}. \quad (5)$$

3. Experiments

Our methodology is benchmarked against the state of the art on select synthetic and real-world experiments; Figure 1 and Figure 2 captures the performance for sequential ($q = 1$) and batch ($q > 1$) settings respectively. Further experiments can be found in (17). Our software implementation is publicly available at <https://github.com/csdlpsu/qpots>.

Algorithm 1 qPOTS: Batch Pareto optimal Thompson sampling

Input: With data sets $\mathcal{D}_n^{1:K+C}$, fit $K + C$ GP models \mathcal{M}_1 through \mathcal{M}_{K+C} and GP hyperparameters $\Omega_1, \dots, \Omega_{K+C}$.

Parameters B (total budget)

Output: Pareto optimal solution \mathcal{X}^* , \mathcal{Y}^* .

for $i = n + 1, \dots, B$, **do**

1. Sample the posterior GP sample paths: $Y_i(\mathbf{x}, \omega)$, $\forall i = 1, \dots, K$.
2. Solve the cheap multiobjective optimization problem Equation (3) (or (5) for constrained problems) via evolutionary algorithms.
3. Choose q candidate points from X^* according to (??).
4. Observe oracles and constraints and append to data set $\mathcal{D}_i^{1:K+C}$.
5. Update GP hyperparameters $\Omega_1, \dots, \Omega_{K+C}$

end for

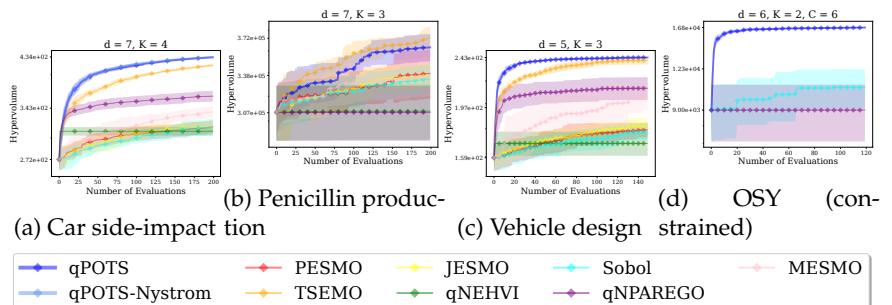


Figure 1: **Sequential acquisition.** Hypervolume Vs. iterations for sequential ($q = 1$) acquisition; plots show mean and ± 1 standard deviation out of 10 repetitions. qPOTS outperforms all competitors or is amongst the best. Bottom right shows constrained case on the OSY problem (15). Additional experiments are provided in the supplementary material.

4. Summary

We present a novel method for derivative-free constrained multiobjective optimization. Our method, qPOTS, leverages Thompson sampling and Bayesian optimization, to propose a novel acquisition policy. qPOTS hybridizes classical evolutionary algorithms that are accurate but not sample efficient, with multiobjective Bayesian optimization which are sample efficient but suffer from several challenges that ultimately impair their sample

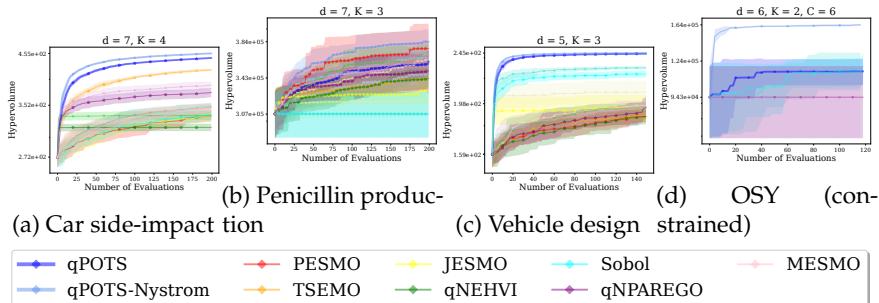


Figure 2: Batch acquisition. Hypervolume Vs. iterations for batch ($q > 1$) acquisition; plots show mean and ± 1 standard deviation out of 10 repetitions. qPOTS outperforms all competitors, but the benefit is more pronounced in the batch case. Additional experiments, including constrained problems, are shown in the supplementary material.

efficiency, thereby getting the best of both worlds. *qPOTS* is endowed with theoretical guarantees and demonstrates strong empirical performance based on a variety of experiments.

References

- [1] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value Entropy Search for Multi-Objective Bayesian Optimization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [2] Eric Bradford, Artur M Schweidtmann, and Alexei Lapkin. Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm. *Journal of global optimization*, 71(2):407–438, 2018.
- [3] Ivo Couckuyt, Dirk Deschrijver, and Tom Dhaene. Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *Journal of Global Optimization*, 60(3):575–594, November 2014.
- [4] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 9851–9864. Curran Associates, Inc., 2020.
- [5] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864, 2020.
- [6] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian Optimization of Multiple Noisy Objectives with Expected Hypervolume Improvement. In *Advances in Neural Information Processing Systems*, volume 34, pages 2187–2200. Curran Associates, Inc., 2021.
- [7] Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective bayesian optimization over high-dimensional search spaces. In *Uncertainty in Artificial Intelligence*, pages 507–517. PMLR, 2022.
- [8] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings 6*, pages 849–858. Springer, 2000.

- [9] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [10] Michael Emmerich. The computation of the expected improvement in dominated hypervolume of Pareto front approximations. January 2008.
- [11] Daniel Hernández-Lobato, José Miguel Hernández-Lobato, Amar Shah, and Ryan P Adams. Predictive Entropy Search for Multi-objective Bayesian Optimization.
- [12] Carl Hvarfner, Frank Hutter, and Luigi Nardi. Joint entropy search for maximally-informed bayesian optimization. *Advances in Neural Information Processing Systems*, 35:11494–11506, 2022.
- [13] Joshua Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.
- [14] Mina Konakovic Lukovic, Yunsheng Tian, and Wojciech Matusik. Diversity-Guided Multi-Objective Bayesian Optimization With Batch Evaluations. In *Advances in Neural Information Processing Systems*, volume 33, pages 17708–17720. Curran Associates, Inc., 2020.
- [15] Andrzej Osyczka and Sourav Kundu. A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm. *Structural optimization*, 10:94–99, 1995.
- [16] Victor Picheny. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25(6):1265–1280, 2015.
- [17] Ashwin Renganathan and Kade Carlson. qpots: Efficient batch multiobjective bayesian optimization via pareto optimal thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 4051–4059. PMLR, 2025.
- [18] Shinya Suzuki, Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Multi-objective bayesian optimization using pareto-frontier entropy. In *International Conference on Machine Learning*, pages 9279–9288. PMLR, 2020.
- [19] Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective bayesian optimization. *Advances in Neural Information Processing Systems*, 35:9922–9938, 2022.

- [20] Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Multi-objective bayesian global optimization using expected hyper-volume improvement gradient. *Swarm and evolutionary computation*, 44:945–956, 2019.
- [21] Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by moea/d with gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3):456–474, 2009.

Session 10





A Bi-level Service Capacity Planning Model with Interdependent MDP Followers

Seokwoo Kim¹, Dong Gu Choi*

¹*Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea,*
¹seokwookim@postech.ac.kr, *dgchoi@postech.ac.kr

Abstract In many on-demand service systems such as ride-hailing or meal-delivery platforms, capacity planning with self-interested drivers presents unique challenges. We study novel modeling and solution methodologies for the integrated problem of staffing and compensating drivers under a service network, considering drivers' best-response behaviors. This problem is formulated as a bi-level program where interdependent followers solve a semi-Markov decision process problem. Extending on a fluid equilibrium model developed in the literature, we relax this problem into another bi-level program with complementarity constraints which admits a mixed-integer bilinear single-level reformulation. Based on problem-specific insight, we derive a novel mixed-integer linear reformulation and derive useful problem-specific valid inequalities to tighten the LP relaxation. Further, a decomposition based exact method is investigated.

Keywords: Bi-level Programming, Equilibrium Problems, Workforce Management

1 Introduction

In on-demand service systems such as ride-hailing and meal-delivery platforms, capacity planning faces unique challenges as servers are gig drivers with strategic autonomy—unlike traditional systems where servers are static resources confined to fixed locations and deterministically serve orders. These drivers can selectively reject orders or relocate to maximize their profits, requiring planners to determine both optimal staffing and

compensation levels that account for drivers' best-response behaviors to ensure adequate service coverage.

The strategic behavior of gig drivers has been extensively investigated within game-theoretic frameworks, generating substantial recent literature [1, 2, 5, 8]. While driver equilibrium concepts have been developed on closed queueing networks with fluid-based approximations [1, 3, 7], solution approaches for platform control of driver equilibrium through compensation and staffing decisions remain limited. Our approach presents a bi-level model in which each follower's problem is formulated as a semi-Markov decision process (SMDP), where the sojourn time of the SMDP reflects the congestion among drivers. Exploiting the linear program (LP) counterpart of the SMDP model, this research investigates novel reformulation and decomposition strategies.

2 Problem Description

Let $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ be a directed graph, where \mathcal{V} is a set of service regions (nodes) and $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of arcs. Every customer order has an origin and destination for each arc $(l, k) \in \mathcal{A}$ and we assume the order arrival process follows a Poisson process with rate Λ_{lk} , $(l, k) \in \mathcal{A}$. The total demand rate at region l is denoted by $\Lambda_l := \sum_{k \in \mathcal{V}} \Lambda_{lk}$. We model the on-demand service network as a closed queueing network as proposed by [1, 3], where drivers represent jobs relocating through the network \mathcal{G} . Further, the service regions are classified into 'hot' and 'cold' regions, i.e., $\mathcal{V} = \mathcal{V}^h \cup \mathcal{V}^c$, $\mathcal{V}^h \cap \mathcal{V}^c = \emptyset$ and we define the set of arcs $\mathcal{A}^h \subseteq \mathcal{V}^h \times \mathcal{V}$. We assume that orders are only arriving in hot regions, i.e., $\Lambda_j = 0 \forall j \in \mathcal{V}^c$. We assume that drivers can accept or reject the allocated order $(l, k) \in \mathcal{A}^h$, and relocate from their current region $l \in \mathcal{V}$ to other regions $k \in \mathcal{V}^h \setminus \{l\}$ after completing the order having destination l .

In anticipation of drivers' strategic decisions, a decision-making problem of the platform is to jointly decide the compensation level for each order and staffing level of drivers to minimize the labor cost subject to certain service level constraints, which can be modeled as a bi-level program. The *upper-level* problem determines locational compensation $\{\theta_{lk}\}_{(l,k) \in \mathcal{A}^h}$ and staffing level N . The *lower-level* problem is an equilibrium problem which is formulated as a mixed-complementarity problem (MCP). MCP consists of (i) equilibrium-related constraints to be discussed below, (ii) a representative driver's profit maximization problem formulated as SMDP. Proposed by [1, 3], the resulting equilibrium is well-defined in fluid-based steady-state queueing networks. Note that total N drivers' interdependent behavior is approximated by 'fluid-regime', similar to *non-atomic* congestion games.

3 Linear programming counterpart of a single driver's problem

We formulate the LP counterpart of driver SMDP as follows:

Lemma 1. *Under mild assumptions, the representative driver SMDP is a unichain SMDP which can be formulated to following LP.*

$$\begin{aligned} \max_y & \sum_{l \in \mathcal{V}^h} \sum_{k \in \mathcal{V}} \left\{ -c\tau_l^{en}y_{assigned,lk}(accept) + (\theta_{lk} + (\phi - c)\tau_{lk}^v)y_{ontrip,lk} \right\} + \\ & \sum_{m \in \mathcal{V}} \sum_{n \in \mathcal{V}^h} -c\tau_{mn}^r y_{arrived,m}(n) \end{aligned} \quad (1)$$

$$\text{s.t. } y_{idle,l} = \sum_{j \in \mathcal{V}} y_{arrived,j}(l) + \sum_{k \in \mathcal{V}^c} y_{assigned,l \rightarrow k}(reject) \quad \forall l \in \mathcal{V}^h \quad (2)$$

$$y_{assigned,lk}(accept) + y_{assigned,lk}(reject) = P_{lk}y_{idle,l} \quad \forall l \in \mathcal{V}^h, k \in \mathcal{V}^c \quad (3)$$

$$y_{assigned,lj}(accept) = P_{lj}y_{idle,l} \quad \forall l \in \mathcal{V}^h, j \in \mathcal{V}^h \quad (4)$$

$$y_{ontrip,lk} = y_{assigned,lk}(accept) \quad \forall l \in \mathcal{V}^h, k \in \mathcal{V} \quad (5)$$

$$\sum_{j \in \mathcal{V}^h} y_{arrived,k}(j) = \sum_{l \in \mathcal{V}^h} y_{ontrip,l \rightarrow k} \quad \forall \mathcal{V} \quad (6)$$

$$\begin{aligned} \sum_{l \in \mathcal{V}^h} & \left[\tau_l^w y_{idle,l} + \left(\sum_{k \in \mathcal{V}} \tau_l^{en} y_{assigned,lk}(accept) + \tau_{lk}^v y_{ontrip,lk} \right) + \sum_{j \in \mathcal{V}} \tau_{jl}^r y_{arrived,j}(l) \right] \\ & = N \end{aligned} \quad (7)$$

First, the decision variable y is an *occupancy-measure* of the SMDP ([4]). $y_{idle,l}$ represents the rate of drivers joining the queue in region l . $y_{assigned,lk}(accept)$ ($reject$) represents the rate of drivers accepting (rejecting) the assigned orders (l, k) . $y_{ontrip,lk}$ represents the rate of drivers delivering the order l to k . $y_{arrived,j}(l)$ represents the rates of drivers relocating from region j to l . Second, the average sojourn times are defined as follows: τ_l^w represents the idling time spent waiting for order assignment at queue l . τ_l^{en} represents the en-routing time to reach a customer at origin l , τ_{lk}^v denotes the service time for delivering an order from l to k , and τ_{mn}^r is the relocation time from region m to n . The remaining model parameters include: c as the fuel cost per unit time, ϕ as the baseline compensation per unit time, P_{lk} as transition probability of order (l, k) assignment.

The objective function (1) consists of two parts: (i) fuel cost of en-route, delivery, and relocation, (ii) reward of completing the order (l, k) . Constraints (2)–(6) denotes to be the state-transition dynamics of drivers, which can be interpreted as *flow-balance* equation of network-flow models. Constraint (7) ensures the stationary distribution of occupancy measure.

4 Bi-level model

We formulate the bi-level service capacity planning (BSCP) model as follows (an optimistic modeling is assumed):

$$\min_{b_\theta, N, y} \sum_{(l, k) \in \mathcal{A}^h} (\theta_{lk} + \phi \tau_{lk}^v) y_{ontrip, lk} \quad (8)$$

$$s.t. \theta_{lk} = \sum_{j \in \mathcal{J}} \theta_j b_{\theta, lkj}, \sum_{j \in \mathcal{J}} b_{\theta, lkj} = 1 \forall (l, k) \in \mathcal{A}^h, N^L \leq N \leq N^U, \quad (9)$$

$$y_{ontrip, lk} \geq (1 - \alpha_{lk}) \Lambda_{lk} \forall (l, k) \in \mathcal{A}, \quad (10)$$

$$\left(\{y_{idle, l}\}_{l \in \mathcal{V}^h}, \{y_{ontrip, lk}\}_{l \in \mathcal{V}^h, k \in \mathcal{V}} \right) \in S(\theta, N) \quad (11)$$

$$b_{\theta, lkj} \in \{0, 1\}, \forall j = 1, N \in \mathbb{Z}_+, \tau_l^w \geq 0 \quad (12)$$

where $S(\theta, N)$ is the set of optimal solutions of problem

$$\max_{y, \tau_l^w} 0 \quad (13)$$

$$s.t. y_{idle, l} \leq \Lambda_l, (\Lambda_l - y_{idle, l}) \tau_l^w = 0, \tau_l^w \leq \tau^U \forall l \in \mathcal{V}^h, (r^\top y)/N \geq \mu^L \quad (14)$$

$$y \in \arg \max_{\hat{y}} \{(1) : (2) - (7)\} \quad (15)$$

In this study, we assume the compensation levels are finite and we define $\mathcal{J} := \{1, \dots, J\}$ as the index set of candidate compensation level. The decision variable $b_{\theta, lkj}$ represents whether level θ_j is selected for compensation θ_{lk} . N represents the staffing level. The upper-level objective function (8) represents the total labor cost of the platform. Constraint (9) determines the compensation and staffing level in pre-defined sets. (10) is the service-level constraint which ensures that at least $(1 - \alpha_{lk}) \Lambda_{lk}$ demand is covered.

In the lower-level equilibrium, τ_l^w represents the idling time at *driver-equilibrium*. (14)-(15) capture the *driver-equilibrium* condition proposed by [1, 3]. It indicates that idling drivers at region l cannot be assigned in excess of total order ($y_{idle, l} \leq \Lambda_l$), and drivers incur positive waiting time τ_l^w only if demand orders are not lost at all ($(\Lambda_l - y_{idle, l}) \tau_l^w = 0$), i.e., drivers are congested in region l which generates the congestion waiting time $\tau_l^w (> 0)$. And driver's average profit must be greater than or equal to minimum income μ^L ($(r^\top y)/N \geq \mu^L$), where we define r as the objective coefficient vector of (1).

5 Exact solution approach

For brevity, let us define $y_{idle}, y_{\sim idle}$ as the vector of aggregated variables $\{y_{idle, l}\}_{l \in \mathcal{V}^h}$ and remaining y . We let $\tau^{\sim w}$ as the vector of aggregated sojourn time $(\tau^{en}, \tau^v, \tau^r)$ with appropriate dimension without τ^w . We define $r_{idle}, r_{\sim idle}$ as the vector of aggregated driver objective coefficients of $y_{idle}, y_{\sim idle}$ in (1) with appropriate dimensions. And we define $A_{idle}, A_{\sim idle}$

as constraint matrices of constraints (2) and (4)-(6) respectively. We first derive the single-level reformulation based on KKT-condition ([6]) from the fact that the problem (1)-(7) is convex at fixed (θ, N, τ^w) . The equivalent single-level formulation is as follows:

$$\min (8) \text{ s.t. } (2) - (7), (9) - (10), (12), (14), , \quad (16)$$

$$(A_{idle})^\top \eta + \tau^w \mu - z_{idle} = r_{idle}, \quad (A_{\sim idle})^\top \eta + \tau^{\sim w} \mu - z_{\sim idle} = r_{\sim idle}, \quad (17)$$

$$y \leq M_1 b, z \leq M_2(1 - b), z \geq 0, \eta, \mu \text{ free} \quad (18)$$

where η, μ are dual variables of constraints (2)-(6) and (7) and z are slack variables. Note that all variables except (θ, N) are continuous and all constraints and products of variables can be linearized, except continuous bilinear terms $(\tau^w \mu)$ in (17), which makes the model as a mixed-integer bilinear program (MIBP).

The idea of our reformulation is based on *Benders decomposition* (BD), by projecting out τ_l^w into a subproblem. We found out that inner minimization over τ^w reduces to the *continuous knapsack-type* feasibility problem. Further, we found a valid mixed-integer-linear reformulation with additional binary variables and constraints that ensure subproblem feasibility without requiring iterative generation of Benders cuts. Finally, we identified useful super-valid and valid inequalities based on problem-specific insights which can significantly improve the LP relaxation. Motivated by network-flow structure of (2)-(6), one example is to ‘split’ the variable $y_{arrived,j}(i)$ into $y_{arrived,lj}(i) \forall l \in \mathcal{V}^h$, tracking the origin of driver flows. Then we add so-called *strong inequalities*: $y_{arrived,lj}(i) \leq M_1 b_{arrived,j}(i) \forall l \in \mathcal{V}^h$. Computational experiments on small-scale 40 instances showed that the MILP formulation achieved a median runtime of 0.23 seconds versus 7.37 seconds for MIBP formulation ($32\times$ speedup), while also providing much more stable performance with substantially lower runtime variability (0.45s vs. 454.20s standard deviation). Currently, we are investigating a Dantzig-Wolfe decomposition approach to solve large-scale instances in a computationally scalable manner.

References

- [1] Philipp Afeche, Zhe Liu, and Costis Maglaras. Ride-hailing networks with strategic drivers: The impact of platform control capabilities on performance. *Rotman School of Management Working Paper*, (3120544):18–19, 2022.
- [2] Omar Besbes, Francisco Castro, and Ilan Lobel. Spatial capacity planning. *Operations Research*, 70(2):1271–1291, 2022.
- [3] Anton Braverman, Jim G Dai, Xin Liu, and Lei Ying. Empty-car routing in ridesharing systems. *Operations Research*, 67(5):1437–1452, 2019.

- [4] Eugene A Feinberg. Constrained semi-Markov decision processes with average rewards. *Zeitschrift für Operations Research*, 39:257–288, 1994.
- [5] Itai Gurvich, Martin Lariviere, and Antonio Moreno. *Operations in the on-demand economy: Staffing services with self-scheduling capacity*. Springer, 2019.
- [6] Thomas Kleinert, Martine Labb  , Ivana Ljubi  , and Martin Schmidt. A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization*, 9:100007, 2021.
- [7] Yunpeng Li, Costas A Courcoubetis, and Antonis Dimakis. Repositioning, ride-matching, and abandonment in on-demand ride-hailing platforms: A mean field game approach. *arXiv preprint arXiv:2504.02346*, 2025.
- [8] Kenan Zhang, Archak Mittal, Shadi Djavadian, Richard Twumasi-Boakye, and Yu Marco Nie. RIdE-hail vehicle routing (RIVER) as a congestion game. *Transportation Research Part B: Methodological*, 177:102819, 2023.

Mathematics of Gas Mixtures on Networks – Modeling, Optimization, Existence and Uniqueness*

Pascal Börner, Marc E. Pfetsch, Stefan Ulbrich

TU Darmstadt, Germany, {boerner|pfetsch|ulbrich}@mathematik.tu-darmstadt.de

Abstract We consider the mathematics and global optimization of stationary binary gas mixtures on networks. We use a Weymouth-based model adapted for mixtures and extend the existence results of single gas flow to mixtures. To solve the problem we propose a solution algorithm and compare it to a mixed-integer nonlinear program (MINLP) formulation. Additionally we develop a sufficient condition for uniqueness on general networks and prove uniqueness for natural gas hydrogen mixtures at *low* hydrogen concentrations.

Keywords: Gas network optimization, gas mixing, global optimization, MINLP

1. Introduction

The modeling of mixtures of gases is important in practice and has been discussed in the literature as follows. A general framework for the continuum thermodynamics of multiple component mixtures is provided in [3]; see also [7]. In order to accurately capture the changes in the mixtures properties we use the model for binary mixtures presented in [2, 6–1]. The model is based on a mass fraction averaged speed of sound and easy to compute. There is evidence for its accuracy and consistency with thermodynamics [2]. Thus, it is suitable to be used for global optimization on networks. The first result on the uniqueness of the solution in mixtures was

*Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project A01, Project-ID 239904186 – TRR 154.

established by [6], which is restricted to tree shaped networks. Yet, they do not analyze the existence of solutions, which was proven for tree shaped networks by [1]. Moreover, [1] also shows uniqueness of solutions on tree shaped networks with one cycle. We extend the existence result of steady state solutions to general arbitrary networks. Addressing the question of uniqueness, we develop a sufficient condition for the uniqueness of solutions on general networks. For the optimization, we develop an algorithm, which iteratively solves the stationary gas flow problem with mixing on networks. The algorithms performance is compared to a MINLP formulation by computational results. We assume constant temperature (isothermal), a constant compressibility factor, and *perfect mixing*, meaning mixing occurs only at nodes, producing homogeneous gas at the outlets.

2. Preliminaries: Modeling and differences to single gas flow

We model the gas flow in a single pipe with the well known WEYMOUTH equation, which is adapted to the mixture [2]. Networks are modeled as directed graphs $G = (V, A)$, where V denotes the set of vertices and A the set of arcs. Vertices represent junctions of pipes, sources and sinks. Arcs represent pipes or components like valves, compressors and other. On each pipe arc, the gas flow is modeled with (WM), which implies that the variables we use are the mass flow q and the pressures p at the start and end of the pipe. In networks the gases mix at nodes, and by our assumption of perfect mixing, the mixing ratio has to be calculated at each node. For each arc a , we introduce variables μ_a for the mass percentage of constituent 1 in the mixture. The resulting model for the passive network, i.e., without other components, of the stationary gas flow problem with mixing then is:

$$p_u^2 - p_v^2 = \lambda c(\mu_a)^2 q_a |q_a| \quad \forall a = (u, v) \in A, \quad (\text{WM})$$

$$\sum_{a \in \delta^+(v)} q_a - \sum_{a \in \delta^-(v)} q_a = b_v \quad \forall v \in V, \quad (1a)$$

$$c(\mu_a)^2 = c_1^2 \mu_a + (1 - \mu_a) c_2^2 \quad \forall a = (u, v) \in A, \quad (1b)$$

$$\mu(v) = \frac{\sum_{a \in I(v)} \mu_a |q_a| + \mu_v (b_v)_+}{\sum_{a \in I(v)} |q_a| + (b_v)_+} \quad \forall v \in V, \quad (1c)$$

$$\mu_a = \mu(v) \quad \forall a \in O(v), v \in V. \quad (1d)$$

Here λ is a pipe specific resistance coefficient, $I(v)$ and $O(v)$ denote the set of arcs with in- and outflow into node v respectively. Flow directions as well as $I(v)$ and $O(v)$ are modeled with binary variables. We highlighted the dependence on the gas constitution with $c^2(\mu)$, which is the speed of sound in the gas mixture. Following [2], the mixture dependent speed of sound is given by (1b), where c_1 and c_2 denote the speed of sound in constituent 1 and 2 respectively. This implies that we consider binary mixtures. Thus the difference to single gas are immediately clear: The gas flow now also depends on the gas constitution. Furthermore, the larger the difference between c_1 and c_2 the larger the impact of the gas constitution. For an example see [2, Section 4.2]. A natural question is how to solve (WM)–(1d) if not implemented as MINLP. For this we present the following Algorithm:

Algorithm 1 Iterative Objective Update for Mixing

```

 $\mu^0 \leftarrow \mu^{\text{init}}$ 
for  $k = 0, 1, \dots$  do
   $q^k(\mu^k) \leftarrow \text{solution of (WM) ... (1d) with } \mu \text{ fixed to } \mu^k$ 
   $\mu^{k+1} \leftarrow \text{exact } \mu \text{ w.r.t. } q^k(\mu^k)$ 

```

The strength of the algorithm is that the non-linear mixing equation (1c) is *eliminated* and that in case of a passive network, the problem can be solved as strictly convex program [4]. We analyze the algorithm in terms of its convergence and convergence speed (see [2] for details). Furthermore, we compare its performance to a MINLP implementation.

For the passive stationary gas transport problem with mixing (WM)–(1d) we establish the following results:

Theorem 1 ([2]). *For the stationary gas flow problem with mixing, i.e. (WM)–(1d), a solution exists, if no flow and pressure bounds are enforced.*

Theorem 2 ([2]). *Let $(\tilde{q}, \tilde{\pi}, \tilde{\mu})$ and (q, π, μ) be two solutions of (WM)–(1d) with $\text{sgn}(\tilde{q}) = \text{sgn}(q)$. If $\bar{\mu} = A_{+}^T \frac{1}{2}(\tilde{\nu} + \nu)$ is close enough to a constant, which is implied if $\tilde{\nu} = \nu$ at the inflows is close enough to a constant, then $(\tilde{q}, \tilde{\pi}, \tilde{\nu}) = (q, \pi, \nu)$.*

These results are of importance since they provide a deeper understanding of the underlying system dynamics. A practical relevance is for gas network operators. Furthermore we present a convex relaxation of the mixing equation (1c), which is tighter than the single term McCORMICK [8] relaxation, which is claimed to be best in [5]. Finally we present computational results and discuss the time increase when solving (WM)–(1d) instead of single gas flow on networks.

References

- [1] Simone Göttlich Alena Ulke, Michael Schuster. Steady state blended gas flow on networks: Existence and uniqueness of solutions. 2024.
- [2] Pascal Börner, Marc E. Pfetsch, and Stefan Ulbrich. Mathematics of gas mixtures on networks – modeling, optimization, existence and uniqueness. Technical report, 2025.
- [3] Dieter Bothe and Wolfgang Dreyer. Continuum thermodynamics of chemically reacting fluid mixtures. *Acta Mech.*, 226(6):1757–1805, 2015.
- [4] M. Collins, L. Cooper, R. Helgason, J. Kennington, and L. LeBlanc. Solving the pipe network analysis problem using optimization techniques. *Management Science*, 24(7):747–760, 1978.
- [5] Akshay Gupte, Shabbir Ahmed, Santanu S. Dey, and Myun Seok Cheon. Relaxations and discretizations for the pooling problem. *Journal of Global Optimization*, 67(3):631–669, 2017.
- [6] Saif R. Kazi, Kaarthik Sundar, Shriram Srinivasan, and Anatoly Zlotnik. Modeling and optimization of steady flow of natural gas and hydrogen mixtures in pipeline networks. *International Journal of Hydrogen Energy*, 54:14–24, 2024.
- [7] Josef Málek and Ondřej Souček. A thermodynamic framework for heat-conducting flows of mixtures of two interacting fluids. *ZAMM Z. Angew. Math. Mech.*, 102(11):Paper No. e202100389, 27, 2022.
- [8] Garth P. McCormick. Computability of global solutions to factorable nonconvex programs: Part i — convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

Gradient-free optimization via integration

Christophe Andrieu¹, Ettore Fincato ¹, Nicolas Chopin ², Mathieu Gerber¹

¹*University of Bristol,* ettore.fincato@bristol.ac.uk

²*ENSAE, Institut Polytechnique de Paris*

Abstract We develop and analyze a novel approach to optimize functions $\ell: \mathbb{R}^d \rightarrow \mathbb{R}$ not assumed to be convex, differentiable or even continuous. The idea is to fit recursively ℓ to distributions from an exponential family, using a Bayesian update followed by a reprojection back onto the chosen family. Remarkably, reprojection boils down to computing expectations, which can be simply approximated through Monte Carlo. We show that this approach can be interpreted as a time-inhomogeneous gradient descent algorithm on a sequence of smoothed approximations of ℓ . We also establish new results for inhomogeneous gradient descent algorithms of independent interest. We illustrate performance of the algorithm on a challenging classification task in machine learning. This work is based on a manuscript currently under review at a leading journal in probability and statistics.

Keywords: Gradient-Free Optimisation, Bayesian Updating, Variational methods, Smoothing.

1. Introduction

1.1 Motivation

Let $\ell: \mathcal{X} := \mathbb{R}^d \rightarrow \mathbb{R}$ be a lower-semicontinuous, potentially non-differentiable function such that $\inf_{x \in \mathbb{R}^d} \ell(x) > -\infty$ and hence $\arg \min_{x \in K} \ell(x) \neq \emptyset$ for any compact set $K \subset \mathcal{X}$. This work is concerned with gradient-free algorithms to minimize functions within this class, when ℓ can be evaluated pointwise.

The main idea behind the class of algorithms we present is as follows. Let ϕ_d be the density of the standard normal distribution $\mathcal{N}(0, \mathbf{I}_d)$ and define

$$\pi_{\theta, \gamma}(x) := \frac{1}{\gamma^{d/2}} \phi_d \left(\frac{1}{\sqrt{\gamma}} (x - \theta) \right) \quad (1)$$

for $\theta \in \Theta := \mathbb{R}^d$, $\gamma > 0$. Then, for a sequence $\{\gamma_n \geq 0, n \in \mathbb{N}\}$ such that $\gamma_n \downarrow 0$ define sequentially the families of distributions $\{\pi_n, n \in \mathbb{N}\}$ and $\{\tilde{\pi}_n, n \in \mathbb{N}\}$ following the Algorithm below. We refer to [2].

1. **Input:** $x \mapsto \ell(x), \theta_0, \{\gamma_k; k \geq 0\}, \pi_0(x) = \pi_{\theta_0, \gamma_0}(x)$

2. **while** $n \geq 0$, **do**

- (a) $\tilde{\pi}_{n+1}(x) \propto \exp(-\ell(x)) \pi_n(x)$
- (b) $\theta_{n+1} \in \operatorname{argmin}_{\theta \in \Theta} \text{KL}(\tilde{\pi}_{n+1}, \pi_{\theta, \gamma_n})$
- (c) $\pi_{n+1} = \pi_{\theta_n, \gamma_{n+1}}$

end while

3. **Output:** sequence of distributions $\tilde{\pi}_n, \pi_n$ and parameters θ_n .

An iteration of the Algorithm above therefore consists of the application of Bayes' rule, where ℓ plays the role of a negative log-likelihood and π_n that of the prior distribution, followed by a "projection" onto the normal family $\pi_{\theta, \gamma_{n+1}}$, using the Kullback-Leibler divergence as a criterion.

In practice, one may approximate the ideal Algorithm by replacing $\tilde{\pi}_n$ with a cloud of $N \in \mathbb{N}$ weighted random samples $\{X_n^i, i \in [N]\}$ propagated along the iterations. The current work focuses on the ideal Algorithm, corresponding to the scenario $N \rightarrow \infty$. We see the study of such ideal algorithms as a prerequisite to the study of their implementable versions, seen as perturbations of the ideal algorithms.

1.2 Connection with gradient descent

In a standard statistical context, repeated application of Bayes rule is known to lead to a concentration phenomenon around particular maximum points of the posteriors, under general conditions [1]; a similar phenomenon occurs in the present setup and is illustrated in Figure 1. The update considered here differs from standard Bayesian updating in that it involves a reprojection step, therefore necessitating a new approach to establishing ability of the algorithm to find minima of $\ell(\cdot)$. This reprojection step is motivated by practical considerations: it circumvents the need to propagate the sequence of distributions obtained by repeated use of Bayes' update and greatly facilitates implementation (as we elaborate below). It turns out that this reprojection step also facilitates theoretical analysis. The crucial observation allowing us to prove convergence of the Algorithm, in the sense that π_n concentrates on local minima of $\ell(\cdot)$, is that it implicitly implements a steepest descent algorithm tracking the minima of a sequence of differentiable approximations $\{\ell_n: \mathbb{R}^d \rightarrow \mathbb{R}, n \geq 1\}$ of $\ell(\cdot)$.

When such approximations converge to ℓ , validity of the procedure should ensue. More precisely, the reprojection step can be shown to correspond to so-called moment matching, a fact extensively used in variational inference [3]. Taking into account that in the present setup θ_{n+1} is the first order moment of π_{n+1} , or mean, moment matching takes the form

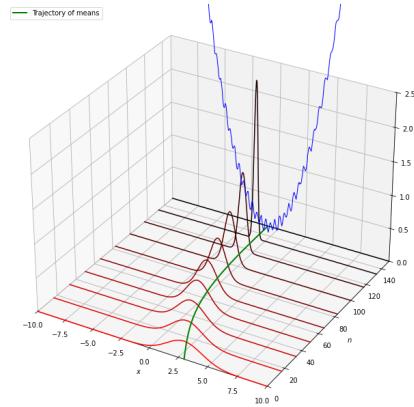


Figure 1: Illustration of the algorithm in the Gaussian case, when $\ell(x)$ (blue line in the background) is a nondifferentiable function. The light red curve is the initial Gaussian density ($n = 0$). The red to black curves are the Gaussian densities π_n , which are recursively obtained by KL minimisation. (The mean of these Gaussian densities is plotted in green.) We observe that these Gaussian densities are progressively attracted to low values of ℓ .

$$\theta_{n+1} = \int x \tilde{\pi}_{n+1}(x) dx = \theta_n + \frac{\int (x - \theta_n) \exp\{-\ell(x)\} \phi\left(\frac{x-\theta_n}{\sqrt{\gamma_n}}\right) dx}{\int \exp\{-\ell(x)\} \phi\left(\frac{x-\theta_n}{\sqrt{\gamma_n}}\right) dx}. \quad (2)$$

It is the evaluation of these expectations which in practice requires a Monte Carlo approximation with weighted samples. For $(\theta, \gamma) \in \Theta \times \mathbb{R}_+$ let

$$\ell_\gamma(\theta) := -\log \left(\int \exp(-\ell(x)) \phi\left(\frac{x-\theta}{\sqrt{\gamma}}\right) dx \right),$$

and for $n \in \mathbb{N}$ and $\theta \in \Theta$ let $\ell_n(\theta) := \ell_{\gamma_n}(\theta)$. Then one can write (2) in the familiar form

$$\theta_{n+1} = \theta_n - \gamma_n \nabla \ell_n(\theta_n), \quad (3)$$

and recognize a time inhomogeneous steepest-descent algorithm tracking the sequence of stationary points of the sequence of functions $\{\ell_n, n \in \mathbb{N}\}$, again smoothed versions of $\ell(\cdot)$. It is remarkable that while this interpretation provides us with an additional rational for the Algorithm and a route to establishing its convergence for a large class of non-differentiable functions $\ell(\cdot)$, implementation does not require differentiation but instead integration.

About convergence, the most stringent assumption we need on $\ell(\cdot)$ is simply that its ‘jumps’ are bounded and that its variations are at most quadratic for large increments of θ . This condition is always satisfied for $\ell(\cdot)$ bounded. The manuscript is under review by Bernoulli.

The algorithm can be easily generalised to target stochastic loss functions $\ell : \mathbb{R}^d \times U \rightarrow \mathbb{R}$, with a given probability distribution P on a space U . Convergence of the ideal algorithm in this case is studied via a stochastic inhomogeneous descent.

References

- [1] Kleijn, B. and van der Vaart, A. (2012). The Bernstein-Von-Mises theorem under misspecification. *Electronic Journal of Statistics*, 6(none):354 – 381.
- [2] Andrieu, C., Chopin, N., Fincato, E., and Gerber, M. (2024). Gradient-free optimization via integration
- [3] Wainwright, M. and Jordan, M. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305.

Session 11





On optimization over a polyhedral set and Augmented Lagrangians*

Eligius M. T. Hendrix,¹ Pablo Guerrero-García,¹ Ana Maria A. C. Rocha²

¹*Universidad de Málaga, Spain, {eligijs, pguerrero}@uma.es*

²*University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal, arocha@dps.uminho.pt*

Abstract Linear and box constraints, which have a long tradition in mathematical optimization, imply that the feasible region of a problem is a polyhedral set, i.e., it can be described as a polytope. We will focus on the so-called Augmented Lagrangian approach where constraints are captured in Lagrangian terms and penalties. We show some first numerical analysis with the easiest case of having a linear objective.

Keywords: Polytope, Polyhedral set, Augmented Lagrangian, Linear Program

1. Introduction

Using linear constraints is a powerful modeling tool to formulate practical optimization problems. One can think of simple constraints, such as a symmetry-breaking constraint defining $x_2 \geq x_1$ or blending constraint $\sum_i x_i = 1$, [5]. Equality constraints such as the latter, introduce additional challenges as they lead to a feasible set that is not full dimensional..

Exact methods in Global Optimization focus usually on low dimensional problems that are box constrained. Adding linear constraints will then lead to a polytope feasible set \mathcal{P} . There are two ways to represent \mathcal{P} .

*This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Unit Project of ALGORITMI Centre

The first one is as the convex hull $\text{conv}(\mathcal{V})$ of its vertex set $\mathcal{V} := \{v_1, \dots, v_q\}$;

$$\mathcal{P} := \{x \in \mathbb{R}^n \mid \sum_{j=1}^q \lambda_j v_j, \sum_{j=1}^q \lambda_j = 1, \lambda_j \geq 0, j = 1, \dots, q\}. \quad (1)$$

This notation allows us to describe the relative interior and face graph; a global optimum can be found in the relative interior of \mathcal{P} or on one of its faces. The founding handbook [6] starts by focusing on concave optimization. This implies that minimum points are attained at the vertices. Algorithmically, this means we can go for a so-called vertex enumeration.

A provided feasible set \mathcal{P} is usually represented by a set of inequalities, i.e. the intersection of half spaces.

$$\mathcal{P} := \{x \in \mathbb{R}^n \mid a_j x \leq b_j, j = 1, \dots, m\}. \quad (2)$$

In this context, the term polyhedral set is more appropriate. \mathcal{P} could have directions d_k for which it is unbounded; the term $\sum_k \mu_k d_k, \mu_k \geq 0$ is added to (1). For box constrained GO, \mathcal{P} is bounded and with that a polytope.

In our recent investigation [1], we compared traditional Interval Arithmetic B&B, that use box shaped partition sets (iBB, [4]) with simplicial partition set B&B (sBB). The latter ones are more effective for low-dimensional polytope feasible set problems, where the minimum is not interior. However, they suffer from slower convergence due to volume reduction. In our finding, the bad behavior of iBB has to do with working with decimal accuracies. Even having exact representations of vertices V or constraint matrix A and vector b , e.g. $x_1 - x_2 \geq 0$, standard routines like `scipy.spatial` to transform (1) into (2) and vice versa, will normalize representation introducing a non-exact representation including $\sqrt{2}$. We found that this makes it harder for exact methods to verify feasibility of simple linear inequalities.

2. Augmented Lagrangians

We investigate the question whether it is possible to obtain exact algorithms using Lagrangian theory. In [9] the use of Augmented Lagrangians (AL) to solve generic constrained GO was investigated. The handbook [7] provides an introduction to the concepts of nonlinear optimization including AL. Consider a NLP description with objective function $g(y)$ and set of equalities $s(y) = 0$ which we will call the dual (D): $\min_{y \geq 0} g(y), s_i(y) = 0, i = 1, \dots, n$. Let x be an estimate of the dual solution (so a solution of the primal) and ρ a penalty parameter. A quadratic penalty based AL objective

with respect to the equality constraints is

$$\mathcal{L}_D(y; x, \rho) := g(y) - x^T s(y) + \frac{\rho}{2} s(y)^T s(y). \quad (3)$$

Using nonnegativity restrictions, a basic algorithm can be based on minimizing $\min_{y \geq 0} \mathcal{L}_D(y; x_k, \rho_k)$ slowly increasing the penalty parameter ρ_k and updating the dual estimate according to $x_{k+1} = x_k - \rho_k s(y_k)$.

We initially tested this approach on a collection of sparse LP problems (details in [3]), as we wanted to investigate performance on a set of comparatively "easy" optimization problems. Consider the dual $\min_{y \geq 0} b^T y, Ay = c$ in [3]. To find a basic feasible dual solution y , one of the used algorithms Clp in the open source COIN-OR software is the so-called "idiot crash" algorithm (IC). It has been outlined in [2]. The basic iterate direction d_k in the primal space relates to the AL (3) of the dual:

$$y_{k+1} = \operatorname{argmin}_{y \geq 0} \left(-b^T y - d_k^T (c - Ay) + \frac{\rho_k}{2} (c - Ay)^T (c - Ay) \right). \quad (4)$$

We found that surprisingly IC decreases the penalty parameter. A much older AL algorithm (in Russian) is due to [8] (PT), which increases the penalty. We investigated what happens if we add to PT a first stage solving (3) with $b = 0$, called AFS (added first stage). This usually does not provide a basic solution, but generates a certificate of infeasibility of the dual (unboundedness of the primal) in case no feasible solution exists.

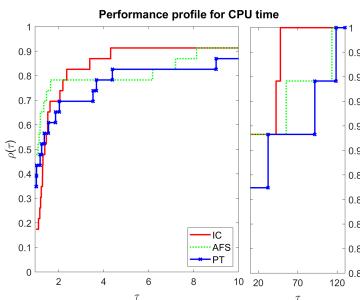


Figure 1: Comparing IC (red), PT (blue) and AFS (green) in CPU.

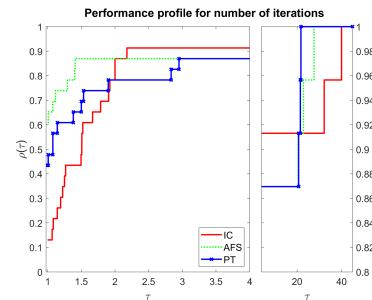


Figure 2: Comparing IC (red), PT (blue), AFS (green) in iterations.

A first sketch of numerical results based on 28 netlib problems is given in Figures 1 and 2. It illustrates that the older PT algorithm is most of the

time better than the IC algorithm and that our suggestion to add a first stage (AFS) seems to make sense.

3. Summary

The target of the workshop presentation is to sketch findings with respect to optimization over polyhedral sets. On one hand sketching challenges for B&B based methods and on the other hand questions on applying AL.

References

- [1] B. G.-Tóth, E.M.T. Hendrix, L.G. Casado, and F. Messine. On dealing with minima at the border of a simplicial feasible area in simplicial branch and bound. *Journal of Optimization Theory and Applications*, 203:1794–1819, 2024.
- [2] Ivet L. Galabova and Julian A. J. Hall. The ‘idiot’ crash quadratic penalty algorithm for linear programming and its application to linearizations of quadratic assignment problems. *Optimization Methods and Software*, 35(3):488–501, 2020.
- [3] Pablo Guerrero-García and Eligius M. T. Hendrix. Experiments with active-set LP algorithms allowing basis deficiency. *Computers*, 12(3):1–23, 2023.
- [4] Eldon R. Hansen and G. William Walster. *Global Optimization Using Interval Analysis*. Marcel Dekker Inc., New York, 2nd edition, 2004.
- [5] Eligius M.T. Hendrix, Leocadio G. Casado, and Inmaculada García. The semi-continuous quadratic mixture design problem: Description and branch-and-bound approach. *European Journal of Operational Research*, 191(3):803–815, 2008.
- [6] Reiner Horst and Herbert Tuy. *Global Optimization (Deterministic Approaches)*. Springer Berlin Heidelberg, 1990.
- [7] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, 2e edition, 2006.
- [8] Boris T. Polyak and Nikolai V. Tretjakov. An iterative method of linear programming and its economic interpretation. *Ekon. Mat. Metody*, 8(5):740–751, 1972. in Russian.
- [9] Ana Maria A.C. Rocha and Edite M.G.P. Fernandes. Numerical study of Augmented Lagrangian algorithms for constrained global optimization. *Optimization*, 60(10-11):1359–1378, 2011.

Anomalies of the Scholtes regularization for mathematical programs with complementarity constraints

Sebastian Lämmel,¹ Vladimir Shikhman,¹

¹*Chemnitz University of Technology, Faculty of Mathematics, Reichenhainer Str. 41, 09126 Chemnitz, Germany, vladimir.shikhman@mathematik.tu-chemnitz.de*

Abstract For mathematical programs with complementarity constraints (MPCC), we refine the convergence analysis of the Scholtes regularization. Our goal is to relate nondegenerate C-stationary points of MPCC with nondegenerate Karush-Kuhn-Tucker points of its Scholtes regularization. We detected the following anomalies: (i) in a neighborhood of a nondegenerate C-stationary point there could be degenerate Karush-Kuhn-Tucker points of the Scholtes regularization; (ii) even if non-degenerate, they might be locally non-unique; (iii) if nevertheless unique, their quadratic index potentially differs from the C-index of the C-stationary point under consideration. Thus, a change of the topological type for Karush-Kuhn-Tucker points of the Scholtes regularization is possible. In particular, a nondegenerate minimizer of MPCC might be approximated by saddle points. In order to bypass the mentioned anomalies, an additional generic condition for nondegenerate C-stationary points of MPCC is identified. Then, we uniquely trace nondegenerate Karush-Kuhn-Tucker points of the Scholtes regularization and successively maintain their topological type.

Keywords: mathematical programs with complementarity constraints, Scholtes regularization method, nondegeneracy, non-uniqueness, index shift, topological type

1. Introduction

We consider mathematical programs with complementarity constraints:

$$\text{MPCC : } \min_x f(x) \quad \text{s. t.} \quad x \in M$$

with

$$M = \{x \in \mathbb{R}^n \mid F_{1,j}(x) \cdot F_{2,j}(x) = 0, F_{1,j}(x) \geq 0, F_{2,j}(x) \geq 0, j = 1, \dots, \kappa\},$$

where the defining functions $f \in C^2(\mathbb{R}^n)$, $F_1, F_2 \in C^2(\mathbb{R}^n)$ are twice continuously differentiable. MPCCs have been widely studied in the literature for several decades, see e.g. [3]. They naturally appear if dealing with generalized Nash equilibrium problems, bilevel optimization, and semi-infinite programming to name a few applications.

The main difficulty of dealing with MPCCs is that their feasible set has kink-like structure. This nonsmoothness of the MPCC feasible set motivates the introduction of regularization methods. In the seminal paper [7], such a regularization was suggested first. According to the latter, the Scholtes regularization of MPCC is given as follows:

$$\mathcal{S}_t : \quad \min_x f(x) \quad \text{s. t.} \quad x \in M^{\mathcal{S}}$$

with

$$M^{\mathcal{S}} = \{x \in \mathbb{R}^n \mid F_{1,j}(x) \cdot F_{2,j}(x) \leq t, F_{1,j}(x) \geq 0, F_{2,j}(x) \geq 0, j = 1, \dots, \kappa\},$$

where the parameter $t > 0$ is positive and is suppressed in the notation for the readers' convenience.

Obviously, the Scholtes regularization \mathcal{S}_t falls into the scope of nonlinear programming (NLP). Under suitable assumptions, two types of results – on its convergence and well-posedness – have been proven there and further in [6, 1]: (1) the Karush-Kuhn-Tucker points of the Scholtes regularization converge to a C-stationary point of MPCC; (2) such a sequence of Karush-Kuhn-Tucker points exists in a neighborhood of a C-stationary point. For numerical aspects of the Scholtes regularization we refer to [4, 5].

In this paper, we refine the convergence analysis of the Scholtes regularization performed in the literature so far. Our goal is to relate nondegenerate C-stationary points of MPCC with nondegenerate Karush-Kuhn-Tucker points of the corresponding Scholtes regularization up to their topological type. To put it simply, nondegeneracy refers to the validity of tailored constraint qualifications, non-vanishing of certain multipliers, and

second-order regularity. In the framework of NLP, the topological type of a nondegenerate Karush-Kuhn-Tucker point is characterized by its quadratic index QI , i.e. the number of negative eigenvalues of the restricted Hessian matrix of the Lagrange function, see e.g. [2]. For a nondegenerate C-stationary point of MPCC, the number of negative pairs of biactive multipliers is additionally involved. This biactive part BI , together with the quadratic part QI as before, yields the C-index CI . Again, it can be unambiguously deduced from the C-index if the corresponding C-stationary point is a minimizer, maximizer, or a saddle of a certain type.

2. Main results

Our approach to the study of convergence and well-posedness properties of the Scholtes regularization reveals however several anomalies:

- (i) Nondegeneracy of a sequence of Karush-Kuhn-Tucker points is not necessarily preserved in the limiting point, i.e. the obtained C-stationary point of MPCC might be degenerate. Vice versa, it can not be assured to find nondegenerate Karush-Kuhn-Tucker points of the Scholtes regularization in any arbitrarily small neighborhood of a nondegenerate C-stationary point of MPCC.
- (ii) The Scholtes regularization might cause bifurcation, i.e. in a neighborhood of a C-stationary point of MPCC there are potentially multiple Karush-Kuhn-Tucker points of the former. This includes cases where they form a continuum or the C-stationary point under consideration is a minimizer of MPCC.
- (iii) The topological type might change while regularizing, which follows from the index shift phenomenon. The latter means that the C-index of the limiting C-stationary point might decrease if compared to the quadratic index of the approximating Karush-Kuhn-Tucker points. E.g., saddle points of the Scholtes regularization may converge to a nondegenerate minimizer of MPCC.

In order to avoid the mentioned anomalies (i)-(iii), an additional generic condition, called by us NDC4, for nondegenerate C-stationary points of MPCC is identified. It requires that the multipliers associated with nonbiactive constraints do not vanish either. By additionally assuming this, we uniquely trace then nondegenerate Karush-Kuhn-Tucker points of the Scholtes regularization and successively maintain their topological type in comparison with the limiting C-stationary point of MPCC.

Theorem 1 shows that it may come to a reduction of the C-index if compared with the quadratic index of the approximating Karush-Kuhn-Tucker points. In other words, the convergence behavior of the Scholtes regularization is anomalous in the sense that the topological type of the limiting C-stationary point of MPCC may change.

Theorem 1 (Convergence). *Suppose a sequence of nondegenerate Karush-Kuhn-Tucker points $x^t \in M^S$ of \mathcal{S}_t with quadratic index q converges to \bar{x} for $t \rightarrow 0$. If $\bar{x} \in M$ is a nondegenerate C-stationary point of MPCC with multipliers $(\bar{\sigma}, \bar{\varrho})$, then we have for its C-index:*

$$\max \{ q - |\{j \in a_{01}(\bar{x}) \mid \bar{\sigma}_{1,j} = 0\}| - |\{j \in a_{10}(\bar{x}) \mid \bar{\sigma}_{2,j} = 0\}|, 0 \} \leq CI \leq q.$$

If additionally NDC4 holds at \bar{x} , then the indices coincide, i.e. $CI = q$.

Next, we turn our attention to the well-posedness of Scholtes relaxation in the vicinity of C-stationary points of the corresponding MPCC. Also here the validity of NDC4 becomes crucial.

Theorem 2 (Well-posedness). *Let $\bar{x} \in M$ be a nondegenerate C-stationary point of MPCC with C-index c , additionally, fulfilling NDC4. Then, for all sufficiently small t there exists a nondegenerate Karush-Kuhn-Tucker point $x^t \in M^S$ of \mathcal{S}_t within a neighborhood of \bar{x} , which has the same quadratic index c . Moreover, for any fixed t sufficiently small, such x^t is the unique Karush-Kuhn-Tucker point of \mathcal{S}_t in a sufficiently small neighborhood of \bar{x} .*

The appearance of anomalies mentioned in (i)-(iii) will be illustrated by numerical examples.

References

- [1] Bouza G. and Still G. (2007). Mathematical programs with complementarity constraints: convergence properties of a smoothing method. *Mathematics of Operations Research*. 32. 467-483.
- [2] Jongen H. Th., Jonker P. and Twilt F. (2000). *Nonlinear Optimization in Finite Dimensions*. Kluwer Academic Publishers, Dordrecht.
- [3] Facchinei F. and Pang J.-S. (2003). *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer New York, NY.
- [4] Hoheisel T., C. Kanzow and A. Schwartz (2013). Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints. *Mathematical Programming*. 137. 257-288.
- [5] Kanzow C. and Schwartz A. (2015). The Price of Inexactness: Convergence Properties of Relaxation Methods for Mathematical Programs with Comple-

- mentarity Constraints Revisited. *Mathematics of Operations Research.* 40. 253-275.
- [6] Ralph D. and Wright S. J. (2004). Some properties of regularization and penalization schemes for MPECs. *Optimization Methods and Software.* 19. 527-556.
- [7] Scholtes S. (2001). Convergence properties of a regularization scheme for mathematical programs with complementarity constraints. *SIAM Journal on Optimization.* 11. 918-936.

The Story of Hansen’s Octagon^{*}

Charles Audet¹, Frédéric Messine²

¹GERAD, École Polytechnique de Montréal, Québec, Canada, Charles.Audet@gerad.ca

²LAPLACE, University of Toulouse, Toulouse, France, messine@laplace.univ-tlse.fr

Abstract This paper tells the story of the discovery of *Hansen’s Octagon*, the solution to the problem of the *largest small octagon*—the convex octagon of maximal area under the constraint that the distance between any pair of vertices does not exceed one. Such polygons, called *small polygons*, were introduced by Karl Reinhardt in 1922. While Reinhardt solved the case for odd n , even values remained open. In 1975, Ron Graham solved the hexagon case. Inspired by this, Pierre Hansen proposed using deterministic global optimization to tackle the octagon, leading to a collaborative success more than a decade later. Hansen’s Octagon exemplifies the power of computational methods in modern geometry. Ultimately, Pierre Hansen’s key idea was that when problems, even seemingly simple ones in planar geometry, remained unsolved, computers—and particularly deterministic global optimization methods—could supplement the human mind.

Keywords: Small Polygons, Octagon, Numerical Proof, Reliable Computations

1. Historical Background

The study of small polygons began with Karl Reinhardt in 1922 [8]. He considered the problem of finding, for a given number of vertices n , the convex small polygon that maximizes area. He showed that for odd n , the regular polygon is optimal. For even n , however, the problem remains more nuanced.

*Work of the first author was supported by NSERC grant RGPIN-2020-04448.

The case $n = 4$ (quadrilateral) was solved by Reinhardt, who showed that the square is one of the optimal solution.

For $n = 6$, the problem is more complex, and remained open until Ron Graham [6] solved it in 1975 through the use of diameter graphs. The nodes of the graphs are the vertices of the polygon, and the edges are the unit-length diagonals. His hexagon possesses an axis of symmetry, and exceeds the regular hexagon's area by nearly 4%.

2. From Hexagon to Octagon

Graham conjectured that when $n \geq 4$ is even, the largest small polygon diameter graph is composed of a cycle with $n - 1$ vertices together with an additional edge attached to a vertex of the cycle. The conjecture was formally shown by Foster and Szabo [5] for arbitrary polygons.

It was during a talk by Ron Graham in the late 1980s, in which he discussed his famous hexagon, that Pierre Hansen got the idea to use deterministic global optimization methods to tackle the more complex case of the octagon ($n = 8$) using deterministic global optimization methods. He shared this idea with Graham, who encouraged the effort despite its difficulty.

With his team [2], Hansen applied Graham strategy to decompose the problem into 31 possible configurations based on the analysis of diameter graphs. Most of the configurations were shown to be suboptimal using elementary geometrical arguments. The remaining ones were modeled as non-convex quadratically constrained optimization problems with a non-convex objective function. A state-of-the-art (at the time) branch-and-cut algorithm [1] found the optimal area up to four decimals through computations that exceeded 100 hours. A few years later, a group of researchers applied an efficient semidefinite programming approach [7] to certify 7 digits of accuracy, in only 5 seconds of computational time. They also pushed the computations to solve the decagon and the dodecagon instances, with computational times of 1 and 25 minutes.

The square (and many other optimal quadrilateral) as well as Graham's octagon have an axis of symmetry, through the pending edge of the diameter graph. The numerical solutions suggest that Hansen's Octagon also has an axis of symmetry. Assuming this to be true, reduces the dimensionality of the search space. Using symbolic computations on a complex-plane representation led to a single-variable root-finding problem, where the area A of the symmetric octagon is the unique root in $[0.7, \frac{\pi}{4}]$ of a degree-42 polynomial with large integer coefficients [3]:

$$\begin{aligned}
 p_8(A) = & 147573952589676412928 A^{42} - 442721857769029238784 A^{41} + \dots \\
 & + 3773041038347596515021000956 A - 478425365462547737405343343
 \end{aligned}$$

In 2010, the Global Optimization Workshop TOGO10 [4] brought together researchers to discuss the state-of-the-art in global optimization, covering theoretical, algorithmic, and application-oriented topics. A central theme of the workshop was the 70th birthday celebration of Pierre Hansen. A hand-made whisky glass in the shape of Hansen's Octagon (that maximizes volume) was offered to all participants. The area of this axially symmetrical octagon is within the error bounds proposed in [7], which suggests that Hansen's Octagon has an axis of symmetry. Figure 1 displays the regular octagon together with Hansen's Octagon and the whisky glass.

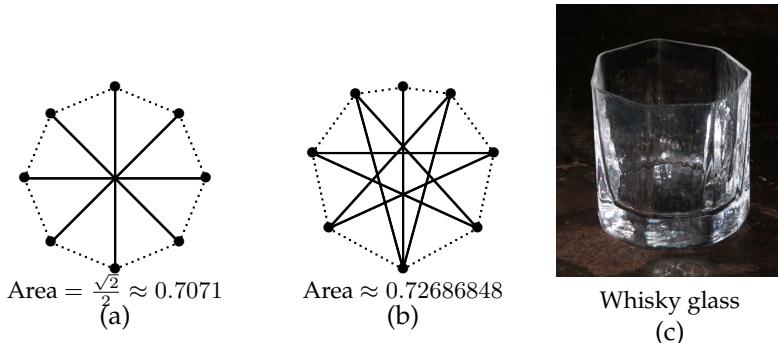


Figure 1: Diameter graph of (a) the regular octagon and (b) Hansen's Octagon, and (c) the whisky glass of TOGO10

3. Discussion

Hansen's Octagon and its associated variants highlights the rich interplay between classical geometry, combinatorial structure, and modern computational optimization. What began as a purely theoretical problem introduced by Reinhardt a century ago has evolved into a fruitful research area where advanced algorithmic tools reveal subtle and surprising optimal shapes.

This journey demonstrates how long-standing questions in planar geometry –such as maximizing area, perimeter, or width under distance or length constraints– can be resolved through global optimization techniques like branch-and-bound, semidefinite programming, and symbolic computation. Ultimately, Hansen’s Octagon serves as a compelling example of how modern optimization can provide rigorous solutions to deep, deceptively simple geometric problems.

Acknowledgments

The authors wish to thank Pierre Hansen for all the good times spent together searching for small octagons.

References

- [1] C. Audet, P. Hansen, B. Jaumard, and G. Savard. A branch and cut algorithm for nonconvex quadratically constrained quadratic programming. *Mathematical Programming, Series A*, 87(1):131–152, 2000.
- [2] C. Audet, P. Hansen, F. Messine, and J. Xiong. The largest small octagon. *J. Combin. Theory Ser. A*, 98(1):46–59, 2002.
- [3] C. Audet, P. Hansen, and D. Svrtan. Using symbolic calculations to determine largest small polygons. *Journal of Global Optimization*, 81(1):261–268, 2021.
- [4] S. Cafieri, B. G.-Tóth, E.M.T. Hendrix, L. Liberti, and F. Messine. Proceedings of the Toulouse Global Optimization Workshop, page 166, 2010.
- [5] J. Foster and T. Szabo. Diameter graphs of polygons and the proof of a conjecture of graham. *Journal of Combinatorial Theory, Series A*, 114(8):1515–1525, 2007.
- [6] R.L. Graham. The largest small hexagon. *Journal of Combinatorial Theory, Series A*, 18:165–170, 1975.
- [7] D. Henrion and F. Messine. Finding largest small polygons with GloptiPoly. *Journal of Global Optimization*, 56(3):1017–1028, 2013.
- [8] K. Reinhardt. Extremale polygone gegebenen durchmessers. *Jahresber. Deutsch. Math. Verein*, 31:251–270, 1922.

Furthest Point Problem: Solution Algorithm and Applications

Roozbeh Abolpour,¹ Mohammad Reza Hesamzadeh,²

¹*Technical University of Darmstadt, Darmstadt, Germany,* roozbeh.abolpour@eins.tu-darmstadt.de

²*KTH Royal Institute of Technology, Stockholm, Sweden,* mrhesa@kth.se

Abstract This paper first presents the furthest point problem for a polyhedron based on the l_2 -Norm. Then two solution algorithms are proposed to solve this nonconvex optimization problem. The properties of our proposed solution algorithms are theoretically discussed. Three applications of the furthest point problem which can be solved using our proposed solution algorithms are presented and theoretically discussed. Supporting numerical results are provided demonstrating the promising performance of our proposed approaches.

Keywords: Furthest point, Nonconvex optimization, Feasibility problem

1. Introduction

The Furthest Point Problem for a polyhedron based on the l_2 -norm (Euclidean distance) is a classical geometric optimization problem with both theoretical interest and practical significance. It involves finding the point within a polyhedron that is located farthest (measured based on Euclidean distance) from a given reference point, often the origin. This problem has interesting geometrical properties and arises in various applications such as computational geometry, facility location planning, energy markets, and data analysis. Mathematically, it entails maximizing a convex function (the squared distance) over a convex feasible region, making it a nonconvex optimization problem. This nonconvexity introduces computational chal-

lenges, especially in high-dimensional settings, where advanced algorithms are required to reliably identify global optima. Moreover, the problem is closely related to other complex optimization and feasibility problems, including Mixed-Integer Linear Optimization Problems (MILPs), Nonconvex Quadratically Constrained Quadratic Optimization Problems (QCQPs), and also the Linear Complementarity Problems (LCPs). As such, solving the Furthest Point Problem not only addresses a fundamental geometric question but also provides valuable insights into broader classes of optimization and feasibility problems.

2. Furthest Point Problem

The Furthest Point Problem [1] for a polyhedron based on the l_2 -Norm is defined as $FP(A, b) : \max_{x \in \mathbb{R}^n} \{\|x\|^2 \mid Ax \leq b\}$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Figure 1 presents an example where red point is the farthest from the origin. However, local optimization solvers converging to a boundary point might fail to find the furthest point, as local gradient information becomes uninformative or unreliable near flat faces or sharp corners of the polyhedron. Hence, exact solution algorithms (including ours) for this problem will have computational complexity $O(\binom{m}{n})$ in the worst case. This complexity arises from the need to check all possible corner points combinations of the polyhedron, as the furthest point may reside at any corner point.

3. Solution Algorithms

Cutting-Plane-Based Iterative Linearization Method (CLM): In the CLM, the objective function of $FP(A, b)$ is linearized. Let $L(u) \in \arg \max_x \{u^T x \mid Ax \leq b\}$ in which $u \in \mathbb{R}^n$ is the parameter vector. The CLM starts with an initial point x_0 and iteratively updates the current point x_k such that $x_k = L(x_{k-1})$. This iterative process continues until the convergence.

Theorem 1. Assume $x_0 \in \mathbb{R}^n$ is arbitrarily selected. Then, the CLM generates a sequence that finitely converges to a corner point of polyhedron $Ax \leq b$.

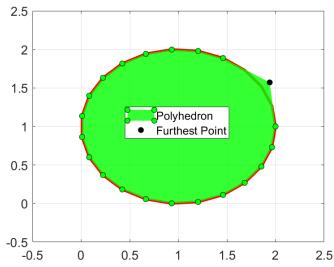


Figure 1: A polyhedron.

The CLM finds a group of corner points until finally reaching equilibrium point \bar{x} satisfying $\bar{x} = L(\bar{x})$. At this point, the CLM is in a pitfall, and to escape it, a cutting plane (a linear inequality) will be added to the problem. As we theoretically prove, this cutting plane will remove the specific equilibrium point while preserving the furthest point. The CLM procedure will continue until the convergence is guaranteed by Theorem 2.

Theorem 2. *The CLM asymptotically converges to either the furthest point or a non-corner point of $P(A, b)$.*

Theorem 2 states that the CLM has a potential to find the furthest point. **Connected Vertices Method (CVM):** The CVM returns the connected vertices to a specific corner point of polyhedron $Ax \leq b$. Let $J(v)$ be the set of indices corresponding to the active constraints at the corner point v . Then, the CVM will find all connected vertices to v as follows for all $J \subset J(v)$, wherein $|J| = n - 1$.

$$w_J = v + t_J n_J : n_J = \text{Null}(A_J), t_J = \min_{t \in \mathbb{R}} \{|t| \mid A(x + tn_J) \leq b\}, \quad (1)$$

where (A_J, b_J) represents the linear inequalities in $Ax \leq b$ corresponding to row index set J .

Our proposed algorithm exploits both CLM and CVM to find the furthest point. The algorithm begins by applying CLM to find a group of corner points (e.g., v_1, \dots, v_n) on the polyhedron. Next, the CVM is used to identify a connected corner point w_1 adjacent to one of these, such as v_1 . Starting from w_1 , the CLM is applied again to explore a new group of corner points. Our new method alternates between CLM and CVM (inspired by classical polyhedral methods) to traverse the polyhedron's corner points. Iterations continue until a lower bound for the furthest distance is found and removed using a proposed cutting plane. Our solution algorithm finds the furthest point under mild conditions (such as the full rank of active constraints at the furthest point and non-degeneracy of polyhedron vertices).

4. Applications

Our following reformulations find a feasible solution under a candidate bound δ ; the optimal solution is then found through bisection on δ .

MILP. A MILP can be reformulated as a furthest point problem as follows (note that $x \in \mathbb{R}^{n_x}$ and $y \in \{0, 1\}^{n_y}$):

$$\begin{array}{lll}
\text{find} & (x, y) & \text{find } (x, y) \\
\text{s.t.} & A_1x + A_2y \leq b & \text{(3a)} \\
& c_1^T x + c_2^T y \leq \delta & \text{(3b)} \\
& c_1^T x + c_2^T y \leq \delta & \text{(3c)} \\
& y \in [0, 1]^{n_y}, \sum_{i=1}^{n_y} y_i \leq \|y\|^2 & \text{(3d)}
\end{array}$$

$$\begin{array}{lll}
\max & \|y\|^2 - \sum_{i=1}^{n_y} y_i & \text{(4a)} \\
\text{s.t.} & A_1x + A_2y \leq b & \text{(4b)} \\
& c_1^T x + c_2^T y \leq \delta & \text{(4c)} \\
& y \in [0, 1]^{n_y} & \text{(4d)}
\end{array}
\quad
\begin{array}{ll}
\max & \|z\|^2 \\
\text{s.t.} & A_1x + A_2(z + 0.5e) \leq b \\
& c_1^T x + c_2^T (z + 0.5e) \leq \delta \\
& (z + 0.5e) \in [0, 1]^{n_y}
\end{array}$$

where $e \in \mathbb{R}^{n_y}$ is a vector whose entries are 1 and $\delta \in \mathbb{R}$ is a candidate upper bound for the MILP's optimal solution. This bound is satisfied if and only if problem (5) has a feasible point whose norm is larger than $\frac{\sqrt{n_y}}{2}$.

In (5), term $0.5e$ centers binary variables at 0, allowing the squared norm to serve as a continuous measure of binary deviation.

Nonconvex QCQP. Using the proposed reformulation technique in [2], we can find a solution satisfying an upper bound δ for a nonconvex QCQP.

$$\begin{array}{ll}
\text{find} & x \\
\text{s.t.} & Ax \leq b, c^T x \leq \delta
\end{array}
\quad
\begin{array}{ll}
\text{(6a)} & \max \|\hat{x}\|^2 \\
\text{(6b)} & \text{s.t. } \hat{A}\hat{x} \leq 0, \hat{c}^T \hat{x} \leq 0
\end{array}$$

$$\begin{array}{ll}
\left[\begin{array}{c} x \\ 1 \end{array} \right]^T Q_i \left[\begin{array}{c} x \\ 1 \end{array} \right] + 1 \leq 0 & \text{(6c)} \\
\hat{x}^T \hat{Q}_i \hat{x} \leq 1 & \text{(7c)}
\end{array}$$

$$\text{with } \hat{Q}_i = \begin{bmatrix} \frac{1}{\sqrt{\gamma}} I_n & 0 \\ 0 & \frac{1}{\sqrt{\gamma-1}} \end{bmatrix}^T (Q_i + \gamma I) \begin{bmatrix} \frac{1}{\sqrt{\gamma}} I_n & 0 \\ 0 & \frac{1}{\sqrt{\gamma-1}} \end{bmatrix}, \hat{c} = \begin{bmatrix} \sqrt{\gamma-1}c \\ -\sqrt{\gamma}\delta \end{bmatrix}, \hat{A} = \begin{bmatrix} \sqrt{\gamma-1}A & -\sqrt{\gamma}b \end{bmatrix}, \text{ and } \gamma = \max_{i \in \{1, \dots, m_2\}} \lambda(-Q_i).$$

Nonconvex LCP: The LCP is to find $x \in \mathbb{R}^n$ such that $0 \leq x \perp Mx + q \geq 0$, where M and q are matrix and vector with proper dimensions. By rewriting the complementarity constraints as $x^T(Mx + q) \leq 0$ and $x^T(Mx + q) \geq 0$, the LCP can be reformulated in a form analogous to (6) with (7) representing its furthest point formulation.

5. Conclusions

This paper addresses the Furthest Point Problem, emphasizing its non-convex nature and relevance to optimization problems like MILPs and QC-QPs and feasibility problems like LCPs. Our proposed algorithm exploits both the CLM and CVM to improve computational efficiency while providing exact solutions for distance-based optimization.

References

- [1] O. Cheong and A. Vigneron, "Computing the maximum distance between a point and a polyhedron," *Computational Geometry*, vol. 40, no. 2, pp. 69–75, 2008.
- [2] Abolpour, R., Dehghani, M., and Hesamzadeh, M.R. (2023). Inside-ellipsoid outside-sphere (IEOS) model for general bilinear feasibility problems: feasibility analysis and solution algorithm. *Automatica*, 147, 110738.

Session 12





Global Optimization of ACOPF in Unbalanced Distribution Networks

M. R. Karimi Gharigh,¹ M. R. Hesamzadeh,¹ J. Kronqvist,¹ G. Dán¹

¹*KTH Royal Institute of Technology, Stockholm, Sweden,* mrkg@kth.se, mrhesa@kth.se, jankr@kth.se, gyuri@kth.se

Abstract We propose an algorithm to find the global solution of the optimal power flow (OPF) problem in unbalanced alternating current (AC) distribution networks. First, the unbalanced ACOPF problem is formulated as a nonconvex quadratically constrained quadratic optimization problem (QCQP). The nonconvex QCQP is transformed into a convex QCQP along with a single nonconvex quadratic constraint by using the eigenvalue decomposition technique. The Lagrangian dual decomposition is employed to decompose the reformulated model into two convex subproblems, each formulated as a semi-definite program (SDP). By iterating between the two SDPs, a computationally efficient and tight lower bound for the original ACOPF model is then obtained. The lower bound combined with a proper upper bound can be used in a branch-and-bound framework to find the ϵ -optimal solution of the unbalanced ACOPF model.

Keywords: Unbalanced ACOPF, QCQP, Eigen-value decomposition, Lagrangian dual decomposition

1. Introduction

The cost-effective operation of power systems is crucial for maximizing efficiency, minimizing operational costs, and ensuring sustainability in energy management. At the transmission level, this has traditionally been achieved through the well-established ACOPF (Alternating Current Optimal Power Flow) problem, which benefits from balanced three-phase configurations and uniform load distribution.

However, when focusing on the distribution system, the situation becomes even more complex. Distribution networks typically consist of single-phase connections in residential and small commercial areas, which leads to uneven phase loading and, hence, to voltage and current imbalances. The imbalance is the result of a practical trade-off: balancing loads would require costly infrastructure and continuous monitoring, which are impractical for large-scale dynamic networks. As a result, distribution systems use a mix of single-, two-, and three-phase feeders, further complicating operational modeling. In this setting, the classical ACOPF problem must be extended to reflect the characteristics of unbalanced distribution systems, resulting in a nonconvex quadratically constrained quadratic program (QCQP)—known as the unbalanced ACOPF [2]. The resulting nonconvex QCQP is known to be computationally challenging [3].

In this paper, we introduce a novel reformulation of the unbalanced ACOPF problem. By leveraging eigenvalue decomposition, we convert the problem into a convex QCQP with a single nonconvex quadratic constraint. The goal of our method is to offer a scalable solution for complex real-world distribution networks.

2. Modeling the unbalanced ACOPF

The general unbalanced ACOPF problem can be formulated as

$$\begin{aligned} \min \quad & \mathbf{x}^T \mathbf{U}_0 \mathbf{x} + \mathbf{c}_0^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x}^T \mathbf{U}_i \mathbf{x} + \mathbf{c}_i^T \mathbf{x} + d_i \leq 0, \quad \forall i \in \{1, 2, \dots, m\} \\ & \mathbf{A} \mathbf{x} \leq \mathbf{b}, \quad \underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}}. \end{aligned} \quad (1)$$

Here, the matrices $\mathbf{U}_0, \mathbf{U}_i \in \mathbb{R}^{n \times n}$ are symmetric, while the vectors $\mathbf{c}_0, \mathbf{c}_i \in \mathbb{R}^n$ and scalars $d_i \in \mathbb{R}$ are defined for each $i \in \{1, 2, \dots, m\}$. The matrix $\mathbf{A} \in \mathbb{R}^{k \times n}$ and the vector $\mathbf{b} \in \mathbb{R}^k$ characterize the system's linear constraints. The vectors $\underline{\mathbf{x}}$ and $\bar{\mathbf{x}} \in \mathbb{R}^n$ represent the lower and upper bounds on the variable vector \mathbf{x} , respectively.

The vector \mathbf{x} encompasses key variables in the unbalanced ACOPF problem, including active and reactive power consumption and generation for local loads and generators, the real and imaginary components of the currents of local loads and distribution lines, and the real and imaginary parts of the nodal voltages. Accurately modeling the imbalance in these variables is essential to capturing the system's behavior.

Given the nonconvexity and NP-hardness of the unbalanced ACOPF problem, obtaining an optimal solution is computationally challenging. In the following section, we introduce an iterative algorithm that leverages

the nonconvex reformulation of (1) with pure convex constraints and a single nonconvex constraint, enabling an efficient relaxation.

3. Joint Lagrangian Dual and Eigen-Value Decomposition Approach

Recall that in the unbalanced ACOPF problem (1) each matrix \mathbf{U}_i , for $i \in \{0, 1, 2, \dots, m\}$, is symmetric, resulting in real eigenvalues. For each \mathbf{U}_i , we can find a scalar λ_i such that the matrix $\mathbf{U}_i + \lambda_i \mathbf{I}_n$ is positive semi-definite, i.e., $\mathbf{U}_i + \lambda_i \mathbf{I}_n \succeq 0$, where \mathbf{I}_n is the identity matrix of size n .

Let us now define $\mathbf{V}_i = \mathbf{U}_i + \lambda_i \mathbf{I}_n$ for each i . Using \mathbf{V}_i and the auxiliary variables \mathbf{w}, y, z such that $\mathbf{w} = \mathbf{x}$, $z = y$, and $y = \mathbf{w}^T \mathbf{w}$, the original problem in (1) can be reformulated as

$$\begin{aligned} \min \quad & \mathbf{x}^T \mathbf{V}_0 \mathbf{x} + \mathbf{c}_0^T \mathbf{x} - \lambda_0 y \\ \text{s.t.} \quad & \mathbf{x}^T \mathbf{V}_i \mathbf{x} + \mathbf{c}_i^T \mathbf{x} + d_i - \lambda_i y \leq 0, \quad \forall i \in \{1, 2, \dots, m\} \\ & \mathbf{A} \mathbf{x} \leq \mathbf{b}, \quad \underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}}, \\ & y \geq \mathbf{x}^T \mathbf{x}, \quad z \leq \mathbf{w}^T \mathbf{w} \\ & \mathbf{w} = \mathbf{x} \leftrightarrow (\alpha), \quad z = y \leftrightarrow (\beta). \end{aligned} \tag{2}$$

By using the Lagrangian relaxation for the last two constraints with the Lagrange multipliers α and β , the relaxed problem of (2) splits into

$$\begin{aligned} \mathcal{L}_1(\alpha, \beta) = \min_{\mathbf{x}, y} \quad & \mathbf{x}^T \mathbf{V}_0 \mathbf{x} + \mathbf{c}_0^T \mathbf{x} - \lambda_0 y + \alpha^T \mathbf{x} + \beta y \\ \text{s.t.} \quad & \mathbf{x}^T \mathbf{V}_i \mathbf{x} + \mathbf{c}_i^T \mathbf{x} + d_i - \lambda_i y \leq 0, \quad \forall i \in \{1, 2, \dots, m\} \\ & \mathbf{A} \mathbf{x} \leq \mathbf{b}, \quad \underline{\mathbf{x}} \leq \mathbf{x} \leq \bar{\mathbf{x}}, \quad y \geq \mathbf{x}^T \mathbf{x}, \end{aligned} \tag{3}$$

$$\begin{aligned} \mathcal{L}_2(\alpha, \beta) = \min_{\mathbf{w}, z} \quad & -\alpha^T \mathbf{w} - \beta z \\ \text{s.t.} \quad & z \leq \mathbf{w}^T \mathbf{w}. \end{aligned} \tag{4}$$

Note that subproblem (3) is a convex QCQP and can be solved to optimality under mild Slater conditions. Also, the subproblem (4) is a non-convex QCQP with one nonconvex quadratic constraint, and using the S-lemma [4], it can be solved. We denote SDP1 as the SDP reformulation of subproblem (3), and SDP2 as the dual optimization problem of subproblem (4). Accordingly, for given values of α and β , the Lagrange functions \mathcal{L}_1 and \mathcal{L}_2 can be found from solving SDP1 and SDP2. The dual functions $\mathcal{L}_{1D} = \max_{\alpha, \beta} \mathcal{L}_1(\alpha, \beta)$ and $\mathcal{L}_{2D} = \max_{\alpha, \beta} \mathcal{L}_2(\alpha, \beta)$ can be solved in parallel, where $\mathcal{L}_D = \mathcal{L}_{1D} + \mathcal{L}_{2D}$ provides a computationally efficient and strong lower bound of the unbalanced ACOPF problem. We will investigate (sufficient) conditions under which the computed \mathcal{L}_D has zero duality gap with the primal optimization problem given the physical properties of

the unbalanced AC distribution network. Under such conditions, our approach provides an exact relaxation with zero-duality gap. Our suggested approach can also be used in a Branch-and-Bound framework to find the global solution of the unbalanced ACOPF in distribution networks.

4. Numerical results

We will investigate the effectiveness and scalability of the proposed method through numerical experiments on standard IEEE 13-bus, 37-bus, and 123-bus unbalanced distribution feeders. These benchmark cases capture a range of phase imbalances, load complexities, and network sizes.

5. Conclusion and future work

This study presents a novel global optimization method for the ACOPF problem in unbalanced distribution networks. It reformulates the nonconvex QCQP into a convex structure with a single nonconvex constraint by leveraging eigenvalue decomposition and Lagrangian dual decomposition. Numerical results on standard IEEE feeders confirm the algorithm's effectiveness, global optimality, and scalability, making it a promising tool for advanced distribution system operations.

We will investigate sufficient conditions for a zero duality gap of the Lagrangian relaxation. With zero duality gap, the method guarantees an ϵ -optimal solution. We will further investigate parallelization and other techniques to increase the suitability for solving large-scale problems.

References

- [1] Kersting, W. H. (2018). Distribution system modeling and analysis. In Electric power generation, transmission, and distribution (pp. 26-1). CRC press.
- [2] He, M. et. al. (2023). A SOCP-based ACOPF for operational scheduling of three-phase unbalanced distribution systems and coordination of PV smart inverters. *IEEE Transactions on Power Systems*, 39(1), 229-244.
- [3] Yamada, S. et. al. (2018). Successive Lagrangian relaxation algorithm for non-convex quadratic optimization. *Journal of Global Optimization*, 71, 313-339.
- [4] Boyd, S. et. al. (2004). Convex optimization. Cambridge University Press.

The Strengthened Semidefinite Relaxation for Nordic Day-ahead Energy Market*

Shudian Zhao,¹

Jan Kronqvist,¹ Mohammad Reza Hesamzadeh²

¹*KTH, Department of Mathematics, Stockholm, Sweden,* shudian@kth.se, jankr@kth.se

²*KTH, Department of Electrical Engineering and Computer Science, Stockholm, Sweden,* mrhesa@kth.se

Abstract Social welfare maximization problems in the day-ahead energy market can be formulated as large-scale mixed-integer programming (MIP) problems. These problems are equivalent to the NP-hard completely positive programming (CPP) problems, which are computational intractable. Instead, we solve a RLT-strengthened doubly nonnegative (DNN) problems which relaxes the CPP cones to achieve a relative small optimality gap. Preliminary results proves that the DNN+RLT relaxations have significant improved bounds compared to LP relaxations and basic DNN relaxations.

Keywords: Semidefinite Programming, Mixed-Integer Programming

1. Introduction

The maximum welfare problem for the European day-ahead energy market aim to clear the market by selecting and pricing energy bids to maximize total social welfare, considering supply and demand orders, network constraints, and technical limitations for regional policies [4]. This problem is NP-hard in general due to the integrality nature for discrete orders. Previous studies have focused on developing linear relaxation based approaches [5]. Semidefinite Programming (SDP) has shown a stronger modeling power for mixed-integer programming problems, and its been ap-

*This project is funded by Energimyndigheten project number 2022-200021.

plied to problems from power system networks, such as unit commitment problems [2]. Reference [3] has shown that the unit commitment problems from the US market can be reformulated as a complete positive programming (CPP) problems which is NP-hard but convex with zero duality gap. In this set-up, a pricing scheme can be designed using dual solutions. SDP is commonly-used to relax CPP as CPP cone is a subset of the positive semidefinite (PSD) cone intersected with the nonnegative orthant [1]. We aim to tackle the intractability of solving CPPs by a RLT-straightened SDP, which can achieve a relative small optimality gap in numerical experiments.

2. The day-ahead energy market problems

The day-ahead energy market problems can be formulated as a MILP:

$$\begin{aligned}
 & \min c_E^\top x_E + c_B^\top x_B + c_{FHB}^\top x_{FHB} \\
 & \text{s.t } A_E x_E + A_B x_B + A_{FHB} x_{FHB} = \mathbf{0}_{|\mathcal{T}|}, \\
 & x_E \in \mathcal{X}_E \subset \mathbb{R}_+^{n_E \cdot |\mathcal{T}|}, \\
 & x_B \in \mathcal{X}_B \subset \mathbb{R}_+^{n_{PB}} \times \{0, 1\}^{n_{RB}}, \\
 & x_{FHB} \in \mathcal{X}_{FHB} \subset \{0, 1\}^{n_{FHB} \cdot |\mathcal{T}|},
 \end{aligned} \tag{MIP}$$

where \mathcal{X}_E , \mathcal{X}_B , \mathcal{X}_{FHB} denote the feasible set for elementary orders x_E , block orders x_B , and flexible hourly orders x_{FHB} respectively. \mathcal{X}_E is a bounded polyhedron, but X_B is a polyhedron intersected with disjunctive sets and integer variables. Those three categories of orders are coupled only by the balance constraint which indicate the demand and supply order are cleared at each time period $t \in \mathcal{T}$, and constraints coefficient A_E, A_B, A_{FHB} with row vectors $a_{E,t}, a_{B,t}, a_{FHB,t}$ denoting positive (negative) quantity for demand (supply) orders of each category. The objective parameter c_E, c_B, c_{FHB} denote the profit (cost) for demand (supply, resp.) order from each category.

3. The strengthened DNN relaxation

The maximum welfare problem is a nonconvex quadratically constrained quadratic programming (QCQP) problem that has a equivalent CPP formulation [1]. CPP problems are convex but still NP-hard, but we can relax the CPP cone to be doubly nonnegative cone (DNN), i.e., a PSD cone

intersected by nonnegative orthant. To this end, we have the basic DNN relaxation for (MIP) as

$$\begin{aligned}
 & \min c_E^\top x_E + c_B^\top x_B + c_{FHB}^\top x_{FHB} \\
 & \text{s.t. } A_E x_E + A_B x_B + A_{FHB} x_{FHB} = \mathbf{0}_{|\mathcal{T}|}, \\
 & \quad \langle A_t, X_t \rangle = 0 \quad \forall t \in \mathcal{T}, \\
 & \quad x_t := \begin{pmatrix} x_{E,t} \\ x_{FHB,t} \\ x_B \end{pmatrix}, \\
 & \quad 0 \leq x_t \leq 1, \forall t \in \mathcal{T}, \\
 & \quad x_E \in \mathcal{X}_E, x_B \in \bar{\mathcal{X}}_B, x_{FHB} \in \bar{\mathcal{X}}_{FHB}, \\
 & \quad \begin{pmatrix} 1 & x_t^\top \\ x_t & X_t \end{pmatrix} \in \mathcal{S}_+ \cap \mathbb{N}_+, \forall t \in \mathcal{T}, \\
 & \quad \text{diag}(X_{t,ii}) = x_{t,i}, \forall i \in \mathcal{B} \cup \mathcal{FHB},
 \end{aligned} \tag{DNN-basic}$$

where $\mathcal{B} \cup \mathcal{FHB}$ denote the index sets for x_B and $X_{FHB,t}$, $\bar{\mathcal{X}}_B$ and $\bar{\mathcal{X}}_{FHB}$ denote the linear relaxations, and A_t is the squared matrix

$$A_t = a_t a_t^\top, a_t := (a_{E,t}^\top \quad a_{B,t}^\top \quad a_{FHB,t}^\top)^\top.$$

To further strengthen the DNN relaxation, we include the RLT inequalities from the following groups:

$$1 - x_i - x_j \leq X_{t,ij} \leq \min\{x_i, x_j\}, \tag{1a}$$

$$W_B X_B \leq f_B, \tag{1b}$$

$$W_{FHB} X_{FHB} \leq f_{FHB}, \tag{1c}$$

where (1a) derived by the inner product between $x_t \leq 1$ and $x_t \geq 0$, (1b) ((1c)) are derived by inner products between all the constraints inducing $\bar{\mathcal{X}}_B$ ($\bar{\mathcal{X}}_{FHB}$, resp.) with substitution of bilinear terms by PSD matrix X_B (X_{FHB} , resp.).

4. Numerical results

We illustrate the tightness of various SDP relaxations with an exemplary instance with 2 elementary order, 2 profile block orders, 2 regular block orders, and 2 flexible hourly orders for both demand and supply sides over $|\mathcal{T}| \in \{2, 3, 24\}$. We compare the upper bounds from various relaxations and the objective function for (MIP) Table 1 shows that the direct LP re-

$ \mathcal{T} $	LP	DNN-basic	DNN+RLT	MIP
2	676582.50	575341.34	158175.41	150264.60
3	940736.50	780850.49	260439.97	245008.60
24	5655408.67	4232999.29	1023351.76	1019784.00

Table 1: Comparison between objective values from different models

laxations can only provide very weak bounds for (MIP), and improvement from DNN-basic are limited, but the DNN + RLT relaxations can reduce the optimality to within 1%.

5. Conclusion

The DNN+RLT relaxation can yield relatively small optimality gaps in practice; however, solving large-scale SDP problems remains computationally intractable for real-world applications. To address this challenge, we aim to develop a decomposition-based approach that reduces computational bottlenecks and demonstrate its effectiveness on representative instances from the Nordic energy market.

References

- [1] Samuel Burer. On the copositive representation of binary and continuous non-convex quadratic programs. *Mathematical Programming*, 120(2):479–495, 2009.
- [2] Salar Fattah, Morteza Ashraphijuo, Javad Lavaei, and Alper Atamtürk. Conic relaxations of the unit commitment problem. *Energy*, 134:1079–1095, 2017.
- [3] Cheng Guo, Merve Bodur, and Joshua Taylor. Copositive duality for discrete energy markets. *Available at SSRN 4964985*, 2024.
- [4] Ioan Alexandru Puiu and Raphael Andreas Hauser. On market clearing of day ahead auctions for european power markets: Consumer payment minimisation versus social welfare maximisation. *Energy Economics*, 139:107880, 2024.
- [5] Ryohei Yokoyama, Yuji Shinano, Syusuke Taniguchi, Masashi Ohkura, and Tetsuya Wakui. Optimization of energy supply systems by milp branch and bound method in consideration of hierarchical relationship between design and operation. *Energy conversion and management*, 92:92–104, 2015.

A Modified SAVLR Method for Zero Duality Gap in MILPs of Electricity Markets *

Zhen Wang,¹ Shudian Zhao,²
Jan Kronqvist,² Mohammad Reza Hesamzadeh¹

¹*KTH, Department of Electrical Engineering and Computer Science, Stockholm, Sweden,*
zhenwa@kth.se, mrhesa@kth.se

²*KTH, Department of Mathematics, Stockholm, Sweden,* shudian@kth.se, jankr@kth.se

Abstract This paper modifies the standard SAVLR method to close the duality gap in MILPs of Electricity Markets based on augmented Lagrangian and dual (ALD) framework. The Lagrangian multipliers are computed by the (surrogate) subgradient method. We use the l_1 -norm as a penalty function and the associated penalty parameter is obtained by calculating a threshold where any value larger than this threshold does not increase the augmented Lagrangian function value. The zero-duality gap has been verified in a case study of Nordic MILP electricity market, which property will be used for a new pricing rule.

Keywords: MILP, Strong duality, New pricing rule, Nordic electricity market

1. Introduction

We aim for a new pricing rule of the electricity market, which is modeled as a MILP. Due to the integer variables, the optimization problem is not convex, causing a duality gap when dualizing the energy balance equation. Therefore, the Lagrangian multiplier of the energy balance equation is not sufficient for the pricing mechanism, and additional money is needed to make each market participant satisfactory, as is standard in the literature,

*This project is funded by Energimyndigheten project number 2022-200021.

[3]. For the pricing rule in the energy market, the dual problem is more important, and we assume the primal problem has been solved.

There is a two-step procedure to achieve our purpose. First, to close the duality gap. Second, to interpret the individual rationality. In this paper, we focus on closing the duality gap through an Augmented Lagrangian and Dual (ALD) framework.

2. The day-ahead electricity market problems

The day-ahead electricity market problem to maximize the overall welfare in one bidding area can be formulated as a MILP [3]:

$$\begin{aligned} & \max c_E^\top x_E + c_B^\top x_B + c_{FHB}^\top x_{FHB} \\ & \text{s.t } A_E x_E + A_B x_B + A_{FHB} x_{FHB} = 0_{|\mathcal{T}|}, \\ & x_E \in \mathcal{X}_E \subset [0, 1]^{n_E \cdot |\mathcal{T}|}, \\ & x_B \in \mathcal{X}_B \subset [0, 1]^{n_{PB}} \times \{0, 1\}^{n_{RB}}, \\ & x_{FHB} \in \mathcal{X}_{FHB} \subset \{0, 1\}^{n_{FHB} \cdot |\mathcal{T}|}, \end{aligned} \quad (1)$$

where \mathcal{X}_E , \mathcal{X}_B , \mathcal{X}_{FHB} denote the feasible sets for the elementary orders x_E , the block orders x_B , the flexible hourly orders x_{FHB} , respectively, and they are coupled only by the power balance constraints which indicate the demand and supply orders are cleared at each time slot $t \in \mathcal{T}$. Note that there are 96 time slots during one day (one period = 15 minutes).

3. Approach for closing the duality gap of the ALD-based framework

We denote $c := [c_E^\top, c_B^\top, c_{FHB}^\top, c_{P,L_{AC},L_{DC}}^\top]^\top$, $x := [x_E^\top, x_B^\top, x_{FHB}^\top]^\top$, $A := \text{Diag}([A_E, A_B, A_{FHB}])$, $\mathcal{X} := \mathcal{X}_E \times \mathcal{X}_B \times \mathcal{X}_{FHB}$. In this way, the centralized augmented Lagrangian relaxation with ℓ_1 -norm is $L(x, \lambda) = c^\top x + \lambda^\top Ax + \rho \|Ax\|_1$, where $\rho > 0$ is the penalty coefficient, and $\lambda \in \mathbb{R}^{|\mathcal{T}|}$ is the Lagrangian multiplier, and $x \in \mathcal{X}$. After dualizing the system-wide constraint, the subproblems of the SAVLR method [1], has an expression: $x_i^{k+1} = \arg \max_{x_i \in \mathcal{X}_i} \tilde{L}(x_i, \lambda^{k+1}) = \arg \max_{x_i \in \mathcal{X}_i} (c_i + A_i^\top \lambda^{k+1})^\top x_i + \rho \|A_i x_i + A_{-i} x_{-i}\|_1$, where $x_i \in \{x_E, x_B, x_{FHB}\}$, and x_{-i} includes all the other vari-

ables except x_i . The update of x^{k+1} requires checking the surrogate optimality condition, $\tilde{L}(x_i^{k+1}, \lambda^{k+1}) > \tilde{L}(x_i^k, \lambda^{k+1})$ [1].

Remark 1. It has been proven in [1, Proposition 1] that for the standard SAVLR method, the penalty coefficient cannot be increased such that the system-wide constraint $\|Ax\|$ is enforced, where the surrogate optimality condition is violated for all the subproblems. What's more, under that condition, the duality gap exists.

Our Algorithm is based on the standard SAVLR method, and the complete procedure is presented in Algorithm 1.

Theorem 2. Give x^{k+1} from the current iteration, if the stepsize σ^{k+1} satisfies,

$$\sigma^{k+1} \leftarrow \left(1 - \frac{1}{M(k+1)^{p^k}}\right) \frac{c^k \|Ax^k\|}{\|Ax^{k+1}\|},$$

where $p^k \leftarrow 1 - \frac{1}{(k+1)^r}$ and $M \geq 1, r \in (0, 1)$, then the iteration of λ^k of Algorithm 1 will converge to the dual optimal solution λ^* [2].

Algorithm 1 Solution Algorithm of ALD

```

1: Input data: Find  $\lambda^1, \rho^1 > 0, \beta > 1$  by heuristic, and compute the primal solution  $x^*$  by
   MILP solvers.
2: Initialization:  $x^1 \leftarrow \arg \max \tilde{L}(x, \lambda^1), \hat{q} > \min_{\lambda} \max_{x \in \mathcal{X}} L(\lambda^1, x), \sigma^1 \leftarrow \frac{\hat{q} - \tilde{L}(\lambda^1, x^1)}{\|Ax^1\|_1^2}, j \leftarrow 1, \rho_1$  that the standard SAVLR method provides.
3: Output:  $\lambda^j, \rho^j, x^j$ 
4: while  $\|L(x^{j+1}, \lambda^{j+1}) - c^\top x^* \|_1 / \|c^\top x^*\|_1 > \epsilon_b$  do
5:    $k \leftarrow 1, \rho^{j+1} \leftarrow \rho^j \beta$  ▷ Increase the penalty coefficient
6:   while  $\|\lambda^{k+1} - \lambda^k\| > \epsilon_d$  do ▷ Dual update
7:      $\lambda^{k+1} \leftarrow \lambda^k - \sigma^k g(x^k)$ 
8:     for  $x_i \in \{x_E, x_B, x_{FHB}\}$  do
9:       Solve  $x_i^{k+1} = \arg \max_{x_i \in \mathcal{X}_i} \tilde{L}(x_i, \lambda^{k+1}),$ 
10:      if  $\tilde{L}(x_i^{k+1}, \lambda^{k+1}) > \tilde{L}(x_i^k, \lambda^{k+1})$  then
11:         $x^{k+1} \leftarrow (x_i^{k+1}, x_{-i}^k)$ , and break ▷ Primal update
12:      end if
13:    end for
14:    if  $\forall i, [x_i^{k+1}, x_{-i}^k] == [x_i^k, x_{-i}^k]$  then
15:       $x^{k+1} \leftarrow \arg \max_{x \in \mathcal{X}} L(x, \lambda^{k+1})$ 
16:    end if
17:    Update  $\sigma^{k+1}$  by Theorem 2, and let  $k \leftarrow k + 1$ .
18:  end while
19:  Given  $\lambda^{k+1}$ , find  $x^{k+1}$  by heuristic.
20:   $\lambda^{j+1} \leftarrow \lambda^{k+1}, x^{j+1} \leftarrow x^{k+1}$ 
21: end while

```

Corollary 3. *It has been proven in [4, Theorem 4] that for the optimal dual solution λ^* , there exists $\rho^* < \infty$ which can close the duality gap of the ALD framework. Therefore, after a finite number of iterations, the outer loop in Algorithm 1, i.e. the step 4 can be terminated.*

4. Numerical results

Algorithm 1 has been tested in the Nordic electricity market model. Taking an example, the penalty coefficient that the standard SAVLR method provides is 40.8, while the duality gap is 7.78%. The one computed by Algorithm 1 is 170.43, and the gap is 0.37%. What is more, the computation time in all the instances tested are less than 5 minutes, and the duality gaps are less than 0.5%, indicating that the duality gap is numerically closed.

5. Conclusion

In this paper, the scalable structure of the electricity market model is employed to formulate the subproblems for the standard SAVLR method. We have modified the standard SAVLR method to close the duality gap.

For future study, as pointed out in Introduction, a new pricing rule will be designed after the interpretation of the individual rationality. In addition, the exploration of the complete electricity market model is in process.

References

- [1] Mikhail A Bragin, Peter B Luh, Bing Yan, and Xiaorong Sun. A scalable solution methodology for mixed-integer linear programming problems arising in automation. *IEEE Transactions on Automation Science and Engineering*, 16(2):531–541, 2018.
- [2] Mikhail A Bragin, Peter B Luh, Joseph H Yan, and Gary A Stern. An efficient approach for solving mixed-integer programming problems under the monotonic condition. *Journal of Control and Decision*, 3(1):44–67, 2016.
- [3] Dimitris I Chatzigiannis, Grigoris A Dourbois, Pandelis N Biskas, and Anastasios G Bakirtzis. European day-ahead electricity market clearing model. *Electric Power Systems Research*, 140:225–239, 2016.
- [4] Mohammad Javad Feizollahi, Shabbir Ahmed, and Andy Sun. Exact augmented lagrangian duality for mixed integer linear programming. *Mathematical Programming*, 161:365–387, 2017.



2025