MINLP Solver Software

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

February 18, 2010

Abstract

In this article we will give a brief overview of the start-of-the-art on 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.

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

1 Introduction

The general form of a 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. The variables x and y are the decision variables, where y is required to be integer valued. $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^s$ are bounding-box-type restrictions on the variables. In addition to integer requirements on variables, other kinds of discrete constraints are commonly used. These are (for example) special-ordered-set constraints (only one (SOS type 1) or two adjacent (SOS type 2) variables in a set are allowed to be nonzero), 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 (linear) 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.

Computational tractability depends significantly on whether the functions $f(x,y)$ and $g(x,y)$ are convex or not. In this chapter, we say a MINLP is *convex* if both $f(x,y)$ 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[1] [56].

[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 `vigerske@math.hu-berlin.de`

[1]Let $X \subseteq \mathbb{R}^n$ be a convex set. A differentiable function $h : X \to \mathbb{R}$ is *pseudo-convex* on X if for every

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 [5, 26, 46]. 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 problem solution via mixed-integer linear programming (MIP). In the mid 1980's Grossmann and Kocis developed DICOPT, a general purpose algorithm for convex MINLP based on the outer approximation method [18]. Since then, a number of academic and commercial codes for convex MINLP have emerged, either based on outer approximation using MIP relaxations [18], an integration of outer approximation into a linear programming (LP) relaxation based branch and cut [44], or nonlinear programming (NLP) relaxation based branch and bound algorithms [34]. For nonconvex MINLP, the first general purpose solvers were ALPHABB and BARON, both based on convexification techniques for nonconvex constraints [3, 48].

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 [11, 12, 20]. 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. 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 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 a MIP, LP, or NLP solver plugin. However, it is most often the case that even 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, a MIP or an NLP solver builds the basis for an extension towards MINLP. Solvers which can be categorized as extending a MIP solver towards handling of nonlinear objectives and constraints are BONMIN, COUENNE, CPLEX, FILMINT, LINDOBB, MOSEK, SCIP, and XPRESS. On the other hand, solvers where an NLP solver was extended to handle integrality restrictions are BNB, 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

$x, y \in X$ 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.

which may solve LP, MIP, or NLP subproblems. In this category we have alphaBB, AlphaECP, AOA, BARON, DICOPT, LaGO, and LindoGlobal.

3.3 Algorithms

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, fminconset, Knitro, LindoBB, MILANO, MINLP_BB, MOSEK, SBB, and XPRESS. For all these solvers except alphaBB, the NLP relaxation is obtained by relaxing the integrality restriction in (P). 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). These solvers 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 [18, 23], 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 a MIP solver. Solvers in this class are AOA, BONMIN (in B-OA mode) and DICOPT. 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) [55]. 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 [44]. 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 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. Such a spatial branch and cut algorithm is implemented by BARON, Couenne, LaGO, LindoGlobal, 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. Finally, OQNLP applies a randomized approach by sampling starting points and fixations 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 programs (SOCP) are CPLEX, MOSEK, SCIP, and XPRESS. CPLEX and MOSEK further require that quadratic constraints can be reformulated as second order cone constraints. XPRESS further requires that quadratic functions are convex.

Solvers that guarantee global optimal solutions only for convex general MINLPs are AlphaECP, AOA, bnb, BONMIN, DICOPT, FilMINT, fminconset, Knitro, LaGO, LindoBB, MILANO, MINLP_BB, 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.

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. The solvers alphaBB, BARON, Couenne, and LindoGlobal belong into this category.

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 modeling systems like AIMMS [45], AMPL [27], and GAMS [28]) 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]. This solver has been developed by the research group of C. A. Floudas at the Computer-Aided Systems Laboratory of Princeton University.

alphaBB can be applied for convex and nonconvex MINLPs. It implements a branch-and-bound algorithm that utilizes convex NLPs for bounding. Convexifications are obtained by using known tight underestimators for specially structured nonconvex terms (e.g., binomial or fractional) and α-underestimators in the generic (twice differentiable) case. 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).

AlphaECP (α-Extended Cutting Plane) [54, 56]. 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 a 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) [45]. 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 a MIP outer approximation of (P) and can solve NLP subproblems to find feasible solutions early. In contrast to AlphaECP, AOA constructs a MIP outer approximation of the feasible region of (P) by linearizing nonlinear functions in solutions of NLP subproblems [18]. 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.

BARON (Branch And Reduce Optimization Navigator) [48, 49]. This solver has been developed by N. Sahinidis and M. Tawarmalani at the Department of Chemical Engineering of the University of Illinois at Urbana-Champaign. 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 factorable reformulation of (P) which allows the application of known convex underestimators for all nonconvex terms that appear in the problem. The algorithm is enhanced by using advanced box reduction techniques and new convexification techniques for multilinear terms [4].

bnb (Branch 'n Bound) [32]. This solver has been developed by K. Kuipers of the Department of Applied Physics at the University of Groningen. It is available as Matlab [39] source.

bnb ensures global optimal solutions for convex MINLPs. It implements a branch-and-bound algorithm utilizing nonlinear relaxations for the bounding step [34]. The NLPs are solved by the Matlab Optimization Toolbox routine fmincon.

BONMIN (Basic Open-source Nonlinear Mixed Integer Programming) [10]. This open-source solver has been developed primarily by P. Bonami in a cooperation of Carnegie Mellon University and IBM Research. It is available in source code and as standalone binaries from COIN-OR (Computational Infrastructure for Operations Research) [38], 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 a MIP outer approximation of (P) [18], B-QG is a branch-and-bound algorithm that utilizes LPs for bounding [44], B-BB is a branch-and-bound algorithm that utilizes NLPs for bounding [34], 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 [25] and can use filterSQP [24] and IPOPT [53] as NLP solvers.

Couenne (Convex Over and Under Envelopes for Nonlinear Estimation) [7]. This open-source solver has been developed primarily by P. Belotti in a cooperation of Carnegie Mellon University, Lehigh University, and IBM Research. 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 factorable reformulation of (P). The algorithm is enhanced by box reduction techniques and disjunctive cuts [6]. Couenne is implemented on top of BONMIN.

CPLEX [30]. This solver has been developed by CPLEX Optimization, Inc. (later acquired by ILOG and recently acquired by IBM). It is generally accepted as one of the leading LP and MIP solvers. It is available as standalone binaries and as a component in almost every modeling system.

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.

DICOPT (Discrete and Continuous Optimizer) [28, 31]. This solver has been developed by the research group of I. E. Grossmann at the Engineering Research Design Center of Carnegie Mellon University. It is available as a commercial solver within GAMS.

DICOPT ensures global optimal solutions for convex MINLPs. It alternates between solving MIP outer approximations and NLP subproblems of (P) [18]. To accommodate also nonconvex MINLPs, nonlinear equality constraints can be relaxed and linearizations of nonlinear functions are allowed to move away from its support point in the MIP relaxation.

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 [44]. FilMINT is implemented on top of the MIP solver MINTO [42] and the NLP solver filterSQP [24].

fminconset [47]. 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 [34]. The NLPs are solved by the Matlab Optimization Toolbox routine fmincon.

Knitro [14]. 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 [34, 44].

LaGO (Lagrangian Global Optimizer) [43]. 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.

LindoBB (LINDO Systems Mixed Integer Nonlinear Solver) [37]. This solver has been developed by LINDO Systems, Inc. It is available within the LINDO environment and as a commercial solver within GAMS.

LINDOBB ensures global optimal solutions for convex MINLPs. It was the first commercially available solver implementing a branch-and-bound algorithm utilizing nonlinear relaxations for the bounding step [34]. The NLP relaxations are solved by CONOPT [17, 28].

LindoGlobal (LINDO Systems Global Solver) [37]. This solver has been developed by LINDO Systems, Inc. It is available within the LINDO environment and as a commercial solver within GAMS.

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

MILANO (Mixed-Integer Linear and Nonlinear Optimizer) [8]. 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 [34]. The NLPs are solved by LOQO [51], where special emphasis is put on how to warmstart this interior-point solver.

MINLP_BB (Mixed Integer Nonlinear Programming Branch-and-Bound) [34]. 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 [29].

MINLP_BB ensures global optimal solutions for convex MINLPs. It implements a branch-and-bound algorithm utilizing nonlinear relaxations for the bounding step [34]. The NLPs are solved by filterSQP.

MISQP (Mixed Integer Sequential Quadratic Programming) [19]. 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.

MOSEK [41]. 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 QCP/SOCPs for bounding [44].

OQNLP (OptQuest Nonlinear Programming) [28, 50]. 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) [28]. 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 [34]. The NLP relaxations are solved by one (or several) of the NLP solvers available with GAMS. Using the GAMS Branch-Cut-and-Heuristic facility [12], SBB allows the user to implement a model-specify heuristic in the GAMS language.

SCIP (Solving Constraint Integer Programs) [2, 9]. This solver has been developed by the Optimization Department at the Zuse Institute Berlin. For academic institutions, it is available in source code and as standalone binary. Also a GAMS interface is available.

SCIP ensures global optimal solutions for convex 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 or convexified quadratic functions.

XPRESS [22]. This solver has been developed by Dash Optimization (later been acquired by FICO). It is available as standalone binaries and as a component in many modeling systems.

XPRESS can solve convex MIQCPs. It implements a branch-and-bound algorithm that utilizes QCPs for bounding.

5 Outlook and Summary

Combining the world of discrete and nonlinear optimization results in a rich modeling paradigm applicable to many real world optimization problems. On the other hand, 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 implement advanced automatic reformulation and preprocessing algorithms, such techniques are in todays MINLP solvers only available in a limited form. Therefor, a good problem formulation delivered by the modeler is still very important to solve a MINLP. However, software for guided automatic model reformulations and relaxations have been recently developed. LogMIP [52], one of the first systems available, translates a MINLP with disjunctions into a standard MINLP by applying bigM and convex hull reformulations. More recently, frameworks like GAMS/EMP (Extended Mathematical Programming) [21] and ROSE (Reformulation/Optimization Software Engine) [36] provide a growing toolbox for reformulating MINLPs. Other recent activities, like libMC [40] and parts of MINOTAUR (Mixed-Integer Nonconvex Optimization Toolbox – Algorithms, Underestimators, Relaxations) [35] focus on (convex) relaxations for (nonconvex) MINLP.

Another important area is the collection and dissemination of MINLP models. Instance collections like MacMINLP [33] and MINLPLib [13] provide valuable test cases for solver developers. The new Cyber-Infrastructure for MINLP [15] 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 state-of-the-art MINLP solvers. We have established several groupings with respect to various features of the software. We hope that the groupings and the individual descriptions give sufficient information to guide the selection for the best solver for a particular MINLP problem.

Acknowledgments. We like to warmly thank Steven P. Dirkse and Arne S. Drud for proofreading this paper. 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. Technical Report ANL/MCS-P1374-0906, Argonne National Laboratory, Mathematics and Computer Science Division, 2006.

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

[3] 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.

[4] 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.

[5] 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, 1980.

[6] P. Belotti. Disjunctive cuts for non-convex MINLP. Technical report, Lehigh University, 2009. `http://coral.ie.lehigh.edu/~belotti/papers/belotti-disj-MINLP.pdf`.

[7] 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.

[8] H.Y. Benson. Mixed integer nonlinear programming using interior-point methods. available at `http://www.pages.drexel.edu/~hvb22/minlp.pdf`, 2007.

[9] T. Berthold, S. Heinz, and S. Vigerske. Extending a CIP framework to solve MIQCPs. Technical Report 09-23, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), 2009.

[10] 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.

[11] C. Bragalli, C. D'Ambrosio, J. Lee, A. Lodi, and P. Toth. Practical water network design by MINLPs. Technical Report OR-09-21, University of Bologna, 2009.

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

[13] 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.

[14] 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.

[15] CMU-IBM Cyber-Infrastructure for MINLP. `http://www.minlp.org`, 2009.

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

[18] 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.

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

[20] 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.

[21] 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.

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

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

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

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

[26] 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.

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

[28] GAMS Development Corp. *GAMS – The Solver Manuals*. Washington DC, 2009.

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

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

[31] 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.

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

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

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

[35] S. Leyffer, J. Linderoth, and T. Munson. MINOTAUR: Mixed-Integer Nonconvex Optimization Toolbox – Algorithms, Underestimators, Relaxations.

[36] 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.

[37] Lindo Systems, Inc. Lindo API 6.0, 2009.

[38] 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`.

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

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

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

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

[43] 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.

[44] 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.

[45] M. Roelofs and J. Bisschop. *AIMMS 3.8 – Language Reference*. Paragon Decision Technology B.V., Haarlem, The Netherlands, 2007.

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

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

[48] 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.

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

[50] 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.

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

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

[53] 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`.

[54] 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`.

[55] T. Westerlund and F. Petterson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 21:131–136, 1995.

[56] 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
alphaBB	[3]	–	–	–	–	`http://titan.princeton.edu`
AlphaECP	[54, 56]	–	–	✓	✓	`http://www.abo.fi/~twesterl`
AOA	[45]	✓	–	–	–	`http://www.aimms.com`
BARON	[48, 49]	✓	–	✓	✓	`http://archimedes.cheme.cmu.edu/baron/baron.html`
bnb	[32]	–	–	–	–	`http://www.mathworks.com/matlabcentral/fileexchange/95`
BONMIN	[10]	–	✓	✓	✓	`https://projects.coin-or.org/Bonmin`
Couenne	[7]	–	✓	✓	✓	`https://projects.coin-or.org/Couenne`
CPLEX	[30]	✓	✓	✓	–	`http://www-01.ibm.com/software/integration/optimization/cplex`
DICOPT	[28, 31]	–	–	✓	✓	`http://www.gams.com/solvers`
FilMINT	[1]	–	✓	–	✓	`http://www-neos.mcs.anl.gov/neos/solvers/minco:FilMINT/AMPL.html`
fminconset	[47]	–	–	–	–	`http://www.mathworks.com/matlabcentral/fileexchange/96`
Knitro	[14]	✓	✓	✓	✓	`http://www.ziena.com`
LaGO	[43]	–	✓	✓[a]	–	`https://projects.coin-or.org/LaGO`
LindoBB	[37]	–	–	✓	–	`http://www.lindo.com`
LindoGlobal	[37]	–	–	✓	✓	`http://www.lindo.com`
LogMIP	[52]	–	–	✓	–	`http://www.logmip.ceride.gov.ar`
MILANO	[8]	–	–	–	–	`http://www.pages.drexel.edu/~hvb22/milano`
MINLP_BB	[34]	–	✓	–	✓	`http://wiki.mcs.anl.gov/leyffer/index.php/Sven_Leyffer's_Software#MINLPBB`
MISQP	[19]	–	–	–	–	`http://www.math.uni-bayreuth.de/~kschittkowski/MISQP.pdf`
MOSEK	[41]	✓	✓	✓	–	`http://www.mosek.com`
OQNLP	[28, 50]	–	–	✓	✓	`http://www.gams.com/solvers`
SBB	[28]	–	–	✓	✓	`http://www.gams.com/solvers`
SCIP	[2, 9]	–	–	✓	✓	`http://scip.zib.de`
XPRESS	[22]	✓	✓	✓	–	`http://www.fico.com/xpress`

Table 1: An overview on solvers for MINLP.

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