

# A Comparative Analysis on Combinatorial Optimization: Minlp Issues and Justification

Meenaxi R. Sangam<sup>1</sup> Dr. R. A. Khan<sup>2</sup>

<sup>1</sup>Research Scholar, CMJ University, Shillong, Meghalaya

<sup>2</sup>Professor of Mathematics, CMJ University, Shillong, Meghalaya

**Abstract** – Recently, the area of Mixed Integer Nonlinear Programming (MINLP) has experienced tremendous growth and a flourish of research activity. In this article we will give a brief overview of past developments in the MINLP arena and discuss some of the future work that can foster the development of MINLP in general and, in particular, robust solver technology for the practical solution of problems.

Mixed-Integer Programs (MIP's) involving logical implications modeled through big-M coefficients, are notoriously among the hardest to solve. In this paper we propose and analyze computationally an automatic problem reformulation of quite general applicability, aimed at removing the model dependency on the big-M coefficients. Our solution scheme defines a master Integer Linear Problem (ILP) with no continuous variables, which contains combinatorial information on the feasible integer variable combinations that can be "distilled" from the original MIP model. The master solutions are sent to a slave Linear

Program (LP), which validates them and possibly returns combinatorial inequalities to be added to the current master ILP. The inequalities are associated to minimal (or irreducible) infeasible subsystems of a certain linear system, and can be separated efficiently in case the master solution is integer. The overall solution mechanism resembles closely the Benders' one, but the cuts we produce are purely combinatorial and do not depend on the big-M values used in the MIP formulation. This produces an LP relaxation of the master problem which can be considerably tighter than the one associated with original MIP formulation. Computational results on two specific classes of hard-to-solve MIP's indicate the new method produces a reformulation which can be solved some orders of magnitude faster than the original MIP model.

Numerous industrial problems can be modeled as MINLP problems combining both numeric and integer variables. Several methods were proposed to solve these problems. But industrial applications need more than solving problems: dynamic problems, over-constrained problems, or explaining solver behavior are features required by industrial applications. Explanation-based constraint programming offers such tools. In this paper, we show how to apply explanation-based mechanisms for mixed problems thanks to a generic framework. Last, some first experimental results are exposed: the overhead due to explanation managing is acceptable and can even speed up some resolutions.

In this work we deal with exponential sum models coming from data acquisition in the empirical sciences. We present a two step approach based on Tikhonov regularization and combinatorial optimization, to obtain stable parameter estimations, which fit the data.

---

## INTRODUCTION

Mixed Integer Nonlinear Programming (MINLP) refers to mathematical programming with continuous and discrete variables and nonlinearities in the objective function and constraints. The use of MINLP is a natural approach of

formulating problems where it is necessary to simultaneously optimize the system structure (discrete) and parameters (continuous).

MINLPs have been used in various applications, including the process industry and the financial, engineering,

management science and operations research sectors. It includes problems in process flow sheets, portfolio selection, batch processing in chemical engineering (consisting of mixing, reaction, and centrifuge separation), and optimal design of gas or water transmission networks. Other areas of interest include the automobile, aircraft, and VLSI manufacturing areas.

The needs in such diverse areas have motivated research and development in MINLP solver technology, particularly in algorithms for handling large-scale, highly combinatorial and highly nonlinear problems.

The general form of a MINLP is

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

The function  $f(x, y)$  is a nonlinear objective function and  $g(x, y)$  a nonlinear constraint function. The variables  $x, y$  are the decision variables, where  $y$  is required to be integer valued.  $X$  and  $Y$  are bounding-box-type restrictions on the variables.

Numerous industrial problems can be modelled as MINLP (Mixed Integer Non-Linear Programming) problems combining both numeric and integer variables: design of water or gas networks, automobile, aircraft, etc.. These problems are really hard to solve: they combine the combinatorial nature of mixed integer programming and the intrinsic difficulty of nonlinear programs. Several methods were proposed to solve such problems: branch-and-bound, extended cutting plane methods, generalized Bender's decomposition, etc.

But industrial applications need more than solving problems. Problems can be dynamic, this implies that constraints may be added or removed dynamically. Moreover, if no solution is found, the user often needs to know why the problem is over-constrained, or why the expected solution is inconsistent.

Constraint programming offers generic models and tools to solve combinatorial problems. Furthermore, explanation-based constraint programming provides tools to solve dynamically such problems and maintain explanations about the resolution: why a problem has no solution, why the optimum bound is reached, or how to improve a solution are informations that explanation-based constraint programming can provide. Such features are now well known for constraint programming over integer variables.

However only few works proposed solutions to extend it to real variables. proposed to extend mac-dbt to solve numeric problems thanks to a dynamic domain splitting mechanism. But these works solve separately integer problems and numeric problems. Moreover, no solution is proposed for the main drawback about explanations for numeric problems: slow convergence of propagation may need to store a huge amount of explanations. This may make prohibitive using explanations with these problems.

Here, we propose both a generic framework to solve MINLP problems with explanation-based constraint programming and some ideas to decrease the number of explanations to store, by filtering redundant or useless explanations.

In the following, we first define MINLP problems, explanations and explanation-based constraint programming. Then, we introduce and extend propositions of. Some propositions are made in order to reduce the number of explanations the solver must store. Last, we present some experimentation about these propositions and explanation-based resolution of MINLP problems.

## A COMBINATORIAL MODEL

In the spirit of the Tikhonov regularization scheme, with previous or historical information about the problem, we get additional elements in order to improve our model (3.1).

We know that the amplitudes  $\rho_j$  should be non-negatives and that only  $k$  of them are strictly positive (represented by peaks at  $\bar{\rho}$ ). This behavior can be modeled by imposing to each point of the sequence  $\rho_j$  the condition  $\rho_j = x_j q_j$  with

$$x_j \geq 0, \quad q_j \in \{0, 1\} \quad \text{and} \quad e^T q \leq k,$$

In this way the solutions should be less smooth than  $\bar{\rho}$ , but closer to the characteristics of the desired solutions; because in the case of  $q_j = 0$ , the amplitude should be zero, but if  $q_j = 1$  more accuracy of  $\rho_j$  should be achieved. We deal with the condition that only  $k$  of the variables should be positives by imposing  $\sum q_i \leq k$  ( $e^T q \leq k$ ) as a constraint. In addition, if we calculate in a first step  $\bar{\rho}(\bar{\lambda}) = \bar{\rho}$ , then we can use this approximate solution as a default solution for the Tikhonov problem, resulting

$$\begin{aligned}
 & \text{minimize} && \frac{1}{2} \|y - M\rho\|^2 + \frac{\lambda^2}{2} \|\rho - \bar{\rho}\|^2 \\
 & \text{subject to} && \rho = x \bullet q \\
 & && e^T q \leq k \\
 & && x \geq 0, \rho \geq 0, q \in \{0, 1\}^N,
 \end{aligned} \tag{1}$$

where  $\bullet$  denotes the componentwise product between vectors. Problem (1) is a mixed integer nonlinear programming problem (MINLP).

In practice, constraints of the type  $\rho_j = x_j q_j$  can produce tricky values. For instance, if  $\rho_j$  is calculated as zero, this could happen either because  $q_j = 0$  or because  $q_j = 1$ , but  $x_j$  approaches zero. The last behavior is undesired, since we should be detecting a peak at  $q_j$ , but with the  $x_j$  value near zero. On the other hand, if  $q_j = 0$ , then  $\rho_j$  is also calculated as zero, but this time with a large value for the corresponding  $x_j$ , distorting the relationship among the variables. In order to prevent the problem of these undesired behaviors, we can model the regularization term by adding

$$\frac{\lambda^2}{2} \|x - \bar{x}\|^2 + \frac{\lambda^2}{2} \|q - \bar{q}\|^2$$

to the objective function at (1). This allows us to keep values of  $x$  and  $q$  close to a known approximate solution, and so avoid those tricky behaviors. We establish our improved MINLP model as

$$\begin{aligned}
 & \text{minimize} && f(\rho, x, q) = \frac{1}{2} \|y - M\rho\|^2 + \frac{\lambda^2}{2} \|\rho - \bar{\rho}\|^2 + \frac{\lambda^2}{2} \|x - \bar{x}\|^2 + \frac{\lambda^2}{2} \|q - \bar{q}\|^2 \\
 & \text{subject to} && \rho = x \bullet q \\
 & && e^T q \leq k \\
 & && x \geq 0, \rho \geq 0, q \in \{0, 1\}^N.
 \end{aligned}$$

(2)

Note that this formulation leads to an NP-hard problem. Since the original problem is essentially easier, we can ask, why to construct a harder one? Two answers appear: First, we get more accuracy in calculating the amplitudes, by avoiding assigning positive values to amplitudes corresponding to the nonlinear parameter values close to the chosen ones; and secondly, we can use heuristics, or approximating solvers to deal with the combinatorial problem, which are computationally less expensive, providing efficient tools to cope with the problem.

## ALGORITHMS

MINLP problems are precisely so difficult to solve, because they combine all the difficulties of both of their subclasses: the combinatorial nature of mixed integer programs (MIP)

and the difficulty in solving non convex (and even convex) nonlinear programs (NLP). Because subclasses MIP and NLP are among the class of theoretically difficult problems (NP-complete), so it is not surprising that solving MINLP can be a challenging and daring venture. Fortunately, the component structure of MIP and NLP within MINLP provides a collection of natural algorithmic approaches, exploiting the structure of each of the subcomponents.

**Solution Approaches :** Methods for solving MINLPs include innovative approaches and related techniques taken and extended from MIP. Outer Approximation (OA) methods, Branch-and-Bound (B&B), Extended Cutting Plane methods, and Generalized Bender's Decomposition (GBD) for solving MINLPs have been discussed in the literature since the early 1980's. These approaches generally rely on the successive solutions of closely related NLP problems. For example, B&B starts out forming a pure continuous NLP problem by dropping the integrality requirements of the discrete variables (often called the relaxed MINLP or RMINLP). Moreover, each node of the emerging B&B tree represents a solution of the RMINLP with adjusted bounds on the discrete variables.

In addition, OA and GBD require the successive solution of a related MIP problem. Both algorithms decompose the MINLP into an NLP sub problem that has the discrete variables fixed and a linear MIP master problem. The main difference between GBD and OA is in the definition of the MIP master problem. OA relies on tangential planes (or linearizations), effectively reducing each sub problem to a smaller feasible set, whereas the master MIP problem generated by GBD is given by a dual representation of the continuous space.

The approaches described above only guarantee global optimality under (generalized) convexity. Deterministic algorithms for global optimization of non convex problems require the solution of sub problems obtained via convex relaxations of the original problem in a branch-and-bound context, and have been quite successful in solving MINLPs.

## ISSUES IN MINLP

In this section the issues and problems that arise in MINLP are discussed by first examining two special cases which are embedded in P, namely MILP and NLP problems. MINLP problems possess also a number of features which are unique in the sense that they do not occur either in NLP or MILP problems and these are listed at the end of this section.

MILP problems are combinatorial optimization problems with an exponential number of integer feasible points. By

fixing the integer variables and solving the resulting LP sub problem in the continuous variables only, it is comparatively easy to find a local solution to an MILP, that is a point that satisfies the first order Kuhn–Tucker conditions for a fixed assignment of integer variables.

The number of these local solutions is exponential and since unlike in LP no optimality conditions exist the task of choosing the optimum among the local minima is very hard. The lack of suitable optimality conditions for MILP implies that any MILP algorithm faces the double task of finding and verifying an optimal solution. Thus even though the algorithm were started at the global optimum it would still require a possibly exponential number of iterations to recognize the optimality. In the case of a branch–and–bound algorithm it is therefore usually necessary to examine further nodes in the tree once the optimum has been found before the optimality is verified.

A consequence of the combinatorial nature of the problem is that for most algorithms there exists a worst case example for which the algorithm has to solve an exponential number of sub problems.

Jeroslow gives a trivial class of integer problems for which any branch–and–bound algorithm has to expand an exponential number of nodes before it discovers that the problem is infeasible.

His result is valid for a wide range of enumerative schemes and can be modified to give worst case behavior for almost any other algorithm. These worst case examples agree with practical experience with integer programming algorithms which indicates an exponential growth in the computing time as the number of variables is increased. Thus, while MILP problems can be solved in a finite number of steps, this number grows usually exponentially in the number of variables.

On the other hand, LP and QP problems can be solved in polynomial time. For example, Karmarkar gives a polynomial time algorithm for LP problems and Gill, Murray, Saunders, Tomlin and Wright show that Karmarkar's algorithm is related to the logarithmic barrier function. However, many practical algorithms for LP and QP problems use the Simplex method or an active set method. For those algorithms there exist worst case examples in which they visit a number of vertices which increases exponentially with the problem size, but this is not very likely to occur in practice. Thus a major difference between MILP and LP problems is that – in practice – the former require an exponential number of iterations while the latter can be solved in a number of iterations that is bounded by a polynomial in the problem size.

LP and QP problems form special classes of NLP problems. However, more general NLP problems cannot be solved in a finite number of steps and usually an iterative scheme has to be applied to solve them. Nevertheless, finding a local solution to an NLP problem is a much simpler task than finding a solution to an MILP problem, since methods like SQP ultimately exhibit a second order rate of convergence and the solver terminates once the first order conditions are satisfied to sufficient accuracy.

Another difference between MILP and NLP problems is that while every local solution of an MILP is a global solution, the same is only true for NLP problems under an additional convexity assumption.

MINLP problems combine the two aspects of MILP and NLP problems but also have some features which are unique. While a strict convexity assumption ensures the global uniqueness of an NLP solution, the same does not hold for MINLP problems. Loh and Papalambros give a very good illustration of such a situation which is given in Figure 1 in a slightly more general form.

The picture shows the contours of a convex objective function and a nonlinear constraint which is indicated by the broken thick line and the two axes. The continuous feasible region lies between the constraint and the axis while the discrete feasible points are the dots that lie within the feasible region. The two discrete optima are indicated by circled dots and it is also shown where the unique continuous minimizer lies.

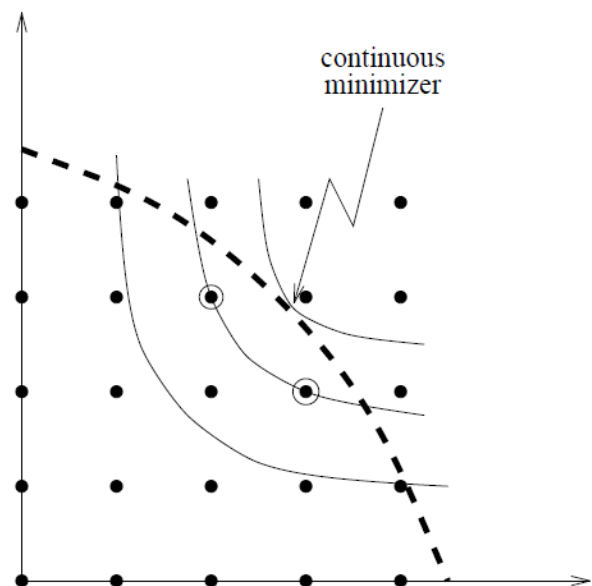


Figure 1: Multiple Optima for a strictly convex MINLP



It is possible to derive upper bounds on the difference between an optimal solution to an MILP problem and an optimal solution to its LP relaxation. However, these bounds are usually very weak since they involve the dimension of the problem and a constant that depends on the largest absolute subdeterminant of the integral coefficient matrix. Werman and Magagnosc generalize these bounds to MINLP problems with a separable and convex objective and linear constraints. Having discussed some of the issues involved in MINLP programming, the next two sections provide an overview of methods for solving MINLP problems.

## SOFTWARE

Although theoretical algorithmic ideas for solving MINLP have been around for a while, the practical implementation of such concepts is much more difficult. Memory limitations, efficient numerical linear algebra routines, suitable algorithmic tolerances, and determining default solver options are some of the key issues faced when extending algorithms to large-scale, general-purpose software.

In this section we give a brief and possibly incomplete historical overview of practical general purpose MINLP software. Commercial MINLP Software Packages : Best to our knowledge, the earliest commercial software package that could solve MINLP problems was SCICONIC in the mid 1970's. Rather than handling nonlinearities directly, linked SOS variables provided a mechanism to represent discretized nonlinear functions and allowed solving the problem via MIP. In the mid 1980's Grossman and Kocis developed GAMS/DICOPT, a general purpose MINLP algorithm based on the outer approximation method. In the early 1990's LINDOs and What's Best B&B code using the Generalized Reduced Gradient (GRG) code for sub problems was extended to solve MINLPs.

Since then a number of excellent academic as well as commercial codes have surfaced, including alpha ECP and mittlp, both of which are based on extended cutting plane methods, and MINLP BB and SBB, which use branch-and-bound to solve relaxed NLP sub problems. Even on the frontier of global MINLP, reliable and large-scale packages have materialized including alpha BB and BARON, which use convex relaxations in a branch-and bound framework.

Modeling Languages : The emergence of algebraic modeling languages in the mid to late 1980's and early 1990's has greatly simplified the process of modeling, in particular the formulation of MINLP type problems. Also, from a MINLP solver perspective, a modeling system delivers reliable black-box-type function evaluations and first and second order derivative information. Finally, the

common solver interface of a modeling system allows MINLP algorithms to deploy existing NLP and MIP solvers to solve sub problems in a seamless way. The latter is available as part of the MINLP World. MINLP World is a forum for discussion and dissemination of information about all aspects of MINLP.

## CONCLUSION AND FUTURE WORK

In this paper, we presented results about using explanations with MINLP resolutions. Indeed, giving explanations to the user to explain the result is a feature that many industrials need. These first results are encouraging since the over-head is quite acceptable, all the more that heuristics for filtering explanations seems to be efficient to decrease this overhead.

Further works should be led both to test as precisely as possible the efficiency of the different heuristics and to improve the first implementation used in these experimentations. Moreover, this work could be extended to apply it to solver collaboration and not only for intra-solver cooperation. Indeed, explanations (or at least no goods) could be valuable information to exchange between different kinds of solvers since they give information about past computations of each solver.

We propose a two step procedure to find estimations of linear and nonlinear parameters in exponential sum models. For the first step we solve a Tikhonov regularized problem which gives us the information to be refined in a second step. At this second stage we solve a mixed integer nonlinear programming problem obtaining good estimates for the parameters. We study properties of the solutions for this problem by exposing the combinatorial nature of the problem. Our procedure requires less initial information to successfully estimate the solution. It is enough to have the data, and an upper bound on the nonlinear parameters. We do not require the initial values to be known.

Progress in the MINLP arena has been significant in recent years, and we are now able to solve large-scale problems efficiently using a wide variety of approaches. However, MINLP has yet to reach the level of maturity that MIP has achieved. While the MIP community has benefited greatly from preprocessing to reduce model sizes and to detect special structure, MINLP technology is still lagging behind. NLP and MINLP preprocessing, similar to global methods, will require the delivery of structural information from the modeling languages.

Progress on reliable large-scale NLP codes with restarting capabilities will have an immediate impact on MINLP. Furthermore, combining individual algorithms (e.g. branch-and-bound and extended cutting plane method) with

sophisticated search strategies (e.g. non-trivial B&B selection strategies) and heuristics to quickly determine integer solutions will help to close the gap. If research and development continues at the current level of activity, MINLP will soon achieve a stage of maturity enjoyed by the other areas in mathematical programming.

## REFERENCES

- Bemporad, D. Mignone, MIQP.M: A Matlab function for solving Mixed Integer Quadratic Programs, Technical Report, vol. Autoo-22, Switzerland, 2001.
- Brooke, D. Kendrick, and A. Meeraus (1988). GAMS: A User's Guide, The Scientific Press, San Francisco, CA.
- C.A. Floudas (2000). Deterministic Global Optimization: Theory, Methods and Applications, Nonconvex Optimization And Its Applications series, 37, Kluwer Academic Publishers, Boston MA.
- Fisher, M. L., The Lagrangean relaxation method for solving integer programming problems. *Management Science* 1981, 27, (1), 18.
- Grossmann, I. E., Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques. *Optimization and Engineering* 2002, 3, 227-252.
- I.P. Androulakis, C.D. Maranas, and C.A. Floudas (1995). A Global Optimization Method for Generalized Constrained Nonconvex Problems, *J. Global Optim.*, 7, 337–363.
- J. Bisschop and M. Roelofs (2002). AIMMS - The User's Guide, Paragon Decision Technology B.V., Haarlem, The Netherlands.
- Kesavan, P.; Allgor, R. J.; Gatzke, E. P.; Barton, P. I., Outer Approximation Algorithms for Separable Nonconvex Mixed-Integer Nonlinear Programs. *Mathematical Programming* 2004, 100, (3), 517-535.
- Leyffer, S., Integrating SQP and branch and bound for mixed integer nonlinear programming. *Computational Optimization and Applications* 2001, 18, 295-309.
- M.A. Duran and I.E. Grossmann (1986). An Outer-Approximation Algorithm for a Class of Mixed-integer Nonlinear Programs, *Math. Program.*, 36, 307–339.
- Michael R. Bussieck and Armin Pruessner. Mixed-integer nonlinear programming. In *SIAG/OPT Newsletter: Views & News*, 2003.
- M.R. Bussieck, A.S. Drud, and A. Meeraus (2002). MINLPLib - A Collection of Test Models for Mixed-Integer Nonlinear Programming, *INFORMS J. Comput.*, to appear.
- R. Fletcher and S. Leyffer (1994). Solving Mixed Integer Programs by Outer Approximation, *Math. Program.*, 66, 327–349.