BRACE YOURSELVES
INTEGER PROGRAMMING IS COMING
imgflip.com

Integer Programming (IP)

IP

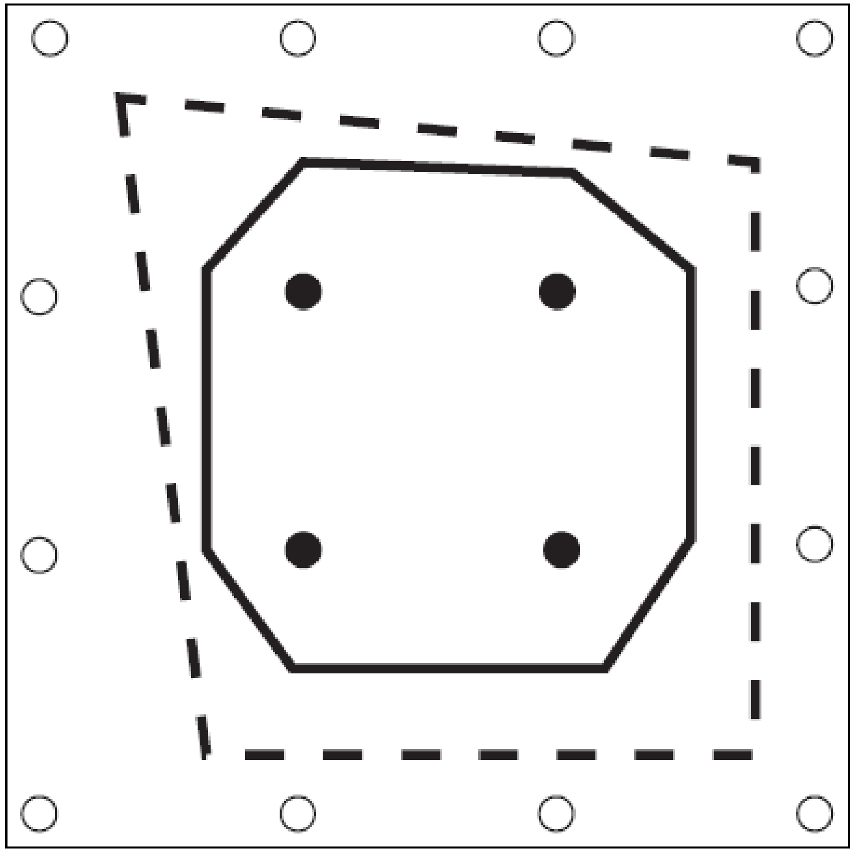
$$\max \mathrm{f} = c^{\mathrm{T}} x$$

$$s.a.$$

$$Ax \leq \mathrm{b}$$

$$\mathrm{x} \geq 0$$

$$\boxed{x \in Z^n}$$



Usually, integer programming problems are very complex and it is difficult to obtain the optimal solution in useful time.

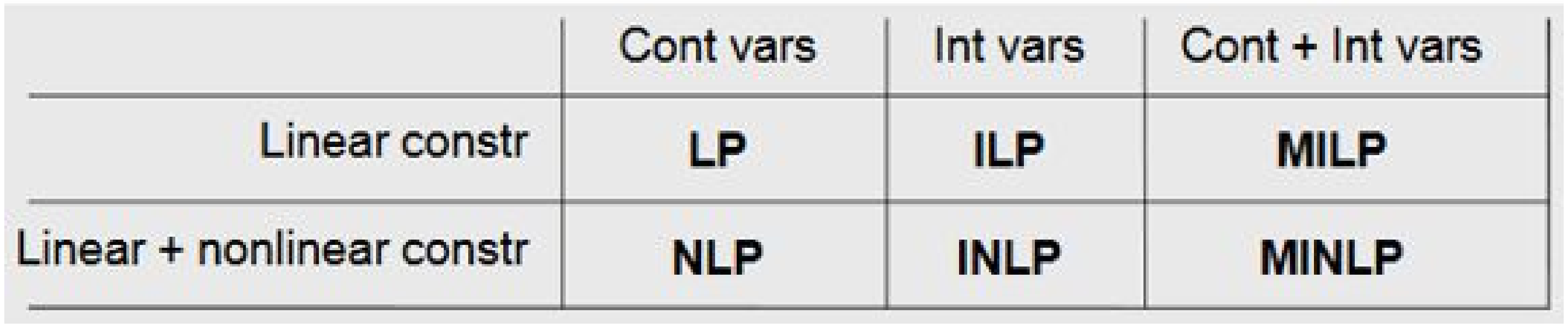
So, it is very important to obtain lower and upper bounds for the optimal solution through **Linear Relaxation** and **Branch&Bound** method.

History of Integer Programming (IP)

- Introduced in 1951 (Dantzig)
- TSP as special case in 1954 (Dantzig)
- First convergent algorithm in 1958 (Gomory)
- General branch-and-bound technique 1960 (Land and Doig)
- Frequently used to prove bounds on approximation algorithms (late 90s)

Decision Variables in IP Models

- An IP is a mixed integer program (MIP) when some but not all of the decision variables are integer.
- If all decision variables are integer we have a pure IP.
- A binary decision variable must be 0 or 1 (a yes-no decision variable).
- If all decision variables are binary, then the IP is a binary IP (BIP).
- Decision variables that are not required to be integer-valued are continuous variables.



	Cont vars	Int vars	Cont + Int vars
Linear constr	**LP**	**ILP**	**MILP**
Linear + nonlinear constr	**NLP**	**INLP**	**MINLP**

Example of IP: Facility Location

A company is thinking about building new facilities in PO and LX.

	capital needed	expected profit
1. factory in PO	€6M	€9M
2. factory in LX	€3M	€5M
3. warehouse in PO	€5M	€6M
4. warehouse in LX	€2M	€4M

Total capital available for investment: €10M

Question: Which facilities should be built to maximize the total profit?

Example of IP: Facility Location

- Define decision variables (*i* = 1, 2, 3, 4):

$$x_i = \begin{cases} 1 & \text{if facility } i \text{ is built} \\ 0 & \text{if not} \end{cases}$$

- Then the total expected benefit: $9x_1+5x_2+6x_3+4x_4$

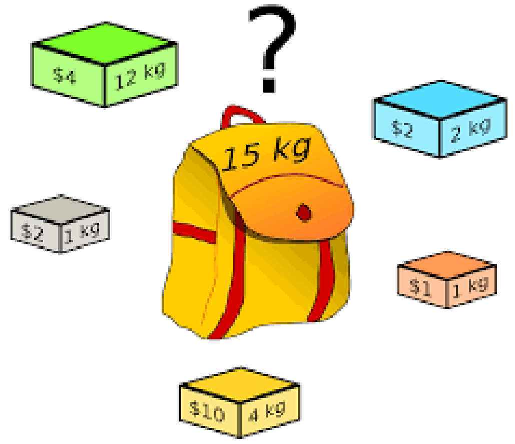
  the total capital needed: $6x_1+3x_2+5x_3+2x_4$

➢ Summarizing, the IP model is:

$$\max 9x_1+5x_2+6x_3+4x_4$$
$$\text{s.t. } 6x_1+3x_2+5x_3+2x_4 \leq 10$$
$$x_i \in \{0,1\}$$

The Knapsack Problem

- Knapsack with volume 15
- What should you take with you to maximize utility?



Items	1	2	3	4	5
	Paper	Book	Bread	Smartphone	Water
Utility	8	12	7	15	12
Volume	4	8	5	2	6

Knapsack problem

- The ***knapsack problem*** is inspired in the situation where a hiker has to decide which objects to include inside the knapsack for his trip.

- c_j is the ‘‘value’’ or utility of including object j that weighs $a_j > 0$ kilos;
- n items to be packed in a knapsack.
- The knapsack can hold up to b kg of items.
- **Goal:** Pack the knapsack such that the total benefit is maximized.

- Any IP, which has only one constraint, is referred to as a knapsack problem .
- The knapsack problem is NP hard.

IP model for Knapsack problem

- Define decision variables (*j* = 1, …, n): $x_j = \begin{cases} 1 & \text{if item } j \text{ is packed} \\ 0 & \text{if not} \end{cases}$
- Total benefit: $\sum_{j=1}^{n} c_i x_j$
- Total weight: $\sum_{j=1}^{n} a_j x_j$
- IP model:

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} c_j x_j \\ s.t. \quad & \sum_{j=1}^{n} a_j x_j \le b \\ & x_j \text{ binary } (j = 1, \ldots, n) \end{aligned}$$

Connection between problems

This version of the facility location problem is a special case of the knapsack problem.

Important modeling skill:

- Suppose you know how to model Problem $A_1,\dots,A_p$;
- You need to solve Problem B;
- Notice the similarities between Problems A_i and B;
- Build a model for Problem B, using the model for Problem A_i as a prototype.

The Facility Location Problem: adding new requirements

x_1: factory PO; x_2: factory LX
x_3 : warehouse PO; x_4: warehouse LX

- Extra requirement: build *at most one* of the two warehouses.
  The corresponding constraint is:
  $$x_3 + x_4 \leq 1$$

- Extra requirement: build *at least one* of the two factories.
  The corresponding constraint is:
  $$x_1 + x_2 \geq 1$$

Modeling Technique: Restrictions on the number of options

- Suppose in a certain problem, n different options are considered. For i=1,…,n

$$x_i = \begin{cases} 1 & \text{if option } i \text{ is chosen} \\ 0 & \text{if not} \end{cases}$$

- *Restrictions:* At least *p* and at most *q* of the options can be chosen.

- The corresponding constraints are:

$$\sum_{i=1}^{n} x_i \geq p \qquad \sum_{i=1}^{n} x_i \leq q$$

Modeling Technique:
Contingent Decisions

Back to the facility location problem:

x_1: factory PO; x_2: factory LX
x_3 : warehouse PO; x_4: warehouse LX

- *Requirement:* Can't build a warehouse *unless* there is a factory in the same city.

  The corresponding constraints are:

  $x_3 \leq x_1$ (PO) $x_4 \leq x_2$ (LX)

- *Requirement:* Can't select option 3 (x_3) *unless* at least one of options 1 and 2 (x_1 or x_2) is selected.

  The constraint: $x_3 \leq x_1 + x_2$

- *Requirement:* Can't select option 4 (x_4) *unless* at least two of options 1, 2 and 3 are selected ($x_1 + x_2 + x_3$)

  The constraint: $2x_4 \leq x_1 + x_2 + x_3$

Modeling Technique: Variables with k possible values

- Suppose variable y should take one of the values $d_1, d_2, \ldots, d_k$.
  How to achieve that in the model?
- Introduce new decision variables. For i=1,…,k,

$$x_i = \begin{cases} 1 & \text{if } y \text{ takes value } d_i \\ 0 & \text{otherwise} \end{cases}$$

- Then we need the following constraints.

$$\sum_{i=1}^{k} x_i = 1 \;\; (y \text{ can take only one value})$$

$$y = \sum_{i=1}^{k} d_i x_i \;\; (y \text{ should take value } d_i \text{ if } x_i = 1)$$

Logical constraints: Alternative Constraints

Consider a situation with the alternative constraints:

$f_1(x_1, x_2, \ldots, x_n) \le b_1,$

$f_2(x_1, x_2, \ldots, x_n) \le b_2.$

At least one, but not necessarily both, of these constraints must be satisfied.

This restriction can be modeled by combining the technique just introduced with a multiple-choice constraint as follows:

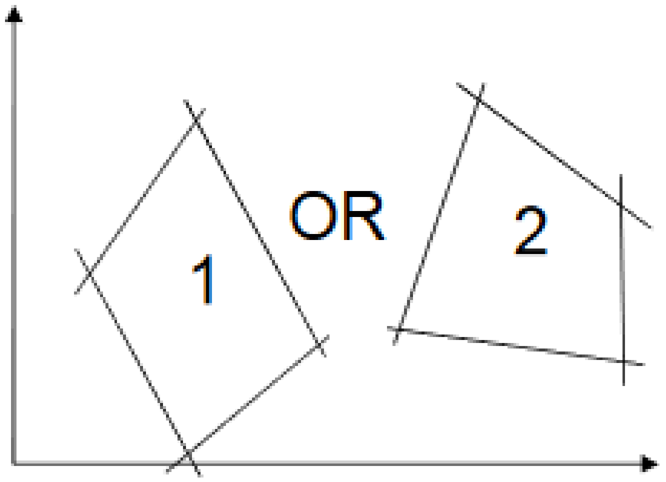
$f_1(x_1, x_2, \ldots, x_n) \le b_1 + M\,y_1$

$f_2(x_1, x_2, \ldots, x_n) \le b_2 + M\,y_2$

$y_1 + y_2 \le 1,$

y_1, y_2 binary

and M is chosen to be large enough



$$[x_1 - x_2 \le -2] \vee [-x_1 + x_2 \le -2]$$

$$0 \le x_1, x_2 \le 4$$

Big-M (M=6)

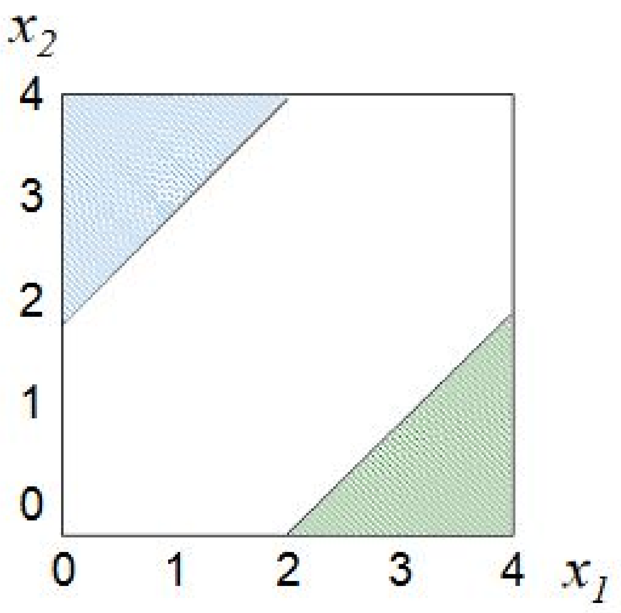
$$x_1 - x_2 \le -2 + M(1 - y_1)$$

$$-x_1 + x_2 \le -2 + M(1 - y_2)$$

$$y_1 + y_2 = 1$$

$$0 \le x_1, x_2 \le 4$$

$$y_1, y_2 \in \{0,1\}$$



Logical constraints: K out of N constraints must hold

N constraints

$$f_1(x_1,\dots,x_n) \leq b_1$$

□

$$f_N(x_1,\dots,x_n) \leq b_N$$

At least *K* out of *N* must be satisfied.

$$f_1(x_1,\dots,x_n) \leq b_1 + M(1 - y_1)$$

□

$$f_N(x_1,\dots,x_n) \leq b_N + M(1 - y_N)$$

$$\sum_{i=1}^{N} y_i = K$$

$$y_i \in \{0,\ 1\}, \quad i = 1,\dots,N$$

M is chosen to be large enough

Example of *K* out of *N* Constraints

- A production system has *N* potential quality control inspection strategies.
- Management has decided that *K* of these strategies should be adopted.

Logical constraints: Conditional Constraints

These constraints have the form:

$$f_1(x_1, x_2, \ldots, x_n) > b_1 \text{ \underline{implies} that } f_2(x_1, x_2, \ldots, x_n) \le b_2.$$

Nota:
$a \Rightarrow b \Leftrightarrow \sim a \vee b$

The conditional constraint is logically equivalent to the alternative constraints:

$$f_1(x_1, x_2, \ldots, x_n) \le b_1 \text{ or } f_2(x_1, x_2, \ldots, x_n) \le b_2,$$

where at least one must be satisfied.

Hence, this situation can be modeled by alternative constraints as indicated before.

Modelling minimum constraints

Consider that our integer variable X must satisfy **X = min(A,B)**.

Then, we may write the following equivalent constraints:

1. $X \le A$
2. $X \le B$
3. $X = A$ OR $X = B$

Then we have:

1. $X \le A$
2. $X \le B$
3. Create two new decision binary variables Y1 and Y2, M is chosen to be large enough
   - 3.1 $X-A-M*Y1 = 0$
   - 3.2 $X-B-M*Y2 = 0$
   - 3.3 $Y1+Y2 = 1$

Modelling maximum constraints

Consider that our integer variable X must satisfy **X = max (A,B)**.

Then, we may write the following equivalent constraints:

1. $X \geq A$
2. $X \geq B$
3. $X = A$ OR $X = B$

Then we have:

1. $X \geq A$
2. $X \geq B$
3. Create two new decision binary variables Y1 and Y2, M is chosen to be large enough
   - 3.1 $X-A-M*Y1 =0$
   - 3.2 $X-B-M*Y2 =0$
   - 3.3 $Y1+Y2 =1$

Modelling Absolute value

Consider that our integer variable X must satisfy **X = |A|**.

Then, we may write the following equivalent constraints:

1. $A \leq X$
2. $-A \leq X$
3. $X \leq A$ OR $X \leq -A$

The first two constraints force X to be a non-negative number, of size at least |A|.

For the third constraint, if $A = 0$, the two “or” clauses are the same, and $X = 0$. If $A \neq 0$, as X is non-negative it can only be less than whichever of A and −A is non-negative.

Uncapacitated Facility Location Problem

Problem: Open a set of facilities and assign each customer to one facility such that cost is minimized.

The cost could be a function of distance from facility to customer or could be based on a response time (e.g., locating fire stations).

Data:

f_i : cost of opening a facility at location i

c_{ij} : cost of assigning customer j to facility i

Decision variables:

y_i : open facility at location i (1 = yes, 0 = no)

x_{ij} : assign customer j to location i (1 = yes, 0 = no)

Uncapacitated Facility Location Problem IP formulation

$$\text{Min} \quad \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} f_i y_i$$

$$\text{s.t.} \quad \sum_{i \in I} x_{ij} = 1, \qquad \forall\, j \in J$$

Each customer is assigned to exactly one facility

$$x_{ij} \leq y_i, \qquad \forall\, i \in I,\; j \in J$$

Setup constraint

$$x_{ij} \in \{0,1\}\; y_i \in \{0,1\}, \quad \forall\, i \in I,\; j \in J$$

Can assign customer *j* to facility *i* only if we open facility *i*.

Example: Facility Location Problem

Problem: A company has *m* potential warehouse sites and *n* customers. A manager must decide which warehouses to use to meet the demand of the customers.

Data:

f_i = fixed operating cost for warehouse **i** , if opened (for example, a cost to lease the warehouse),

c_{ij} = per-unit operating cost at warehouse i plus the transportation cost for shipping from warehouse **i** to customer **j**.

d_j : demand for customer *j*

s_i : capacity (supply) of warehouse *i*

Decision variables:

y_i : build a warehouse at site *i* (1 = yes, 0 = no)

x_{ij} : shipments from warehouse *i* to customer *j*

Facility Location IP Model

$$\text{Min} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} f_i y_i$$

$$\text{s.t.} \quad \sum_{i=1}^{m} x_{ij} = d_j \qquad j = 1,\ldots,n$$

$\longleftarrow$ Satisfy each customer's demand

$$\sum_{j=1}^{n} x_{ij} \leq s_i y_i \qquad i = 1,\ldots,m$$

$\longleftarrow$ each warehouse can ship no more than its supply if it is built

$$x_{ij} \geq 0, \quad i = 1,\ldots,m, \quad j = 1,\ldots,n$$

$$y_i \in \{0,1\}, \quad i = 1,\ldots,m$$

Set Covering Example

A simple example of a covering problem is hiring sports coaches at a school. Each applicant can coach some, but not all, of the sports that the school needs coaches for. For example:

Coach	Sports
Adams	Soccer, Rugby, Swimming
Brown	Soccer, Baseball, Swimming
Caroll	Baseball, Rugby
Devin	Swimming
Ernst	Basketball

The school wishes to hire as few coaches as possible, yet still have teams in Baseball, Basketball, Soccer, Swimming, and Rugby.

Since this is a small problem, it's easy to see that hiring coaches Brown, Caroll, and Ernst would be the best choice. Coach Adams, although he coaches three sports, duplicates all of his sports with coaches Brown and Caroll who can coach other sports as well, so Adams is not selected.

Set Covering Problem

A hospital ER needs to keep doctors on call, so that a qualified individual is available to perform every medical procedure that might be required (there is an official list of such procedures). For each of several doctors available for on-call duty, the additional salary they need to be paid, and which procedures they can perform, is known. The goal to choose doctors so that each procedure is covered, at a minimum cost.

Example:

	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6
Procedure 1	X			X		
Procedure 2	X				X	
Procedure 3		X	X			
Procedure 4	X					X
Procedure 5		X	X			X
Procedure 6		X				

Set Covering Problem

Data representation: incidence matrix.

With m procedures and n available doctors, the data can be represented as

A (mxn), where *aij = 1 if doctor j can perform procedure i and*
aij = 0 otherwise.

Also, let c_j , $j = 1, .., n$ be the additional salary that will need to be paid to doctor j for on-call duty.

Variables: xj = 1 if doctor j is on call, and 0 otherwise.

Set Covering and Partitioning

Given n sets and m items:

n columns = sets

m Rows = items

$$
\begin{aligned}
A_{ij} &= \begin{cases} 1, & \text{if set } j \text{ includes item } i \\ 0, & \text{otherwise} \end{cases} \\
c_j &= \text{cost of set } j \\
x_j &= \begin{cases} 1, & \text{if set } j \text{ is included} \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
$$

Set covering:

minimize:	$c^T x$
subject to:	$Ax \geq 1$, x binary

Set partitioning:

minimize:	$c^T x$
subject to:	$Ax = 1$, x binary

Solving Integer Linear Problems

Methods to solve integer programming problems

Whereas the simplex method is effective for solving linear programs, there is no single technique for solving integer programs.

Instead, a number of procedures have been developed, and the performance of any particular technique appears to be highly problem-dependent.

Methods to date can be classified as following one of three approaches:

- i) rounding techniques
- ii) cutting-plane techniques;
- iii) group-theoretic techniques
- **iv) enumeration techniques, including the branch-and-bound procedure**
- v) heuristics and metaheuristics

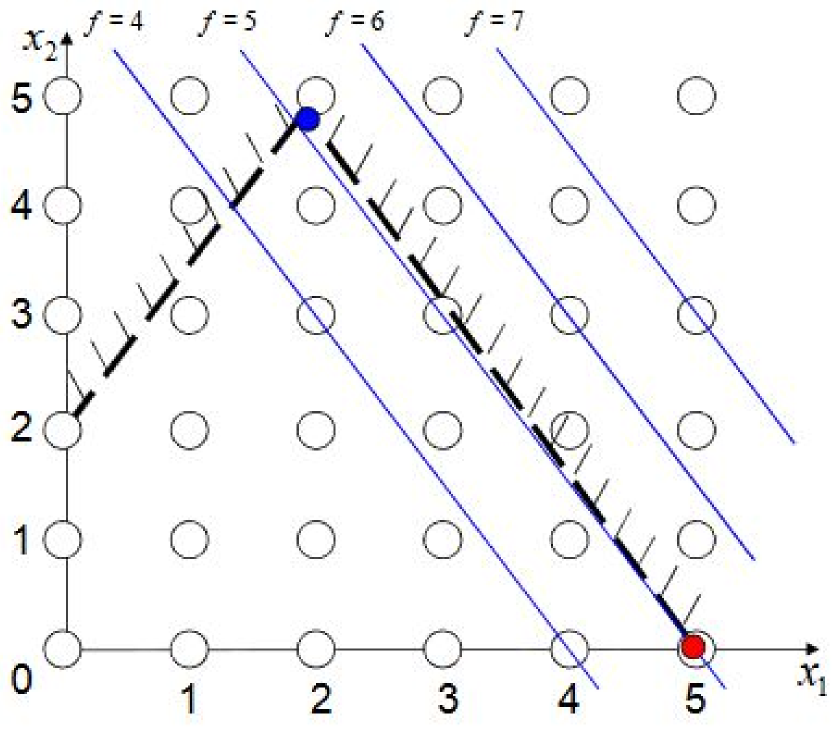
In addition, several composite procedures have been proposed, which combine techniques using several of these approaches.

IP Current Status

- Has become "dominant" over linear programming in past decade
- Saves billions of dollars/year in industry
- Provides benchmarks for TSP problems
- 1 million variable LP approximations
- Branch-and-bound, cutting planes, and separation all used in practice
- General purpose packages do not tend to work as well as with linear programming -- knowledge of the domain is critical.

Rounding

Rounding is a simple solution that forgets about discrete variables, solve the LP problem and use some round-off strategy



$$\max f(x_1,x_2) = x_1 + 0.64x_2$$
$$50x_1 + 31x_2 \le 250$$
$$-3x_1 + 2x_2 \le 4$$
$$x_1, x_2 \in Z$$

LP-solution: (1.95, 4.92)
where f=5.098

IP-solution: (5,0)
Where f=5

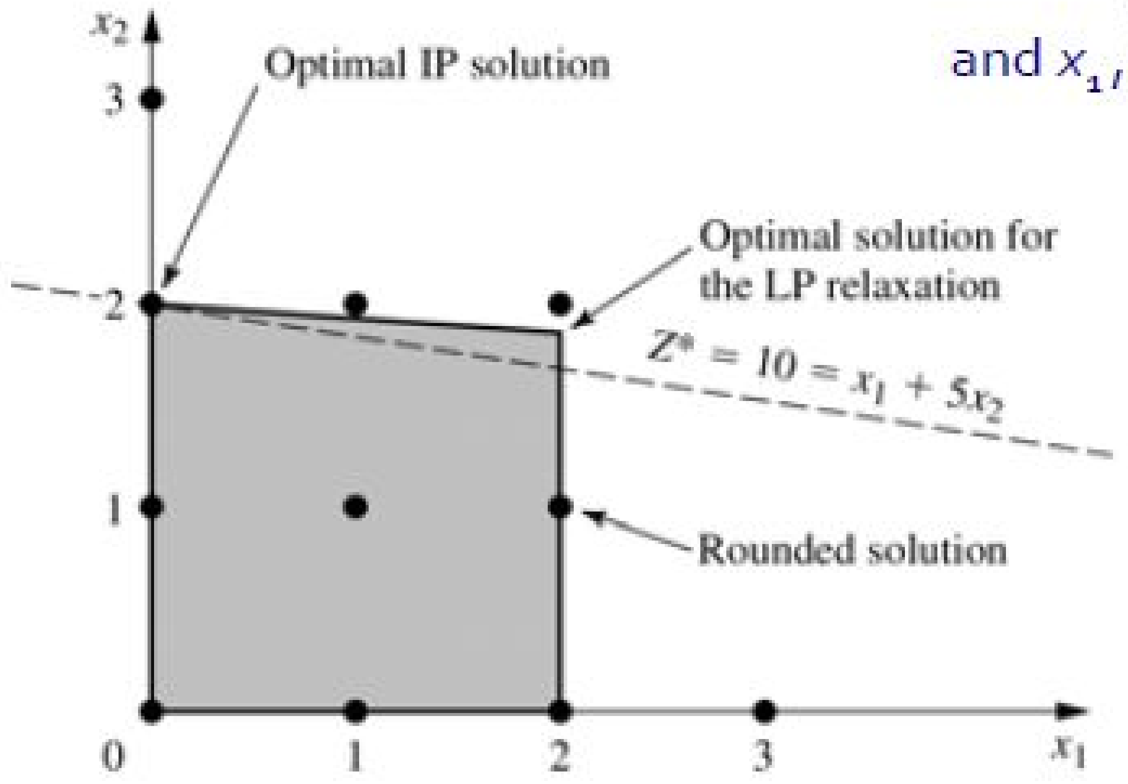
But the rounded LP-solution is not feasible…
and the IP solution is very different from the LP solution ☹

Just another example

Minimize $Z = x_1 + 5x_2$ subject to $x_1 + 10x_2 \le 20$

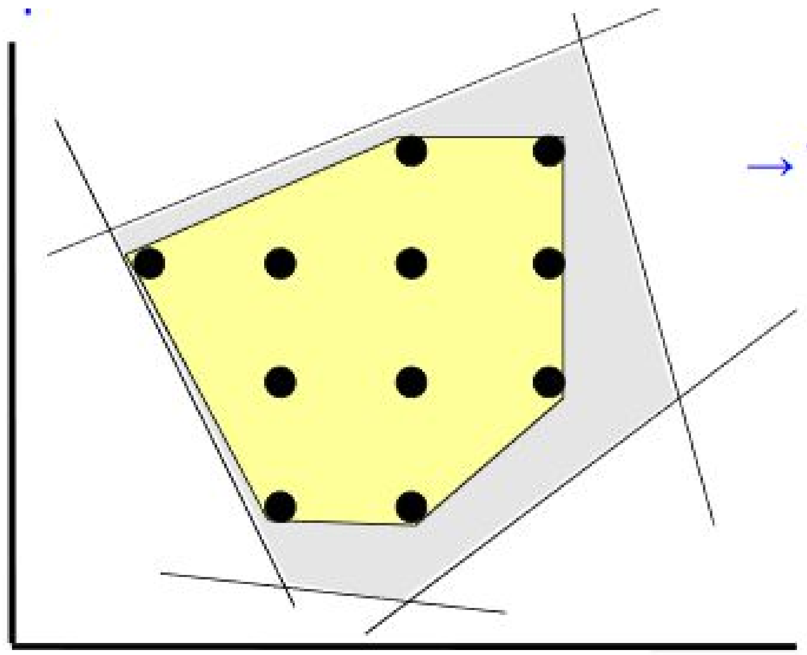
$x_1 \le 2$

and $x_1, x_2 \ge 0$, integers.



The feasible solutions of a IP

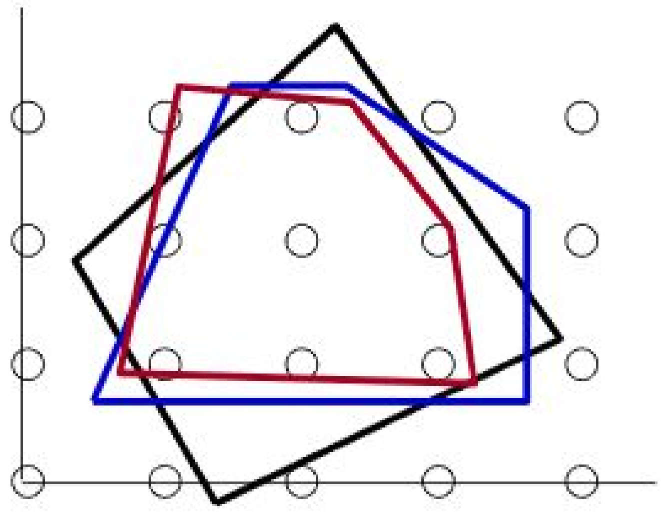
- The feasible region of an integer problem is a subset of the continuous one.
- MILP or IP = initial LP + integer variable constraints
- **A solution can never improve if a constraint is added to the problem**

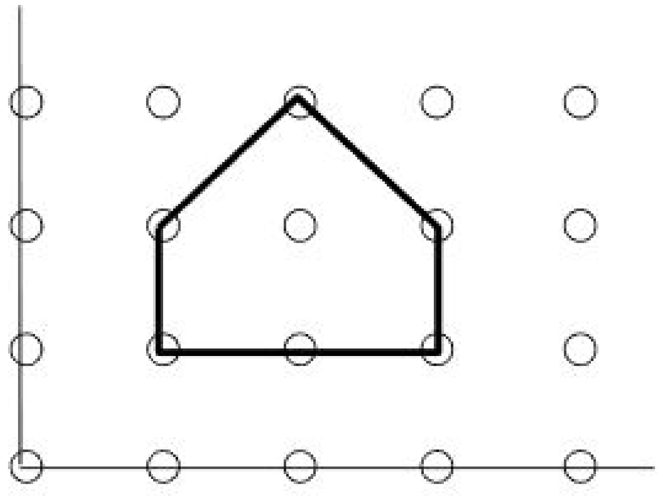


→ The initial IP-solution is never better as the original LP

IP formulation and convex hulls

- An IP can in principle be formulated in several ways.
- Convex hull join all feasible extreme points into a convex envelope.
- Ideal Convex hull = tightest possible IP formulation





Are integer problems easy to solve?

- A difference to LP is that IP has fewer solutions.
- IP problems have a finite number of feasible solutions

However,

- Finite numbers can be very large!!! With n variables, a BIP problem has 2^n solutions, having exponential growth.
- LP assures that a continuous solution can be optimal, showing the efficiency of the Simplex method.
- LP problems are much easier to solve than IP problems!!!

Using optimality conditions and bounds

- Optimality conditions allow us to evaluate the quality of a candidate solution even if we do not know the optimal solution (z).
- One method consists in finding a lower bound $\underline{z} \leq z$

  and an upper bound $\overline{z} \geq z$ such that, if $\overline{z} = \underline{z}$ then z is optimal.
- Usually, we build sequences of upper and lower bounds:

  $$\overline{z}_1 \geq \overline{z}_2 \geq \ldots \geq \overline{z}_k \geq z \qquad \underline{z}_1 \leq \underline{z}_2 \leq \ldots \leq \underline{z}_t \leq z$$

  such that for any ζ >0, we can define an interval $\left|\overline{z_k} - \underline{z}_t\right| < \zeta$

This interval will be the <u>stopping criterium</u> for the algorithm.

Optimality conditions

- When the algorithm stops, we know that the candidate solution z is at most ζ units inferior than the optimal solution.

- To be able to estimate the quality of a candidate solution has a vital importance for the optimization process.

Relaxation:

Transformation of a given problem in a different, simpler, with a wider solution space than the original problem space.
A relaxation provides an upper bound.

In a maximization problem, $\bar{z} \geq z$

Types of relaxation:
- ignore some constraints
- ignore the variable's integrality constraints (Linear relaxation)

LP relaxation

- Consequently, most IP algorithms incorporate the Simplex method. This is called LP relaxation.
- Sometimes the solution of the LP problem is also the solution of the IP problem, such as:

  transportation problem, assignment problem, shortest path problem, maximum flow problem,…
- Special structures, such as mutually exclusive constraints or contigent decisions may sometimes simplify the problem.

Linear Relaxation, upper bounds and lower bounds

$$\boxed{\begin{array}{l} \max\ 4x_1 - x_2 \\ s.to \\ 7x_1 - 2x_2 \le 14 \\ x_2 \le 3 \\ 2x_1 - 2x_2 \le 3 \\ x_1 \ge 0, \text{integer} \\ x_2 \ge 0, \text{integer} \end{array}}$$

Lower bound

$x_1 = 2$ and $x_2 = 1$ is a feasible solution, so $z \ge 7$.

Upper bound

The optimal soluion of the linear relaxation is $x_1 = 2.857$ and $x_2 = 3$, with $z_{LP} = 8.428$

Then, $z \le 8 \le 8.428$

Branch & Bound method (1)

The Branch & Bound is the most common method for the resolution of Integer Programming problems and Mixed Integer Programming problems.

It is a divide and conquer approach: we divide the original problem into several simpler problems.

1. Divide P problem into a set of sub-problems {Pk}.
2. Solve sub-problems Pk.
3. Obtain the solution of P through the solutions of the sub-problems.

Branch & Bound (2)

Consider the problem:

$$P : z = \max \{c^T x : x \in S\}$$

How to divide P into smaller sub-problems that will recombine in such a way that a solution for the original problem can be found?

Proposition

- Let $S = S_1 \cup \ldots \cup S_K$ be a decomposition of S into K smaller sets.
- Let $z_k = \max\{c^T x : x \in S_k\}$ for $k = 1, \ldots, K$.
- Then, $z = \max \{z_k : k = 1, \ldots, K\}$.

Branch & Bound (3)

Solve the Linear Relaxation (LP) of the original problem (IP).
As a result we can have the following situations (fathoming tests):

1. The LP does not have feasible solutions: The IP does not have feasible solutions.
2. The LP optimal solution is a feasible IP solution: we have found the optimal solution.
3. The LP optimal solution is not feasible for the IP: we have an upper bound.

In cases 1 and 2, the algorithm stops.
In case 3, we divide (or branch) the problem and solve the next sub-problems recursively.

Branch & Bound (4)

In order to divide a problem into sub-problems:

- Select a variable i, which value, x_i , is not integer in the LP solution.
- Create two subproblems:
  - ✓ *In one of the sub-problems, impose the constraint:* $x_i \leq \lfloor x_i \rfloor$
  - ✓ *In the other one, impose the constraint* $x_i \geq \lceil x_i \rceil$
- After the branching, solve the subproblems recursively..
- If the optimal solution of the linear relaxation is inferior to the current lower bound, <u>it is not necessary to analyse the subproblem further</u>.
- This is the key of success and efficiency of the method: implicit enumeration.

Some questions

(i) Selecting a branching strategy (which is the next problem to analyze):

- depth-first search:
- breadth-first search
- choose the problem with highest upper bound.

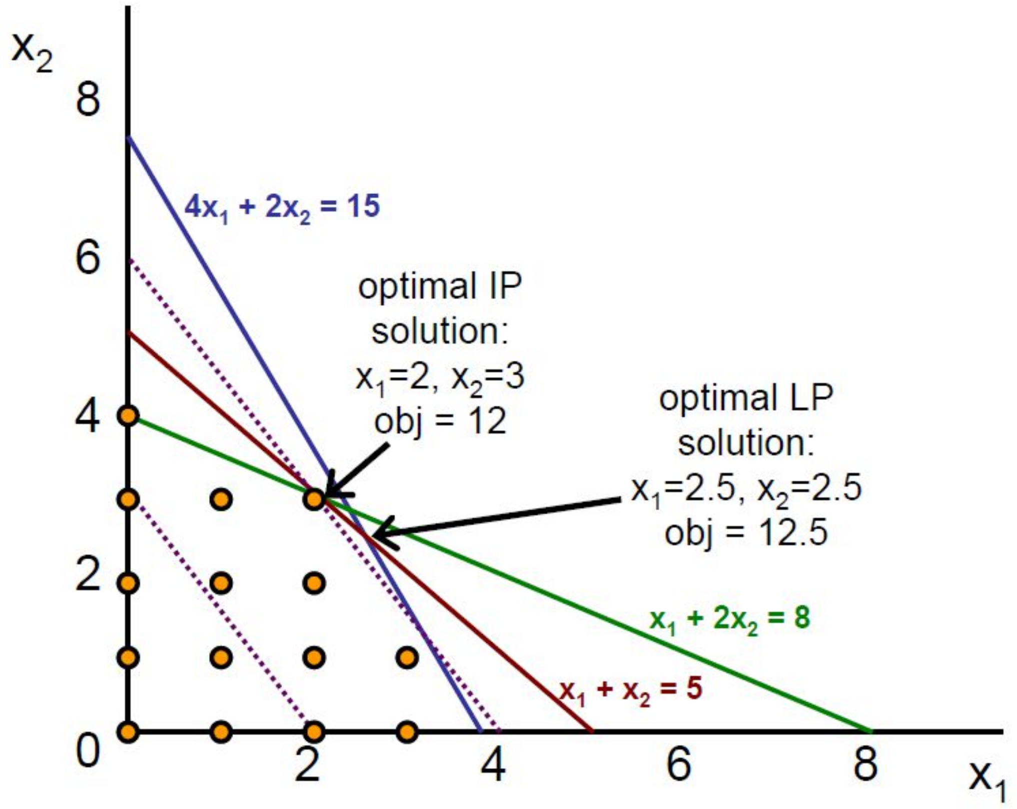
(ii) Once we have chosen the sub-problem, which will be the next variable to analyze, from those that do not have integer values?

- "Best-first" chooses the variable with the highest upper bound.
- Several different strategies have been proposed in literature, but there is no consensus regarding their efficiency.

Example

P_1

$$\boxed{\begin{array}{l} \max 3x_1 + 2x_2 \\ s.to \\ 4x_1 + 2x_2 \le 15 \\ x_1 + 2x_2 \le 8 \\ x_1 + x_2 \le 5 \\ x_1 \ge 0, \text{integer} \\ x_2 \ge 0, \text{integer} \end{array}}$$



P1 optimal solution:	x1 = 2	x2 = 3	obj = 12
PL optimal solution:	x1 = 2.5	x2 = 2.5	obj = 12.5

Example

P1 divides into P11 and P12

PI optimal solution:	x1 = 2	x2 = 3	f.o = 12
LP optimal solution :	x1 = 2.5	x2 = 2.5	f.o = 12.5

Since both x1 and x2 are non integer, we can branch the problem in x1 or x2.
Branching the problem in x1 (we could also have chosen x2)

P_{11}

$$\begin{aligned}&\text{maximize } 3x_1 + 2x_2\\ &\text{subject to}\\ &4x_1 + 2x_2 \leq 15\\ &x_1 + 2x_2 \leq 8\\ &x_1 + x_2 \leq 5\\ &\boxed{x_1 \geq 3}\\ &x_1 \geq 0,\text{ integer; } x_2 \geq 0,\text{ integer}\end{aligned}$$

P_{12}

$$\begin{aligned}&\text{maximize } 3x_1 + 2x_2\\ &\text{subject to}\\ &4x_1 + 2x_2 \leq 15\\ &x_1 + 2x_2 \leq 8\\ &x_1 + x_2 \leq 5\\ &\boxed{x_1 \leq 2}\\ &x_1 \geq 0,\text{ integer; } x_2 \geq 0,\text{ integer}\end{aligned}$$

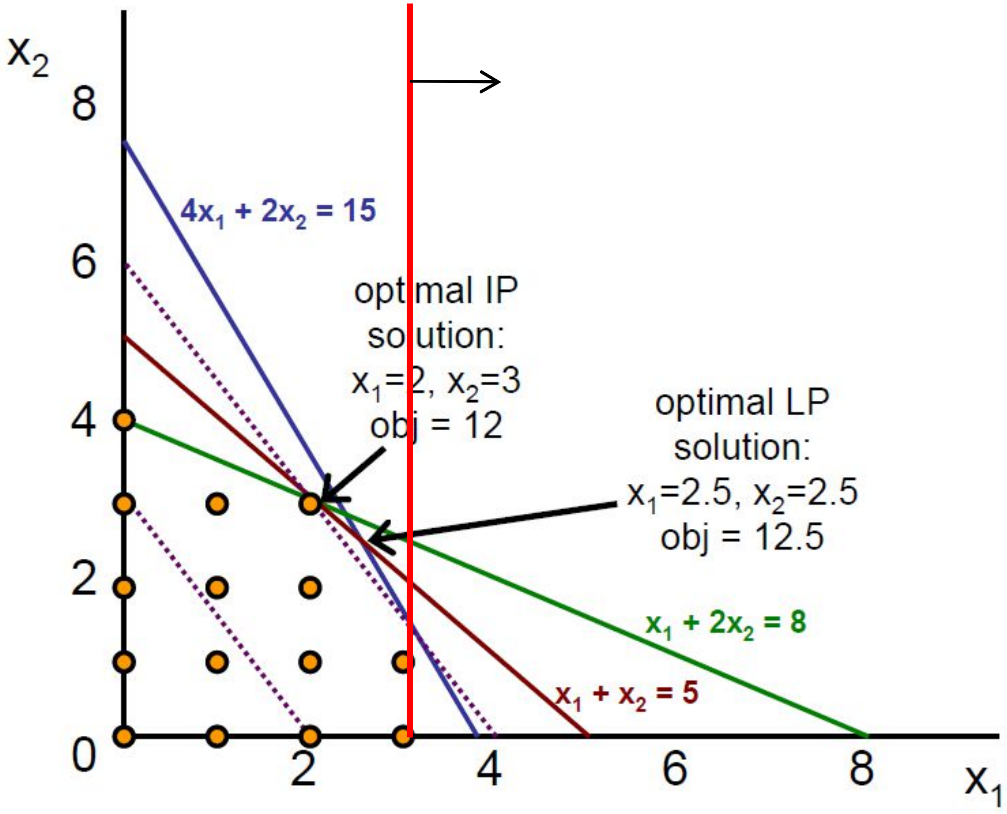
Example

solving P11 and dividing into P111 and P112

P11

$$\begin{aligned}&\textbf{maximize } 3x_1 + 2x_2\\&\textbf{subject to}\\&4x_1 + 2x_2 \le 15\\&x_1 + 2x_2 \le 8\\&x_1 + x_2 \le 5\\&x_1 \ge 3\\&x_1 \ge 0, \text{ integer}; \; x_2 \ge 0, \text{ integer}\end{aligned}$$

Solving the linear relaxation:



X1 = 3
X2 = 1.5
f.o = 12

Since x_2 is not integer, we have to branch the problem again, adding the constraints $x_2 \le 1$ e $x_2 \ge 2$

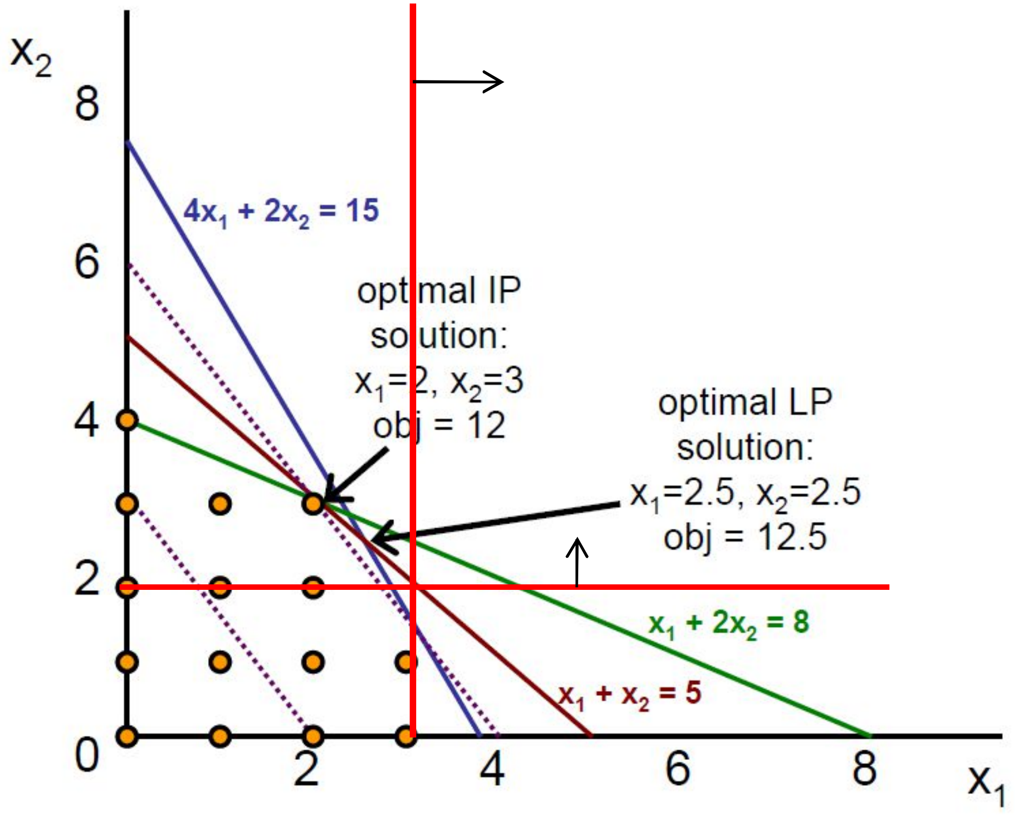
Example
solving P111

P_{111}

$$\begin{aligned}&\textbf{maximize } 3x_1 + 2x_2\\&\textbf{subject to}\\&4x_1 + 2x_2 \le 15\\&x_1 + 2x_2 \le 8\\&x_1 + x_2 \le 5\\&\boxed{x_1 \ge 3}\\&\boxed{x_2 \ge 2}\\&x_1 \ge 0, \text{ integer}; \; x_2 \ge 0, \text{ intege}\end{aligned}$$

This problem does not have feasible solutions.

The search on this brach ends here

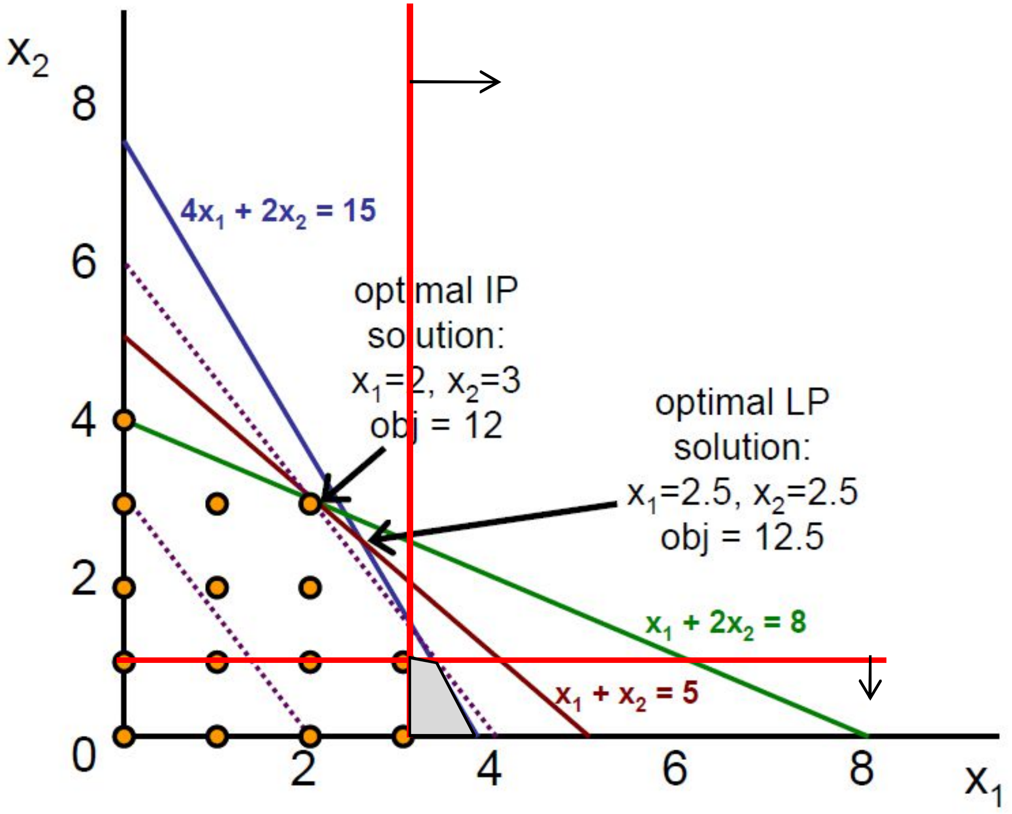


Example

solving P112 and dividing into P1121 and P1122

P_{112}

$$\begin{aligned}
&\text{maximize } 3x_1 + 2x_2 \\
&\text{subject to} \\
&4x_1 + 2x_2 \leq 15 \\
&x_1 + 2x_2 \leq 8 \\
&x_1 + x_2 \leq 5 \\
&\boxed{x_1 \geq 3} \\
&\boxed{x_2 \leq 1} \\
&x_1 \geq 0, \text{ integer}; \; x_2 \geq 0, \text{ integer}
\end{aligned}$$



Solving the linear relaxation:

X1 = 3.25
X2 =1
f.o = 11.75

Since x_1 is not integer, we divide the problem again, adding the constraints $x_1 >= 4$ e $x_1 <= 3$

Example

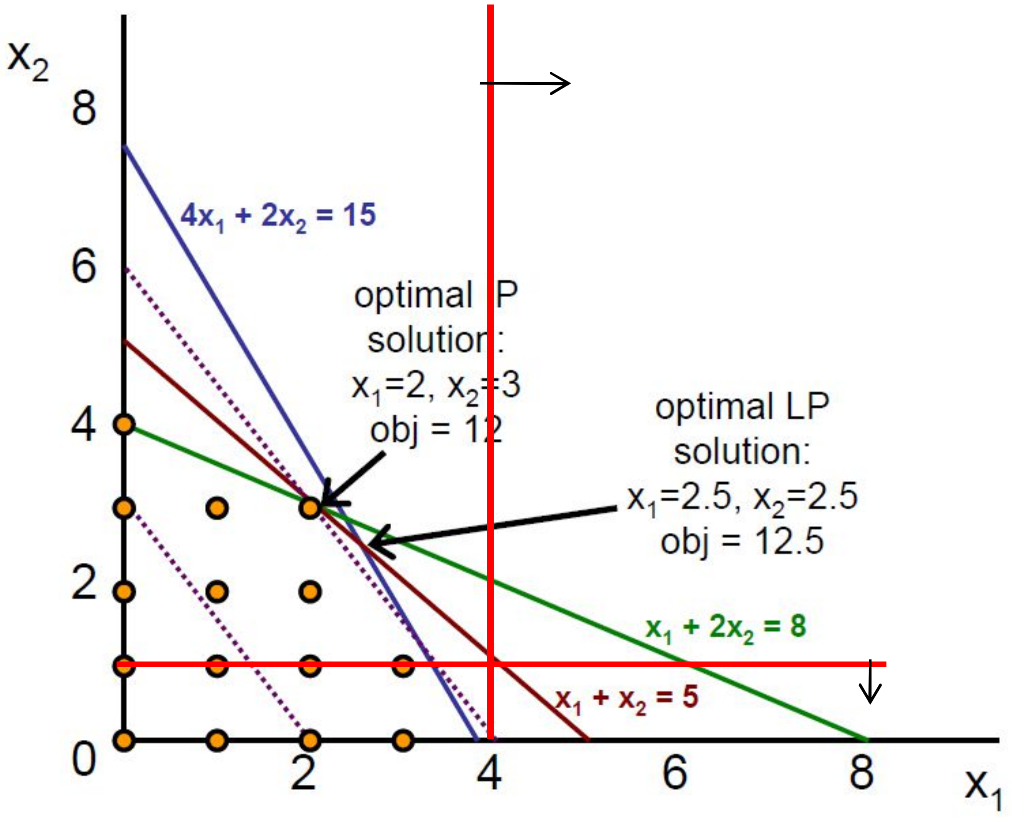
solving P1121

P_{1121}

$$\begin{aligned}
&\text{maximize } 3x_1 + 2x_2\\
&\text{subject to}\\
&4x_1 + 2x_2 \le 15\\
&x_1 + 2x_2 \le 8\\
&x_1 + x_2 \le 5\\
&\boxed{x_1 \ge 3}\\
&\boxed{x_2 \le 1}\\
&\boxed{x_1 \ge 4}\\
&x_1 \ge 0, \text{ integer}; \; x_2 \ge 0, \text{ integer}
\end{aligned}$$

This problem does not have feasible solutions.

The search on this branch ends here.



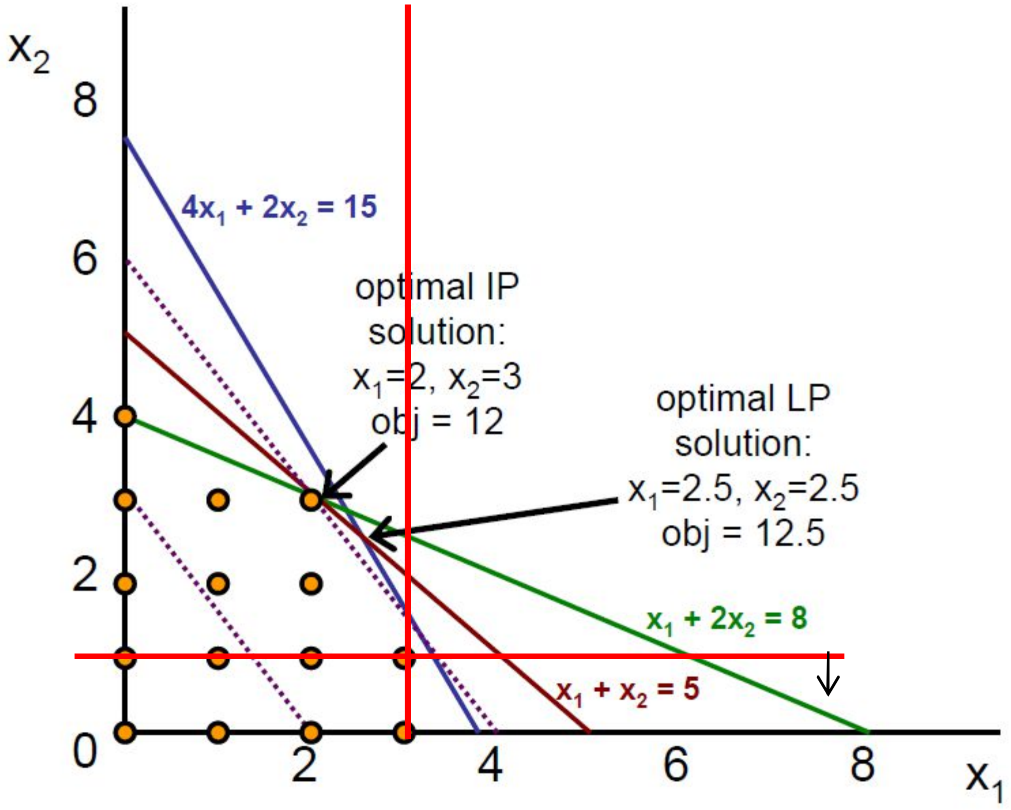
Example
solving P1122- integer solution and lower bound

P_{1122}

$$
\begin{aligned}
&\textbf{maximize } 3x_1 + 2x_2 \\
&\textbf{subject to} \\
&4x_1 + 2x_2 \le 15 \\
&x_1 + 2x_2 \le 8 \\
&x_1 + x_2 \le 5 \\
&\boxed{x_1 \ge 3} \\
&\boxed{x_2 \le 1} \\
&\boxed{x_1 \le 3} \\
&x_1 \ge 0, \text{ integer}; \ x_2 \ge 0, \text{ integer}
\end{aligned}
$$

Solving the linear relaxation:

X1 = 3
X2 = 1
f.o. = 11



x_2
8
6
4
2
0
2
4
6
8
x_1
$4x_1 + 2x_2 = 15$
optimal IP solution: $x_1=2$, $x_2=3$ obj = 12
optimal LP solution: $x_1=2.5$, $x_2=2.5$ obj = 12.5
$x_1 + 2x_2 = 8$
$x_1 + x_2 = 5$

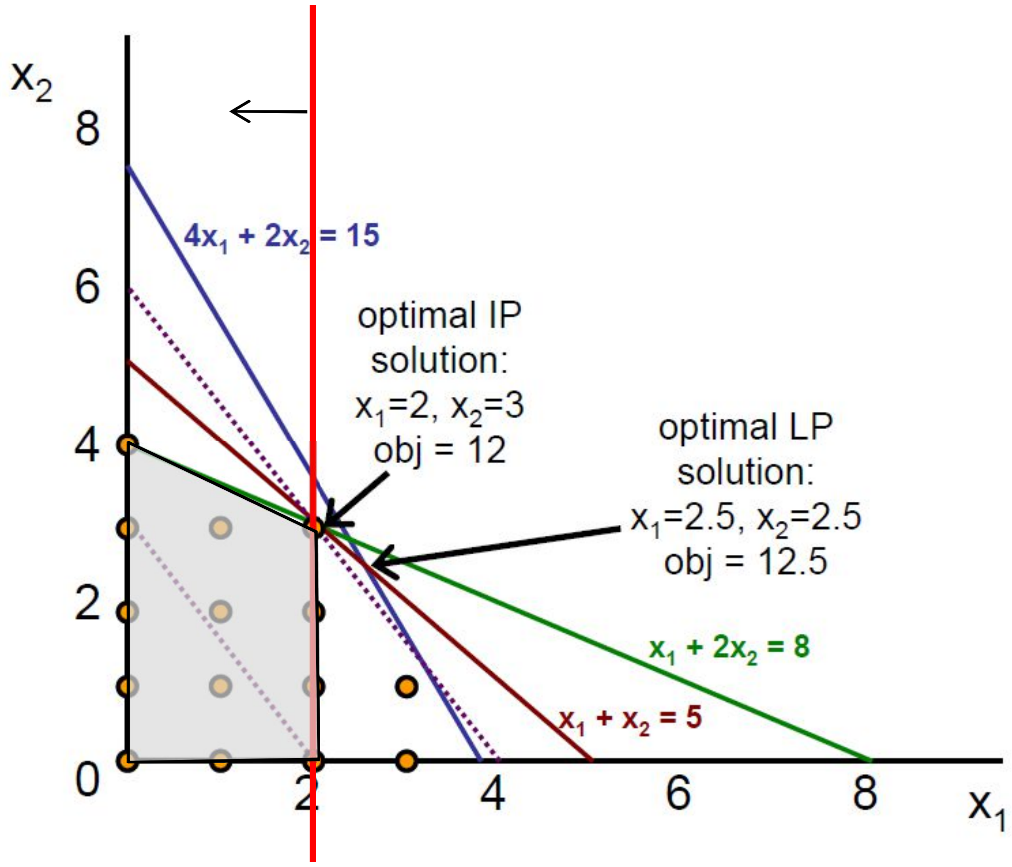
This solution is integer (by far, the only one), so we don't need to branch this problem further…however, the search process has not ended yet…☹

Example
solving P12- optimal solution

$$\begin{aligned}&\text{maximize } 3x_1 + 2x_2\\&\text{subject to}\\&4x_1 + 2x_2 \leq 15\\ P_{12} \quad &x_1 + 2x_2 \leq 8\\&x_1 + x_2 \leq 5\\&\boxed{x_1 \leq 2}\\&x_1 \geq 0, \text{ integer; } x_2 \geq 0, \text{ integer}\end{aligned}$$

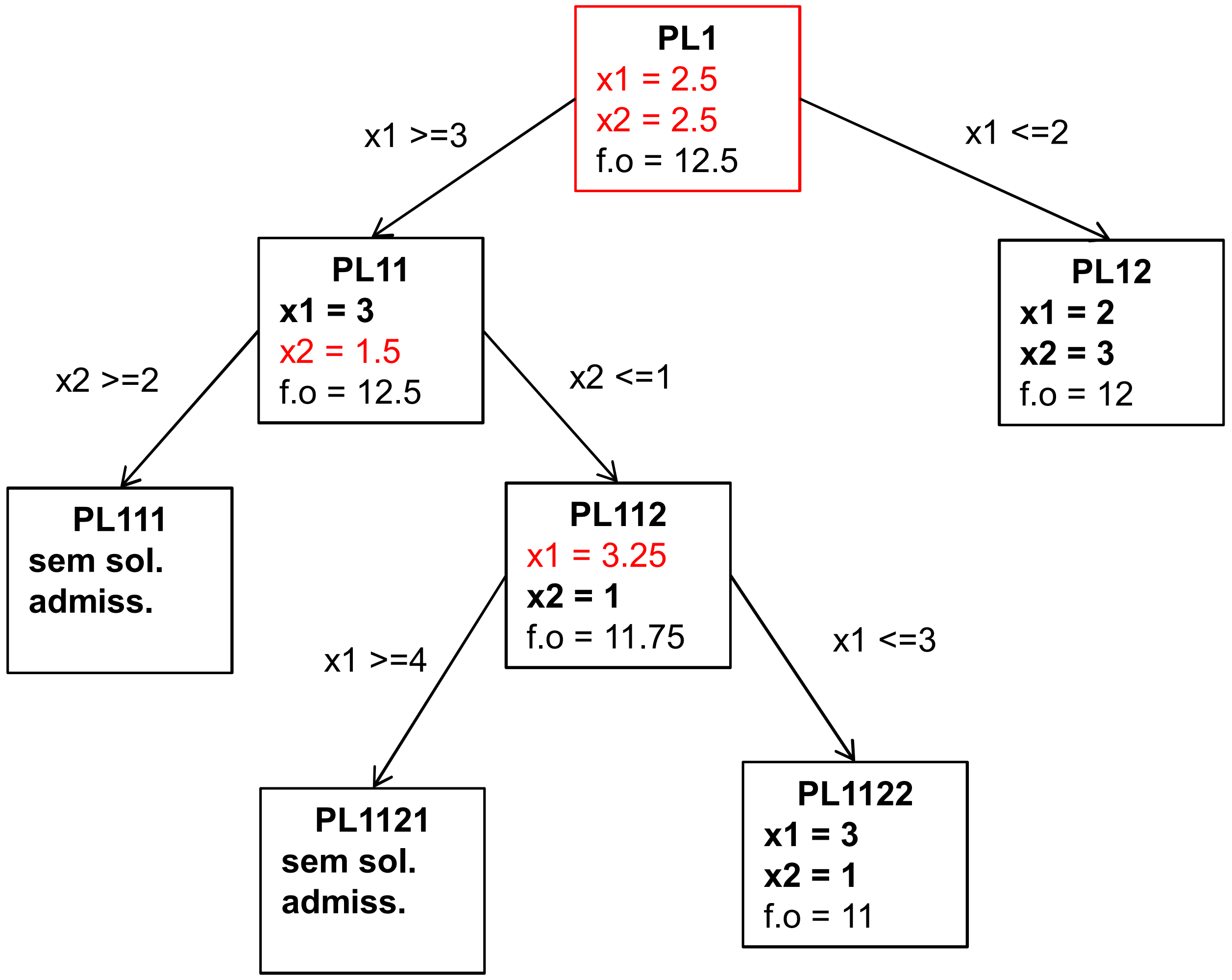
Solving the linear relaxation:

X1 = 2
X2 = 3
f.o = 12



The LP solution is integer, so we don't need to further branch the problem. The solution is better than the other intger solution and we don't have more problems to analyse. Hence this solution is optimal ☺

Branch & Bound search tree


PL1
x1 = 2.5
x2 = 2.5
f.o = 12.5
x1 >=3
x1 <=2
PL11
x1 = 3
x2 = 1.5
f.o = 12.5
PL12
x1 = 2
x2 = 3
f.o = 12
x2 >=2
x2 <=1
PL111
sem sol. admiss.
PL112
x1 = 3.25
x2 = 1
f.o = 11.75
x1 >=4
x1 <=3
PL1121
sem sol. admiss.
PL1122
x1 = 3
x2 = 1
f.o = 11