INTEGER PROGRAMMING

• Pure Integer Programs: all decision variables restricted to integer values
  ⇒ 0-1 programs
  ⇒ general IP problems

• Mixed Integer Programs: some variables restricted to integer values; others may be continuous
  ⇒ 0-1 programs
  ⇒ general IP problems

Very versatile because many “real-world” applications involve variables that are naturally integer valued.

Also, many combinatorial optimization problems (where an optimum combination out of a possible set of combinations must be determined) can be cast as IP’s.
Set Covering/Partitioning/Packing

- $\Gamma=\{A,B,C,D,E,F,G,H,I,J,K\}$

<table>
<thead>
<tr>
<th>$B_1=(A,B,C,D)$</th>
<th>$B_6=(A,C,J,K)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_2=(B,C,F,G)$</td>
<td>$B_7=(D,G)$</td>
</tr>
<tr>
<td>$B_3=(A,H)$</td>
<td>$B_8=(A,H,J,K)$</td>
</tr>
<tr>
<td>$B_4=(B,D,E,I,J)$</td>
<td>$B_9=(E,F,I)$</td>
</tr>
<tr>
<td>$B_5=(D,E,F,K)$</td>
<td>$B_{10}=(G,H,J,K)$</td>
</tr>
</tbody>
</table>

- Set Representation: $(A,G,I,K); (A,E,G); (A,B,D,E,K)$

- Set Covering: $(B_2,B_4,B_8); (B_1,B_2,B_8,B_9); (B_1,B_3,B_5,B_7,B_8,B_9)$

- Set Partitioning: $(B_1,B_9,B_{10})$

- Set Packing: $(B_6,B_7,B_9); (B_1,B_9,B_{10})$

- $F_A=(B_1,B_3,B_6,B_8); F_B=(B_1,B_2,B_4); F_C=(B_1,B_2,B_6); \ldots; F_H=(B_3,B_8,B_{10}); \ldots; F_K=(B_5,B_6,B_8,B_{10});$
<table>
<thead>
<tr>
<th>Application</th>
<th>$\mathcal{I}$</th>
<th>$\mathcal{B}$</th>
<th>$F_i$</th>
<th>$X_j=1$ if...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivery &amp; Routing</td>
<td>Set of delivery locations</td>
<td>Set of routes, each covering some subset of locations</td>
<td>Routes that contain location $i$</td>
<td>...route $j$ is chosen</td>
</tr>
<tr>
<td>Facility Location</td>
<td>Set of areas that require service</td>
<td>Set of locations, each servicing some subset of areas</td>
<td>Locations that service area $i$</td>
<td>...location $j$ is chosen</td>
</tr>
<tr>
<td>Fire Hydrant Location</td>
<td>Set of street blocks</td>
<td>Set of locations, each servicing some subset of street blocks</td>
<td>Locations that service block $i$</td>
<td>...location $j$ is chosen</td>
</tr>
<tr>
<td>Sales force Assignment</td>
<td>Sales areas to be covered</td>
<td>Set of possible assignments, each one covering some subset of areas</td>
<td>Assignments that cover area $i$</td>
<td>...assignment $j$ is chosen</td>
</tr>
<tr>
<td>Crew Scheduling</td>
<td>Set of flight legs or segments</td>
<td>Set of pairings of different sequential flight legs</td>
<td>Pairings that cover flight segment $i$</td>
<td>...pairing $j$ is chosen</td>
</tr>
<tr>
<td>Committee Selection</td>
<td>Set of desired committee characteristics</td>
<td>Set of groups of people (perhaps based on feasibility or availability...)</td>
<td>Groups that contain characteristic $i$</td>
<td>...group $j$ is chosen</td>
</tr>
</tbody>
</table>
Service Districts and Candidate Locations for EMS
FIGURE: Flight Schedule for AA Example

TABLE: Possible Pairings for AA Example

<table>
<thead>
<tr>
<th>j</th>
<th>Flights Sequence</th>
<th>Cost</th>
<th>j</th>
<th>Flights Sequence</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>101-203-406-308</td>
<td>2900</td>
<td>9</td>
<td>305-407-109-212</td>
<td>2600</td>
</tr>
<tr>
<td>2</td>
<td>101-203-407</td>
<td>2700</td>
<td>10</td>
<td>308-109-212</td>
<td>2050</td>
</tr>
<tr>
<td>3</td>
<td>101-204-305-407</td>
<td>2600</td>
<td>11</td>
<td>402-204-305</td>
<td>2400</td>
</tr>
<tr>
<td>4</td>
<td>101-204-308</td>
<td>3000</td>
<td>12</td>
<td>402-204-310-211</td>
<td>3600</td>
</tr>
<tr>
<td>5</td>
<td>203-406-310</td>
<td>2600</td>
<td>13</td>
<td>406-308-109-211</td>
<td>2550</td>
</tr>
<tr>
<td>6</td>
<td>203-407-109</td>
<td>3150</td>
<td>14</td>
<td>406-310-211</td>
<td>2650</td>
</tr>
<tr>
<td>7</td>
<td>204-305-407-109</td>
<td>2550</td>
<td>15</td>
<td>407-109-211</td>
<td>2350</td>
</tr>
<tr>
<td>8</td>
<td>204-308-109</td>
<td>2500</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**ROUNDING OFF**

Why not solve the Integer LP as a regular LP and “round off intelligently?”

• Not an illogical approach when all variables must be general integers. May turn out to be a practical approach, especially if all the integer variables are expected to be “large” at the optimum and feasibility is easy to maintain when rounding off

However...

• Usually doesn’t make sense with 0-1 programs
• Many times all “rounded-off” solutions may be infeasible:

\[ \hat{x} \]

 feasable region

© 2014, Jayant Rajgopal
• In other cases, the rounded solution may be quite far away from the integer optimum:

LP opt. = \( x^* \), “rounded-off” to \( x^i \), IP opt. = \( x^2 \) !!
Maximize \( 120x_1 + 96x_2 \)

subject to \( 6x_1 + 13x_2 \leq 67 \)
\( 8x_1 + 5x_2 \leq 55 \)

and \( x_1, x_2 \geq 0 \)
\( x_1, x_2 \) integer valued

© 2014, Jayant Rajgopal
Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_1, x_2 \geq 0$ & integer valued

$X_2 \geq 3$

1A

Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_2 \geq 3$
$x_1, x_2 \geq 0$ & integer valued

$X_2 \leq 2$

1B

Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_2 \leq 2$
$x_1, x_2 \geq 0$ & integer valued

$X_2 \geq 6$

2A

Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_1 \geq 6$
$x_2 \geq 2$
$x_1, x_2 \geq 0$ & integer valued

$X_2 \leq 1$

2B

Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_2 \leq 2$
$x_1 \geq 6$
$x_1, x_2 \geq 0$ & integer valued

$X_2 \geq 2$

3A

Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_1 \geq 6$ (Corrected from $x_2 \geq 6$)
$x_2 \geq 2$
$x_1, x_2 \geq 0$ & integer valued

$X_2 \leq 1$

3B

Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_2 \leq 2$
$x_1 \geq 6$ (Corrected from $x_2 \geq 2$)
$x_1, x_2 \geq 0$ & integer valued

$X_1 \geq 5$

4A

Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_2 \geq 3$
$x_1 \geq 5$
$x_1, x_2 \geq 0$ & integer valued

$X_1 \leq 4$

4B

Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_2 \geq 3$
$x_1 \leq 4$
$x_1, x_2 \geq 0$ & integer valued

$X_1 \geq 7$

5A

Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_2 \leq 2$ (Corrected from $x_2 \geq 2$)
$x_1 \geq 6$
$x_2 \leq 1$
$x_1 \geq 7$
$x_1, x_2 \geq 0$ & integer valued

$X_1 \leq 6$

5B

Maximize $120x_1 + 96x_2$

st $6x_1 + 13x_2 \leq 67$
$8x_1 + 5x_2 \leq 55$
$x_2 \leq 2$ (Corrected from $x_2 \geq 2$)
$x_1 \geq 6$
$x_2 \leq 1$
$x_1 \leq 6$
$x_1, x_2 \geq 0$ & integer valued

© 2014, Jayant Rajgopal
Iteration 0
Incumbent: None
$Z_{lower} = -\infty$
$Z_{Upper} = 883.459$

Iteration 1
Incumbent: None
$Z_{lower} = -\infty$
Max $Z_{Upper}$ at leaves = 867

Iteration 2
Incumbent: $(x_1, x_2) = (5, 2)$
$Z_{lower} = 792$
Max $Z_{Upper}$ at leaves = 854.4
Max % error = $(854.4 - 792) / 854.4 = 7.30\%$

Iteration 3
Incumbent: $(x_1, x_2) = (5, 2)$
$Z_{lower} = 792$
Max $Z_{Upper}$ at leaves = 848
Max % error = $(848 - 792) / 848 = 6.60\%$

Iteration 4
Incumbent: $(x_1, x_2) = (5, 2)$
$Z_{lower} = 792$
Max $Z_{Upper}$ at leaves = 846
Max % error = $(846 - 792) / 846 = 6.38\%$

Iteration 5
Incumbent: $(x_1, x_2) = (6, 1)$
$Z_{lower} = 816$
Max $Z_{Upper}$ at leaves = 816
Max % error = $(816 - 816) / 816 = 0\%$

© 2014, Jayant Rajgopal
Maximize $120x_1 + 96x_2$

Subject to:

$6x_1 + 13x_2 \leq 67$

$8x_1 + 5x_2 \leq 55$

$x_1, x_2 \geq 0$ and integer valued

$Z = 883.459$
\[ x_2 \geq 3 \]
\[ x_2 \leq 2 \]
\[ x_1 \leq 5 \]
\[ x_1 \geq 6 \]
\[ z = 848 \]
\[ z = 792 \]
\[ z = 846 \]