IV. Violations of Linear Programming Assumptions

Some types of Mathematical Programming problems violate at least one condition of strict Linearity

- **Deterministic Nature**
- **Additivity**
- **Direct Proportionality**
- **Fractionality**

Or do not satisfy the additional conditions of:

- **Nonnegativity**
- **Single goal**

We now look at how to handle such situations.

Violations of:	Are Handled With:
Deterministic Nature	**Sensitivity Analysis, Parametric Programming (PP), Stochastic Programming (SP) Statistical Programming(STP)**
Additivity	**Revise/Correct Formulation**
Direct Proportionality	**Nonlinear Programming (NLP)**
Fractionality	**Integer Programming (IP)**
Nonnegativity	**Creative Decision Variables**
Single Goal	**Goal Programming (GP)**

A. Integer Programming

- Integer Programming (IP): Branch of linear programming in which at least one decision variable is restricted to integer values.
- Mixed Integer Program (MIP): - A linear program in which the values of some decision variables are limited to integer values and others are allowed to assume fractional values.
- 0/1 Integer Program (Binary Program or BIP) - A linear program in which the values of decision variables are limited to binary values (zero or one). Often used for problems that involve Yes/No types of decisions.

The Television Production Problem is an excellent example of a scenario for which the decision variables should be restricted to integer values - we were fortunate that this happened naturally in that problem!

Consider the following example.

Riggs Paint & Supply has available 3 different processes for producing standard white house paint. Each process has a capacity and a per gallon processing cost given below:

Process	Processing Cost/gal.	Maximum Daily Capacity (Gals.)
1	$5	2000
2	$4	3000
3	$3	4000

Daily demand is 3500 gals. What processes should be used? How many gallons should be made by each process?

If we allow non integer quantities of paint to be produced by each process (certainly reasonable since all processes are making the same paint), then an appropriate formulation would be:

minimize $C= 5x_1 + 4x_2 + 3x_3$

subject to:
$$\begin{array}{llll} x_1 + & x_2 + & x_3 & = 3500 \text{ (daily demand)} \\ x_1 & & & \leq 2000 \text{ (process 1 capacity)} \\ & x_2 & & \leq 3000 \text{ (process 2 capacity)} \\ & & x_3 & \leq 4000 \text{ (process 3 capacity)} \\ x_1, & x_2, & x_3 & \geq \quad 0 \text{ (nonnegativity)} \end{array}$$

Where x_1 is the number of gallons produced using process 1
x_2 is the number of gallons produced using process 2
x_3 is the number of gallons produced using process 3

What if, on the other hand, each process produced a different quality of paint?

If each process produced a different quality of paint, then we might wish to restrict production by each process to whole cans (gallons). An appropriate formulation would be:

minimize $C= 5x_1 + 4x_2 + 3x_3$

subject to:
$$\begin{array}{llll} x_1 + & x_2 + & x_3 & = 3500 \text{ (daily demand)} \\ x_1 & & & \leq 2000 \text{ (process 1 capacity)} \\ & x_2 & & \leq 3000 \text{ (process 2 capacity)} \\ & & x_3 & \leq 4000 \text{ (process 3 capacity)} \\ x_1, & x_2, & x_3 & \geq \quad 0 \text{ (nonnegativity)} \\ x_1, & x_2, & x_3 & \text{are integer} \end{array}$$

Where x_1 is the number of gallons produced using process 1
x_2 is the number of gallons produced using process 2
x_3 is the number of gallons produced using process 3

This is an Integer Programming formulation.

Now consider a fixed cost for setting up each of the processes. If the setup costs are $100 for process 1, $200 for process 2, and $300 for process 3, an appropriate formulation would be:

minimize $C = 5x_1 + 4x_2 + 3x_3 + 100y_1 + 200y_2 + 300y_3$

subject to: $x_1 + x_2 + x_3 = 3500$ **(daily demand)**
$x_1 \leq 2000y_1$ **(process 1 capacity)**
$x_2 \leq 3000y_2$ **(process 2 capacity)**
$x_3 \leq 4000y_3$ **(process 3 capacity)**
$x_1, x_2, x_3 \geq 0$ **(nonnegativity)**
x_1, x_2, x_3 **are integer**
y_1, y_2, y_3 **are binary**

Where x_i **is the number of gallons produced using process i**
y_i **is 1 if process i is used and 0 otherwise**

This is a Binary Integer Programming formulation.

- **Solving Integer Programming problems graphically**

LP Relaxation: the version of an integer programming problem that results when the integer restrictions are ignored (relaxed).

IP problems with two decision variables can be solved graphically. Consider the following LP problem:

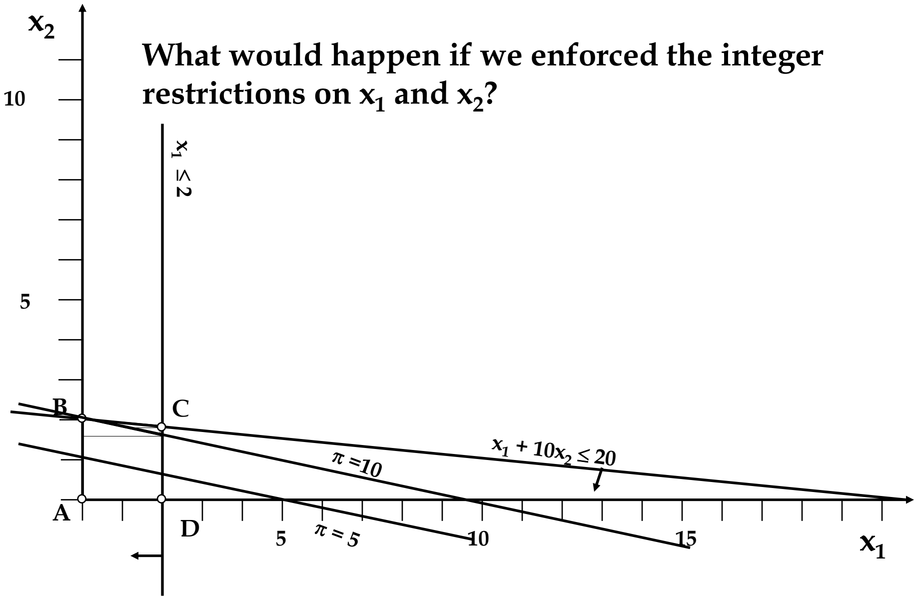
maximize $\pi = x_1 + 5x_2$

subject to: $x_1 + 10x_2 \leq 20$ **(constraint 1)**
$x_1 \leq 2$ **(constraint 2)**
$x_1, x_2 \geq 0$ **(nonnegativity)**
x_1, x_2 **are integer**

Graphically, the Linear Programming relaxation problem looks like this:

The optimal solution is at Extreme Value C (x_1 = 2.0, x_2= 1.8, π=11.0).

What would happen if we enforced the integer restrictions on x_1 and x_2?



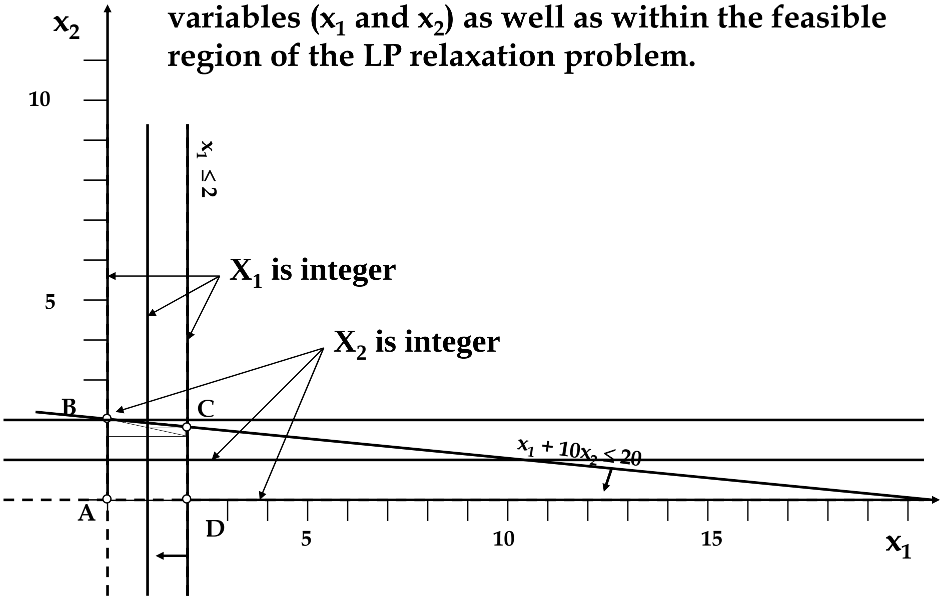
Why don't we simply round the LP optimal values of the decision variables (x_1 =2.0, x_2= 1.8)?

Rounding up yields x_1 = 2, x_2= 2, π=12 - this is a better value of the objective function, but is unfortunately infeasible.

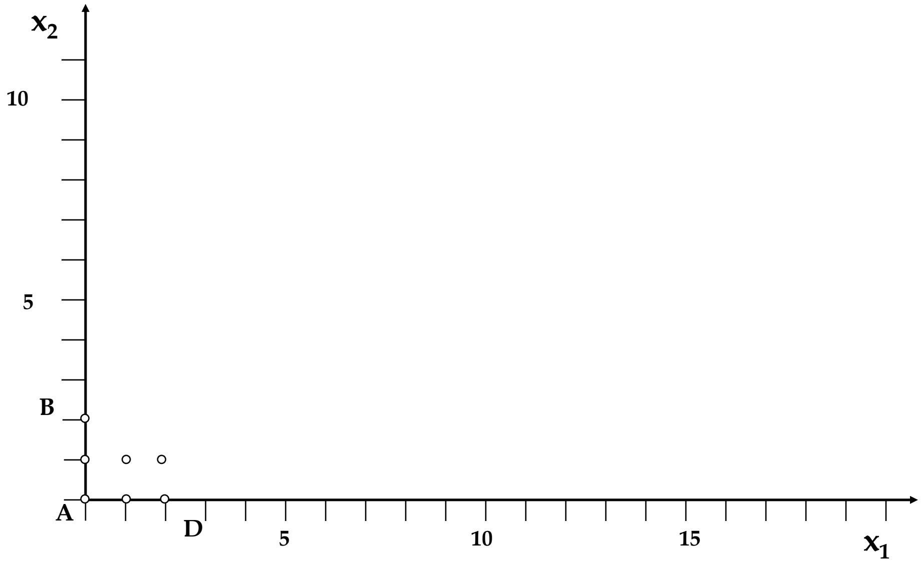
Rounding down yields x_1 = 2, x_2= 1, π=7 - this is feasible but is unfortunately infeasible a much worse value of the objective function (can we find a better feasible solution?).

What changes do the integer restrictions graphically imply for our Linear Programming relaxation problem?

The feasible region now must lie on the lines that represent the integer restrictions on the decision variables (x_1 and x_2) as well as within the feasible region of the LP relaxation problem.

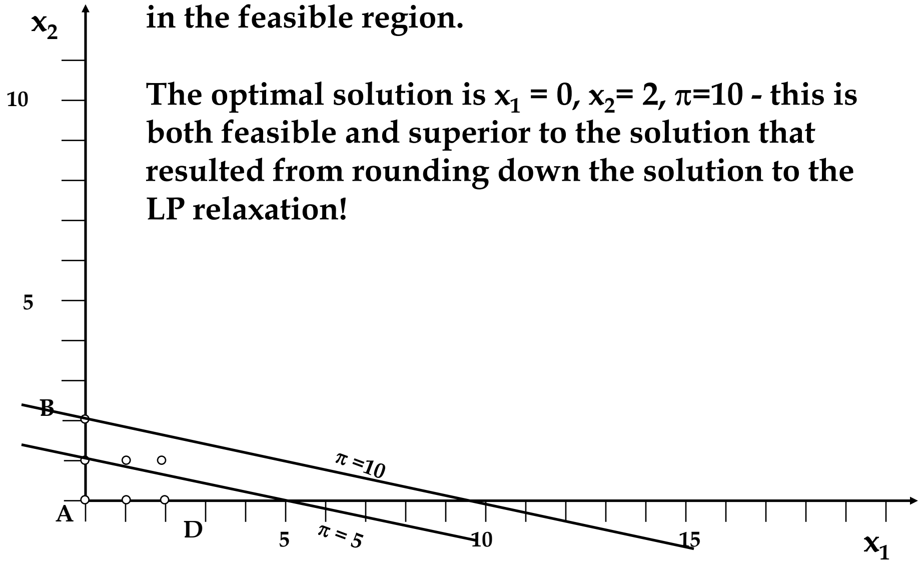


The feasible region is now a series of points (that were formed by the feasible *lattice*).

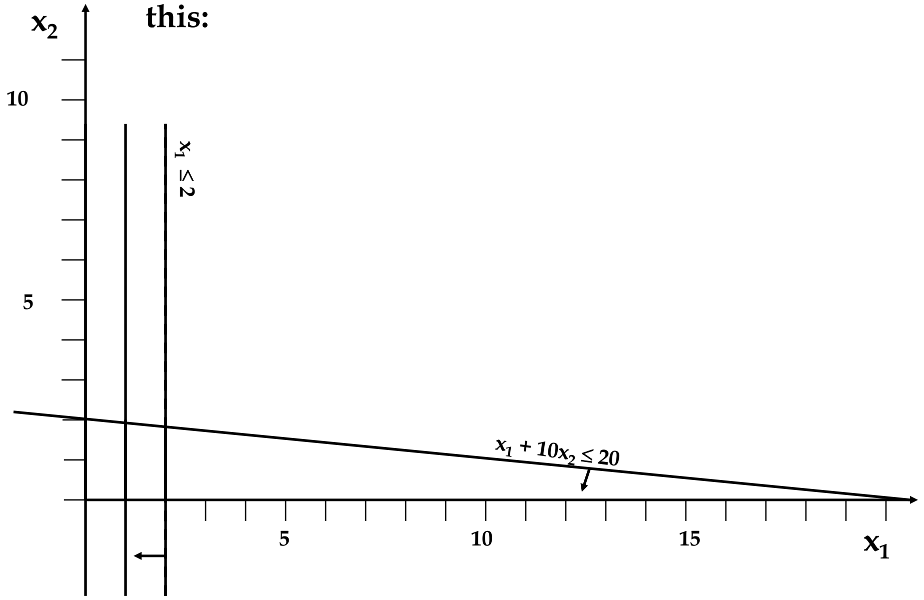


We now move the objective function in the direction of improvement until we hit the last point in the feasible region.

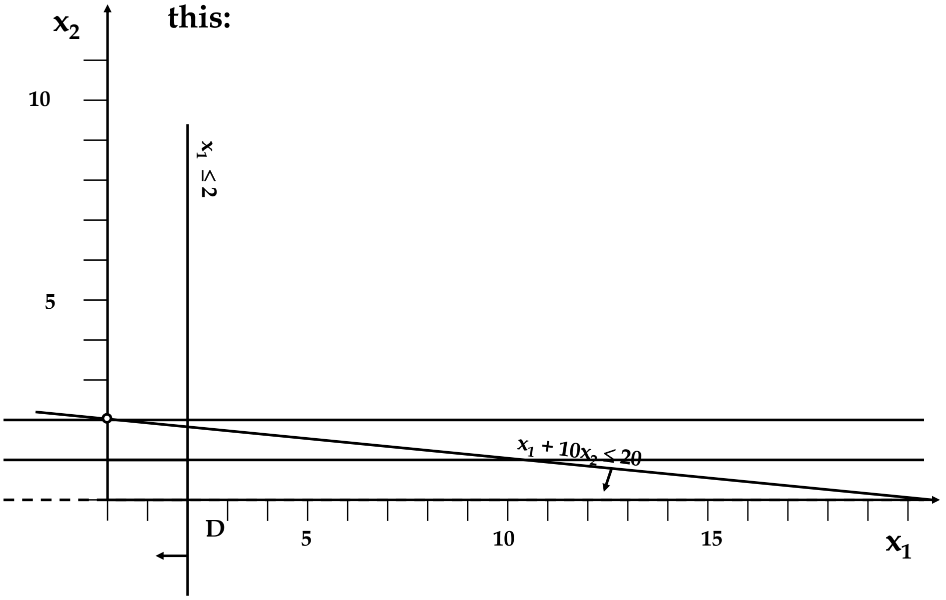
The optimal solution is $x_1 = 0$, $x_2 = 2$, $\pi=10$ - this is both feasible and superior to the solution that resulted from rounding down the solution to the LP relaxation!



If only x_1 was restricted to integer values, the the feasible region for this IP problem would look like this:



If only x_2 was restricted to integer values, the the feasible region for this IP problem would look like this:



- **Solving Integer Programming problems algorithmically**

Cutting Plane Algorithm - Set of methods for solving IP problems for which successive constraints (cuts) are added to the LP relaxation to force the extreme points of the feasible region to be at integer points.

Branch & Bound Method - A cutting plane approach to solving IP problems through systematic partial enumeration. Basically:

- **A Parent Problem (the LP relaxation with some integrality constraints enforced) is solved**
- **Then additional constraints (which cut away parts of the feasible region that violate integrality constraints) are added to form varying Descendent Linear Programs which are then solved.**
- **These become Parent Problems, and the cycle is repeated until the optimal solution to the IP is found.**

The steps of the Branch & Bound Method are:

1. Solve the Parent Problem (the LP relaxation of the original Integer Programming problem in the first iteration of this algorithm). This solution provides:

    the current upper bound for its direct Descendents in maximization problems.

    the current lower bound for its direct Descendents in minimization problems.

2. Check the status of all decision variables that have integrality constraints in the original IP problem:

    If all integrality constraints are satisfied, the solution is feasible. If this is the best current integer solution, retain as the lower bound (for maximizations) or as the upper bound (for minimizations). Do not proceed down this branch.

    Proceed if any integrality constraint is violated.

---

3. Arbitrarily select a decision variable that has an integrality constraint in the original IP problem that is unmet by the current solution. Create two Descendent Problems by *branching*:

    Round up the current value of the chosen variable to the nearest integer and add a constraint that forces the chosen variable to be at least as great (≥) as the result.

    Round down the current value of the chosen variable to the nearest integer and add a constraint that forces the chosen variable to be no more than (≤) as the result.

4. **For each descendent problem**

   If the resulting solution does not satisfy all integrality constraints from the original IP but is superior to the lower bound (for maximizations) or the upper bound (for minimizations), continue down this path (i.e., use this as a future Parent Solution).

   If the solution satisfies all integrality constraints from the original IP, do not continue down this path. If this is the best current integer solution, retain this solution as the lower bound (for maximization problems) or as the upper bound (for minimization problems).

   If the resulting solution is not superior to the lower bound (for maximizations) or the upper bound (for minimizations), do not continue down this path.

5. **Continue until all branches result in either**

   i) **a solution that is inferior to the lower bound (for maximizations) or the upper bound (for minimizations) or**

   ii) **a solution that satisfies all integrality constraints from the original IP.**

   At this point the remaining lower bound (for maximizations) or upper bound (for minimizations) is the optimal solution.

Example - Recall the IP problem we solved graphically:

maximize $\pi = x_1 + 5x_2$

subject to:
$x_1 + 10x_2 \leq 20$ **(constraint 1)**
$x_1 \leq 2$ **(constraint 2)**
$x_1, \; x_2 \geq 0$ **(nonnegativity)**
$x_1, \; x_2$ **are integer**

To solve this problem using the Branch & Bound Method, we must first solve the LP Relaxation:

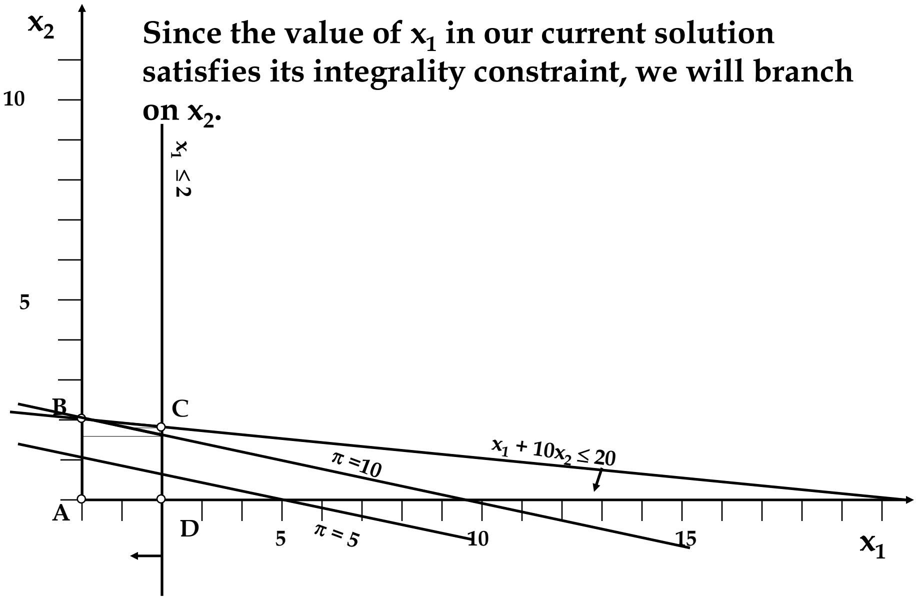
maximize $\pi = x_1 + 5x_2$

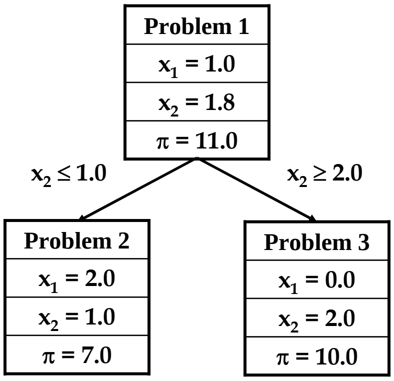
subject to:
$x_1 + 10x_2 \leq 20$ **(constraint 1)**
$x_1 \leq 2$ **(constraint 2)**
$x_1, \; x_2 \geq 0$ **(nonnegativity)**

The optimal solution is at Extreme Value C ($x_1 = 1.0$, $x_2 = 1.8$, $\pi = 11.0$).

Now we must systematically enforced the integer restrictions on x_1 and x_2.

Since the value of x_1 in our current solution satisfies its integrality constraint, we will branch on x_2.





Problem 1
$x_1 = 1.0$
$x_2 = 1.8$
$\pi = 11.0$

$x_2 \leq 1.0$

$x_2 \geq 2.0$

Problem 2
$x_1 = 2.0$
$x_2 = 1.0$
$\pi = 7.0$

Problem 3
$x_1 = 0.0$
$x_2 = 2.0$
$\pi = 10.0$

The formulation for descendent problem 2 is:

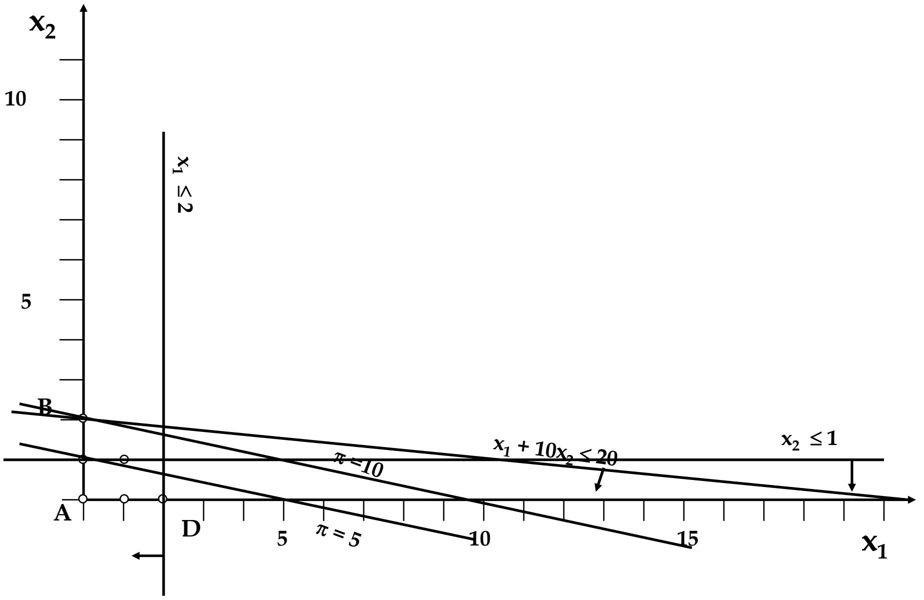
maximize $\pi = x_1 + 5x_2$

subject to:
$x_1 + 10x_2 \leq 20$ **(constraint 1)**
$x_1 \leq 2$ **(constraint 2)**
$x_2 \leq 1$ **(cutting plane constraint)**
$x_1, x_2 \geq 0$ **(nonnegativity)**

The optimal solution is at $x_1 = 2.0$, $x_2 = 1.0$, $\pi = 7.0$. This satisfies all integrality constraints (and so is feasible), so we will no longer branch in this direction.

This is also the best current solution that satisfies all integrality constraints, so it constitutes the current lower bound for our maximization problem.

The Cutting Plane for Problem 2 looks like this:



The formulation for descendent problem 3 is:

maximize $\pi = x_1 + 5x_2$

subject to:

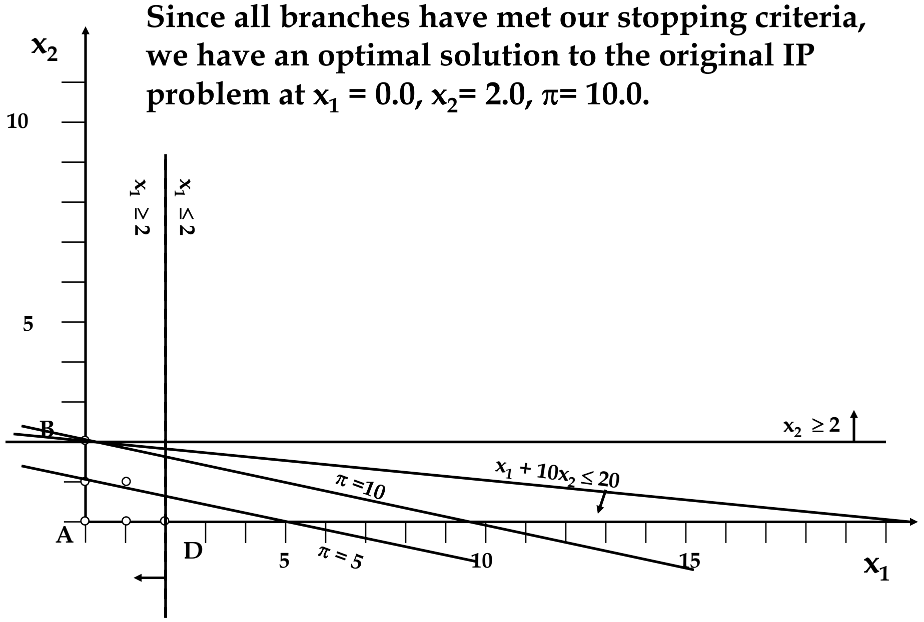
$$
\begin{aligned}
x_1 + 10x_2 &\le 20 \text{ (constraint 1)}\\
x_1 \quad &\le 2 \text{ (constraint 2)}\\
x_2 &\ge 2 \text{ (cutting plane constraint)}\\
x_1,\ x_2 &\ge 0 \text{ (nonnegativity)}
\end{aligned}
$$

The optimal solution is at $x_1 = 0.0$, $x_2 = 2.0$, $\pi = 10.0$. This satisfies all integrality constraints (and so is feasible), so we will no longer branch in this direction.

This is also now the best current solution that satisfies all integrality constraints, so it supercedes our previous lower bound and constitutes the current lower bound for our maximization problem.

The Cutting Plane for Problem 3 looks like this:

Since all branches have met our stopping criteria, we have an optimal solution to the original IP problem at $x_1 = 0.0$, $x_2 = 2.0$, $\pi = 10.0$.



Another example - consider this IP problem:

$$
\begin{array}{llcl}
\text{maximize } \pi = & x_1 + 5x_2 + 7x_3 + 3x_4 & & \\
\text{subject to:} & 7x_1 + 3x_2 + 2x_3 + 4x_4 & \leq 15 & \text{(constraint 1)} \\
 & 8x_1 + 2x_2 + 3x_3 + 5x_4 & \leq 17 & \text{(constraint 2)} \\
 & x_1 & \leq 4 & \text{(constraint 3)} \\
 & x_2 & \leq 4 & \text{(constraint 4)} \\
 & x_3 & \leq 1 & \text{(constraint 5)} \\
 & x_4 & \leq 1 & \text{(constraint 6)} \\
 & x_1, x_2, x_3, x_4 & \geq 0 & \text{(nonnegativity)} \\
 & x_1, x_2, x_3, x_4 & \in Z & \text{(integrality)}
\end{array}
$$

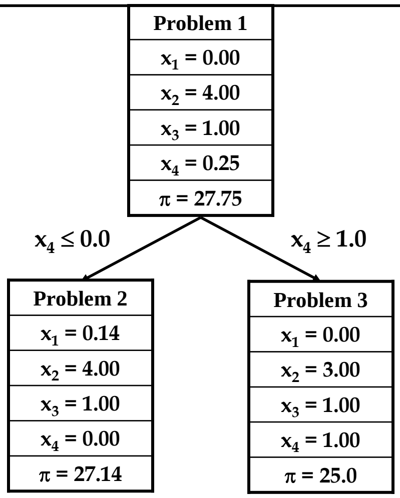
To solve this problem using the Branch & Bound Method, we must first solve the LP Relaxation:

$$
\begin{array}{llcl}
\textbf{maximize } \pi = & x_1 + 5x_2 + 7x_3 + 3x_4 & & \\
\textbf{subject to:} & 7x_1 + 3x_2 + 2x_3 + 4x_4 & \leq 15 & \text{(constraint 1)} \\
 & 8x_1 + 2x_2 + 3x_3 + 5x_4 & \leq 17 & \text{(constraint 2)} \\
 & x_1 & \leq 4 & \text{(constraint 3)} \\
 & x_2 & \leq 4 & \text{(constraint 4)} \\
 & x_3 & \leq 1 & \text{(constraint 5)} \\
 & x_4 & \leq 1 & \text{(constraint 6)} \\
 & x_1, x_2, x_3, x_4 & \geq 0 & \text{(nonnegativity)}
\end{array}
$$

The optimal solution is $x_1 = 0.00$, $x_2 = 4.00$, $x_3 = 1.00$, $x_4 = 0.25$, $\pi = 27.75$.

Now we must systematically enforced the integer restrictions on x_1, x_2, x_3, and x_4.

Since the values of x_1, x_2, and x_3 in our current solution satisfy their integrality constraints, we will branch on x_4.



Problem 1

$x_1 = 0.00$

$x_2 = 4.00$

$x_3 = 1.00$

$x_4 = 0.25$

$\pi = 27.75$

$x_4 \leq 0.0$

$x_4 \geq 1.0$

Problem 2

$x_1 = 0.14$

$x_2 = 4.00$

$x_3 = 1.00$

$x_4 = 0.00$

$\pi = 27.14$

Problem 3

$x_1 = 0.00$

$x_2 = 3.00$

$x_3 = 1.00$

$x_4 = 1.00$

$\pi = 25.0$

The formulation for descendent problem 2 is:

$$\begin{array}{lrrrrll}
\text{maximize } \pi = & x_1 + & 5x_2 + & 7x_3 + & 3x_4 & & \\
\text{subject to:} & 7x_1 + & 3x_2 + & 2x_3 + & 4x_4 & \le 15 & \text{(constraint 1)} \\
 & 8x_1 + & 2x_2 + & 3x_3 + & 5x_4 & \le 17 & \text{(constraint 2)} \\
 & x_1 & & & & \le 4 & \text{(constraint 3)} \\
 & & x_2 & & & \le 4 & \text{(constraint 4)} \\
 & & & x_3 & & \le 1 & \text{(constraint 5)} \\
 & & & & x_4 & \le 1 & \text{(constraint 6)} \\
 & & & & x_4 & \le 0 & \text{(cutting plane)} \\
 & x_1, & x_2, & x_3, & x_4 & \ge 0 & \text{(nonnegativity)}
\end{array}$$

The optimal solution is $x_1 = 0.14$, $x_2 = 4.00$, $x_3 = 1.00$, $x_4 = 0.00$, $\pi = 27.14$. This does not satisfy all integrality constraints, so we will continue to branch in this direction.

The formulation for descendent problem 3 is:

$$\begin{array}{lrrrrll}
\text{maximize } \pi = & x_1 + & 5x_2 + & 7x_3 + & 3x_4 & & \\
\text{subject to:} & 7x_1 + & 3x_2 + & 2x_3 + & 4x_4 & \le 15 & \text{(constraint 1)} \\
 & 8x_1 + & 2x_2 + & 3x_3 + & 5x_4 & \le 17 & \text{(constraint 2)} \\
 & x_1 & & & & \le 4 & \text{(constraint 3)} \\
 & & x_2 & & & \le 4 & \text{(constraint 4)} \\
 & & & x_3 & & \le 1 & \text{(constraint 5)} \\
 & & & & x_4 & \le 1 & \text{(constraint 6)} \\
 & & & & x_4 & \ge 1 & \text{(cutting plane)} \\
 & x_1, & x_2, & x_3, & x_4 & \ge 1 & \text{(nonnegativity)}
\end{array}$$

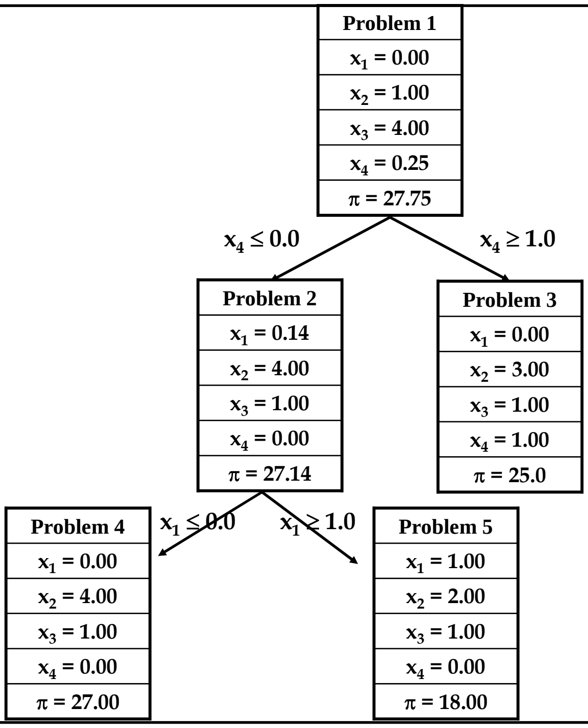
The optimal solution is $x_1 = 0$, $x_2 = 3$, $x_3 = 1$, $x_4 = 1$, $\pi = 25$. This does satisfy all integrality constraints, so we will not branch in this direction any further.

This is also the best current solution that satisfies all integrality constraints, so it constitutes the current lower bound for our maximization problem.

We continue by branching on descendent problem 2:

$$\begin{array}{llllll}
\text{maximize } \pi = & x_1 + & 5x_2 + & 7x_3 + & 3x_4 & \\
\text{subject to:} & 7x_1 + & 3x_2 + & 2x_3 + & 4x_4 \le 15 & \text{(constraint 1)} \\
& 8x_1 + & 2x_2 + & 3x_3 + & 5x_4 \le 17 & \text{(constraint 2)} \\
& x_1 & & & \le 4 & \text{(constraint 3)} \\
& & x_2 & & \le 4 & \text{(constraint 4)} \\
& & & x_3 & \le 1 & \text{(constraint 5)} \\
& & & & x_4 \le 1 & \text{(constraint 6)} \\
& & & & x_4 \le 0 & \text{(cutting plane)} \\
& x_1, & x_2, & x_3, & x_4 \ge 0 & \text{(nonnegativity)}
\end{array}$$

The optimal solution here is $x_1 = 0.14$, $x_2 = 4.00$, $x_3 = 1.00$, $x_4 = 0.00$, $\pi = 27.14$, so we now branch on x_1.



The formulation for descendent problem 4 is:

maximize $\pi = x_1 + 5x_2 + 7x_3 + 3x_4$

subject to: $7x_1 + 3x_2 + 2x_3 + 4x_4 \le 15$ (constraint 1)
$8x_1 + 2x_2 + 3x_3 + 5x_4 \le 17$ (constraint 2)
$x_1 \le 4$ (constraint 3)
$x_2 \le 4$ (constraint 4)
$x_3 \le 1$ (constraint 5)
$x_4 \le 1$ (constraint 6)
$x_4 \le 0$ (cutting plane)
$x_1 \le 0$ (cutting plane)
$x_1, x_2, x_3, x_4 \ge 0$ (nonnegativity)

The optimal solution is $x_1 = 0$, $x_2 = 4$, $x_3 = 1$, $x_4 = 0$, $\pi = 27$. This does satisfy all integrality constraints, so we will not branch in this direction any further.

This is also the best current solution that satisfies all integrality constraints, so it now constitutes the current lower bound for our maximization problem.

The formulation for descendent problem 5 is:

maximize $\pi = x_1 + 5x_2 + 7x_3 + 3x_4$

subject to: $7x_1 + 3x_2 + 2x_3 + 4x_4 \le 15$ (constraint 1)
$8x_1 + 2x_2 + 3x_3 + 5x_4 \le 17$ (constraint 2)
$x_1 \le 4$ (constraint 3)
$x_2 \le 4$ (constraint 4)
$x_3 \le 1$ (constraint 5)
$x_4 \le 1$ (constraint 6)
$x_4 \le 0$ (cutting plane)
$x_1 \ge 1$ (cutting plane)
$x_1, x_2, x_3, x_4 \ge 0$ (nonnegativity)

The optimal solution is $x_1 = 1$, $x_2 = 2$, $x_3 = 1$, $x_4 = 0$, $\pi = 18$. This does satisfy all integrality constraints, so we will not branch in this direction any further.

However, this solution is inferior to our current lower bound (π=27 at problem 5) and so this solution is terminal (and we abandon our search down this path).

Since all branches have ended in solutions that satisfy all integrality constraints (or have feasible solutions worse than the current lower bound) we are finished branching.

The optimal solution to the original IP problem is the solution for descendent problem 5:

$x_1 = 0$
$x_2 = 4$
$x_3 = 1$
$x_4 = 0$
$\pi = 27$

- Uses for Integer Restricted Variables:

Either-Or Variables - Suppose there is a choice as to which of two projects that can occur, so that it is necessary that at most one of two corresponding DV's (say x_1 or x_2) is equal to one. This could be accomplished by incorporating the following constraint into the model:

$$x_1 + x_2 \leq 1$$
$$x_1, x_2 = 0 \text{ or } 1$$

If it is necessary that exactly one of two projects must be chosen, this could be accomplished by incorporating the following constraint into the model:

$$x_1 + x_2 = 1$$
$$x_1, x_2 = 0 \text{ or } 1$$

k out of r Variables can be Chosen - Generalizes either-or variables. Given a set of r binary variables (say $x_1,\dots,x_r$), at most k can be equal to one. The appropriate constraint is:

$$\sum_{i=1}^{r} x_i \le k$$

$$x_1, x_2, \dots, x_r = 0 \text{ or } 1$$

If it is necessary that exactly k of the r projects must be chosen, this could be accomplished by incorporating the following constraint into the model:

$$\sum_{i=1}^{r} x_i = k$$

$$x_1, x_2, \dots, x_r = 0 \text{ or } 1$$

Either-Or Constraints - Suppose there is a choice as to which of 2 resources to use for a certain purpose, so that it is necessary only that one of 2 constraints hold. Suppose that one of these constraints must hold:

$$3x_1 + 2x_2 \le 18$$

or

$$x_1 + 4x_2 \le 16$$

Define M as an extremely large number and y as a binary variable (= 0 or 1). Then

$$3x_1 + 2x_2 \le 18 + yM$$

$$x_1 + 4x_2 \le 16 + (1-y)M$$

will guarantee that only one constraint is binding.

k out of r Constraints must Hold - Generalizes either-or constraints. Given a set of r constraints, only k of them must hold. The r constraints are

$$
\begin{aligned}
&f_1 (x_1,\ldots,x_n) \le b_1 \\
&f_2 (x_1,\ldots,x_n) \le b_2 \\
&\quad\vdots \\
&f_r (x_1,\ldots,x_n) \le b_r
\end{aligned}
$$

Let y_i = 0 or 1 for i = 1 to r. By the same logic as for either-or constraints,

$$
\begin{aligned}
&f_1 (x_1,\ldots,x_n) \le b_1 + My_1 \\
&f_2 (x_1,\ldots,x_n) \le b_2 + My_2 \\
&\quad\vdots \\
&f_r (x_1,\ldots,x_n) \le b_r + My_r
\end{aligned}
$$

$$\sum_{i=1}^{r} y_i = r - k$$

B. Goal Programming

- **Goal Programming (GP): Branch of Linear Programming associated with solving multiobjective problems by minimizing the (weighted) sum of deviations from the goals/targets. Attributed to Charnes and Cooper [1950's].**

  Target Value - The desired value of attainment specified in a goal.

  Goal Deviation Variable - Often denoted Y_i^+ (over achievement of the target) and Y_i^- (underachievement of the target), these represent the solution's positive and negative deviations from the ith goal.

  Preemptive Priorities - A method of assigning priorities to the goals so that the relative satisfaction of a higher priority goal can not be sacrificed in the satisfaction of a lower priority goal.

Priority Weights - Assignment of weights to the goals so that the weighted sum of the deviations from the targets is minimized. Also called Penalty Points.

Omnibus Objective Function - Mathematical statement that reflects all goals in a GP problem

---

Example - American Electronics, Inc.: American Electronics, Inc. produces color television sets on two production lines. The production rate of line 1 is two sets per hour, whereas it is 1.5 sets per hour in line 2. The regular production capacity is 40 hours a week for both lines. The expected profit from an average color television set is $100. The top management of the firm has the following goals for the week in ordinal ranking:

G1 Meet the production goal of 180 sets for the week.

G2 Limit the overtime of line 1 to ten hours.

G3 Avoid the underutilization of regular working hours for both lines (assign differential weights according to the production rate of each line).

G4 Limit the sum of overtime operation for both lines.

Assign weights according to the relative cost of overtime hour. Formulate the above problem as a GP model.

Define: X_j = # production hours spent in line j per week, j = 1,2

Goal Deviation Constraints:

Production (G1)	$1.5x_1 + 2.0x_2 - (Y_1^+ - Y_1^-) = 180$
Line 1 OT (G4)	$x_1 - (Y_2^+ - Y_2^-) = 40$
Line 1 OT > 10 hrs (G2)	$Y_2^+ - (Y_3^+ - Y_3^-) = 10$
Line 2 OT (G4)	$x_2 - (Y_4^+ - Y_4^-) = 40$
Combined Line Limit (G3)	$Y_2^- + Y_4^- - (Y_5^+ - Y_5^-) = 0$

Note that Goal 3 is be dealt with by using deviation variables from Goals 2 and 4.

Now the omnibus objective function is

$$\text{Minimize } Z = Y_1^- + Y_3^+ + Y_5^- + (Y_2^+ + Y_4^+)$$

Now we must still add our goal constraints

$1.5x_1 + 2.0x_2 - (Y_1^+ - Y_1^-) = 180$ (Production Target Constraint)
$x_1 - (Y_2^+ - Y_2^-) = 40$ (Line 1 OT Limitation)
$Y_2^+ - (Y_3^+ - Y_3^-) = 10$ (Line 1 OT > 10 hrs)
$x_2 - (Y_4^+ - Y_4^-) = 40$ (Line 2 OT Limitation)
$Y_2^- + Y_4^- - (Y_5^+ - Y_5^-) = 0$ (Combined Line Limit)
$x_1, x_2, Y_1^+, Y_1^-, Y_2^+, Y_2^-, Y_3^+, Y_3^-, Y_4^+, Y_4^-, Y_5^+, Y_5^- \geq 0$ (Nonnegativity)

The optimal solution is

$$\begin{aligned} x_1 &= 40, & x_2 &= 60 \\ Y_1^+ &= 0, & Y_1^- &= 0 \\ Y_2^+ &= 0, & Y_2^- &= 0 \\ Y_3^+ &= 0, & Y_3^- &= 10 \\ Y_4^+ &= 20, & Y_4^- &= 0 \\ Y_5^+ &= 0, & Y_5^- &= 0 \end{aligned}$$

In other words, produce 40 hour's worth (60 units) on line 1 and 60 hour's worth (120) units on Line 2 (with no deviation from the target production) with no Line 1 Overtime or Underutilization (and so 10 units under the maximum line 1 Overtime), no Line 2 Underutilization, and 20 hours of Line 2 Overtime.

If we are also told that we only have 190 picture tubes available and 951 legs (it takes four legs to build our televisions), we would also have the following constraints:

$$1.5x_1 + 2.0x_2 \leq 190 \quad \text{(Picture Tube constraint)}$$
$$1.5x_1 + 2.0x_2 \leq 951/4 \quad \text{(Leg constraint)}$$

Note that these constraints are often referred to as *Hard Constraints* while Goal Constraints are commonly called *Soft Constraints*.

The resulting formulation is

$$\text{Minimize } Z = Y_1^- + Y_3^+ + Y_5^- + (Y_2^+ + Y_4^+)$$

Subject To:

$1.5x_1 + 2.0x_2 - (Y_1^+ - Y_1^-) = 180$ **(Production Target Constraint)**
$x_1 - (Y_2^+ - Y_2^-) = 40$ **(Line 1 OT Limitation)**
$Y_2^+ - (Y_3^+ - Y_3^-) = 10$ **(Line 1 OT > 10 hrs)**
$x_2 - (Y_4^+ - Y_4^-) = 40$ **(Line 2 OT Limitation)**
$Y_2^- + Y_4^- - (Y_5^+ - Y_5^-) = 0$ **(Combined Line Limit)**
$x_1 + x_2 \leq 190$ **(Picture Tube constraint)**
$x_1 + x_2 \leq 237.75$ **(Leg constraint)**
$x_1, x_2, Y_1^+, Y_1^-, Y_2^+, Y_2^-, Y_3^+, Y_3^-, Y_4^+, Y_4^-, Y_5^+, Y_5^- \geq 0$ **(Nonnegativity)**

The new optimal solution is

$$\begin{aligned} x_1 &= 40, & x_2 &= 60 \\ Y_1^+ &= 0, & Y_1^- &= 0 \\ Y_2^+ &= 0, & Y_2^- &= 0 \\ Y_3^+ &= 0, & Y_3^- &= 10 \\ Y_4^+ &= 20, & Y_4^- &= 0 \\ Y_5^+ &= 0, & Y_5^- &= 0 \end{aligned}$$

which is the exact same solution we achieved when we used only the Soft Constraints (why?).

Note that in this problem all goals are equally weighted. If we thought the fourth goal was six times as important as the other three goals, we could reformulate this problem as:

$$\text{Minimize } Z = Y_1^- + Y_3^+ + Y_5^- + 6(Y_2^+ + Y_4^+)$$

Subject To:

$x_1 + x_2 - (Y_1^+ - Y_1^-) = 180$ **(Production Target Constraint)**

$.5x_1 - (Y_2^+ - Y_2^-) = 40$ **(Line 1 OT Limitation)**

$Y_2^+ - (Y_3^+ - Y_3^-) = 10$ **(Line 1 OT > 10 hrs)**

$.67x_2 - (Y_4^+ - Y_4^-) = 40$ **(Line 2 OT Limitation)**

$Y_2^+ + Y_3^+ - (Y_5^+ - Y_5^-) = 0$ **(Combined Line Limit)**

$x_1 + x_2 \leq 190$ **(Picture Tube constraint)**

$x_1 + x_2 \leq 237.75$ **(Leg constraint)**

$x_1, x_2, Y_1^+, Y_1^-, Y_2^+, Y_2^-, Y_3^+, Y_3^-, Y_4^+ \geq 0$ **(Nonnegativity)**

In both of these formulations we are utilizing Priority Weights (i.e., Penalty Points).

The new optimal solution is

$x_1 = 40,\ x_2 = 40$

$Y_1^+ = 0,\ Y_1^- = 40$

$Y_2^+ = 0,\ Y_2^- = 0$

$Y_3^+ = 0,\ Y_3^- = 10$

$Y_4^+ = 0,\ Y_4^- = 0$

$Y_5^+ = 0,\ Y_5^- = 0$

which is the exact same solution we achieved when we used only the Soft Constraints (why?).

If we wish to use a Preemptive Priority approach, we would:

rate the goals in order of importance

solve the problem minimizing deviation from the first target value (ignoring all other goals)

add a constraint to the formulation forcing the first constraint to be satisfied at the level achieved in the previous step and solve the problem minimizing deviation from the second target value (ignoring all other goals)

Continue this process until each goal has been addressed.

C. Negative Decision Variables

- Decision Variables that can assume negative values can easily be handled in a manner similar to deviation variables. Say that decision variable x_i can assume either a positive or negative value. Then:

Define $x_i = (x_i^+ - x_i^-)$,

Substitute $(x_i^+ - x_i^-)$ for x_i everywhere in the formulation, and

Make both x_i^+ and x_i^- nonnegative

Note that x_i^+or x_i^- (or both) must equal zero

x_i^+ is the amount that x_i exceeds 0

x_i^- is the amount that x_i is below 0

D. Nonlinear Programming

- **Nonlinear Programming (NLP) - An approach to solving problems for which the objective function and/or at least one constraint has a term that involves i) an exponent that is a function of a decision variable and/or the product of multiple decision variables.**