Integer Linear Programming

• Integer L P problem

(P) Min
$$\sum_{j=1}^{n} c_j x_j$$

s. t. $\sum_{j=1}^{n} a_{ij} x_j = b_i$ $i = 1, \dots, m$
 $x_j \ge 0$, integer $j = 1, \dots, n$

- Exemple
 - $\begin{array}{ll} \text{Min } z = -x_1 5x_2\\ \text{s. t. } x_1 + 10x_2 \leq 20\\ x_1 &\leq 2\\ x_1, x_2 \geq 0, \text{integer} \end{array}$

• F(P) = feasible domain of P

• Integer L P problem

(P) Min
$$\sum_{j=1}^{n} c_j x_j$$

s. t. $\sum_{j=1}^{n} a_{ij} x_j = b_i$ $i = 1, \dots, m$
 $x_j \ge 0$, integer $j = 1, \dots, n$

• F(P) = feasible domain of P

Exemple ٠ Min $z = -x_1 - 5x_2$ s. t. $x_1 + 10x_2 \le 20$ ≤ 2 X_1 $x_1, x_2 \ge 0$, integer $|x_1 = 2|$ $x_1 + 10x_2 = 20$ 0 b C σ b $F(P) = \{(0,0), (1,0), (2,0), (0,1), (1,1), (2,1), (0,2)\}$

• Integer L P problem

$$(\overline{P}) \operatorname{Min} \sum_{j=1}^{n} c_{j} x_{j}$$

s. t.
$$\sum_{j=1}^{n} a_{ij} x_{j} = b_{i} \quad i = 1, \dots, m$$
$$x_{j} \ge 0, \qquad j = 1, \dots, n$$

• F(P) = feasible domain of P

• Exemple



• (\overline{P}) denote the relaxation of (P) where the integrality constraints are relaxed

• Integer L P problem

$$(\overline{P}) \operatorname{Min} \sum_{j=1}^{n} c_{j} x_{j}$$

s. t.
$$\sum_{j=1}^{n} a_{ij} x_{j} = b_{i} \quad i = 1, \dots, m$$
$$x_{j} \ge 0, \quad : j = 1, \dots, n$$

• F(P) = feasible domain of P

• (\overline{P}) denote the relaxtion of (P) where the integrality constraints are relaxed • Exemple



•

Exemple

• Integer L P problem

(P) Min
$$\sum_{j=1}^{n} c_j x_j$$

s. t. $\sum_{j=1}^{n} a_{ij} x_j = b_i$ $i = 1, \dots, m$
 $x_j \ge 0$, integer $j = 1, \dots, n$

• How to solve the problem? Why dont we solve the relaxed problem, and take the rounded solution?

s.t. $x_1 + 10x_2 \le 20$ ≤ 2 X_1 $x_1, x_2 \ge 0, 0$ $\cdot \cdot \cdot \cdot \cdot x_1 = 2$ $\bar{x}_1 + 10x_2 = 20$ FP Solution of the relaxed problem: (2, 9/5) et z = -11Rounded solution: (2, 1) et z = -7But (0, 2) is feasible with z = -10

Min $z = -x_1 - 5x_2$

Solution methods

 Foundamental principle
 Generate a set of linear constraints to be added to (*P*)



Solution methods

 Foundamental principle
 Generate a set of linear constraints to be added to (P) in order to generate a new problem (PR) such that

$$F\left(\overline{PR}\right) \subset F\left(\overline{P}\right)$$
$$F\left(PR\right) = F\left(P\right)$$

Furthermore, when solving the problem \overline{PR} , the variables take integer values, and hence it is optimal for (*P*).

• Exemple Min $z = -x_1 - 5x_2$ s.t. $x_1 + 10x_2 \le 20$ $x_1 \le 2$ $x_1 + 2x_2 \le 4$ $x_1, x_2 \ge 0$, integer $x_1 = 2$ $x_1 + 10x_2 = 20$ • $x_1 + 10x_2 = 20$

Cutting plane algorithm

- General statement of the cutting plane algorithm Introduce new linear constraints to the problem in order to reduce the feasible domain of the relaxed problem but keep all the feasible points of the problem.
- In this procedure, we solve a sequence of relaxed problems until an integer optimal solution is reached
- Any problem of the sequence is obtained by adding an additional linear constraint (a cut) to the current problem.

Consider the following integer linear programming problem:

(P) Min
$$\sum_{j=1}^{n} c_j x_j$$

s. t. $\sum_{j=1}^{n} a_{ij} x_j = b_i$ $i = 1, \dots, m$
 $x_j \ge 0$, integer, $j = 1, \dots, m$

How to generate a Gomory cut.

Let *B* be an optimal basis of (*P*), and let x_k be the basic variable in the i^{th} row of the optimal tableau that is not integer.

The optimal tableau is as follows:

basic
 r.h.s

 var.

$$x_1$$
 $x_2 \dots x_k \dots x_{j_1} \dots x_j \dots x_{j_m} \dots x_n - z$
 x_{j_1}
 t_{11}
 $t_{12} \dots 0 \dots 1 \dots t_{1j} \dots 0 \dots t_{1n}$
 $\overline{b_1}$
 x_k
 t_{i1}
 $t_{i2} \dots 1 \dots 0 \dots t_{ij} \dots 0 \dots t_{in}$
 $\overline{b_i}$
 x_k
 t_{i1}
 $t_{i2} \dots 0 \dots 0 \dots t_{ij} \dots 0 \dots t_{in}$
 $\overline{b_i}$
 x_{j_m}
 t_{m1}
 $t_{m2} \dots 0 \dots 0 \dots t_{mj} \dots 1 \dots t_{mn}$
 $\overline{b_m}$
 $-z$
 $\overline{c_1}$
 $\overline{c_2} \dots 0 \dots 0 \dots \overline{c_j} \dots 0 \dots \overline{c_n}$
 $\overline{c_n}$
 1
 $-\overline{z}$

The corresponding row of the tableau is as follows:

$$x_k + \sum_{j \in J} t_{ij} x_j = \overline{b}_i \tag{1}$$

where $J = \{j : j \text{ is an index of a non basic variable}\}$ and \overline{b}_i is not integer. The corresponding row of the tableau is as follows:

$$x_k + \sum_{j \in J} t_{ij} x_j = \overline{b}_i \tag{1}$$

where $J = \{j : j \text{ is an index of a non basic variable}\}$ and \overline{b}_i is not integer.

Let $\lfloor d \rfloor$ = the largest integer (floor value) $\leq d$.

Since
$$x_j \ge 0 \ \forall j$$
, then

$$\sum_{j \in J} \left\lfloor t_{ij} \right\rfloor x_j \le \sum_{j \in J} t_{ij} x_j$$

and consequently

$$x_{k} + \sum_{j \in J} \left\lfloor t_{ij} \right\rfloor x_{j} \le x_{k} + \sum_{j \in J} t_{ij} x_{j} = \overline{b}_{i}.$$
(2)

The cooresponding row of the tableau is as follows:

$$x_k + \sum_{j \in J} t_{ij} x_j = \overline{b}_i \tag{1}$$

where $J = \{j : j \text{ is an index of a non basic variable}\}$ and \overline{b}_i is not integer.

Let $\lfloor d \rfloor$ = the largest integer (floor value) $\leq d$.

Since
$$x_j \ge 0 \quad \forall j$$
, then

$$\sum_{j \in J} \left\lfloor t_{ij} \right\rfloor x_j \le \sum_{j \in J} t_{ij} x_j$$

and consequently

$$x_k + \sum_{j \in J} \left[t_{ij} \right] x_j \le x_k + \sum_{j \in J} t_{ij} x_j = \overline{b}_i.$$

$$(2)$$

If we account for the fact that the variables x_j must take integer values, it follows from (2) that

$$x_{k} + \sum_{j \in J} \left\lfloor t_{ij} \right\rfloor x_{j} \le \left\lfloor \overline{b}_{i} \right\rfloor.$$
(3)

Hence all solutions of (P) satisfy (3).

Now consider the relation obtained by substracting (1) from (3):

$$\sum_{j \in J} \left(\left\lfloor t_{ij} \right\rfloor - t_{ij} \right) x_j \le \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right)$$
(4)

Note that

$$\left(\left\lfloor t_{ij} \rfloor - t_{ij}\right) \le 0$$
 et $\left(\left\lfloor \overline{b}_{i} \rfloor - \overline{b}_{i}\right) < 0$

Since any solution of (P) satisfies (1) and (3), then it also satifies (4), and we can introduce this constraint in (P) without eliminating any solution of (P).

Now consider the relation obtained by substracting

(1) from (3):

$$\sum_{j \in J} \left(\left\lfloor t_{ij} \right\rfloor - t_{ij} \right) x_j \le \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right)$$
(4)

Note that

$$\left(\left\lfloor t_{ij} \rfloor - t_{ij}\right) \le 0$$
 et $\left(\left\lfloor \overline{b}_{i} \rfloor - \overline{b}_{i}\right) < 0$

Since any solution of (P) satisfies (1) and (3), then it also satifies (4),

and we can introduce this constraint in (P) without eliminating any solution of (P).

But the current optimal solution of the relaxed problem (\overline{P}) where $x_j = 0 \quad \forall j \in J$ does not satisfies constraint (4), and thus introducing (4) reduces the size of the feasible domain of the relaxed problem (\overline{P}) .

To continue the solution process, we introduce the constraint

$$\sum_{j \in J} \left(\left\lfloor t_{ij} \right\rfloor - t_{ij} \right) x_j \le \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right) \iff \sum_{j \in J} \left(\left\lfloor t_{ij} \right\rfloor - t_{ij} \right) x_j + x_{\tau} = \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right)$$

where x_{τ} is a slack variable having a cost equal to 0, in the last simplex tableau to generate a basic solution for the new problem by looking at x_{τ} as the basic variable in the new row of the tableau.



$$\sum_{j \in J} \left(\left\lfloor t_{ij} \right\rfloor - t_{ij} \right) x_j + x_{\tau} = \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right) \text{ in the last row}$$
$$x_{\tau} \text{ basic variable in the last row}$$

To continue the solution process, we introduce the constraint

$$\sum_{j \in J} \left(\left\lfloor t_{ij} \right\rfloor - t_{ij} \right) x_j \le \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right) \iff \sum_{j \in J} \left(\left\lfloor t_{ij} \right\rfloor - t_{ij} \right) x_j + x_{\tau} = \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right)$$

where x_{τ} is a slack variable having a cost equal to 0, in the last simplex tableau to generate a basic solution for the new problem by looking at x_{τ} as the basic variable in the new row of the tableau.

This basic solution is not feasible since $x_{\tau} = \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right) < 0.$

Then we continue the solution process using the dual simplex method.



To continue the solution process, we introduce the constraint

$$\sum_{j \in J} \left(\left\lfloor t_{ij} \right\rfloor - t_{ij} \right) x_j \le \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right) \iff \sum_{j \in J} \left(\left\lfloor t_{ij} \right\rfloor - t_{ij} \right) x_j + x_{\tau} = \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right)$$

where x_{τ} is a slack variable having a cost equal to 0, in the last simplex tableau to generate a basic solution for the new problem by looking at x_{τ} as the basic variable in the new row of the tableau.

This basic solution is not feasible since $x_{\tau} = \left(\left\lfloor \overline{b}_i \right\rfloor - \overline{b}_i \right) < 0.$

Then we continue the solution process using the dual simplex method. Remarks:

1) If
$$\lfloor t_{ij} \rfloor = t_{ij}$$
 (i.e., t_{ij} is integer) $\forall j \in J$, et si \overline{b}_i is not integer, then
 $x_k + \sum_{j \in J} t_{ij} x_j = \overline{b}_i$ (1)

indicates that (P) is not feasible since the left hand side take an integer value for all feasible solution of (P) while the right hand element \overline{b}_i is not integer.

2) The same process is applied at each iteration.

Consider the following exemple Min $-21x_1 - 11x_2$ s.t. $7x_1 + 4x_2 + x_3 = 13$ $x_1, x_2, x_3 \ge 0$, integer





Iteration 2:





Iteration 3:

Solve the problem



Convergence of Gomory cutting plane algorithm

Reference:

A. Schrijver, Theory of Linear and Integer Programming, Wiley & Sons, 1986, 354 - 357

Under some assumptions for selecting the row of the tableau to specify the next cut, this author shows that: "... the cutting plane method terminates"

Solution methods

 Foundamental principle
 Generate a set of linear constraints to be added to (P) in order to generate a new problem (PR) such that

$$F\left(\overline{PR}\right) \subset F\left(\overline{P}\right)$$
$$F\left(PR\right) = F\left(P\right)$$

Furthermore, when solving the problem \overline{PR} , the variables take integer values, and hence it is optimal for (*P*).

• Exemple Min $z = -x_1 - 5x_2$ s.t. $x_1 + 10x_2 \le 20$ $x_1 \le 2$ $x_1 + 2x_2 \le 4$ $x_1, x_2 \ge 0$, integer $x_1 = 2$ $x_1 + 10x_2 = 20$ • $x_1 + 10x_2 = 20$

Branch & Bound method

(P) Min
$$\sum_{j=1}^{n} c_j x_j$$

(P) s.t. $\sum_{j=1}^{n} a_{ij} x_j = b_i$ $i = 1, \dots, m$
 $x_j \ge 0$, integer $j = 1, \dots, n$

Branch & Bound method

$$\begin{array}{ll} \text{Min} & \sum_{j=1}^{n} c_{j} x_{j} \\ (\overline{P}) & \text{s.t.} & \sum_{j=1}^{n} a_{ij} x_{j} = b_{i} \quad i = 1, \cdots, m \\ & x_{j} \geq 0, \qquad j = 1, \cdots, n \end{array}$$

- Here also we solve a sequence of relaxed problems.
- First we solve (\overline{P}) . If the optimal solution \overline{x} is integer, then this solution is optimal for the original problem (*P*).
- Otherwise we use a variable x_i having value \overline{x}_i which is not integer.
- Consider two new constraints

$$x_{i} \leq \left\lfloor \overline{x_{i}} \right\rfloor \quad (\text{floor of } x_{i})$$

and
$$x_{i} \geq \left\lceil \overline{x_{i}} \right\rceil \quad (\text{ceeling of } x_{i})$$





New constraints to consider:

$$x_1 \le \lfloor 3.75 \rfloor = 3$$

and
 $x_1 \ge \lfloor 3.75 \rfloor = 4$

With these two new constraints
feasible point of (*P*) are maintained
a slice of the feasible domain of the relaxed problem is eliminated.





• Next iteration

Select one of the two problems (P_2) or (P_3) Solve this selected problem as we did for (P).

• In our exemple, we select the problem (P_3)



• Next iteration

Select a problem in the set $\{P_2, P_4, P_5\}$ which have not been solved yet.

Solve this selected problem as we did for (P).

• In our exemple, we select the problem (P_5)



Since the optimal solution of the relaxed sub problem is integer, then it is a feasible solution of (P).

Do not generate new sub problems since we have identified the best feasible solution in this part of the feasible domain of (P).

During this solution process, we keep the best feasible integer solution found so far, and its value becomes an upper bound BS for the optimal value of (P).

• Next iteration

Select a problem in the set $\{P_2, P_4\}$ which have not been solved yet.

Solve this selected problem as we did for (P).

• In our exemple, we select the problem (P_4)



Stop looking for feasible solutions in the part since it is empty.

• Next iteration

Select a problem in the set $\{P_2\}$ which has not been solved yet. Solve this selected problem as we did for (*P*).

• Then we select the problem (P_2)



The optimal solution of the relaxed problem is not integer but its value z = -22.333 > BS = -23Stop looking for an integer solution in the part of the feasible domain of (*P*) since it is not possible to find one having a value smaller than the BS = -23.

- The procedure stops when all the relaxed problems have been solved.
- The integer solution having its value equal to *BS* is an optimal solution of (*P*).

Summary of the Branch & Bound approach

- Iterative approach.
- At each iteration,
 - a list of candidate problems is available to be solved. When the procedure starts, the list includes only the problem (*P*)
 - a candidate problem is selected, and the correponding relaxed problem is solved
 - the optimal solution of the relaxed problem allows to update the list of candidate problems or the upper bound and the best integer solution found so far.

The Branch & Bound method

• Initialisation

The list of candidate problem includes only the problem (*P*) $BS = \infty$ Go to Step 2.

• Step 2

Select the first candidate problem (PC) on the top of the list.

• Step 2

Select the first candidate problem (PC) on the top of the list.

• Step 3

To analyse (PC), solve the relaxed problem (\overline{PC}) . If $F(\overline{PC}) = \Phi$, go to Step1. If $v(\overline{PC}) \ge BS$, go to Step 1. If the optimal solution of (\overline{PC}) is integer, then if $v(\overline{PC}) < BS$, then $BS := v(\overline{PC})$, go to Step1. • Step 3

To analyse (PC), solve the relaxed problem (\overline{PC}) .

If
$$F(\overline{PC}) = \Phi$$
, go to Step1.

If
$$v(\overline{PC}) \ge BS$$
, go to Step 1.

If the optimal solution of
$$(\overline{PC})$$
 is integer, then
if $v(\overline{PC}) < BS$, then $BS := v(\overline{PC})$,
go to Step1.

• Step 4

Select a variable x_i which is not integer.

Generate a first new problem by including the constraint

 $x_j \ge \lceil \overline{x}_j \rceil$ to the problem (*PC*) and place it on the top of the list. Generate a second new problem by including the constraint

 $x_j \leq \lfloor \overline{x}_j \rfloor$ to the problem (*PC*), and place it on the top of the list. Go to Step 2.

• Step 2

Select the first candidate problem (PC) on the top of the list.

• Step 3

To analyse (PC), solve the relaxe problem (\overline{PC}) . If $F(\overline{PC}) = \Phi$, go to Step1. If $v(\overline{PC}) \ge BS$, go to Step 1. If the optimal solution of (\overline{PC}) is integer, then $if v(\overline{PC}) < BS$, then $BS := v(\overline{PC})$, go to Step1.

Enumeration tree of the Branch & Bound



How to search in the tree (how to select the next candidate problem)

- a) Depth first search (select the last generated candidate problem): decend rapidly in the tree in order to reach as soon as possible a feasible integer solution
- b) Search using the best node (using the best candidate problem):
 require to complete some iterations to solve the candidate
 problem at each node in order to identify the node having
 the best potentiel to improve the upper bound

Selection of the separating variable $x_j (x_j \ge \lceil \overline{x}_j \rceil, x_j \le \lfloor \overline{x}_j \rfloor)$ The variable x_j such that

a)
$$\overline{x}_{j} - \lfloor \overline{x}_{j} \rfloor$$
 is the largest
b) $\overline{x}_{j} - \lfloor \overline{x}_{j} \rfloor$ is the smallest
c) $\overline{x}_{j} - \lfloor \overline{x}_{j} \rfloor$ is closer to 0.5

Solving the relaxed candidate problem using the dual simplex method

The optimal solution of the current relaxation of the candidate problem (at the current node) is not feasible for the relaxed problem obtained by adding a constraint of the type $x_j \ge \lceil \overline{x}_j \rceil$ ou $x_j \le \mid \overline{x}_j \mid$.

Then we can use the dual simplex method to solve the relaxed candidate problem using the optimal solution of the current problem to generate the initial basic solution. Reference.

A. Atamturk, M.W.P. Savelsbergh, "Integer-Programming Software Systems", Annals of Operations Research 140, 67-124, 2005.