MINLP Solver Software

Michael R. Bussieck[1] and Stefan Vigerske[2]

February 21, 2012

Abstract

In this article we give a brief overview of the start-of-the-art in software for the solution of mixed integer nonlinear programs (MINLP). We establish several groupings with respect to various features and give concise individual descriptions for each solver. The provided information may guide the selection of a best solver for a particular MINLP problem.

Keywords: mixed integer nonlinear programming, solver, software, MINLP, MIQCP

1 Introduction

The general form of an MINLP is

$$\begin{aligned} \text{minimize} \quad & f(x,y) \\ \text{subject to} \quad & g(x,y) \leq 0 \\ & x \in X \\ & y \in Y \quad \text{integer} \end{aligned} \tag{P}$$

The function $f : \mathbb{R}^{n+s} \to \mathbb{R}$ is a possibly nonlinear objective function and $g : \mathbb{R}^{n+s} \to \mathbb{R}^m$ a possibly nonlinear constraint function. Most algorithms require the functions f and g to be continuous and differentiable, but some may even allow for singular discontinuities. The variables x and y are the decision variables, where y is required to be integer valued. The sets $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^s$ are bounding-box-type restrictions on the variables. Additionally to integer requirements on variables, other kinds of discrete constraints are commonly used. These are, e.g., special-ordered-set constraints (only one (SOS type 1) or two consecutive (SOS type 2) variables in an (ordered) set are allowed to be nonzero) [8], semicontinuous variables (the variable is allowed to take either the value zero or a value above some bound), semiinteger variables (like semicontinuous variables, but with an additional integer restriction), and indicator variables (a binary variable indicates whether a certain set of constraints has to be enforced). In all cases it is possible to reformulate such constraints into a standard form by introducing additional variables and linear constraints. The purely continuous case ($s = 0$) is not considered here, c.f. the chapter titled “NLP Software” for an overview.

Computational tractability depends significantly on whether the functions $f(x, y)$ and $g(x, y)$ are convex or not, c.f. Sec. 3.3. In this chapter, we say an MINLP is *convex* if both

---

[1] GAMS Development Corp., 1217 Potomac St, NW Washington, DC 20007, USA, `mbussieck@gams.com`

[2] Humboldt–Universität zu Berlin, Department of Mathematics, Unter den Linden 6, 10099 Berlin, Germany, `stefan@math.hu-berlin.de`

$f(x, y)$ and $g(x, y)$ are convex over $X \times Y$. Otherwise the MINLP is said to be *nonconvex*. Note that some solvers for convex MINLPs can also be applied under less strict notions of convexity, e.g., to the case where the set defined by the constraints $g(x, y) \leq 0$ is convex, or where the objective function and constraints are only pseudo-convex [76]. (A differentiable function $h : X \to \mathbb{R}$ is *pseudo-convex* on a convex set $X \subseteq \mathbb{R}^n$ if for every $x, y \in X$ with $h(x) < h(y)$ it follows that $\langle \nabla h(y), x - y \rangle < 0$. An important property of a pseudo-convex function is the convexity of its level-sets.)

2 History

To the best of our knowledge, the earliest commercial software package that could solve MINLP problems was SCICONIC in the mid 1970's [7, 30, 64]. Rather than handling nonlinearities directly, linked Special-Ordered-Set variables provided a mechanism to represent low dimensional nonlinear terms by a piecewise linear approximation and thus allowed to use mixed-integer linear programming (MIP) to obtain solutions to an approximation of the MINLP. In the mid 1980's Grossmann and Kocis developed DICOPT, a general purpose algorithm for convex MINLP based on the outer approximation method [19]. Since then, a number of academic and commercial codes for convex MINLP have emerged, either based on outer approximation using MIP relaxations [19], an integration of outer approximation into a linear programming (LP) relaxation based branch and cut [58], or nonlinear programming (NLP) relaxation based branch and bound algorithms [43]. For the global solution of nonconvex MINLP, the first general purpose solvers were ALPHABB, BARON, and GLOP, all based on convexification techniques for nonconvex constraints [5, 60, 65, 67]. See also Section 3.3 for a small discussion of MINLP algorithms.

3 Groupings

3.1 Embedded vs. independent

Due to the high complexity of MINLP and the wide range of applications that can be modeled as MINLPs, it is sometimes desirable to customize the MINLP solver for a specific application in order to achieve good computational performance [12, 13, 22]. Further, MINLP solvers are often built by combining LP, MIP, and NLP solvers. These are two main reasons for tightly integrating some MINLP solvers into modeling systems (general systems like AIMMS [59], AMPL [31], and GAMS [32] or vendor specific systems like FICO XPRESS-MOSEL [25], LINGO [63], and OPL [17]). For example, the AIMMS Outer Approximation solver AOA allows modifications of its algorithm by the user. Further, the solvers DICOPT and SBB are exclusively available for GAMS users since they revert to MIP and NLP solvers in the GAMS system for the solution of subproblems. Also for an efficient use of the solver OQNLP it is preferable to use one of the GAMS NLP solvers.

On the other side, there are many solvers that can be used independently of a modeling system, even though they may still require the presence of an LP, MIP, or NLP solver plugin. However, often also these "independent" solvers are used within a modeling system, since the modeling system typically provides evaluators for nonlinear functions, gradients, and Hessians and gives easy access to algebraic information about the problem.

3.2 Extending MIP vs. extending NLP vs. starting from scratch

MINLP solvers are seldom developed completely from scratch. In many cases, an MIP or an NLP solver builds the basis for an extension towards MINLP. Solvers which can be categorized as extending an MIP solver towards handling of nonlinear objectives and constraints are BONMIN, COUENNE, CPLEX, FICO XPRESS-OPTIMIZER, FILMINT, LINDOAPI without global solver option, MOSEK, and SCIP. On the other hand, solvers where an NLP solver was extended to handle integrality restrictions are BNB, FICO XPRESS-SLP, FMINCONSET, KNITRO, MILANO, MINLP_BB, MISQP, OQNLP, and SBB.

Finally, there is a group of solvers which were more-or-less developed from scratch, but which may solve LP, MIP, NLP, or MINLP subproblems. In this category we have ALPHABB, ALPHAECP, AOA, BARON, DICOPT, GLOMIQO, LAGO, LINDOAPI, and MIDACO.

3.3 Algorithms

Algorithms for solving MINLPs are often build by combining algorithms from Linear Programming, Integer Programming, and Nonlinear Programming, e.g., branch and bound, outer approximation, local search, global optimization. We refer to the chapters titled "Fundamental Techniques", "Nonlinear Programming and Global Optimization", and "Models and Algorithms" for an introduction into these topics.

Most of the solvers implement one (or several) of three algorithmic ideas to tackle MINLPs. First, there are branch and bound solvers that use NLP relaxations: ALPHABB, BNB, BONMIN (in B-BB mode), CPLEX, FICO XPRESS-OPTIMIZER, FICO XPRESS-SLP (in "SLP within MIP" mode), FMINCONSET, KNITRO, LINDOAPI without global solver option, MILANO, MINLP_BB, MOSEK, and SBB. For all these solvers except ALPHABB, the NLP relaxation is obtained by relaxing the integrality restriction in (P), thus it may be a nonconvex NLP. Since the NLP solver used to solve the NLP relaxation usually ensures only local optimal solutions, these solvers work as heuristics in case of a nonconvex MINLP. For the solver ALPHABB, however, a convex NLP relaxation is generated by using convex underestimators for the functions $f(x, y)$ and $g(x, y)$ in (P). This solver can therefore be applied also to nonconvex MINLPs.

As an alternative to relaxing integrality restrictions and keeping nonlinear constraints, some solvers keep the integrality constraints and instead replace the nonlinear functions $f(x, y)$ and $g(x, y)$ by a linear relaxation. In an outer-approximation algorithm [19, 27], a relaxation is obtained by using gradient-based linearizations of $f(x, y)$ or $g(x, y)$ at solution points of NLP subproblems. The resulting MIP relaxation is then solved by an MIP solver. Solvers in this class are AOA, BONMIN (in B-OA mode), DICOPT, MISQP (with OA extension), and FICO XPRESS-SLP (in "MIP within SLP" mode). Since gradient-based linearizations yield an outer-approximation only for convex MINLPs, these solvers are only applicable for convex MINLPs. In contrast to outer-approximation based algorithms, an extended cutting plane algorithm solves a sequence of MIP relaxations which encapsulate optimal solutions of (P) by cutting planes and supports of $f(x, y)$ rather than outer-approximating the whole feasible region of (P) [75]. This algorithm is implemented by the solver ALPHAECP, which can be applied to convex as well as pseudo-convex MINLPs.

A third class of solvers are those which integrate the linearization of $f(x, y)$ and $g(x, y)$ into the branch and cut process [58]. Thus, here an LP relaxation is successively solved, new linearizations of $f(x, y)$ and $g(x, y)$ are generated to improve the relaxation, and integrality constraints are enforced by branching on the y variables. Solvers which use gradient-based linearizations are AOA, BONMIN (in B-QG mode) and FilMINT.

Since the use of gradient-based linearizations in a branch and cut algorithm ensures global solutions only for convex MINLPs, solvers for nonconvex MINLPs use convexification techniques to compute linear underestimators of a nonconvex function. However, the additional convexification step may require to branch also on continuous variables in nonconvex terms (so called *spatial* branching). Such a branch and cut algorithm is implemented by BARON, Couenne, GloMIQO, LaGO, LindoAPI, and SCIP.

The remaining solvers implement a different methodology. BONMIN (in B-Hyb mode) alternates between LP and NLP relaxations during one branch and bound process. MISQP integrates the handling of integrality restrictions into the solution of a nonlinear program via sequential quadratic programming, i.e., it ensures that $f(x,y)$ and $g(x,y)$ are only evaluated at points where y is integral. MIDACO applies an extended ant colony optimization method and can use MISQP as a local solver. Finally, OQNLP applies a randomized approach by sampling starting points and fixings of integer variables for the solution of NLP subproblems.

3.4 Capabilities

Not every solver accepts general MINLPs as input. Solvers that currently handle only MINLPs where the objective function and constraints are quadratic (so-called MIQCPs) or second order cone (SOC) programs are CPLEX, FICO Xpress-Optimizer, GloMIQO, and MOSEK. All solvers support convex quadratic functions. Further, nonconvex quadratic functions that involve only binary variables are supported by CPLEX and FICO Xpress-Optimizer. Quadratic constraints that permit a SOC representation are supported by CPLEX. SOC constraints are supported by MOSEK. CPLEX and GloMIQO support quadratic constraints in any form, but CPLEX ensures global optimality only for the cases mentioned before.

Solvers that guarantee global optimal solutions for convex general MINLPs are AlphaECP, AOA, bnb, BONMIN, DICOPT, FICO Xpress-SLP, FilMINT, fminconset, Knitro, LaGO, LindoAPI without global solver option, MILANO, MINLP_BB, MISQP with OA extension, and SBB. In case of a nonconvex MINLP, these solvers can still be used as a heuristic. Especially branch and bound based algorithms that use NLPs for bounding often find good solutions also for nonconvex problems, while pure outer approximation based algorithms may easily run into infeasible LP or MIP relaxations due to wrong cutting planes. Note, that AlphaECP ensures global optimal solutions also for pseudo-convex MINLPs.

Solvers that also guarantee global optimality for nonconvex general MINLPs require an algebraic representation of the functions $f(x,y)$ and $g(x,y)$ for the computation of convex underestimators. That is, for each function a representation as a composition of basic arithmetic operations and functions (addition, multiplication, power, exponential, trigonometric, ...) on constants and variables need to be provided. The solvers alphaBB, BARON, Couenne, LindoAPI, and SCIP belong into this category.

MIDACO, MISQP, and OQNLP can handle general MINLPs, but do not guarantee global optimality even on convex problems.

4 MINLP solvers

In the following we briefly discuss individual solvers for MINLPs. We have excluded solvers from this list that are clearly no longer available (e.g., SCICONIC). The solvers listed below have different levels of reliability and activity with respect to development and maintenance. Wide availability through modeling systems and other popular software indicates that a solver has reached a decent level of maturity. Hence, in this list, we mention availability (e.g., open source, standalone binary, interfaces to general modeling systems) in addition to a solver's developer, capability, and algorithmic details. Table 1 summarizes the list of solvers and indicates for each solver the availability via AIMMS, AMPL, GAMS, and the NEOS server [16].

alphaBB (α-Branch-and-Bound) [3, 5]. This solver has been developed by the research group of C. A. Floudas at the Computer-Aided Systems Laboratory of Princeton University. It is available to this group and to their collaborators.

alphaBB can be applied for convex and nonconvex MINLPs. It implements a branch and bound algorithm that utilizes convex NLPs for bounding. Convex envelopes and tight convexifications are obtained for specially structured nonconvex terms (e.g., bilinear, trilinear, multilinear, univariate concave, edge concave, generalized polynomials, fractional), and α-convex underestimators for general twice continuously differentiable functions. The latter are determined by adding a non-positive convex function to the original nonconvex function such that the Hessian of the sum is guaranteed to be positive semidefinite (PSD) [4]. Various interval arithmetic based techniques for estimating rigorous bounds on the minimal eigenvalue of the Hessian of the original nonconvex function are available.

AlphaECP (α-Extended Cutting Plane) [74, 76]. This solver has been developed by the research group of T. Westerlund at the Process Design and Systems Engineering Laboratory of the Åbo Akademi University, Finland. It is available as a commercial solver within GAMS.

AlphaECP ensures global optimal solutions for convex and pseudo-convex MINLPs. It generates and successively improves an MIP outer approximation of a neighborhood of the set of optimal solutions of (P) and can solve NLP subproblems to find feasible solutions early. The MIP is here refined by linearizing nonlinear constraints at solutions of the MIP outer approximation. By shifting hyperplanes, pseudo-convex functions can also be handled.

AOA (AIMMS Outer Approximation) [59]. This solver has been developed by Paragon Decision Technology. AOA is available as an "open solver" inside AIMMS. The open solver approach allows the user to customize the algorithm for a specific application.

AOA ensures global optimal solutions only for convex MINLPs. It generates and successively improves an MIP outer approximation of (P) and can solve NLP subproblems to find feasible solutions early. In contrast to AlphaECP, AOA constructs an MIP outer approximation of the feasible region of (P) by linearizing nonlinear functions in solutions of NLP subproblems [19]. Since for a nonconvex constraint such a linearization may not be valid, the MIP relaxation is modified such that the corresponding hyperplane is allowed to move away from its support point. Recently, also a branch and bound algorithm that utilizes LPs for bounding [58] has been added to AOA.

BARON (Branch-And-Reduce Optimization Navigator) [67, 68]. This solver was originally developed by the group of N.V. Sahinidis at the University of Illinois at Urbana-Champaign and is currently developed by N.V. Sahinidis at Carnegie Mellon University and M. Tawarmalani at Purdue University. It is available as a commercial solver within AIMMS and GAMS.

BARON can be applied to convex and nonconvex MINLPs. It implements a spatial branch and bound algorithm that utilizes LPs for bounding. The linear outer-approximation is based on a reformulation of (P) that it constructed (by adding auxiliary variables) in a way that it contains only nonconvex terms for which a convex underestimator (or concave overestimator) is known. The algorithm is enhanced by using advanced box reduction techniques and new convexification techniques for quadratic functions [6]. Further, BARON is able to use NLP relaxations for bounding [34], even though this option is not encouraged.

bnb (Branch 'n Bound) [40]. This solver has been developed by K. Kuipers of the Department of Applied Physics at the University of Groningen. It is available as Matlab [51] source.

bnb ensures global optimal solutions for convex MINLPs. It implements a branch and bound algorithm utilizing nonlinear relaxations for the bounding step [43]. The NLPs are solved by the Matlab Optimization Toolbox routine fmincon.

BONMIN (Basic Open-source Nonlinear Mixed Integer Programming) [11]. This open-source solver has been developed primarily by P. Bonami in a cooperation of Carnegie Mellon University and IBM Research, now at University Marseille. It is available in source code and as standalone binaries from COIN-OR (Computational Infrastructure for Operations Research) [48], has an AMPL interface, and is distributed as a free solver within GAMS.

BONMIN ensures global optimal solutions only for convex MINLPs. It implements (at least) four algorithms: B-OA is an outer-approximation algorithm that generates and successively improves an MIP outer approximation of (P) [19], B-QG is a branch and bound algorithm that utilizes LPs for bounding [58], B-BB is a branch and bound algorithm that utilizes NLPs for bounding [43], and B-Hyb is a hybrid of B-QG and B-BB which alternates between LP and NLP relaxations for bounding. BONMIN is implemented on top of the MIP solver CBC [29] and can use filterSQP [28] and IPOPT [73] as NLP solvers.

Couenne (Convex Over and Under ENvelopes for Nonlinear Estimation) [9]. This open-source solver has been developed primarily by P. Belotti, originally in a cooperation of Carnegie Mellon University and IBM Research, and now at Clemson University. It is available in source code and as standalone binaries from COIN-OR, has an AMPL interface, and is distributed as a free solver within GAMS.

Couenne ensures global optimal solutions for convex and nonconvex MINLPs. It implements a spatial branch and bound algorithm that utilizes LPs for bounding. Similar to BARON, the linear outer-approximation is generated from a reformulation of (P). The algorithm is enhanced by box reduction techniques, disjunctive cuts, MINLP heuristics, and symmetry handling. Couenne is implemented on top of BONMIN.

CPLEX [38]. This solver has been developed by CPLEX Optimization, Inc. (later acquired by ILOG and recently acquired by IBM). It is available as standalone binaries and as a component in many modeling systems.

CPLEX can solve convex MIQCPs. For models that only have binary variables in the potentially indefinite quadratic matrices, CPLEX automatically reformulates the problem to an equivalent MIQCP with PSD matrices. It implements a branch and bound algorithm that utilizes LPs or QCPs for bounding. Recently, also an option to solve general nonconvex MIQCPs by a branch and bound algorithm that utilizes NLPs for bounding [43] has been added, but global optimality is not guaranteed for this case.

DICOPT (Discrete and Continuous Optimizer) [32, 39]. This solver has been developed by the research group of I. E. Grossmann at the Engineering Research Design Center at Carnegie Mellon University. It is available as a commercial solver within GAMS.

DICOPT ensures global optimal solutions for convex MINLPs. Starting with the NLP relaxation (obtained from (P) by relaxing the integer requirement on y), it alternates between solving MIP outer approximations and NLP subproblems of (P) to compute lower and upper bounds [19]. To accommodate nonconvex MINLPs, nonlinear equality constraints are relaxed by replacing them with inequalities where the linearizations of the nonlinear functions are allowed to move away from their support point by the use of slack variables and through an augmented penalty function in the MIP relaxation. Since in this case valid lower bounds cannot be obtained, the termination is based on lack of improvement in the objective of the NLP subproblem.

FICO Xpress-Optimizer [26]. This solver has been developed by Dash Optimization (later acquired by FICO). It is available as standalone binaries and as a component in many modeling systems.

FICO Xpress-Optimizer can solve convex MIQCPs. For models that only have binary variables in the potentially indefinite quadratic matrices, FICO Xpress-Optimizer automatically reformulates the problem to an equivalent MIQCP with PSD matrices. It implements a branch and bound algorithm that utilizes QCPs for bounding.

FICO Xpress-SLP [24]. This solver has been developed by Dash Optimization (later acquired by FICO). It is available as standalone binaries and as a FICO Xpress-Mosel module [25].

FICO Xpress-SLP ensures global optimal solutions for convex MINLPs. It implements three algorithms: The (default) “SLP within MIP” variant is a branch and bound algorithm that utilizes NLPs for bounding [43]. The NLP subproblems are solved by Successive Linear Programming (SLP). Solving MIPs as subproblems of the SLP algorithm leads to the “MIP within SLP” variant. It is comparable with an MIP relaxation based outer-approximation algorithm [19]. A third variant (“SLP then MIP”) solves first an NLP relaxation (by SLP), then an MIP relaxation, and finally an NLP subproblem to obtain a feasible solution to (P) [24]. To accommodate also nonconvex constraints, in all variants, the hyperplanes obtained from gradient-based linearizations in SLP are allowed to move away from their support point.

FilMINT (Filter-Mixed Integer Optimizer) [1]. This solver has been developed by the research groups of S. Leyffer at the Laboratory for Advanced Numerical Simulations of Argonne National Laboratory and J. T. Linderoth at the Department of Industrial and Systems Engineering of Lehigh University. It provides an AMPL interface.

FilMINT ensures global optimal solutions only for convex MINLPs. It implements a branch and bound algorithm that utilizes LPs for bounding [58], where different strategies for choosing the linearization point for the nonlinear functions are available. Further, FilMINT includes several variants of disjunctive cutting planes for convex MINLP and a feasibility pump. FilMINT is implemented on top of the MIP solver MINTO [56] and the NLP solver filterSQP [28].

fminconset [66]. This solver had been developed by I. Solberg at the Department of Engineering Cybernetics of the University of Trondheim (now NTNU). It is available as Matlab source.

fminconset ensures global optimal solutions for convex MINLPs. It implements a branch and bound algorithm utilizing nonlinear relaxations for the bounding step [43]. The NLPs are solved by the Matlab Optimization Toolbox routine fmincon.

GloMIQO [52, 53]. This solver has been developed by R. Misener and C. A. Floudas at the Computer-Aided Systems Laboratory of Princeton University. It is available as a commercial solver within GAMS.

GloMIQO ensures global optimal solutions for convex and nonconvex MIQCPs. It implements a spatial branch-and-bound algorithm that utilizes MIPs for bounding and employs a large collection of convexification and bound tightening techniques for quadratic constraints.

Knitro [15]. This solver has been developed by Ziena Optimization, Inc. It is available as standalone binary and as a component in many modeling systems.

Knitro ensures global optimal solutions for convex MINLPs. MINLPs are solved by branch and bound, where both linear or nonlinear problems can be used for the bounding step [43, 58].

LaGO (Lagrangian Global Optimizer) [57]. This open-source solver had been developed by the research group of I. Nowak at the Department of Mathematics of Humboldt University Berlin. It is available in source code from COIN-OR and provides AMPL and GAMS interfaces.

LaGO ensures global optimal solutions for convex MINLPs and nonconvex MIQCPs. It implements a spatial branch and bound algorithm utilizing a linear relaxation for the bounding step. The relaxation is obtained by linearizing convex functions, underestimating quadratic nonconvex functions, and approximating nonconvex nonquadratic functions by quadratic ones.

LindoAPI [47, 45]. This solver library has been developed by LINDO Systems, Inc. It is available within the LINDO environment [47], LINGO [63], What's*Best!* [46], and as a commercial solver within GAMS.

LindoAPI ensures global optimal solutions for convex and nonconvex MINLPs. It implements a branch and cut algorithm that utilizes LPs for bounding [33, 45]. Branching is performed for subproblems that are not provably infeasible and where nonconvex constraints are present or the LP relaxation has a fractional solution. LindoAPI can also handle some nonsmooth or discontinuous functions like `abs(x)`, `floor(x)`, and `max(x,y)`.

Additionally, LindoAPI allows to disable the global solver components, by what the MIP solver is used together with nonlinear relaxations for the bounding step [43]. This option still ensures global optimal solutions for convex MINLPs. It was the first commercially available solver implementing a branch and bound algorithm utilizing nonlinear relaxations for bounding. The NLP relaxations are solved by CONOPT [18, 32].

MIDACO (Mixed Integer Distributed Ant Colony Optimization) [61, 62]. This solver has been developed by M. Schlüter at the Theoretical & Computational Optimization Group of the University of Birmingham. It works as a library with Matlab, C/C++, and Fortran interfaces and is available from the author on request.

MIDACO can be applied to convex and nonconvex MINLPs. It implements an extended ant colony search method based on an oracle penalty function and can be combined with MISQP as solver for local searches in (P). It targets applications where the problem formulation is unknown ($f(x,y)$ and $g(x,y)$ are black-box functions) or involves critical properties like nonconvexities, discontinuities, flat spots, or stochastic distortions. Further, MIDACO can exploit distributed computer architectures by parallelizing function evaluation calls.

MILANO (Mixed-Integer Linear and Nonlinear Optimizer) [10]. This solver is developed by H. Y. Benson at the Department of Decision Sciences of Drexel University. It is still in development and available as Matlab source.

MILANO ensures global optimal solutions for convex MINLPs. It implements a branch and bound algorithm utilizing nonlinear relaxations for the bounding step [43]. The NLPs are solved by LOQO [70], where special emphasis is put on how to warmstart this interior-point solver.

MINLP_BB (Mixed Integer Nonlinear Programming Branch-and-Bound) [43]. This solver had been developed by R. Fletcher and S. Leyffer at the University of Dundee. It provides an AMPL interface and is available for Matlab via the TOMLAB Optimization Environment [37].

MINLP_BB ensures global optimal solutions for convex MINLPs. It implements a branch and bound algorithm utilizing nonlinear relaxations for the bounding step [43]. The NLPs are solved by filterSQP.

MISQP (Mixed Integer Sequential Quadratic Programming) [20, 21]. This solver has been developed by the research group of K. Schittkowski at the Department of Computer Science of the University of Bayreuth. It works as a standalone library with a Fortran interface.

MISQP can be applied to convex and nonconvex MINLPs, but assumes that the values of $f(x,y)$ and $g(x,y)$ do not change drastically as a function of y. MISQP implements a modified sequential quadratic programming (SQP) method, where functions are only evaluated at points (x,y) with y integer. It targets applications where the evaluation of $f(x,y)$ or $g(x,y)$ may be expensive. Additionally, a combination with outer-approximation [19] that guarantees convergence for convex MINLPs is available [41].

MOSEK [55]. This solver has been developed by MOSEK ApS. It is available as a standalone binary, has AMPL and Matlab interfaces, and is distributed as a commercial solver within AIMMS and GAMS.

MOSEK can be applied to convex MIQCPs and to mixed-integer conic programs. It implements a branch and bound method that utilizes QCPs or SOC programs for bounding [58].

OQNLP (OptQuest Nonlinear Programming) [32, 69]. This solver has been jointly developed by OptTek Systems, Inc. and Optimal Methods, Inc. It is available as a standalone library, for MATLAB via the TOMLAB Optimization Environment, and is distributed as a commercial solver within GAMS.

OQNLP is a heuristic that can be applied to any MINLP. It implements a multistart scatter search algorithm which solves NLP subproblems with fixed discrete variables.

SBB (Simple Branch-and-Bound) [32]. This solver had been developed by ARKI Consulting and Development A/S. It is available as a commercial solver within GAMS.

SBB ensures global optimal solutions for convex MINLPs. It implements a branch and bound algorithm utilizing nonlinear relaxations for the bounding step [43]. The NLP relaxations are solved by one (or several) of the NLP solvers available with GAMS. Using the GAMS Branch-Cut-and-Heuristic facility [13], SBB allows the user to implement a model-specify heuristic in the GAMS language.

SCIP (Solving Constraint Integer Programs) [2, 72]. This solver has been developed by the Optimization Department at the Zuse Institute Berlin and its collaborators. For academic institutions, it is available in source code and as standalone binary and is distributed within GAMS.

SCIP ensures global optimal solutions for convex and nonconvex MINLPs. It implements a spatial branch and bound algorithm that utilizes LPs for the bounding step. Similar to BARON, the outer-approximation is generated from a reformulation of the MINLP. Additionally, SCIP includes large-neighborhood search heuristics and a new sub-MIP MINLP heuristic.

5 Outlook and Summary

Combining discrete and nonlinear optimization results in a rich modeling paradigm applicable to many real world optimization problems. At the same time, mixed integer nonlinear programming represents a theoretically and computationally challenging problem class and hence provides many interesting research opportunities. Software for solving MINLP models facilitates co-operation between research and application and explains the popularity and increased level of activity around MINLP.

While state-of-the-art MIP solvers typically implement advanced automatic reformulation and preprocessing algorithms, such techniques are less commonly available in MINLP solvers, and in a limited form. Therefore, the modeler's choice of problem formulation is still very important when solving an MINLP. However, software for guided automatic model reformulations and relaxations has recently been developed. LOGMIP [71], one of the first systems available, translates an MINLP with disjunctions into a standard MINLP by applying bigM and convex hull reformulations [36]. More recently, frameworks like GAMS/EMP (Extended Mathematical Programming) [23] and ROSE (Reformulation/Optimization Software Engine) [44] provide a growing toolbox for reformulating MINLPs. Other recent activities, like libMC [54] and parts of MINOTAUR (Mixed-Integer Nonconvex Optimization Toolbox – Algorithms, Underestimators, Relaxations) [50, 49] focus on (convex) relaxations for (nonconvex) MINLP.

Another important area is the collection and dissemination of MINLP models. Instance collections like MacMINLP [42] and MINLPLib [14] provide valuable test cases for solver developers. The new Cyber-Infrastructure for MINLP [35] features a growing library of problems with high level model descriptions, reformulations, and problem instances.

In this paper we have given a concise description of the state-of-the-art in MINLP solvers and have established several groupings with respect to various features of the software. We hope that these groupings and the individual descriptions give sufficient information to guide the selection of the best solver for a particular MINLP problem.

Acknowledgments. We thank Oliver Bastert, Pietro Belotti, Zsolt Csizmadia, Steve Dirkse, Arne Drud, Christodoulos Floudas, Ignacio Grossmann, Marcel Hunting, Ed Klotz, Nikolaos Sahinidis, Martin Schlüter, Linus Schrage, and Tapio Westerlund for their invaluable comments. The second author was supported by the DFG Research Center Matheon *Mathematics for key technologies* in Berlin.

References

[1] K. Abhishek, S. Leyffer, and J.T. Linderoth. FilMINT: An outer-approximation-based solver for nonlinear mixed integer programs. *INFORMS Journal On Computing*, 22(4):555–567, 2010.

[2] T. Achterberg. SCIP: Solving Constraint Integer Programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[3] C.S. Adjiman, I.P. Androulakis, and C.A. Floudas. Global optimization of mixed-integer nonlinear problems. *Journal of the American Institute of Chemical Engineers*, 46:1769–1797, 2000.

[4] C.S. Adjiman and C.A. Floudas. Rigorous convex underestimators for general twice-differentiable problems. *Journal of Global Optimization*, 9:23–40, 1996.

[5] I.P. Androulakis, C.D. Maranas, and C.A. Floudas. αBB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7:337–363, 1995.

[6] X. Bao, N.V. Sahinidis, and M. Tawarmalani. Multiterm polyhedral relaxations for nonconvex, quadratically-constrained quadratic programs. *Optimization Methods and Software*, 24:485–504, 2009.

[7] E.M.L. Beale. Branch and bound methods for numerical optimization of non-convex functions. In M.M. Barritt and D. Wishart, editors, *COMPSTAT 80 Proceedings in Computational Statistics*, pages 11–20, Vienna, 1980. Physica-Verlag.

[8] E.M.L. Beale and J.A. Tomlin. *Special Facilities in a General Mathematical Programming System for Nonconvex Problems Using Ordered Sets of Variables*, pages 447–454. Number 69 in Operational Research. Tavistock Publishing, London, 1970.

[9] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5):597–634, 2009.

[10] H. Y. Benson. Mixed integer nonlinear programming using interior-point methods. *Optimization Methods and Software*, 26(6):911–931, 2011.

[11] 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. *Discrete Optimization*, 5:186–204, 2008.

[12] C. Bragalli, C. D'Ambrosio, J. Lee, A. Lodi, and P. Toth. On the optimal design of water distribution networks: a practical MINLP approach. *Optimization and Engineering*, to appear, 2011.

[13] M.R. Bussieck. Introduction to GAMS Branch-and-Cut Facility. Technical report, GAMS Development Corp., 2003. `http://www.gams.com/docs/bch.htm`.

[14] M.R. Bussieck, A.S. Drud, and A. Meeraus. MINLPLib - A Collection of Test Models for Mixed-Integer Nonlinear Programming. *INFORMS Journal on Computing*, 15(1):114–119, 2003.

[15] R.H. Byrd, J. Nocedal, and R.A. Waltz. KNITRO: An integrated package for nonlinear optimization. In G. di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, pages 35–59. Springer, 2006.

[16] J. Czyzyk, M. Mesnier, and J. Moré. The NEOS server. *IEEE Journal on Computational Science and Engineering*, 5:68–75, 1998. `http://neos.mcs.anl.gov`.

[17] T. Dong. Efficient modeling with the IBM ILOG OPL-CPLEX Development Bundles. `http://www-01.ibm.com/software/integration/optimization/cplex-dev-bundles/library`, 2009.

[18] A. Drud. CONOPT - a large-scale GRG code. *INFORMS Journal on Computing*, 6:207–216, 1992.

[19] M.A. Duran and I.E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

[20] O. Exler, T. Lehmann, and K. Schittkowski. A comparative study of SQP-type algorithms for nonlinear and nonconvex mixed-integer optimization. `http://www.ai7.uni-bayreuth.de/minlp_comp_study.htm`, 2011.

[21] O. Exler and K. Schittkowski. A trust region SQP algorithm for mixed-integer nonlinear programming. *Optimization Letters*, 1(3):269–280, 2007.

[22] T. Farkas, B. Czuczai, E. Rev, and Z. Lelkes. New MINLP model and modified outer approximation algorithm for distillation column synthesis. *Industrial & Engineering Chemistry Research*, 47:3088–3103, 2008.

[23] M.C. Ferris, S.P. Dirkse, J.-H. Jagla, and A. Meeraus. An extended mathematical programming framework. *Computers & Chemical Engineering*, 33(12):1973–1982, 2009. FOCAPO 2008 - Selected Papers from the Fifth International Conference on Foundations of Computer-Aided Process Operations.

[24] FICO. *Xpress-SLP Program Reference Manual*, 1.41 edition, 2008. `http://www.fico.com/xpress`.

[25] FICO. Xpress-Mosel 7.0, 2009. `http://www.fico.com/xpress`.

[26] FICO. *Xpress-Optimizer Reference manual*, 20.0 edition, 2009. `http://www.fico.com/xpress`.

[27] R. Fletcher and S. Leyffer. Solving Mixed Integer Nonlinear Programs by Outer Approximation. *Mathematical Programming*, 66(3):327–349, 1994.

[28] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–270, 2002.

[29] J.J.H. Forrest. COIN-OR Branch and Cut. `http://projects.coin-or.org/Cbc`.

[30] J.J.H. Forrest and J.A. Tomlin. Branch and bound, integer, and non-integer programming. *Annals of Operations Research*, 149(1):81–87, 2007.

[31] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* Duxbury Press, Brooks/Cole Publishing Company, 1993.

[32] GAMS Development Corp. *GAMS – The Solver Manuals.* Washington DC, 2011.

[33] C. Gau and L. Schrage. Implementation and testing of a branch-and-bound based method for deterministic global optimization: Operations research applications. In C.A. Floudas and P.M. Pardalos, editors, *Frontiers in Global Optimization*, Nonconvex Optimization and Its Applications, pages 145–164. Springer, 2003.

[34] V. Ghildyal and N.V. Sahinidis. Solving global optimization problems with BARON. In A. Migdalas, P.M. Pardalos, and P. Värbrand, editors, *From Local to Global Optimization*, volume 53 of *Nonconvex Optimization and Its Applications*, chapter 10, pages 205–230. Springer, 2001.

[35] I. E. Grossmann and J. Lee. Cyberinfrastructure for mixed-integer nonlinear programming. *SIAG/OPT Views-and-News*, 22(1):8–12, March 2011.

[36] I. E. Grossmann and S. Lee. Generalized disjunctive programming: Nonlinear convex hull relaxation and algorithms. *Computational Optimization and Applications*, (26):83–100, 2003.

[37] K. Holmström. The Tomlab Optimization Environment in Matlab. *Advanced Modeling and Optimization*, 1:47–69, 1999. `http://tomopt.com/tomlab`.

[38] IBM. CPLEX. `http://www.ibm.com/software/integration/optimization/cplex`.

[39] G.R. Kocis and I.E. Grossmann. Computational experience with DICOPT: Solving MINLP problems in process systems engineering. *Computers & Chemical Engineering*, 13:307–315, 1989.

[40] K. Kuipers. bnb. `http://www.mathworks.com/matlabcentral/fileexchange/95`, 2003.

[41] T. Lehmann. *On Efficient Solution Methods for Mixed-Integer Nonlinear and Mixed-Integer Quadratic Optimization Problems*. PhD thesis, Universität Bayreuth, 2011.

[42] S. Leyffer. MacMINLP. `http://wiki.mcs.anl.gov/leyffer/index.php/MacMINLP`.

[43] S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18:295–309, 2001.

[44] L. Liberti, S. Cafieri, and F. Tarissan. Reformulations in mathematical programming: A computational approach. In A. Abraham, A.-E. Hassanien, and P. Siarry, editors, *Global Optimization*, volume 203 of *Studies in Computational Intelligence*, pages 153–234. Springer, New York, 2009.

[45] Y. Lin and L. Schrage. The global solver in the LINDO API. *Optimization Methods & Software*, 24(4–5):657–668, 2009.

[46] Lindo Systems, Inc. What's*Best!* 10.0, 2009. `http://www.lindo.com`.

[47] Lindo Systems, Inc. Lindo API 6.1, 2010. `http://www.lindo.com`.

[48] R. Lougee-Heimer. The Common Optimization INterface for Operations Research. *IBM Journal of Research and Development*, 47(1):57–66, 2003. `http://www.coin-or.org`.

[49] A. Mahajan. Global optimization with MINOTAUR. talk at INFORMS Annual Meeting, November 2011.

[50] A. Mahajan and T. Munson. Exploiting second-order cone structure for global optimization. Technical Report ANL/MCS-P1801-1010, Argonne National Laboratory, 2010.

[51] The MathWorks. *MATLAB User's Guide*, 2009. `http://www.mathworks.com`.

[52] R. Misener and C. A. Floudas. Global optimization of mixed-integer quadratically-constrained quadratic programs (MIQCQP) through piecewise-linear and edge-concave relaxations. Accepted for Mathematical Programming, 2011.

[53] R. Misener and C. A. Floudas. GloMIQO: Global mixed-integer quadratic optimizer. Accepted for Journal of Global Optimization, 2012.

[54] A. Mitsos, B. Chachuat, and P.I. Barton. McCormick-based relaxations of algorithms. *SIAM Journal on Optimization*, 20(2):573–601, 2009.

[55] MOSEK Corporation. *The MOSEK optimization tools manual*, 6.0 edition, 2009. `http://www.mosek.com`.

[56] G.L. Nemhauser, M.W.P. Savelsbergh, and G.S. Sigismondi. MINTO, a Mixed INTeger Optimizer. *Operations Research Letters*, 15:47–58, 1994.

[57] I. Nowak and S. Vigerske. LaGO: a (heuristic) branch and cut algorithm for nonconvex MINLPs. *Central European Journal of Operations Research*, 16(2):127–138, 2008.

[58] L. Quesada and I.E. Grossmann. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers & Chemical Engineering*, 16:937–947, 1992.

[59] M. Roelofs and J. Bisschop. *AIMMS 3.9 – The Language Reference.* Paragon Decision Technology B.V., Haarlem, The Netherlands, 2009.

[60] N.V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201–205, 1996.

[61] M. Schlüter and M. Gerdts. The oracle penalty method. *Journal of Global Optimization*, 47(2):293–325, 2009.

[62] M. Schlüter, M. Gerdts, and J.J. Rückmann. MIDACO: New global optimization software for MINLP. available at `http://www.midaco-solver.com/about.html`, 2010.

[63] L. Schrage. *Optimization Modeling with LINGO.* Lindo Systems, Inc., 2008. `http://www.lindo.com`.

[64] SCICON Ltd. *SCICONIC User Guide Version 1.40.* Scicon Ltd., Milton Keynes, UK, 1989.

[65] E.M.B. Smith and C.C. Pantelides. A symbolic reformulation/spatial branch-and-bound algorithm for the global optimization of nonconvex MINLPs. *Computers & Chemical Engineering*, 23:457–478, 1999.

[66] I. Solberg. fminconset. `http://www.mathworks.com/matlabcentral/fileexchange/96`.

[67] M. Tawarmalani and N.V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications.* Kluwer Academic Publishers, 2002.

[68] M. Tawarmalani and N.V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.

[69] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí. Scatter search and local NLP solvers: A multistart framework for global optimization. *INFORMS Journal on Computing*, 19(3):328–340, 2007.

[70] R.J. Vanderbei and D.F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13:231–252, 1999.

[71] A. Vecchietti and I.E. Grossmann. LOGMIP: A Disjunctive 0-1 Nonlinear Optimizer for Process System Models. *Computers & Chemical Engineering*, 23:555–565, 1999.

[72] S. Vigerske. *Decomposition of Multistage Stochastic Programs and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming.* PhD thesis, Humboldt Universität zu Berlin, 2012. Submitted.

[73] A. Wächter and L.T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. `http://projects.coin-or.org/Ipopt`.

[74] T. Westerlund and K. Lundquist. Alpha-ECP, version 5.04. an interactive MINLP-solver based on the extended cutting plane method. Technical Report 01-178-A, Process Design Laboratory, Åbo Akademi University, Åbo, Finland, 2003. `http://www.abo.fi/~twesterl/A-ECPManual.pdf`.

[75] T. Westerlund and F. Pettersson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 19:S131–S136, 1995.

[76] T. Westerlund and R. Pörn. Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. *Optimization and Engineering*, 3:253–280, 2002.

	solver	literature	AIMMS	AMPL	GAMS	NEOS	URL
MIQCP	CPLEX	[38]	✓	✓	✓	–	`http://www-01.ibm.com/software/integration/optimization/cplex`
	FICO Xpress-Optimizer	[26]	✓	✓	✓	–	`http://www.fico.com/xpress`
	GloMIQO	[52, 53]	–	–	✓	–	`http://helios.princeton.edu/GloMIQO`
	MOSEK	[55]	✓	✓	✓	–	`http://www.mosek.com`
general MINLP	alphaBB	[3, 5]	–	–	–	–	`http://titan.princeton.edu`
	AlphaECP	[74, 76]	–	–	✓	✓	`http://www.abo.fi/~twesterl`
	AOA	[59]	✓	–	–	–	`http://www.aimms.com`
	BARON	[67, 68]	✓	–	✓	✓	`http://archimedes.cheme.cmu.edu/baron/baron.html`
	bnb	[40]	–	–	–	–	`http://www.mathworks.com/matlabcentral/fileexchange/95`
	BONMIN	[11]	–	✓	✓	✓	`https://projects.coin-or.org/Bonmin`
	Couenne	[9]	–	✓	✓	✓	`https://projects.coin-or.org/Couenne`
	DICOPT	[32, 39]	–	–	✓	✓	`http://www.gams.com/solvers`
	FICO Xpress-SLP	[24]	–	–	–	–	`http://www.fico.com/xpress`
	FilMINT	[1]	–	✓	–	✓	`http://www.neos-server.org/neos/solvers/minco:FilMINT/AMPL.html`
	fminconset	[66]	–	–	–	–	`http://www.mathworks.com/matlabcentral/fileexchange/96`
	Knitro	[15]	✓	✓	✓	✓	`http://www.ziena.com`
	LaGO	[57]	–	✓	✓[a]	–	`https://projects.coin-or.org/LaGO`
	LindoAPI	[47, 45]	–	–	✓	✓	`http://www.lindo.com`
	LogMIP	[71]	–	–	✓	–	`http://www.logmip.ceride.gov.ar`
	MIDACO	[61, 62]	–	–	–	–	`http://www.midaco-solver.com`
	MILANO	[10]	–	–	–	–	`http://www.pages.drexel.edu/~hvb22/milano`
	MINLP_BB	[43]	–	✓	–	✓	`http://wiki.mcs.anl.gov/leyffer/index.php/Sven_Leyffer's_Software#MINLPBB`
	MISQP	[20, 21]	–	–	–	–	`http://www.klaus-schittkowski.de/misqp.htm`
	OQNLP	[32, 69]	–	–	✓	✓	`http://www.gams.com/solvers`
	SBB	[32]	–	–	✓	✓	`http://www.gams.com/solvers`
	SCIP	[2, 72]	–	–	✓	✓	`http://scip.zib.de`

Table 1: An overview on solvers for MINLP. The first row indicates whether a solver accepts only problems with quadratic functions (MIQCPs) or general nonlinear functions (MINLPs).

---

[a] interface for MINLPs available, but not included in GAMS distribution