Wednesday, October 29

Program of the day

- Overview of course, exercises
- Introduction to Integer Programming
- Modelling (Williams, chapter 9)
- Applications: Opencast mining
- Ladies or tigers
Purpose of the course

- To learn to build complex models from real life using Mathematical Programming
- To know techniques for solving Mathematical Programming models
- To understand that some problems can be solved efficiently and some cannot
- To learn that the same problem may be formulated in different ways, which are easier/harder to solve
- To know a number of techniques for decreasing solution times (or turn a problem from practically “unsolvable” to “solvable”)
Integer Programming

In first part of course: continuous variables, linear constraints

- Most products are integral (apart from liquids)  
  *Airplane production, Tomato Soups*

- Structure of problem leads to IP  
  *Graph problems*

- Nonlinear objective functions or constraints occur frequently

- Logical conditions  
  "If I use vegetable oil in the blend, then I must also add 5ml of preservatives"
### Integer Programming

General formulation:

\[
\begin{align*}
\text{maximize} & \quad \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \quad \sum_{j=1}^{n} a_{1j} x_j \leq b_1 \\
& \quad \vdots \\
& \quad \sum_{j=1}^{n} a_{mj} x_j \leq b_m \\
x_j & \geq 0, \quad j = 1, \ldots, n, \quad x \text{ integer}
\end{align*}
\]

where

- \( A \) is a \( m \times n \) matrix
- \( b \) is a \( m \)-vector
- \( c \) is a \( n \)-vector

- IP: integer programming model
- PIP: pure integer programming model
- MIP: mixed integer programming model

ILP is not ideal model, but bounds from LP (Edmonds)
Integer Programming

IP powerful method for modelling

- LP easy to solve by e.g. Simplex (polynomial time by interior-point methods).
- General IP is NP-hard
- Many concrete problems may be solved despite NP-hardness
- Specific techniques for individual problems

Special problems

- travelling salesman problem
- project selection
- transportation problem
- assignment problem
- assembly line balancing
- set partitioning problem
- aircrew scheduling
- depot location problem
- sequencing problem
- job-shop scheduling
Hardness of IP

maximize \( x_1 + x_2 \)
subject to \[-2x_1 + 2x_2 \geq 1\]
\[-8x_1 + 10x_2 \leq 13\]
\( x_1, x_2 \geq 0, \text{integer} \)

Solutions are not found in extreme points (or nearby)

Find convex hull
Model building

- Indicator variables
- Non-convex problems
- Nonlinear functions
- Logical expressions
- Transformation of “human text” to ILP
Indicator variables

- Most important modelling tool!
- $\delta \in \{0, 1\}$
- $\delta = 1$ if and only if some event happens.

Model:

$$\delta = 1 \iff x > 0$$
$$\delta \in \{0, 1\}, \ x \geq 0$$

\[\delta = 1 \Rightarrow x > 0\]
\[\delta = 1 \Rightarrow x \geq \varepsilon\] \quad \varepsilon \text{ level for } x \text{ regarded as 0}
\[x - \varepsilon \delta \geq 0, \ \delta \in \{0, 1\}\]

\[x > 0 \Rightarrow \delta = 1\]
\[\delta = 0 \Rightarrow x = 0\]
\[x - M \delta \leq 0, \ \delta \in \{0, 1\}\] \quad M \text{ upper bound on } x
Indicator variables

Logical implications \( X \iff Y \)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
<td>X (\Rightarrow) Y</td>
<td>X (\Leftarrow) Y</td>
<td>(\neg X \Rightarrow \neg Y)</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Fixed-charge problem

cost function

\[ f(x) = \begin{cases} 
ax + b & \text{if } x > 0 \\
0 & \text{if } x = 0
\end{cases} \]

Model:

\[
\begin{align*}
\text{minimize} & \quad ax + \delta b \\
\text{subject to} & \quad x - M\delta \leq 0 \\
& \quad x - \epsilon\delta \geq 0 \\
& \quad \delta \in \{0, 1\}, \quad x \geq 0
\end{align*}
\]
Non-convex problems

constraints:

\[
\begin{align*}
x_1 + x_2 & \leq b \\
x_1 & \geq 1 \text{ or } x_2 & \geq 1 \\
x_1, x_2 & \geq 0
\end{align*}
\]

Modeling tool

\[
\delta = 1 \Rightarrow x \geq \varepsilon
\]

Two indicator variables \( \delta_1, \delta_2 \):

\[
\begin{cases}
x_1 + x_2 & \leq b \\
\delta_1 + \delta_2 & \geq 1 \\
x_1 - 1\delta_1 & \geq 0 \\
x_2 - 1\delta_2 & \geq 0 \\
x_1, x_2 & \geq 0, \\
\delta_1, \delta_2 & \in \{0, 1\}
\end{cases}
\]
Indicator variables

“if A is included in the blend then B is included in the blend”

can be modeled by using constraints

<table>
<thead>
<tr>
<th>$x_A &gt; 0 \Rightarrow \delta = 1$</th>
<th>$x_A - M\delta \leq 0$, $\delta \in {0, 1}$</th>
<th>$M$ upper bound on $x_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta = 1 \Rightarrow x_B &gt; 0$</td>
<td>$x_B - \varepsilon\delta \geq 0$, $\delta \in {0, 1}$</td>
<td>$\varepsilon$ level for $x_B$ regarded as 0</td>
</tr>
</tbody>
</table>

Example

Assume that $x_A$ and $x_B$ are proportions in blend i.e. $x_A + x_B = 1$.

$$M = 1 \quad \varepsilon = 0.01$$

Formulation:

$$\begin{cases} 
    x_A - \delta & \leq 0 \\
    x_B - 0.01\delta & \geq 0 \\
    \delta & \in \{0, 1\}
\end{cases}$$
Indicator variables for linear inequalities

Example

“If resources needed for production of \( x_1, x_2 \) and \( x_3 \) are below the limit of one truck, then use the other truck for some other purpose.”

General form

\[
\sum_{j=1}^{n} a_j x_j \leq b \iff \delta = 1, \quad \delta \in \{0, 1\}
\]

- \( \sum_{j=1}^{n} a_j x_j \leq b \iff \delta = 1 \) has the MIP formulation

\[
\sum_{j=1}^{n} a_j x_j + M\delta \leq M + b
\]

where \( M \) is upper bound on \( \sum_{j=1}^{n} a_j x_j - b \)

\( \delta = 1: \sum_{j=1}^{n} a_j x_j \leq b \)

\( \delta = 0: \sum_{j=1}^{n} a_j x_j - b \leq M \)
Indicator variables for linear inequalities

- $\sum_{j=1}^{n} a_j x_j \leq b \Rightarrow \delta = 1$ has the MIP formulation

\[
\sum_{j=1}^{n} a_j x_j - (m - \varepsilon)\delta \geq b + \varepsilon
\]

where $m$ is lower bound on $\sum_{j=1}^{n} a_j x_j - b$.

\[
\delta = 0 \Rightarrow \sum_{j=1}^{n} a_j x_j \geq b + \varepsilon
\]

$\delta = 0$: $\sum_{j=1}^{n} a_j x_j \geq b + \varepsilon$

$\delta = 1$: $\sum_{j=1}^{n} a_j x_j - m + \varepsilon \geq b + \varepsilon$

$\sum_{j=1}^{n} a_j x_j - b \geq m$
Indicator variables for inequalities, example

Logical condition

\[ 2x_1 + 3x_2 \leq 1 \iff \delta = 1 \]
\[ \delta \in \{0, 1\} \]
\[ 0 \leq x_1 \leq 1, \ 0 \leq x_2 \leq 1 \]

We find

\[ M = \text{u.b.}(\sum_{j=1}^{n} a_j x_j - b) \]
\[ = \text{u.b.}(2x_1 + 3x_2 - 1) = 4 \]

and

\[ m = \text{l.b.}(\sum_{j=1}^{n} a_j x_j - b) \]
\[ = \text{l.b.}(2x_1 + 3x_2 - 1) = -1 \]

choose \( \varepsilon = 0.01 \), i.e. constraint broken when \( 2x_1 + 3x_2 \geq 1.01 \)

Constraints

\[ 2x_1 + 3x_2 + 4\delta \leq 4 + 1 \]
\[ 2x_1 + 3x_2 - (1 - 0.01)\delta \geq 1 + 0.01 \]

Which results in model:

\[
\begin{cases}
2x_1 + 3x_2 + 4\delta \leq 5 \\
2x_1 + 3x_2 + 1.01\delta \geq 1.01 \\
\delta \in \{0, 1\} \\
0 \leq x_1 \leq 1, \\
0 \leq x_2 \leq 1
\end{cases}
\]
**Nonlinear functions**

Frequently, the objective function or some of the constraints may contain nonlinear functions.

Approx. nonlinear function by piecewise linear function

- Split into $m$ intervals
- For each interval $[d_i, d_{i+1}]$
  \[
  d_i \leq x \leq d_{i+1} \iff y = a_i x + b_i
  \]
- Model as

\[
\begin{cases}
  d_i \leq x & \iff \delta_1 = 1 \\
  x \leq d_{i+1} & \iff \delta_2 = 1 \\
  \delta_1 + \delta_2 = 2 & \iff \delta = 1 \\
  y = a_i x + b_i & \iff \delta = 1 
\end{cases}
\]

- Many intervals $m$, better precision but much harder to solve!
Logical conditions and 0-1 variables

a) If no depot is sited here then it will not be possible to supply any of the customers from the depot.

b) If we manufacture product A then we must also manufacture product B or at least one of products C and D.

c) If we do not place an electronic module in this position, then no wires can be connected into this position.

Introduce an indicator variable $\delta_i \in \{0, 1\}$ with each condition $X_i$

<table>
<thead>
<tr>
<th>Condition $X_i$ is true</th>
<th>$\delta_i = 1$</th>
</tr>
</thead>
</table>

In this way we may formulate:

<table>
<thead>
<tr>
<th>$X_1 \lor X_2$</th>
<th>$\delta_1 + \delta_2 \geq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1 \land X_2$</td>
<td>$\delta_1 = 1, \delta_2 = 1$</td>
</tr>
<tr>
<td>$X_1 \Rightarrow X_2$</td>
<td>$\delta_1 - \delta_2 \leq 0$</td>
</tr>
<tr>
<td>$X_1 \iff X_2$</td>
<td>$\delta_1 - \delta_2 = 0$</td>
</tr>
</tbody>
</table>
Transformation to linear form

Write up the text in ordinary mathematical form

\[(\sin(x_1) \leq \frac{1}{2} \lor x_1 x_2 \leq x_3) \Rightarrow (x_3 = 1 \lor x_2 + x_1 \leq 1)\]

Stepwise transformation

1 Arithmetic functions are replaced by piecewise linear approximations of the functions.

2 Products of decision variables are transformed into products of binary variables. Products of binary variables may easily be expressed as logical constraints, and thus put on binary form.

3 Relations are transformed into linear inequalities with boolean variables.

\[ (ax \leq b) \iff (\delta = 1) \]

4 Boolean logics are transformed into linear form.

\[ (B_1 \lor B_2) \iff (\delta' = 1) \]

5 The resulting expression should be true $\delta_{all} = 1$

6 Domains of variables are defined.

\[ \delta_i \in \{0, 1\} \]
Transformation of general constraints to linear form

Step 3:

<table>
<thead>
<tr>
<th>Relation</th>
<th>ILP-constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Ax \leq b)</td>
<td>(Ax + (\delta - 1)M \leq b, \quad Ax + \delta M \geq b + \varepsilon)</td>
</tr>
<tr>
<td>(Ax &lt; b)</td>
<td>(Ax + (\delta - 1)M \leq b - \varepsilon, \quad Ax + \delta M \geq b)</td>
</tr>
<tr>
<td>(Ax &gt; b)</td>
<td>(Ax + (1 - \delta)M \geq b + \varepsilon, \quad Ax - \delta M \leq b)</td>
</tr>
<tr>
<td>(Ax \geq b)</td>
<td>(Ax + (1 - \delta)M \geq b, \quad Ax - \delta M \leq b - \varepsilon)</td>
</tr>
<tr>
<td>(Ax = b)</td>
<td>(Ax \geq b \land Ax \leq b)</td>
</tr>
</tbody>
</table>

Step 4:

<table>
<thead>
<tr>
<th>Relation</th>
<th>Meaning</th>
<th>ILP-constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>(B_1 \lor B_2)</td>
<td>(\delta = 1 \iff \delta_1 = 1 \lor \delta_2 = 1)</td>
<td>(\delta - \delta_1 - \delta_2 \leq 0, \quad \delta_1 + \delta_2 - 2\delta \leq 0)</td>
</tr>
<tr>
<td>(B_1 \land B_2)</td>
<td>(\delta = 1 \iff \delta_1 = 1 \land \delta_2 = 1)</td>
<td>(2\delta - \delta_1 - \delta_2 \leq 0, \quad \delta_1 + \delta_2 - \delta \leq 1)</td>
</tr>
<tr>
<td>(B_1 \rightarrow B_2)</td>
<td>(\delta = 1 \iff (\delta_1 = 1 \implies \delta_2 = 1))</td>
<td>(\delta_1 - \delta_2 + \delta \leq 1, \quad \delta_1 - \delta_2 + 2\delta \geq 1)</td>
</tr>
<tr>
<td>(B_1 \leftrightarrow B_2)</td>
<td>(\delta = 1 \iff (\delta_1 = 1 \iff \delta_2 = 1))</td>
<td>use: ((B_1 \Rightarrow B_2) \land (B_2 \Rightarrow B_1))</td>
</tr>
<tr>
<td>(-B_1)</td>
<td>(\delta = 1 \iff \neg (\delta_1 = 1))</td>
<td>(\delta = 1 - \delta_1)</td>
</tr>
</tbody>
</table>
Opencast mining (williams example 12.15)

Concentration $c_{ijk}$ of pure metal for each block

- **Level 1 (surface)**
  
  \[
  \begin{array}{cccc}
  1.5 & 1.5 & 1.5 & 0.75 \\
  1.5 & 2.0 & 1.5 & 0.75 \\
  1.0 & 1.0 & 0.75 & 0.50 \\
  1.5 & 1.5 & 1.5 & 0.25 \\
  \end{array}
  \]

- **Level 2 (25 ft depth)**
  
  \[
  \begin{array}{ccc}
  4.0 & 4.0 & 2.0 \\
  3.0 & 3.0 & 1.0 \\
  2.0 & 2.0 & 0.5 \\
  \end{array}
  \]

- **Level 3 (50 ft depth)**
  
  \[
  \begin{array}{cc}
  12.0 & 6.0 \\
  5.0 & 4.0 \\
  \end{array}
  \]

- **Level 4 (75 ft depth)**
  
  \[
  \begin{array}{c}
  6.0 \\
  \end{array}
  \]

Cost of extraction

- Level 1: $e_1 = 3,000$ pounds
- Level 2: $e_2 = 6,000$ pounds
- Level 3: $e_3 = 8,000$ pounds
- Level 4: $e_4 = 10,000$ pounds

Revenue from 100% block is $R = 200,000$ pounds
Opencast mining

Introduce variables $x_{ijk}$

- **Level 1** (surface)
  
  \[
  \begin{array}{cccc}
  x_{111} & x_{112} & x_{113} & x_{114} \\
  x_{121} & x_{122} & x_{123} & x_{124} \\
  x_{131} & x_{132} & x_{133} & x_{134} \\
  x_{141} & x_{142} & x_{143} & x_{144} \\
  \end{array}
  \]

- **Level 2** (25 ft depth)
  
  \[
  \begin{array}{ccc}
  x_{211} & x_{212} & x_{213} \\
  x_{221} & x_{222} & x_{223} \\
  x_{231} & x_{232} & x_{233} \\
  \end{array}
  \]

- **Level 3** (50 ft depth)
  
  \[
  \begin{array}{cc}
  x_{311} & x_{312} \\
  x_{321} & x_{322} \\
  \end{array}
  \]

- **Level 4** (75 ft depth)
  
  \[
  \begin{array}{c}
  x_{411} \\
  \end{array}
  \]

Maximize net profit

\[
\sum (R \cdot c_{ijk} - e_i)x_{ijk}
\]

Constraints

\[
\delta_{ijk} = 1 \Rightarrow \begin{cases} 
\delta_{i-1,j,k} = 1 \\
\delta_{i-1,j-1,k} = 1 \\
\delta_{i-1,j,k-1} = 1 \\
\delta_{i-1,j-1,k-1} = 1 
\end{cases}
\]

All variables $x_{ijk} \in \{0, 1\}$. 
Opencast mining

The constraints

\[ \delta_{i,j,k} = 1 \Rightarrow \begin{cases} 
\delta_{i-1,j,k} = 1 \\
\delta_{i-1,j-1,k} = 1 \\
\delta_{i-1,j,k-1} = 1 \\
\delta_{i-1,j-1,k-1} = 1 
\end{cases} \]

can be expressed as

\[
\begin{align*}
\delta_{i,j,k} - \delta_{i-1,j,k} & \leq 0 \\
\delta_{i,j,k} - \delta_{i-1,j-1,k} & \leq 0 \\
\delta_{i,j,k} - \delta_{i-1,j,k-1} & \leq 0 \\
\delta_{i,j,k} - \delta_{i-1,j-1,k-1} & \leq 0 
\end{align*}
\]

Model with 30 binary variables, 56 constraints
Brute-force solution: \(2^{30}\) steps
Opencast mining

\[
\begin{pmatrix}
-1 & -1 & 1 \\
-1 & -1 & 1 \\
-1 & -1 & 1 \\
-1 & -1 & 1 \\
-1 & -1 & 1 \\
-1 & -1 & 1
\end{pmatrix}
\]

The constraint matrix $A$ is totally unimodular (TU)

Solving

\[
\begin{align*}
\max & \quad c\delta \\
\text{s.t.} & \quad A\delta \leq b \\
& \quad \delta \geq 0
\end{align*}
\]

gives integer solutions for any integer vector $b$ and any $c$
Opencast mining

The optimal solution

- Level 1 (surface)

- Level 2 (25 ft depth)

- Level 3 (50 ft depth)

- Level 4 (75 ft depth)

Net profit is 17.500 pounds.