COMPUTATIONAL EXPERIENCE WITH LOGMIP SOLVING LINEAR AND NONLINEAR DISJUNCTIVE PROGRAMMING PROBLEMS

Aldo Vecchietti[(+)] and Ignacio E. Grossmann[(*)]
[(+)]INGAR – Instituto de Desarrollo y Diseño
UTN – Facultad Regional Santa Fe
E-mail: aldovec@ceride.gov.ar
[(*)]Department of Chemical Engineering, Carnegie Mellon University,
Pittsburgh, PA 15213 - USA
E-mail: grossmann@cmu.edu

Abstract

The objectives of this paper are to give a brief overview of the code LogMIP and to report numerical experience on a set of test problems. LOGMIP is currently the only code for disjunctive programming, which has implemented the research work done on this area on the last decade. Major motivations in the development of LogMIP have been to facilitate problem formulation of discrete/continuous optimization problems, and to improve the efficiency and robustness of the solution of these problems, particularly for the nonlinear case. LOGMIP is a software system linked to GAMS for solving problems that are formulated as disjunctive/hybrid programs (Vecchietti and Grossmann, 1999). For linear problems the disjunctive/hybrid model can be automatically reformulated as a mixed-integer (MIP) formulation using either a big-M reformulation, or a convex hull reformulation (Balas, 1979) depending on the choice selected by the user. The other option for solving nonlinear problems is the Logic-Based Outer-Approximation (LBOA) (Turkay and Grossmann, 1996). Computational experience on a set of linear and non-linear problems is reported. These problems correspond to the synthesis of process flowsheets, synthesis, retrofit and design of batch plants, scheduling of a multi-product pipeline, jobshop scheduling, and strip packing problem.

Keywords

Disjunctive Programming, Big-M/Convex Hull Relaxation, Linear and non-linear discrete models.

Introduction

Modeling and solving linear and non-linear discrete continuous program optimization problems is a very challenging task. Models are difficult to build and solve, and the main goal pursued is to generate program models easy to pose, read, understand and solve, goals that frequently compete each other. Modeling frameworks like GAMS, AMPL, AIMMS, etc., facilitate the task of building a model by providing powerful language sentences and management tools. On the other hand, solution algorithms and solvers play an important role on driving the problems to their solutions.

Over the last decade an active research area has been disjunctive programming applied to process system engineering problems (Raman and Grossmann, 1994; Turkay and Grossmann, 1996; Björkqvist and Westerlund; 1999; Lee and Grossmann, 2000). Major motivations of this work have been to facilitate the problem formulation of discrete/continuous optimization problems, and to improve the efficiency and robustness of the solution of these problems, particularly for the non-linear case.

In this paper we report LogMIP, the only code that is currently available for disjunctive programming. Recent theoretical advances and methods in disjunctive programming have been incorporated in the code LogMIP. This is a software system that is implemented in GAMS for solving problems that are formulated as disjunctive/hybrid programs (Vecchietti and Grossmann, 1999). Here by hybrid we mean when part of the model is expressed in disjunctive form, and the other part in mixed-integer form. Recent developments in LogMIP include a language parser for writing and interpreting disjunctive programs. In this sense, LogMIP complements the GAMS modeling framework by adding the capability of writing discrete decisions in the form of disjunctions. The language is based on IF..THEN.. ELSE/ELSIF sentences for representing disjunctions that contain Boolean variables and constraints involving continuous variables (Vecchietti and Grossmann, 2000, Gil and Vecchietti, 2002). The logic operators, implication (=>), equivalence (⇔)), logical AND/OR, negation (~) are used for writing relationships between the Boolean variables in the form of logic propositions. Special sentences such as ATLEAST, ATMOST and EXACTLY, are also used to express relationships for Boolean variables in a more natural and expressive form. Based on these features, LogMIP can currently be used to model linear and non-linear disjunctive/hybrid programs. More detailed description can be found in http://www.ceride.gov.ar/logmip/.

Regarding the solution algorithms and methods, for linear/non-linear problems the disjunctive-hybrid model can be automatically reformulated as a mixed-integer linear(MIP)/non-linear(MINLP) program formulation using either a linear/non-linear big-M or convex hull reformulation (Balas, 1979, Lee and Grossmann, 2000) respectively. Once the reformulation is performed any MIP/MINLP solver can be applied to solve the problem. One option for solving nonlinear problems is the Logic-Based Outer-Approximation (LBOA) (Turkay and Grossmann, 1976). For both linear and nonlinear problems the Disjunctive Branch and Bound (DB&B) method (Lee and Grossmann, 2001), which is currently under development, can also be used to solve disjunctive programs.

In the following section we present the numerical experience with LogMIP on a set of linear and non-linear problems. Most of these problems are in the area of process system engineering and include for instance the synthesis of process flowsheets, synthesis, retrofit and design of batch plants, scheduling of a multi-product pipeline, jobshop scheduling, and strip packing problems. Due to space limitations we briefly describe each problem and in some cases highlight key disjunctions that were used.

Linear Examples

Jobshop scheduling

The first set of examples is the jobshop-scheduling problem (Raman and Grossmann, 1994). In this problem, there is a set of jobs $i \in I$ that must be processed in a sequence of stages but not all jobs require all stages. Zero wait transfer policy is assumed between stages. To obtain a feasible solution is necessary to eliminate all clashes between jobs. It requires that no two jobs be performed at any stage at the same time. This is expressed by the following disjunction:

$$\begin{bmatrix} Y_{ik} \\ t_i + \sum_{\substack{m \in J(i) \\ m \le j}} \tau_{im} \le t_k + \sum_{\substack{m \in J(k) \\ m < j}} \tau_{km} \end{bmatrix} \vee \begin{bmatrix} \neg Y_{ik} \\ t_k + \sum_{\substack{m \in J(k) \\ m \le j}} \tau_{km} \le t_i + \sum_{\substack{m \in J(k) \\ m < j}} \tau_{im} \end{bmatrix} \quad (1)$$

where t_i is the starting time of job i and τ_{ij} the processing time of job i in stage j. The meaning of (1) is that either the job i precede job k or viceversa in the stage j where a clash can occur. The objective is to minimize the makespan.

Strip packing problem

The problem considered in this section deals with the problem of fitting a set of rectangles of fixed dimensions in a strip with fixed width W and variable length L that is to be minimized (Sawaya and Grossmann, 2003). If the N rectangles have height h_i and length l_i the problem can be formulated as follows:

$$\min \quad L$$

$$\text{s.t.} \quad L \ge x_i + l_i$$

$$\begin{bmatrix} Y_{ij1} \\ x_i + l_i \le x_j \end{bmatrix} \vee \begin{bmatrix} Y_{ij2} \\ x_j + l_j \le x_i \end{bmatrix} \vee \begin{bmatrix} Y_{ij3} \\ y_i + h_i \le h_j \end{bmatrix} \vee \begin{bmatrix} Y_{ij4} \\ y_j + h_j \le h_i \end{bmatrix} \quad (2)$$

where $0 \le x_i \le U - l_i$, $h_i \le y_i \le W$. No rotation between the rectangles is assumed, and the meaning of this disjunction is to state that either rectangle j is to the left or to the right of rectangle i, or above or below.

Retrofit

Another problem solved is the one proposed by Jackson and Grossmann (2002) related to the retrofit design of a processes network. This type of problem is difficult to solve due to the constraints involved in the preexisting design and operations imposed to the network. They proposed a disjunctive model involving the following constraints:

$$\bigvee_{m \in M_i} \begin{bmatrix} Y_{im}^t \\ A_{im}^t x_i^t \le b_{im}^t \end{bmatrix} \qquad (3)$$

$$\bigvee_{m \in M_i} \begin{bmatrix} W_{im}^t \\ c_i^t = FC_{im}^t \end{bmatrix} \qquad (4)$$

The first disjunction selects the operation mode m between all possible modes M_i for process i in the planning horizon

$t \in T$, while the second disjunction selects the cost c_i^t corresponding to the first choice made, FC^t_{im} is the cost coefficient corresponding to the operation mode *m* at time *t* for process *i*. Boolean variables Y^t_{im} and W^t_{im} are related by logic propositions. The model also includes mass balances, and other logic proposition constraints.

Multi-product pipeline scheduling

This example was taken from Cafaro and Cerda (2003). It corresponds to a scheduling model of a multi-product petroleum refinery pipeline, where large amounts of different products must be delivered from the refinery to the depots. It is a complex continuous/discrete model. The main discrete decisions of the model are: the product orders to process, the amount of product to pump, the product orders to feed into the pipeline and the product transference (or not) from the pipeline to the depots. It is a large model applied to an interesting industrial problem.

Non-linear examples

The non-linear solution algorithm currently implemented in LogMIP is Logic-Based Outer Approximation. This algorithm was developed mainly for the synthesis of process networks and that is the reason for which is applied for problems formulated by special two-term disjunction formulated in (5).

$$\begin{bmatrix} Y_j \\ mass\ \ balance\ \ constraint \\ energy\ \ balance\ \ constraint \\ cost\ \ equation \end{bmatrix} \vee \begin{bmatrix} \neg Y_j \\ flux = 0 \\ cost = 0 \\ some\ variables = 0 \end{bmatrix} \quad (5)$$

The meaning of (5) is that if the piece of equipment j of the process network is selected the mass and energy balances and the cost equation apply, otherwise the flux, cost and some other variables goes to zero.

Even with the restriction of two terms per disjunction, optimization problems other than the synthesis of process networks can be formulated as for instance, the design and retrofit of multiproduct batch plants, the covariance matrix determination for an infrared spectroscopy experiment.

The problems solved with LogMIP corresponds to the synthesis of a process network, the multiproduct batch plant design with intermediate storage tanks (Vecchietti and Grossmann, 1999), the retrofit of a multiproduct batch plant, the covariance matrix determination for a infrared spectroscopy, and the synthesis of a methanol plant.

Results

LogMIP greatly facilitates the task of posing a discrete model through the use of the disjunctions. The error messages given by the language compiler guide the task of writing a model when it is not formulated according to its syntactic and semantic rules. To write discrete decisions by means of disjunctions make the model more clear and easy to understand.

In LogMIP, the linear models can be transformed into a MILP model by using the Big-M or convex-hull relaxations, and then solved by any B&B solver like OSL, CPLEX, Xpress, etc. The process of transforming and solving the problem is done systematically without user intervention. The Big-M transformation has a couple of options to calculate the best value of the M parameter if appropriate bounds on the variables are given.

Table 1 shows the results obtained on the solution of the disjunctive linear problems of different sizes. The Big-M and convex hull relaxations algorithms were applied to all problems. The first column shows the problem name, and then the number of equations, variables and discrete variables are listed. Finally the CPU time, number of iterations and number of nodes taken to the B&B algorithm to reach the solution for both the Big-M relaxation (BM) and the convex hull (CH) are shown. The solver used was CPLEX 7.5 without preprocessing and cuts, using a Pentium III-850 Mhz processor.

From Table 1 can be seen that the somewhat unexpected result that the Big-M reformulation takes less CPU time, number of nodes, and iterations for the jobshop problems, except for the retrofit, retrofit_big and cut-1 problems. For the cases of the strip packing problems (cut-1 and cut-2) although the Big-M relaxation takes less CPU time and number of iterations, the convex hull relaxations requires fewer number of nodes to reach the solution. It must be noted that no revisions were made on the problem formulation in terms of variables and constraints; the original problem formulations were converted in disjunctive problems. In this sense the problems formulation can be improved by adding/eliminating constraints that accelerate/slow-down the problem solution respectively. In that case the convex hull relaxation algorithm can be as competitive as the Big-M relaxation. Observe also the impressive behavior of the Convex-Hull against the Big-M reformulation for the case of the retrofit_big problem where reduction in orders of magnitude in CPU time has been achieved.

Table 2 shows the results of using LogMIP on solving non-linear discrete-continuous problems by applying a modified version of the-Logic Based Outer-Approximation algorithm. In this modified version, we have included the termination criterion of no improvement in the NLP subproblem as is used in DICOPT++.

Four problems have been solved. In Table 2, the number of equations, variables and discrete variables used in the models are shown, together with the number of initial NLP's sub-problems, number of total NLP's sub-problems and master's sub-problems solved, with the CPU time needed to reach the solution. CONOPT and CPLEX were used as NLP and master sub-problems solvers, respectively, on the same computer described before

Table 1.Results obtained on the solution of linear disjunctive models

Problem	# Equat.	# var.	# disc. var.	CPU BM (sec.)	iter. BM	nodes BM	CPU CH (sec.)	iter CH	nodes CH
cut-1	30	34	24	0.05	64	32	0.11	92	0
cut-2	236	202	180	7.19	29196	4673	43.78	44434	873
jobshop-1	13	21	12	0.05	5	0	0.001	9	0
jobshop-2	78	253	245	0.16	341	86	0.93	2155	200
jobshop-3	219	319	320	3.57	10034	2209	154.45	207605	20600
retrofit	211	160	72	0.72	1449	136	0.11	122	0
retrofit-big	2935	1635	336	5719	4009307	439273	17.19	4898	47
pipeline	3385	1640	387	327.52	89820	13657	940.65	420574	23750

Table 2. Results obtained on the solution of non-linear disjunctive model

Problem	# Equat.	# var.	# disc. var.	# nlp initial.	#nlp total	# master	CPU master (sec.)	CPU nlp (sec.)
8 processes	70	42	8	3	4	1	1.2	0.22
batch-design	217	113	54	1	2	2	0.82	0.18
spectroscopy	162	99	30	1	14	14	1.71	0.74
methanol	557	310	17	2	5	3	0.55	1.27

Conclusions

The development of LogMIP has been part of a research effort that has taken several years. The idea behind this solver is to provide a new tool for the formulation and solution of discrete-continuous problems through the use of disjunctions and logic propositions. As was shown with several test problems, linear disjunctive problems can readily be formulated with LogMIP, which will then automatically perform the transformation into mixed-integer programs either by using the linear Big-M or convex hull relaxation. The numerical results have shown that either option can perform better, although the big-M performed much better than expected. In LogMIP the user can trivially specify any of the two, which allows easily exploring both formulations. For non-linear problems the Logic-Based Outer-approximation algorithm was implemented, although we plan to include in the near future the disjunctive B&B. The numerical results have shown that one can efficiently and reliably solve nonlinear disjunctive problems. This follows from the fact that the NLP subproblems exclude the equations and variables that do not apply for a fixed choice of the Boolean variables.

Acknowledgments

The authors would like to acknowledge financial support from Universidad Tecnologica Nacional project 25-O053 and the National Science Foundation under grant INT-0104315**.** The authors are also grateful for the formulation on the petroleum pipeline scheduling problem provided by Diego Cafaro and Jaime Cerdá.

References

Balas, E.. "*Disjunctive programming*". Discrete Optimizations II, Annals of Discrete Mathematics, 5, North Holland, 1979.

Bjorkqvist, J. and Westerlund T.(1999), *"Automated Reformulation of Disjunctive Constraints in MINLP Optimization"*, Computers and Chemical Engineering, 23,Supp., pp. S11-S14.

Cafaro D. and Cerdá J, *"A ContinuousTime approach to Multi-Product Pipeline Scheduling"*.Proceedings of ESCAPE 13 (2003).

Gil J.J. and Vecchietti A., *"Using design patterns for a compiler modeling for posing disjunctive optimization programs"*. Proceedings of Argentinian Symposium of Software Engineering (ASSE 2002) (2002).

Jackson J. and Grossmann I.E., *"High Level Optimization Model for the retrofit planning of process networks"*,I&EC Research 41,3762-3770 (2002)..

Lee S. and Grossmann I.E. *"New algorithm for Nonlinear Generalized Disjunctive Programming"*. Computers and Chemical Engineering, , 24, 9, 2125-2142, 2000.

Raman R. and Grossmann I.E., *"Modeling and Computational Techniques for Logic Based Integer Programming"*. Comp. Chem. Eng., 18 (7), 563-578, 1994.

Sawaya, N.S.. and Grossmann I.E., *"A cutting plane method for solving linear generalized disjunctive programming problems"*. Proceedings of PSE2003, 1032 (2003).

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

Vecchietti A. and Grossmann I.E. **"***LOGMIP: A Disjunctive 0-1 Nonlinear Optimizer for Process System Models"*. Comp. Chem. Eng., 23, 555-565, 1999.

Vecchietti A. and Grossmann I.E. *"Modeling issues and implementation of language for disjunctive programming"*. Comp. Chem. Eng., 24, 9, 2143-2155, 2000.