A Whole New Look for CONOPT

Lutz Westermann, Renke Kuhlmann

OR 2025 - Bielefeld

Introduction & Outline

- CONOPT history and transition: from ARKI to GAMS
- Technical improvements, e.g.
  - Speed
  - Robustness
  - Debugging, usability
- New APIs for new environments and user expectations
- Conclusion

The Conopt Story

The History of CONOPT

- 1976: CONOPT began as a Ph.D. project by Arne Stolbjerg Drud to develop a nonlinear solver.
- 1979: A crucial partnership was established with Alex Meeraus and the GAMS team at the World Bank, leading to the first practical versions of both GAMS and CONOPT in the early 80s.
- Ongoing Partnership: Drud founded his own company, ARKI Consulting and Development, and continued a close collaboration with GAMS Development Corp.
- Lasting Legacy: The development of the solver has been successfully transferred to a group at GAMS in 2024, ensuring its future and cementing the long-term relationship.

A New Chapter for CONOPT

- Long-Term Partnership: The transfer of CONOPT's development to GAMS represents the natural conclusion of a more than 30-year partnership.
- Ensuring Continuity: This move guarantees the solver's long-term sustainability and ensures a seamless continuity of professional development and user support.
- Leveraging Expertise: By fully integrating CONOPT, GAMS can leverage its extensive experience in solver development and optimization to further enhance the solver.
- Commitment to Users: The primary focus remains on delivering a reliable, high-performance, and user-friendly solver across a wide range of platforms and workflows.

The New Team

- Ksenia Bestuzheva
- Renke Kuhlmann
- Stephen Maher
- Stefan Vigerske
- (Lutz Westermann)






CONOPT nonlinear optimization

Algorithmic History and Evolution

- Initial Algorithm: The solver's foundation is the Generalized Reduced Gradient (GRG) method, designed for highly nonlinear problems.
- Continuous Improvement: Early versions were optimized by removing optimal control components in favor of more efficient sparse procedures.
- Integration of LP Techniques: The solver evolved by incorporating sparse matrix techniques from linear programming to handle large-scale models more efficiently.
- Multi-Method Architecture: Later versions introduced a dynamic "multi-method" architecture that selects the best solution approach for a given model and adjusts it as it solves: GRG, SLP, SQP, …

CONOPT Today

- Proven NLP Solver: CONOPT is a well-established and trusted solver for smooth, continuous nonlinear programming (NLP) problems.
- Large-Scale Performance: Based on a proven active set method, it excels at solving large-scale, structured models with many continuous variables and smooth, differentiable constraints.
- Seamless Integration: It is seamlessly integrated into leading modeling systems like GAMS, GAMSPy, AMPL, AIMMS, and the LINDO API.
- Robust and Reliable: The solver is highly valued in both academic and industrial contexts for its stability, robustness, and adaptability to a wide variety of problems.

CONOPT Today

- Highly Nonlinear Constraints: CONOPT is exceptionally well-suited for models that contain highly nonlinear constraints.
- Models with Few Degrees of Freedom: It has a fast method for finding a first feasible solution, making it particularly effective for "almost square" models where the number of variables is approximately the same as the number of constraints.
- Powerful Preprocessing: CONOPT's advanced preprocessing engine automatically simplifies models by identifying and solving recursive equations and variables. This feature is a significant advantage for models with a structured, sequential design.
- Valuable Diagnostic Tool: It serves as a useful diagnostic tool during model development, with built-in tests for issues like poor scaling. When a model can be improved, CONOPT provides a constructive message to guide the user.

Recent Enhancements

Performance and more

Recent Core Improvements

- Intelligent switching between conjugate and scaled conjugate gradient methods for faster convergence
- Default threading increased from 1 to 4
- Enhanced handling of small bound constraints and edge conditions
- Strengthened LU factorization consistency through added numerical checks
- Improved licensing infrastructure and updated branding reflecting GAMS stewardship
- Numerous multithreading and boundary-related bugs resolved for smoother runs

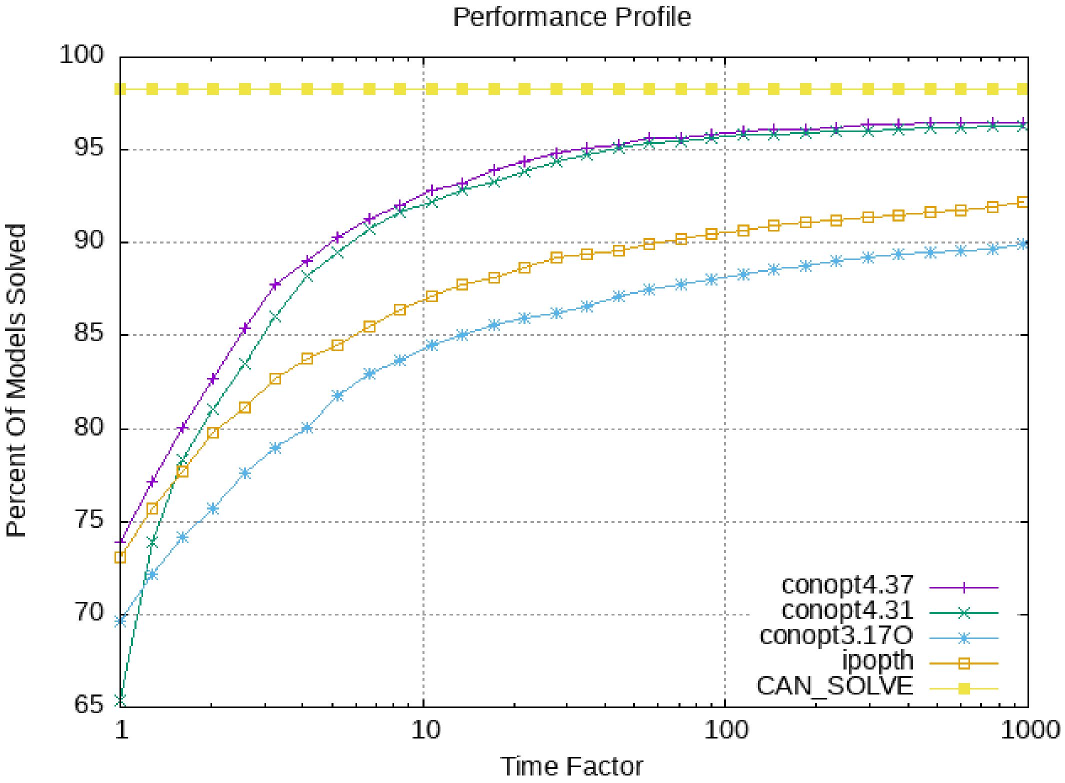
Performance Benchmarks

CONOPT nonlinear optimization

- Evaluated on a comprehensive suite of academic and real-world models (3029 instances from globallib, minlplib (continuous relaxations), princetonlib and more)
- Measurable reductions in runtime across many cases
- Substantial improvements in robustness and iteration efficiency
- Benchmarked on an i9-12900K with 64 GB RAM
- Time limit of 1000s for each run

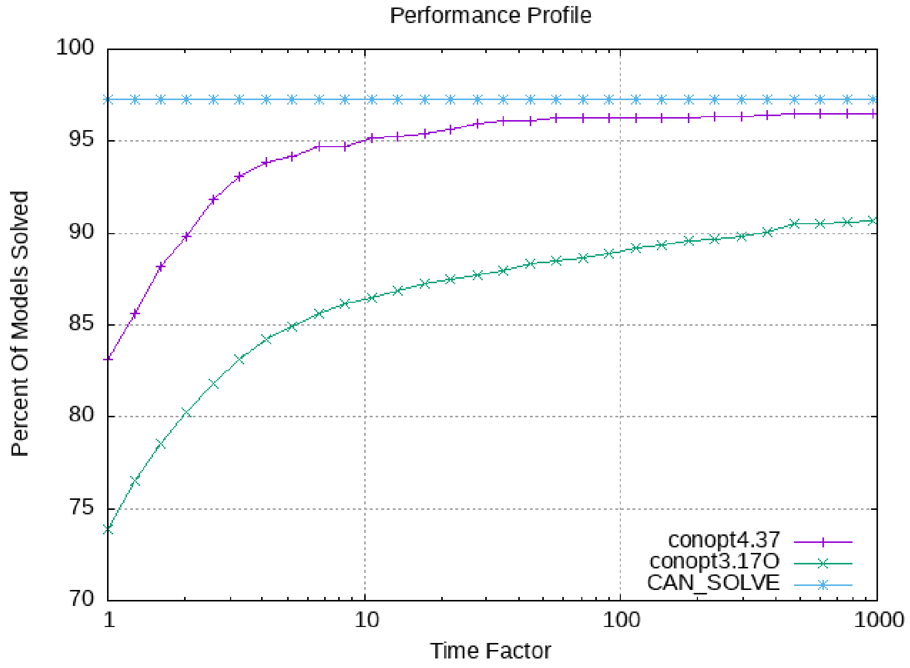
Benchmark Summary

CONOPT
nonlinear optimization



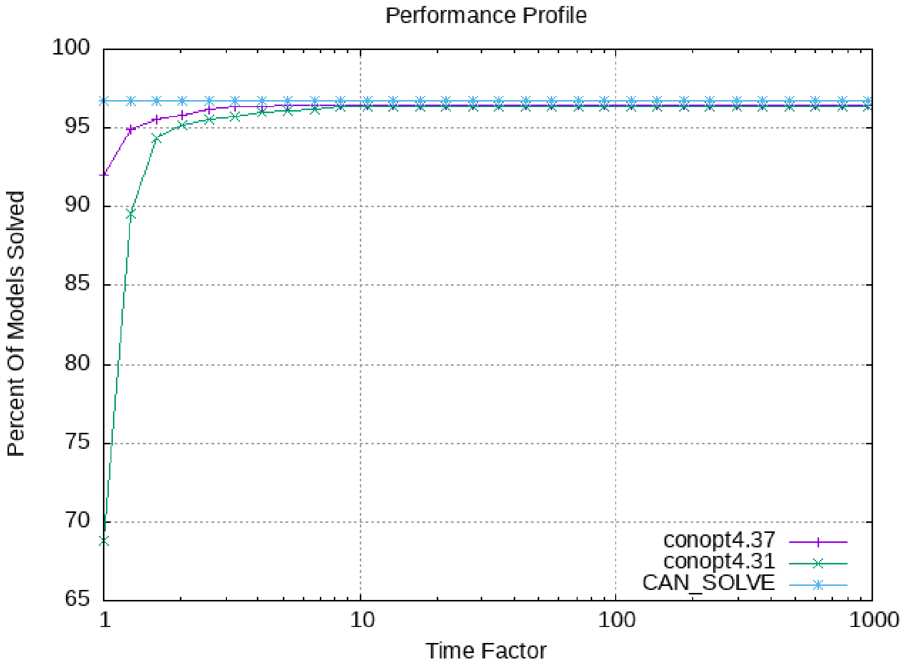
Conopt 4.37 vs. Conopt 3.17O

Speed	Conopt 4.37	Conopt 3.17
Same	1974	1974
10-50% faster	155	132
>50% faster	292	173
The other failed	197	24



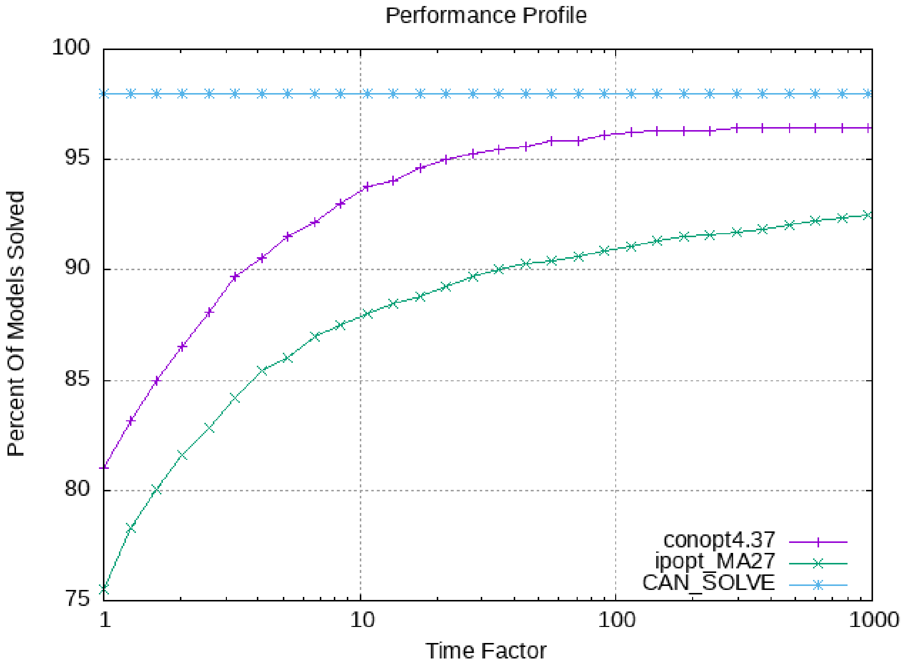
Conopt 4.37 vs. Conopt 4.31

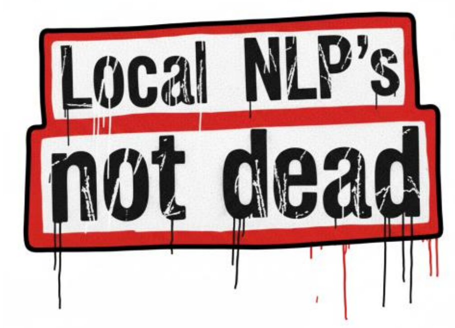
Speed	Conopt 4.37	Conopt 4.31
Same	2271	2271
10-50% faster	536	51
>50% faster	37	19
The other failed	9	3



Conopt 4.37 vs. IPOPT MA27

Speed	Conopt 4.37	IPOPT
Same	1964	1964
10-50% faster	151	125
>50% faster	290	45
The other failed	136	61





Excursus

Neural Net Verification

Excursus: Neural Net Verification

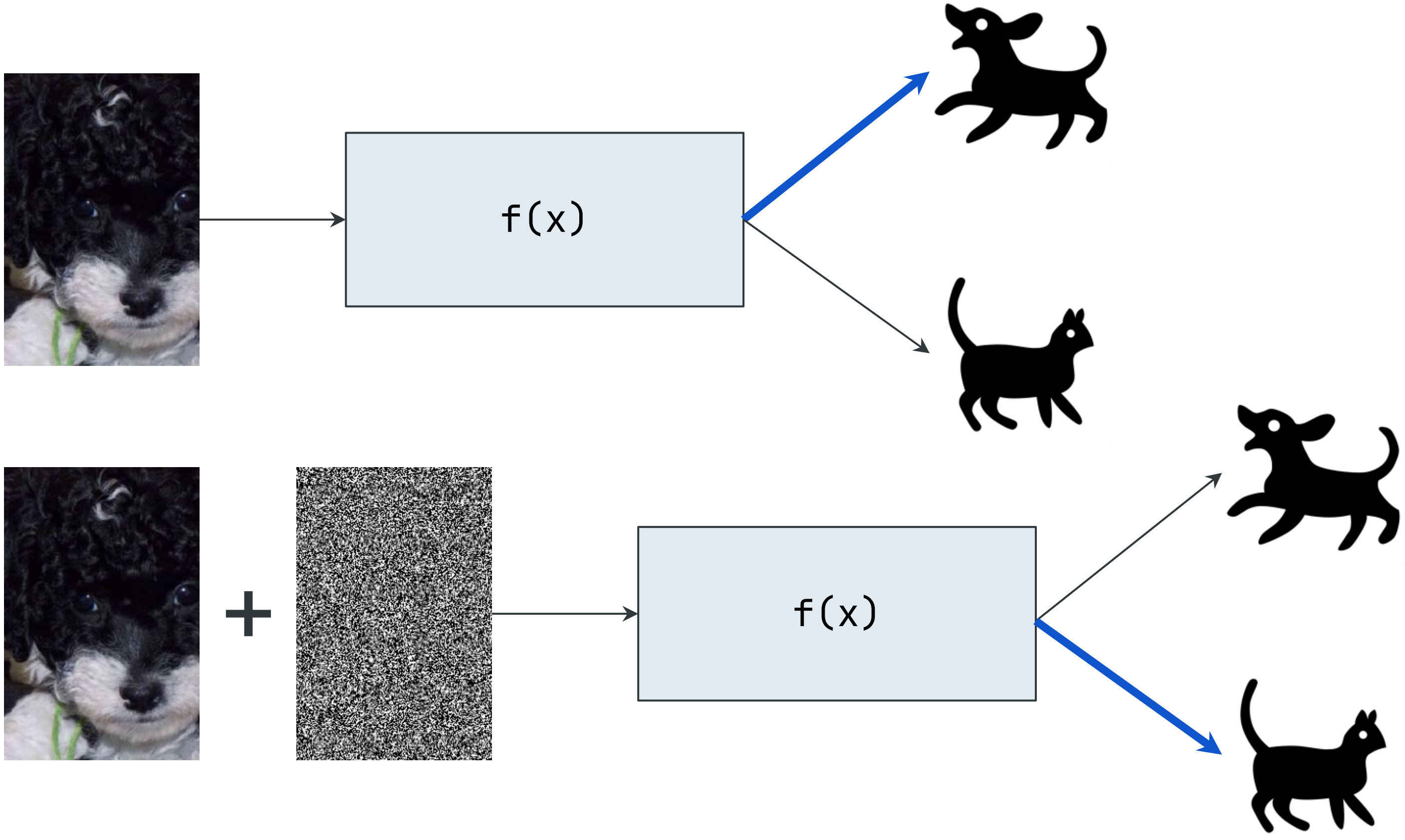
- The use of neural networks and machine learning has become pervasive across many industries
- Combining optimization w… research and application.
- Verification provides a vita… …ty and safety of AI models.
- CONOPT performs exceptio… demonstrating its capabilit… that arise from optimizing a…

Would you like to know more?

WE-09
Wednesday, 16:30-18:00
Room: H15

"Embedding Neural Networks into Optimization Models with GAMSPy"

Robustness



Optimization (Robustness Example)

nonlinear optimization

$$min_{noise} NN(input)_{correct} - NN(input)_{incorrect}$$

$$\|noise\|_\infty \leq \epsilon$$

$$input = image + noise$$

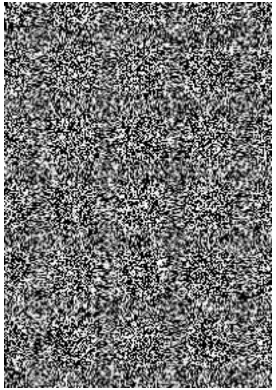
$$noise \in \mathbb{R}^{3\times H\times W}$$

$$image \in \mathbb{R}^{3\times H\times W}$$

$$input \in \mathbb{R}^{3\times H\times W}$$

$$NN : \mathbb{R}^{3\times H\times W} \to \mathbb{R}^2$$

xx



yy

German Sign Recognition Benchmark

CONOPT
nonlinear optimization

43 traffic signs

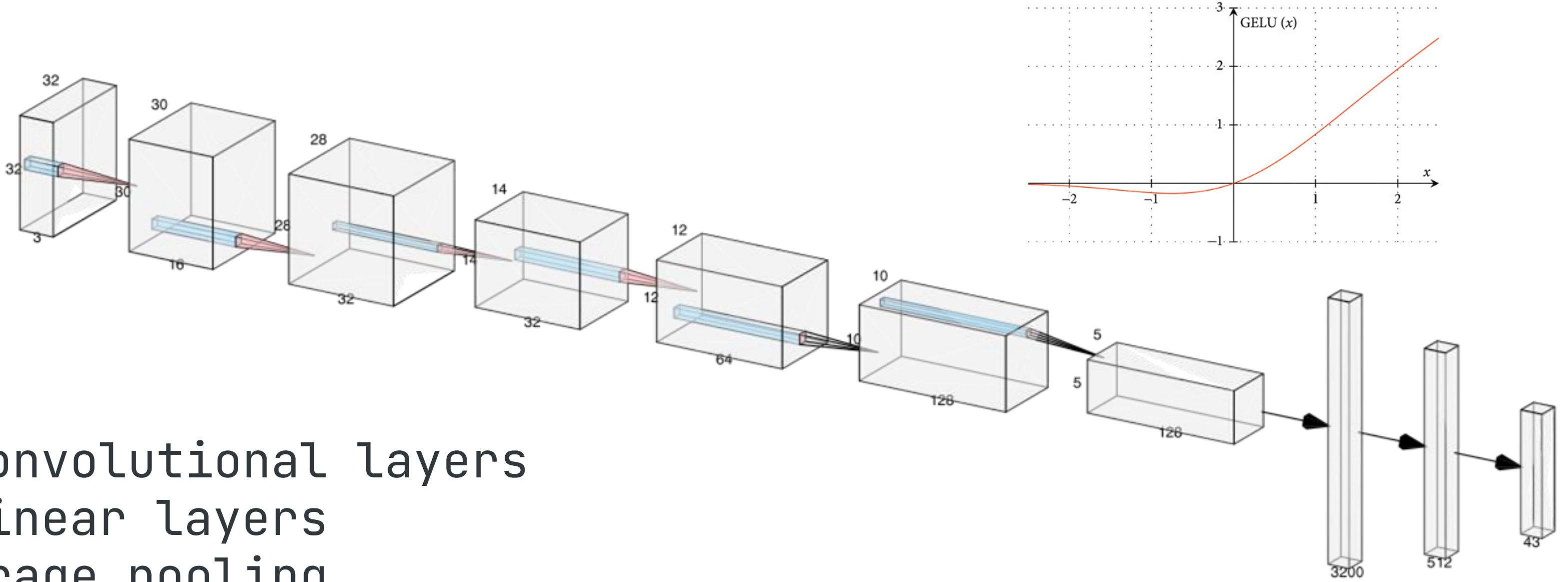
Real life example

Not trivially recognizable

A critical example

>50,000 images

A sample neural network



- 4 convolutional layers
- 2 linear layers
- Average pooling
- GELU
- Batch normalization

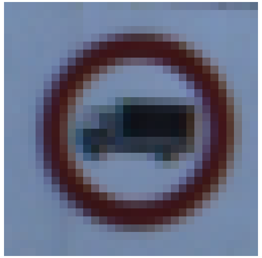
1,758,731 trainable parameters
85% accuracy on test set

Solve statistics

158,508 constraints
155,436 variables
15,969,198 Jacobian elements
62,016 of which are nonlinear

Local optimal solution found in 24 secs w/ CONOPT

No feasible solution found in 9 hours by a global solver



99.8%

No vehicles over 3.5 t permitted

99%

Speed limit 100 km/h

New APIs

And more

New APIs – C++, Python, and Java

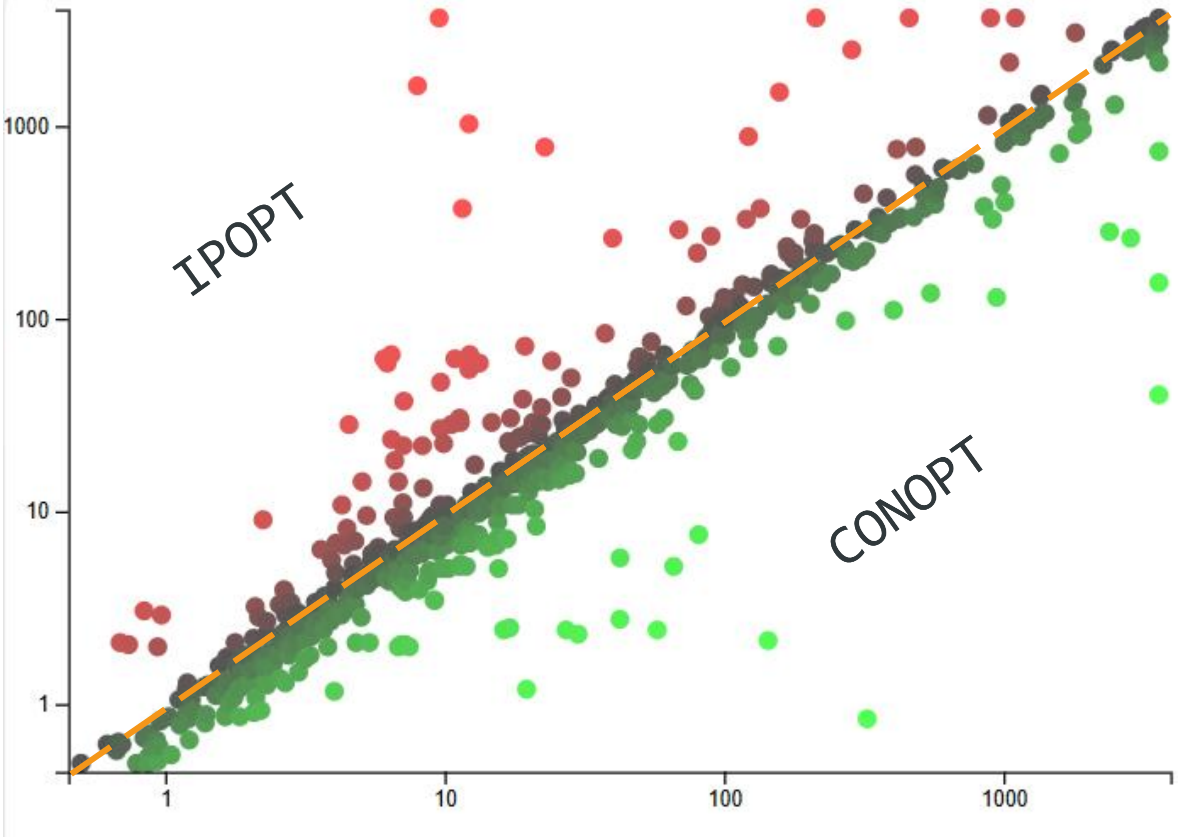
CONOPT
nonlinear optimization

- Support for modern programming languages extends CONOPT's usability
- Designed for seamless embedding in custom workflows and applications
- Ideal for integrating NLP solvers into research, industrial, or enterprise environments
- APIs enable:
  - Python scripting for rapid prototyping and teaching
  - C++ integration for high-performance computing
  - Java support for cross-platform and enterprise applications

Why These APIs Matter

- Provide direct, flexible access to CONOPT without a modeling system
- Enable novel use cases in simulation, optimization pipelines, and software engineering
- Simplify experimentation and research for students and developers alike
- Lower the barrier to adoption in nontraditional environments such as embedded systems or web services
- Allows also us to integrate CONOPT in other systems

Conopt as new subsolver for SCIP



Planned to be released with SCIP 10.0

Supplying Derivatives

CONOPT
nonlinear optimization

- Providing accurate derivatives significantly improves convergence speed and stability!
- CONOPT supports:
  - Analytic derivs via user-written source code
  - Analytic derivs via automatic differentiation (e.g. via a modeling system or external tool)
  - Numerical derivs (approximations: useful but less precise)
- Derivatives include:
  - Functions & Gradients (0th and 1st derivs): required
  - Hessian-vector products (directional derivs): optional
  - Hessians (2nd derivs): optional

www.conopt.com

CONOPT
nonlinear optimization

CONOPT - Powerful Nonlinear Optimization

CONOPT is a feasible path solver based on advanced active set methods, particularly suited for large and sparse models. With over 30 years of development history, it is widely used in engineering, economics, and operations research.

Originally developed by ARKI Consulting & Development A/S in Denmark, CONOPT was acquired by GAMS in 2024. CONOPT continues to evolve as a state-of-the-art tool for solving **large-scale, continuous optimization models**.

The original developer Arne Stolbjerg Drud has kindly written a comprehensive account of the CONOPT Story.

Key Features

Large Scale NLP
Excels in smooth, large NLP models with complex constraints.

Feasible Path Solver
Robust, stable, and proven solver for challenging problems.

Smart Model Preprocessing
Automatically detects and removes redundant equations and variables for better solve efficiency.

$f''(x)$ **Second-Order Derivative Support**
Can leverage second-order information for fast convergence and better accuracy in suitable models.

Flexible licensing
Available for academic and commercial use with flexible options.

Supported Platforms
Integrates seamlessly with modeling systems (AIMMS, AMPL, GAMS, GAMSPy, Pyomo) and programming languages (C++, C, Fortran, Java, Python).

Documentation   Using the CONOPT API   Search

CONOPT Documentation Page

CONOPT is a general-purpose optimization system for large-scale nonlinear models. CONOPT employs a feasible path algorithm tha is based on active set methods, particularly suited to large, sparse models. CONOPT can be used as an dedicated non-linear solver, either stand-alone or integrated with in a larger system, by interfacing directly through the API. When using the CONOPT API, the user must define routines to evaluate non-linear expressions. This can achieved through hand-coded evaluation routines, or the integration with automatic differentiation packages, such as ADOL-C and CppAD. As an alternative, CONOPT can be used through modeling systems such as AIMMS, AMPL, GAMS and LINGO.

This documentation provides numerous resources for getting started with CONOPT. The Introduction provides an overview of the CONOPT library and some details on how to build your own project. Following this, it is recommended to work through the Tutorial and look at the Examples using CONOPT. Finally, the CONOPT API (Fortran, C, C++, Python and Java) is detailed in Public API.

Introduction

- The CONOPT algorithm
- Library Organization
- The Mathematical Model
- Overview over CONOPT routines – Fortran users
- Overview over CONOPT routines – C users
- ChangeLog

Tutorial and Examples

- Tutorial
- Examples using CONOPT

Conclusion

Summary

- Under GAMS, CONOPT has undergone significant modernization and refinement
- Renewed drive for better performance and reliability
- API support opens doors to new workflows beyond traditional modeling systems
- Flexible deployment enables both educational and enterprise adoption
- Positioned for growth in academic research, teaching, and real-world applications

What's Next?

CONOPT nonlinear optimization

- Continued focus on platform support, usability, and documentation
- Key roadmap initiatives:
  - Comprehensive, user-friendly documentation enhancements
  - Expanded ecosystem integration:
    1. nearly completed SCIP support
    2. JuMP integration in development
  - Performance tuning and feature development guided by community input
  - Advanced tools for diagnostics, debugging, and model performance analysis
  - Alternative interface that allows for automatic differentiation without external tools

Academic Program

🎓 Free for Academia

- A full-featured CONOPT license is now available at no cost for academic research and teaching
- No limitations on model size or capabilities
- Also included in the GAMSPy academic package
- It's easy to get with the program:
  - Via the portal: academic.gams.com
  - Via email: sales@gams.com
- Designed to foster education and experimentation with nonlinear optimization

CONOPT
nonlinear optimization

Contact Us

www.conopt.com

academic.gams.com

sales@gams.com

Visit us at our booth!

https://www.linkedin.com/company/gams-development

CONOPT
nonlinear optimization