The Resource Allocation Problem

The Linear Programming Problem

Plants:

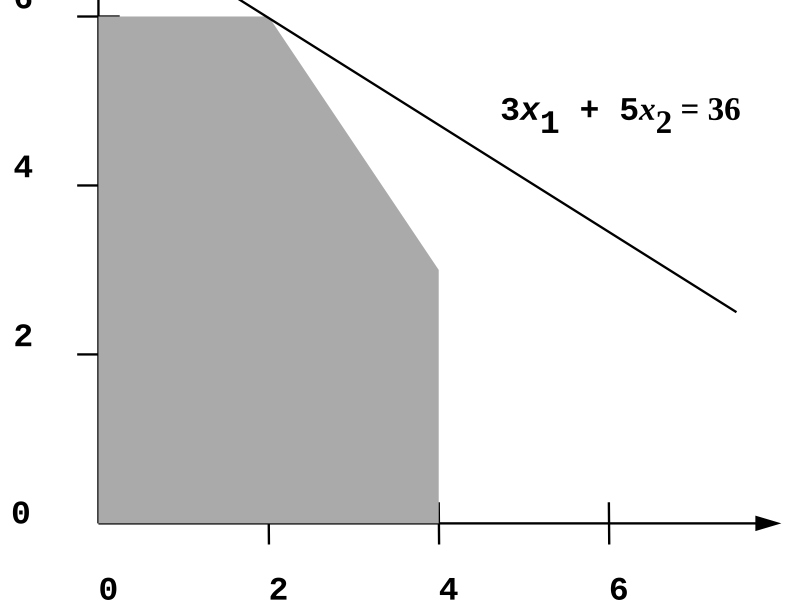
- Plant 1: Aluminum frame production.
- Plant 2: Wood frame production.
- Plant 3: Glass production and assembly.

Production is in batches of 200 units.

Data:

	Hrs/batch		Hrs avail
	Door	Window	
Plant 1	1	0	4
Plant 2	0	2	12
Plant 3	3	2	18
Profit/batch	\$3,000	\$5,000	

$$
\begin{array}{rcrcr}
 & & 2x_2 & \leq & 12 \\
3x_1 & + & 2x_2 & \leq & 18 \\
 & & x_1,\ x_2 & \geq & 0 \quad .
\end{array}
$$



6
4
2
0
$3x_1 + 5x_2 = 36$
0
2
4
6

```
maximize profit: 3 * Batches1 + 5 * Batches2;

subject to P1_Hrs_Avail:   Batches1 <= 4;
subject to P2_Hrs_Avail: 2*Batches2 <= 12;
subject to P3_Hrs_Avail: 3*Batches1 + 2*Batches2 <= 18;

solve;

display Batches1, Batches2, profit;
```

Output:

```
MINOS 5.4: optimal solution found.
2 iterations, objective 36
Batches1 = 2
Batches2 = 6
profit = 36
```

```
param avail  {Resources};
param unit_profit {Activities};
param usage {Resources, Activities};

var amt {Activities} >= 0;

maximize profit:
    sum {j in Activities} unit_profit[j] * amt[j];

subject to capacity {i in Resources}:
    sum {j in Activities} usage[i,j] * amt[j]
                                    <= avail[i];
```

```
param avail :=
    Plant1    4
    Plant2   12
    Plant3   18
    ;

param unit_profit :=
    Door      3
    Window    5
    ;

param usage: Prod1 Prod2 :=
    Plant1     1     0
    Plant2     0     2
    Plant3     3     2
    ;
```

```
ampl: solve;
MINOS 5.4: optimal solution found.
2 iterations, objective 36

ampl: display amt, profit;
amt [*] :=
Prod1  2
Prod2  6
;

profit = 36

ampl: quit
```

$$
\begin{array}{rcrcrcl}
a_{21}x_1 & + & a_{22}x_2 & +\cdots+ & a_{2n}x_n & \leq & b_2 \\
 & & & \vdots & & & \\
a_{m1}x_1 & + & a_{m2}x_2 & +\cdots+ & a_{mn}x_n & \leq & b_m \\
 & & x_1, & x_2, & \ldots\ x_n & \geq & 0.
\end{array}
$$

Much of the course will be devoted to study of such algorithms.

Most real-world problems don’t quite fit the LP paradigm.

They have nonlinearities in either the objective function, the constraints, or both.

Later in the course we will extend the algorithms we develop for LP to some of these more general problems.