LOGMIP: A Discrete Continuous Nonlinear Optimizer

Aldo Vecchietti and Ignacio E. Grossmann

EDRC 06-228-96

LOGMBP : A DISJUNCTIVE 0-1 NONLINEAR OPTIMIZER FOR PROCESS SYSTEMS MODELS

Aldo Vecchietti
Department of Chemical Engineering
and the Engineering Design Research Center
Carnegie Mellon University,
Pittsburgh, PA 15213
e-mail: av2c@andrew.cmu.edu

Ignacio E. Grossmann*
Department of Chemical Engineering
and the Engineering Design Research Center
Carnegie Mellon University,
Pittsburgh, PA 15213
grossmann@cmu.edu

ABSTRACT

Discrete-continuous non-linear optimization models are frequently used to formulate problems in Process System Engineering. Major modeling alternatives and solution algorithms include generalized disjunctive programming and MINLP. Both have advantages and drawbacks depending on the problem they are dealing with. In this work, we describe the theory behind LOGMIP, a new computer code for disjunctive programming and MINLP. We discuss a hybrid modeling framework which combines both approaches, allowing binary variables and disjunctions for expressing discrete choices. An extension of the Logic-Based OA algorithm has been implemented to solve the proposed hybrid model. Computational experience is reported on several examples, which are solved within disjunctive, MINLP and hybrid approaches.

INTRODUCTION

Mathematical programming models for addressing problems in Process Systems Engineering have been extensively used over the last decade. Many applications in the synthesis and design of process networks have been formulated as MINLP models. These models assume an algebraic representation of the equations, and discrete variables are mainly restricted to 0-1 values.The current methods to solve this type of optimization problems are: Branch and Bound , Generalized Benders Decomposition (GBD) and Outer Approximation (OA). An overview of these methods, the relationships between them, and references for process engineering applications can be found in Grossmann and Kravanja (1995). Recently Turkay and Grossmann(1996) have presented logic-based algorithms in which the discrete-continuous problem is modeled as generalized disjunctive program. This model involves logic disjunctions with non-linear equations and pure logic relations. The main advantages of generalized disjunctive programs in structural flowsheet optimization are its robustness and computational efficiency when compared to algebraic MINLP models and algorithms. This approach for modeling discrete-continuous nonlinear problems is based on the work by Raman and Grossmann (1994) who investigated linear disjunctive problems. Starting with the disjunctive programming representation, a subset of the disjunctions is converted into algebraic mixed integer equations using the ^{M}w-MIP" representability criterion. This is a theoretical characterization which establishes conditions of equivalence between the disjunctions and mixed integer algebraic form. A solution algorithm restricted to the linear case was also presented. The above modeling schemes provide several alternatives and solution methods for the same problem. Depending on the representation that is selected, the computational efficiency and robustness to achieve the solution can be greatly affected.

In this work a hybrid modeling formulation for discrete-continuous non-linear problems for process system engineering is proposed. The model can involve disjunctions, binary variables and integer or mixed-integer constraints. It will be shown that from this formulation, the algebraic and the disjunctive formulation can be derived as particular cases. For the case of the hybrid formulation, we also introduce a new solution algorithm. Through the solution of several examples, we show that using disjunctions in the problem formulation is a better alternative for problems where avoiding zero flows is an important issue, or where big-M constraints yield poor relaxations. The hybrid representation is convenient when the ^{H}w-MIP" criteria applies to some disjunctions but not to all, or when it is not natural to express the entire model in terms of disjunctions and logic relations.

The above ideas we have implemented in LOGMIP, a new computer code for solving discrete-continuous nonlinear optimization problems in which the problems can be modeled with either the algebraic, disjunctive or hybrid formulation. The program has a model recognition routine to check the model type, such that, if it is in the MINLP algebraic form, the OA/ER/AP

two sub-problems, the NLP and the master MILP sub-problems. First, fixing the binary (y^k) and Boolean variables (Y;) we obtain the following NLP problem:

$$\min \quad Z = Z\,q + f(x) + d^T y^k$$

$$\text{s.t.} \quad g(x) \leq 0$$

$$r(x) \leq -Dy^k \qquad \text{(NLP)}$$

$$\left.\begin{array}{l} h_i(x) \leq 0 \\ c_i = \gamma_i \end{array}\right] \text{ if } Y_i = \text{True}$$

$$\left.\begin{array}{l} B^i x = 0 \\ c_i = 0 \end{array}\right] \text{ if } Y_j = \text{False}$$

It should be noted that in the solution of the NLP sub-problem, the dimensionality is reduced because only the equations whose Boolean variables are true apply. Therefore, nonlinear equations with zero value variables are avoided reducing difficulties with numerical singularities. On the other hand, depending on the problem, more than one initial NLP sub-problem should be solved, in order to set up the first MILP master problem. This MILP must contain linearizations of all non-linear equations in the disjunctions, to predict new binary and Boolean variable values for the next NLP. An interesting issue arises at the initialization step. While the Boolean variables should be fixed for each initial NLP sub-problem to be solved, the binary variables can be fixed or relaxed for each set of fixed Boolean, and remain constant or variable through the different initial NLPs. For some problems, e.g. synthesis of process networks, the minimum number of initial NLP sub-problems and the Boolean values, can be determined solving a modified set-covering algorithm for CNF Prepositional Logic (Turkay and Grossmann, 1996). In other cases these values should be provided by the user.

The hybrid linear master sub-problem is obtained by the linearizations of the disjunctions and non-linear constraints at the solution point of the NLP sub-problem. Instead of working with a master problem involving linear disjunctions, it is transformed into algebraic form by using the convex-hull representation for the linear disjunctions proposed by Balas (1985). The original set of binary variables is increased by m, the number necessary to replace the Boolean variables by 0-1 variables. In this way the master MILP problem can be solved with the conventional branch and bound method. This algebraic master MILP sub-problem has the following form :

$$\min Z_L = a + Zyiy + d^T y$$

$$\text{s.t} \quad f(x') + Vf(x^l)^T(x - x^l) \wedge a$$

$$g(x) + Vg(x^l)^T(x-x^*) < 0 \qquad \text{(MILP)}$$

$$r(x') + Vr(x')^T (x - x^1) + Dy \le 0$$

$$Vh(x')^T x \le \left[-h(x) + Vh(x')^T x'\right] y$$

$$Ay \ge a \qquad 1 = 1, \ldots, L$$

$$Ey \ge e$$

$$\alpha \in R^1, x \in R^n, y e \{O,l\}^{q+m}$$

where for simplicity we assume that all variables for a given disjunction are set to zero if it is false. The constraint $Ey \ge e$ are integer inequalities corresponding to the logic relationship between the Boolean variables (fi(Y)). After solving the master MILP new values for the binary and Boolean variables are obtained for the next iteration.

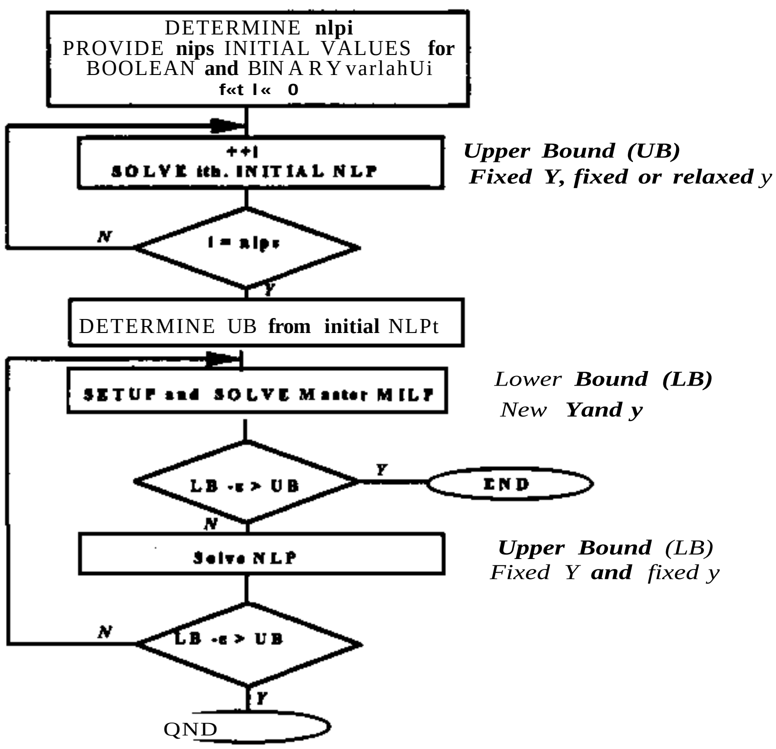

*Figure 1: Algorithm flow **chart***

As can be seen in the algorithm flowchart, the number of initial NLP problems to be solved (nips) has to be determined first, in order to set up the first master problem. Depending on the problem this can be done in a systematic way or simply specified by the user. Having identified the initial NLP sub-problems, the Boolean variables have to be fixed for each problem, and the binary variables can be fixed or relaxed, according to which strategy is specified. After solving the initial NLP sub-problems, we can obtain a valid Upper Bound from this set. Then we setup the MILP master problem to predict the Boolean and binary variables values for the next NLP, and a Lower Bound. If the Upper and the Lower Bound lie within a tolerance we stop; otherwise the iterations continue until convergence is achieved.

LOGMIP FEATURES

LOGMIP is a computer code written in C that allows the specifications of disjunctions in the problem formulation. Logic relations are presently handled as inequalities. The program has been linked to GAMS, whose language is used to express the model in terms of disjunctions and algebraic equations. The model can be specified in form of models (PA), (PD) and (PH). LOGMIP has a model recognition routine that works as follows: if no disjunctions are detected in the model, that means we are in the presence of a MINLP model and the OA/ER/AP is applied. If disjunctions are detected, LOGMIP finds if the model contains binary variables or not. In that way it can decide which algorithm has to be applied, if the Logic-Based OA for the disjunctive model (PD) or the proposed algorithm for the hybrid model (PH). Figure 2 shows the major steps involved in LOGMIP. Since the solution algorithms require the solution of NLP and MILP master problems, the I/O GAMS Library is used to set up and solve the sub-problems. The program has been written in C to assure portability to other platforms. After supplying a GAMS input file with the discrete-continuous model, the input file syntax is checked through the GAMS language compiler. If it is correct, the control is transferred to LOGMIP.

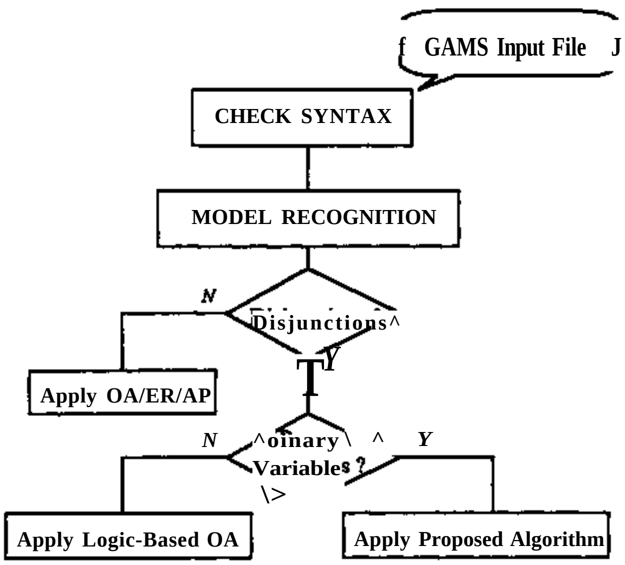

Figure 2: LOGMIP Flowchart

EXAMPLES

A set of synthesis and design problems of different size have been solved with LOGMIP. The first example corresponds to a superstructure of 8 chemical processes (Turkay and Grossmann, 1996). The model is composed of convex constraints. The objective of this example is to find the configuration with the minimum cost. Figure 3 shows the superstructure for this example.

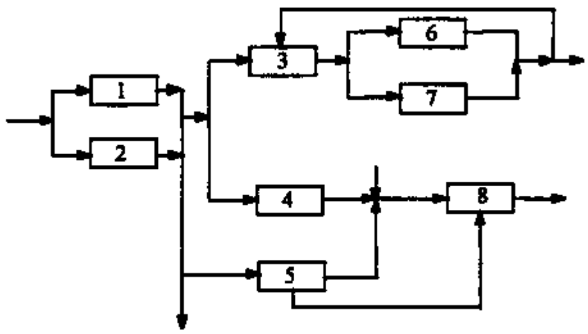

Figure 3: Processes Superstructure

We have modeled this example in three different ways: algebraic, disjunctive and hybrid. The results obtained with this example are in the following table:

RESULTS : PROCESSES SUPERSTRUCTURE

Model -*	PA	PD	PH
Initialization—>	Relaxed	Fixed	Fixed
Constraints	32	52	52
Variables	33	42	42
Discrete V.	8	8	8
Obj. Value	68	68	68
Exec, time	2.7 sec.	2 sec.	0.74 sec
Iterations	4 major*	3 nip 1 major*	2 nip 1 major*

* 1 major iteration => 1 nip + 1 master milp

In this example the constraints were increased in the disjunctive and the hybrid models because the prepositional logic equations (relationships between the Boolean variables) were added to those models. The variables added correspond to the fixed charge costs. Even with this increase in constraints and variables the results obtained with these models are encouraging. The number of iterations have been reduced in the disjunctive and hybrid models compared to the algebraic MINLP. Therefore, the execution time is also reduced. Due to the small execution times the performance is better analyzed through the number of major iterations. In the disjunctive case the difference respect to the algebraic model lies in the master milp number of problems executed. Three master problems have not been executed for the disjunctive model. The transformation from the disjunctive model into the hybrid model, has taken place for the linear models representing the process 3, 4 and 5. These linear models are "w-MIP" representable, and therefore these models have been written in algebraic equation form. The rest of process models that are non-linear remain as disjunctions. This transformation has allowed to reduce in one the initial number of NLP's to be executed to set up the first master problem.

The next example corresponds to a simultaneous model structure determination and parameter estimation for a FTIR-spectroscopy example by Brink and Westerlund (1995). It has a non-linear objective function subject to linear constraints. These linear constraints have been transformed into disjunctions. They are not "w-MIP" representable, so that the hybrid model cannot be applied to this example. The results obtained are shown below :

RESULTS : FTIR-Spectroscopy

Model -»	PA	PA	PD
Initialization—»	Relaxed	Fixed	Fixed
Constraints	112	112	175
Variables	142	142	172
Discrete V.	30	30	30
Obj. Value	13.98	13.98	13.98
Exec, time	55 sec	100 sec.	7 sec
Iterations	6 major*	11 major*	4major*

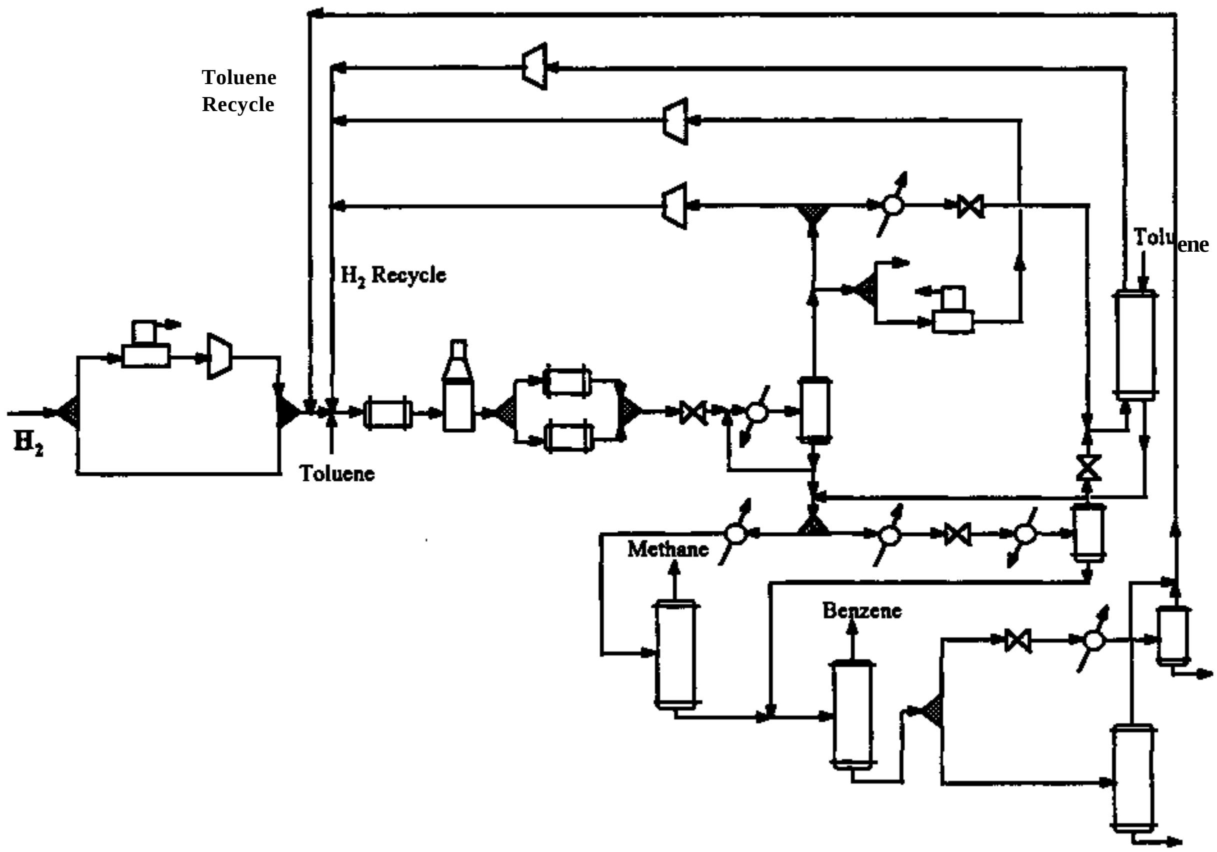

Figure 4. HDA superstructure

The results show for the FTIR-spectroscopy example an impressive performance for the disjunctive model compared to the algebraic MINLP models. The disjunctive model is superior not only in the number of iterations, but also in the execution time, which has been reduced by a factor of eight compared to the algebraic model with relaxed initialization. The explanation for this behavior is that, the convex-hull representation for the linear disjunctions in the master MILP gives a tighter relaxation, improving the prediction of lower bounds.

The third example is a structural flowsheet optimization problem (Kocis and Grossmann, 1989). For this example, the HDA process was modeled in the algebraic and the disjunctive forms. There is a significant number of non-convex non-linear equations in this problem. The augmented penalty strategy (Viswanathan and Grossmann, 1990) was applied to solve it. No w-MIP representable disjunctions were found in this model, therefore, the hybrid model does not apply for this example. The objective for this optimization problem is to obtain the HDA flowsheet with maximum profit. The superstructure for the process can be seen in Figure 4. The results obtained with this problem are shown in the next table.

RESULTS: HDA PLANT

Model -»	PA	PA	PD
Initialization ->	Relaxed	Fixed	Fixed
Constraints	719	719	737
Variables	722	722	717
Discrete V.	13	13	14
Obj. Value	5304.8	5671.4	5810.8
Exec, time	348 sec.	293 sec.	280 sec.
Iterations	lnlp 3 major*	lnlp 2 major*	2 nip 1 major*

* 1 major iteration => 1 nip + 1 master milp

Two initial configurations were given to solve this example in the disjunctive representation. Therefore, two initial NLPs have been solved to obtain the linearizations for all non-linear constraints, in order to set up the first MILP master problem. For this case the disjunctive model (PD) obtained a solution with higher profit than the algebraic models (PA), presumably because zero flows are avoided in (PD). The profit was increased with the disjunctive model in approximately 9.5 % compared to the algebraic model with relaxed MINLP as initial point, and 2.4 % compared to the algebraic with fixed initial point.

The last example is a batch plant design with multiple units in parallel and intermediate storage tanks (Ravemark, 1995). The problem consists of determining the volume of the equipment, the number of units in parallel, and the volume and location of the intermediate storage tanks. The objective is to minimize the plant investment cost. To ensure rigorous lower bounds the equations and constraints were convexified. The problem was modeled in both algebraic and hybrid representation. In the hybrid representation, the disjunctions correspond to the storage tank volume equations and the batch size equations. The example corresponds to a batch plant with five products and six stages. The results are shown in the following table:

RESULTS : BATCH PLANT

Model -»	PA	PA	PH
Initialization ->	Relaxed	Fixed	Fixed
Constraints	1S6	186	187
Variables	112	112	113
Discrete V.	53	53	53
Obj. Value	260383	260383	260383
Exec, time	287 sec.	616 sec.	80 sec.
Iterations	lnlp 10 major*	lnlp 20 major*	lnlp 4 major*

* 1 major iteration => 1 nip -+-1 master milp

For this example, which is difficult to solve in the algebraic mode, the results obtained for the hybrid model are very encouraging. The reduction in time and number of iterations for this example are very significant. The number of sub-problems solved with respect to the better algebraic model (with relaxed MINLP initialization) has been reduced by more than one half. The execution time has been reduced more than 3 times compared to the same model.

The results obtained from the solution of the four examples show that, for some cases, the use of disjunctions is a better modeling alternative compared to the algebraic MINLP. The model and algorithm to be applied for a particular problem depends on the type of equations and constraints the problem has. For convex problem, the disjunctive and hybrid models have shown a very significant improvement compared to the algebraic case. For non-convex problems the main advantage seems to lie in the fact that the disjunctive and hybrid models are less likely to get trapped into poor suboptimal solutions. In contrast, the algebraic model may be trapped more easily due to the difficulties with nonconvex models where flows are set to zero.

CONCLUSIONS

In this work, the solution of hybrid models, with disjunctions and binary variables, for discrete-continuous non-linear optimization problem has been investigated. From this model the pure disjunctive or algebraic model can be derived. The computer code LOGMIP (acronym of Logic Mixed Integer Program) has been developed to deal with this representation. The problem input for LOGMIP can be written in three different forms: hybrid, disjunctive or algebraic form. For the hybrid model an extension of the Logic Based OA algorithm has been presented. LOGMIP has been written in C and linked to GAMS. The possibility to define different starting points has been added. All these capabilities make LOGMIP an important tool for the solution of non-linear discrete-continuous optimizations problems. The novelty of this program is the

capability to handle disjunctions. No other non-linear computer code has been reported in the open literature that can solve problems of this type. The results obtained in the solution of several examples have shown that the disjunctive and hybrid representation outperform the algebraic MINLP in terms of computational time and quality of solutions.

REFERENCES

Brink A. and Westerlund T. (1995). The Joint Problem of Model Structure Determination and Parameter Estimation on Quantitative IR-Spectroscopy. Chemomet. and Intellig. Lab. Sys., 29, 1, pp. 29-36.

Grossmann, I.E. and Kravanja Z.,(1995). Mixed-Integer Nonlinear Programming Thecniques for Process Synthesis Engineering. Comp. Chem. Eng., 19,supl., pp. SI89-204.

Kocis G. and Grossmann I.E.(1989). A Modeling and Decomposition Strategy for the MINLP Optimization of Process Flowsheets. Comp. and Chem. Eng., 13, 7, pp. 797-819.

Ravemark, E. (1995). Optimization Models for Design and Operation of Chemical Batch Processes. Ph. D. Thesis.

Raman R. and Grossmann (1994). Nodelling and Computational Thecniques for Logic Based Integer Programming. Comp. Chem. Eng., 18, 7, pp. 563-578.

Turkay M. and Grossmann I.E.(1996). Logic-Based Algorithms for the Optimal Synthesis of Process Networks. Comp. Chem. Eng., 20, 8, pp. 959-978.

Viswanathan J.V. and Grossmann I.E.(1990). A Combined Penalty Function and Outer-Approximation method for MINLP Optimization. Comp. Chem Eng., 14,7, p. 769-78.