Computational Experience with Logmip
Solving Linear and Nonlinear
Disjunctive Programming Problems

Aldo Vecchietti, INGAR – Instituto de Desarrollo y Diseño
E-mail:aldovec@ceride.gov.ar
Ignacio E. Grossmann, Carnegie Mellon University,
E-mail: grossmann@cmu.edu
Alexander Meeraus, GAMS Development
E-mail:alex@gams.com
Motivation

- Modeling framework for facilitating the formulation of models that can be expressed in terms of algebraic, disjunctive and symbolic logic expressions.

- Language compiler within GAMS for disjunctions expressions, logic constraints and logic propositions.

- Solution algorithms and techniques for solving linear /nonlinear disjunctive programming problems.
Generalized Disjunctive Programming (GDP)

- Raman and Grossmann (1994)

\[
\begin{align*}
\min & \quad Z = \sum_k c_k + f(x) \\
\text{s.t.} & \quad r(x) \leq 0 \\
& \quad \begin{bmatrix}
Y_{jk} \\
g_{jk}(x) \leq 0 \\
c_k = \gamma_{jk}
\end{bmatrix}, \quad k \in K \\
& \quad \Omega(Y) = \text{true} \\
& \quad x \in \mathbb{R}^n, \quad c_k \in \mathbb{R}^1 \\
& \quad Y_{jk} \in \{\text{true, false}\}
\end{align*}
\]

Objective Function
Common Constraints
Disjunction
Constraints
Fixed Charges
Logic Propositions
Continuous Variables
Boolean Variables

Relaxation?
**Big-M MINLP (BM)**

- MINLP reformulation of GDP

\[
\begin{align*}
\text{min } Z &= \sum_{k \in K} \sum_{j \in J_k} \gamma_{jk} y_{jk} + f(x) \\
\text{s.t. } r(x) &\leq 0 \\
g_{jk}(x) &\leq M_{jk} (1 - y_{jk}) , j \in J_k, k \in K \\
\sum_{j \in J_k} y_{jk} &= 1, \quad k \in K \\
\quad Ay &\leq a \\
\quad x &\geq 0, \quad y_{jk} \in \{0, 1\}
\end{align*}
\]

**NLP Relaxation**

\[
0 \leq y_{jk} \leq 1
\]
Convex Hull Formulation

- **Consider Disjunction** \( k \in K \)

\[
\begin{bmatrix}
Y_{jk} \\
g_{jk}(x) \leq 0 \\
c = \gamma_{jk}
\end{bmatrix}
\]

- **Theorem: Convex Hull of Disjunction** \( k\)  *(Lee, Grossmann, 2000)*
  - **Disaggregated variables** \( v^j \)
    \[
    \{(x, c) \mid x = \sum_{j \in J_k} v^j, \quad c = \sum_{j \in J_k} \lambda_{jk} \gamma_{jk},
    
    0 \leq v^j \leq \lambda_{jk} U_{jk}, \quad j \in J_k
    
    \sum_{j \in J_k} \lambda_{jk} = 1, \quad 0 < \lambda_{jk} \leq 1,
    
    \lambda_{jk} g_{jk}(v^j / \lambda_{jk}) \leq 0, \quad j \in J_k\}
    
  - \( \lambda_j \) - weights for linear combination

- Generalization of Stubbs and Mehrotra (1999)
Convex Relaxation Problem (CRP)

**CRP:**

$$\text{min } Z = \sum_{k \in K} \sum_{j \in J_k} \gamma_{jk} \lambda_{jk} + f(x)$$

$$s.t. \quad r(x) \leq 0$$

$$x = \sum_{j \in J_k} v_{jk}, k \in K$$

$$0 \leq v_{jk} \leq \lambda_{jk} U_{jk}, j \in J_k, k \in K$$

$$\sum_{j \in J_k} \lambda_{jk} = 1, k \in K$$

$$\lambda_{jk} g_{jk} (v_{jk} / \lambda_{jk}) \leq 0, \quad j \in J_k, k \in K$$

$$A \lambda \leq a$$

$$x, v_{jk} \geq 0, 0 \leq \lambda_{jk} \leq 1, \quad j \in J_k, k \in K$$

**Property:** The NLP (CRP) yields a lower bound to optimum of (GDP).

**Remark:** MINLP reformulation by setting $\lambda_{jk} = 0,1$
Theorem: The relaxation of (CRP) yields a lower bound that is greater than or equal to the lower bound that is obtained from the relaxation of problem (BM):

\[
\text{RBM: } \quad \min Z = \sum_{k \in K} \sum_{j \in J_k} \gamma_{jk} y_{jk} + f(x) \\
\text{s.t. } \quad r(x) \leq 0 \\
g_{jk}(x) \leq M_{jk} (1 - y_{jk}) \quad j \in J_k, k \in K \\
\sum_{j \in J_k} y_{jk} = 1, \quad k \in K \\
Ay \leq a \\
x \geq 0, \quad 0 \leq y_{jk} \leq 1
\]
Methods Generalized Disjunctive Programming

GDP

Logic based methods

Branch and bound
(Lee & Grossmann, 2000)

Decomposition
Outer-Approximation
Generalized Benders
(Turkay & Grossmann, 1997)

Reformulation MI(N)LP

Outer-Approximation
Generalized Benders
Extended Cutting Plane

Convex-hull Big-M
Cutting plane
(Sawaya & Grossmann, 2004)

LogMIP
LogMIP Language Syntax

Properties of LogMIP language
- Simple syntax
- Semantic close to disjunction expression to model blocks, exclusive between them
- The syntax must be known for a regular user
- The syntax construction must allow the definition of embedded disjunctions

Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; &gt;</td>
<td>The phrase enclosed is a syntactic rule</td>
</tr>
<tr>
<td>Word</td>
<td>Token</td>
</tr>
<tr>
<td>[ ]</td>
<td>Optional expression</td>
</tr>
<tr>
<td>{ }</td>
<td>Expression that can be repeated</td>
</tr>
<tr>
<td>( )</td>
<td>Expression enclosed can be grouped</td>
</tr>
<tr>
<td>::=</td>
<td>Define like</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Rules describing LogMIP syntax

<LOGMIP Model> ::=<Disjunction Declaration> {; <LogMIP Model>} | <Disjunction Definition> {; <LogMIP Model>} ;
<Disjunction Declaration> ::=disjunction identifier {, identifier} ;
<Disjunction Definition> ::=disjunction identifier is <If Sentence> ;
<If Sentence> ::=if <condition> then <components> else <components> end if ;
| if <condition> then <components> {elsif <condition> then <components>} end if ;
| entity {; entity} [; <If Sentence>] ;
Examples

Modeling two terms disjunction

\[
\begin{bmatrix}
\text{True} \\
\text{Constraint } 1
\end{bmatrix} \lor 
\begin{bmatrix}
\text{False} \\
\text{Constraint } 2
\end{bmatrix}
\]

IF (condition\(_1\)) THEN
Constraints to be considered when condition\(_1\) is True
ELSE
Constraints to be considered when condition\(_1\) is False
END IF

Modeling a multi-term disjunction

\[
\begin{bmatrix}
1 \\
\text{Constraints } 1
\end{bmatrix} \lor 
\begin{bmatrix}
2 \\
\text{Constraints } 2
\end{bmatrix} \lor \ldots \lor 
\begin{bmatrix}
N \\
\text{Constraints } N
\end{bmatrix}
\]

IF (condition\(_1\)) THEN
Constraints to be considered when condition\(_1\) is True
ELSIF (condition\(_2\)) THEN
Constraints to be considered when condition\(_1\) is True
ELSIF (condition\(_3\)) THEN
...
ELSIF (condition\(_N\)) THEN
Constraints to be considered when condition\(_1\) is True
END IF
Small Example

\[ \min c + 2x_1 + x_2 \]

s.a.:

\[
\begin{bmatrix}
Y_1 \\
-x_1 + x_2 + 2 \leq 0 \\
c = 5
\end{bmatrix} \lor \begin{bmatrix}
Y_2 \\
2 - x_2 \leq 0 \\
c = 7
\end{bmatrix}
\]

\[
\begin{bmatrix}
Y_3 \\
x_1 - x_2 \leq 1
\end{bmatrix} \lor \begin{bmatrix}
\neg Y_3 \\
x_1 = 0
\end{bmatrix}
\]

\[
Y_1 \land \neg Y_2 \Rightarrow \neg Y_3 \\
\neg(Y_2 \land Y_3)
\]

\[
0 \leq x_1 \leq 5 \quad 0 \leq x_2 \leq 5 \quad c \geq 0
\]

\[ Y_j \in \{true, false\}, j = 1, 2, 3 \]
$ONTEXT BEGIN LOGMIP

DISJUNCTION D1, D2;
D1 IS
IF Y('1') THEN
   EQUAT1;
   EQUAT2;
ELSIF Y('2') THEN
   EQUAT3;
   EQUAT4;
ENDIF;
Y('1') and not Y('2') -> not Y('3');
Y('2') -> not Y('3') ;
Y('3') -> not Y('2') ;

$OFFTEXT END LOGMIP

OPTION MIP=LOGMIPM;
MODEL PEQUE /ALL/;
SOLVE PEQUE USING MIP MINIMIZING Z;

SET I /1*3/;
SET J /1*2/;
BINARY VARIABLES Y(I);
POSITIVE VARIABLES X(J), C;
VARIABLE Z;
EQUATIONS EQUAT1, EQUAT2, EQUAT3,
   EQUAT4, EQUAT5, EQUAT6,
   DUMMY, OBJECTIVE;

EQUAT1.. X('2')- X('1') + 2 =L= 0;
EQUAT2.. C =E= 5;
EQUAT3.. 2 - X('2') =L= 0;
EQUAT4.. C =E= 7;
EQUAT5.. X('1')-X('2') =L= 1;
EQUAT6.. X('1') =E= 0;
DUMMY.. SUM(I, Y(I)) =G= 0;

OBJECTIVE.. Z =E= C + 2*X('1') + X('2');
X.UP(J)=20;
C.UP=7;

GAMS
Recent developments

- Disjunctions specified with IF Then ELSE statements

```
DISJUNCTION D1(I,K,J);
D1(I,K,J)
    with (L(I,K,J)) IS
IF Y(I,K,J) THEN
    NOCLASH1(I,K,J);
ELSE
    NOCLASH2(I,K,J);
ENDIF;
```

- Logic can be specified in symbolic form (⇒, OR, AND, NOT) or special operators (ATMOST, ATLEAST, EXACTLY)
- Linear case: MILP reformulation big-M, convex hull
- Nonlinear: Logic-based OA
LogMIP Compiler Architecture

MBL Compiler -> Lexer -> Parser -> Semantic Analizer -> Model Composer -> Logic Extractor

Input File

Errors Manager

Errors Manager 

LogMIP Symbol Table

LogMIP Solver 

Solution

Filter

Pipe

GAMS
Problem statement: *Hifi (1998)*

- Given a set of small rectangles with width $w_i$ and length $l_i$.
- Large rectangular strip of fixed width $W$ and unknown length $L$.
- Objective is to fit small rectangles onto strip without overlap and rotation while minimizing length $L$ of the strip.

![Diagram of strip-packing problem](image)
## GDP Model For Strip-packing Problem

**Objective function**  
Minimize length

**Disjunctive constraints**  
No overlap between rectangles

**Bounds on variables**  

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min $lt$</td>
<td>Minimize length</td>
</tr>
<tr>
<td>$st. \quad lt \geq x_i + L_i$</td>
<td>Subject to: $lt$ is greater than or equal to $x_i + L_i$ for all $i \in N$</td>
</tr>
<tr>
<td>$x_i + L_i \leq x_j$</td>
<td>Disjunctive constraints: $x_i + L_i$ is less than or equal to $x_j$ for all $i, j \in N, i &lt; j$</td>
</tr>
<tr>
<td>$x_j + L_j \leq x_i$</td>
<td></td>
</tr>
<tr>
<td>$y_i - H_i \geq y_j$</td>
<td></td>
</tr>
<tr>
<td>$y_j - H_j \geq y_i$</td>
<td></td>
</tr>
<tr>
<td>$x_i \leq UB_i - L_i$</td>
<td>Bounds on variables: $x_i$ is less than or equal to $UB_i - L_i$ for all $i \in N$</td>
</tr>
<tr>
<td>$H_i \leq y_i \leq W$</td>
<td></td>
</tr>
</tbody>
</table>

$lt, x_i, y_i \in \mathbb{R}_+ \ , \ Y_{ij}^1, Y_{ij}^2, Y_{ij}^3, Y_{ij}^4 \in \{True, False\}$  

For all $i, j \in N, i < j$
- Set of jobs $i \in I$ must be processed sequentially on a set of consecutive stages $j \in J$.
  - All jobs can be sequenced on a subset of stages $j \in J(i)$.
  - Zero-wait transfer is assumed between stages.
- Objective is to obtain a schedule that *minimizes makespan*. 

![Diagram of stages with A and B tasks]

**Stage 1**
```
A | B
---|---
```

**Stage 2**
```
   | B
---|---
```

**Stage 3**
```
A | B
---|---
```
GDP Model For
Zero-wait Job-shop Scheduling Problem

\[
\begin{align*}
\text{Min} & \quad m_s \\
\text{s.t.} & \quad m_s \geq t_i + \sum_{j \in J(i)} TAU_{ij} \quad \forall i \in I \\
& \quad t_j + \sum_{m \in J(j)} TAU_{jm} \leq t_k + \sum_{m \in J(k)} TAU_{km} \quad \forall j \in C_{ik}, \forall i, k \in I, i < k
\end{align*}
\]

Objective function
Minimize makespan

Disjunctive constraints
No clashes between jobs

\[m_s, t_j \in \mathbb{R}_+, Y_{ik}^1, Y_{ik}^2 \in \{True, False\} \quad \forall i, k \in I, i < k\]
**Retrofit Planning Problem**

**Problem Statement:** *Jackson J. & Grossmann I.E. (2002)*

- Retrofit: Redesign of existing plant.
  - Improvements such as higher yield, increased capacity, energy reduction.
- Objective is to identify modifications that maximize economic potential, given time horizon and limited capital investments.
GDP Model For Retrofit Planning Problem

**Objective function**
Maximize economic potential

**Common constraints**
Mass balances

**Disjunctive constraints**
Conversion/Capacity scenarios

**Disjunctive constraints**
Energy reduction scenarios

**Common constraint**
Investment limit

**Logic constraints**
Connect disjunctions
# Computational Results

## Linear Disjunctive Problems

<table>
<thead>
<tr>
<th>Problem</th>
<th># disc. var.</th>
<th># var.</th>
<th># Equat.</th>
<th>CPU BigM (sec.)</th>
<th>iter. BigM</th>
<th>nodes BigM</th>
<th>CPU CH (sec.)</th>
<th>iter CH</th>
<th>nodes CH</th>
</tr>
</thead>
<tbody>
<tr>
<td>cut-1</td>
<td>24</td>
<td>34</td>
<td>30</td>
<td>0.05</td>
<td>64</td>
<td>32</td>
<td>0.11</td>
<td>92</td>
<td>0</td>
</tr>
<tr>
<td>cut-2</td>
<td>180</td>
<td>202</td>
<td>236</td>
<td>7.19</td>
<td>29,196</td>
<td>4673</td>
<td>43.78</td>
<td>44,434</td>
<td>873</td>
</tr>
<tr>
<td>jobshop-1</td>
<td>12</td>
<td>21</td>
<td>13</td>
<td>0.05</td>
<td>5</td>
<td>0</td>
<td>0.001</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>jobshop-2</td>
<td>245</td>
<td>253</td>
<td>78</td>
<td>0.16</td>
<td>341</td>
<td>86</td>
<td>0.93</td>
<td>2155</td>
<td>200</td>
</tr>
<tr>
<td>jobshop-3</td>
<td>320</td>
<td>319</td>
<td>219</td>
<td>3.57</td>
<td>10,034</td>
<td>2209</td>
<td>154.45</td>
<td>207,605</td>
<td>20,600</td>
</tr>
<tr>
<td>jobshop-4</td>
<td>125</td>
<td>131</td>
<td>106</td>
<td>0.11</td>
<td>288</td>
<td>55</td>
<td>2.25</td>
<td>3035</td>
<td>299</td>
</tr>
<tr>
<td>retrofit</td>
<td>72</td>
<td>160</td>
<td>211</td>
<td>0.72</td>
<td>1449</td>
<td>136</td>
<td>0.11</td>
<td>122</td>
<td>0</td>
</tr>
<tr>
<td>retrofitNS</td>
<td>336</td>
<td>1685</td>
<td>2935</td>
<td>1810.11</td>
<td>6,995,748</td>
<td>494,291</td>
<td>2.08</td>
<td>3019</td>
<td>18</td>
</tr>
<tr>
<td>pipeline</td>
<td>387</td>
<td>1640</td>
<td>3385</td>
<td>327.52</td>
<td>89,820</td>
<td>13,657</td>
<td>940.65</td>
<td>420,574</td>
<td>23,750</td>
</tr>
</tbody>
</table>

Performance of big-M and Convex Hull (CH) is problem dependent

*CPLEX 7.5*
### Computational Results

#### Nonlinear Disjunctive Problems

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

*Logic-based Outer-Approximation (CPLEX 7.5/CONOPT)*
Conclusions

1. LogMIP provides unique capability for formulating and solving discrete/continuous optimization problems (GDPs) in GAMS environment:
   - Handling disjunctions
   - Handling symbolic logic propositions

2. Numerical results show following:
   - For linear GDPs performance of big-M and convex hull is problem dependent
   - For nonlinear GDPs robustness increased with logic-based outer-approximation

3. Future work:
   - Extend big-M and convex hull reformulation to nonlinear GDPs
   - Branch and bound for linear and nonlinear GDPs

LogMIP can be downloaded from http://www.ceride.gov.ar/logmip/