# Integer Linear Programs

Sometimes, we need variables to take integer values:

- The optimal solution of an LP can be fractional.
- In some cases, we can simply round.
- Rounding can be infeasible or suboptimal.
- An Integer Linear Program is an LP with the additional constraint that the solution should be integral.

# McDonald's Example I

McDonald's menu:

| | Hamburger | Big Mac | McChicken | Caesar Salad | French Fries |
|---|---|---|---|---|---|
| Total Cals | 250 | 770 | 360 | 190 | 230 |
| Fat Cals | 81 | 360 | 144 | 45 | 99 |
| Protein (g) | 31 | 44 | 14 | 27 | 3 |
| Sodium (mg) | 480 | 1170 | 800 | 580 | 160 |
| Cost | \$1.00 | \$3.00 | \$2.50 | \$3.00 | \$1.00 |

# McDonald's Example II

Our goal will be to minimize the cost of a meal, with the following constraints:

- Calories are between 600 and 900
- Less than 1150 mg of sodium.
- Less than 40% of calories are from fat.
- At least 30 grams of protein.

# McDonald's Example III

Decision Variables:

- $x_1$: number of hamburgers purchased
- $x_2$: number of big macs purchased
- $x_3$: number of McChickens purchased
- $x_4$: number of caesar salads purchased.
- $x_5$: number of french fries purchased.

Objective:

$$\min_{x \in \mathbb{R}^n} x_1 + 3x_2 + 2.5x_3 + 3x_4 + x_5$$

# McDonald's Example IV

Constraints:

- Calories between 600 and 900

$$250x_1 + 770x_2 + 360x_3 + 190x_4 + 230x_5 \geq 600$$
$$250x_1 + 770x_2 + 360x_3 + 190x_4 + 230x_5 \leq 900$$

- Less than 1150 mg of sodium

$$480x_1 + 1170x_2 + 800x_3 + 580x_4 + 160x_5 \leq 1150$$

- Less than 40% of calories are from fat

$$81x_1 + 360x_2 + 144x_3 + 45x_4 + 99x_5 \leq 0.4\left(250x_1 + 770x_2 + 360x_3 + 190x_4 + 230x_5\right)$$

- At least 30 grams of protein:

$$31x_1 + 44x_2 + 14x_3 + 27x_4 + 3x_5 \geq 30$$

# McDonald's Example V

Constraints:

- Order quantities are non-negative and integral:

$$x \in \mathbb{Z}^5$$
$$x \geq 0$$

# McDonald's Example VI

The LP solution:

| Menu Item | Quantity |
|---|---|
| Hamburger | 1.129207 |
| Big Mac | 0.412595 |
| McChicken | 0.0 |
| Caesar Salad | 0.0 |
| French Fries | 0.0 |

But no feasible integral solution that uses only hamburgers and Big Macs! Actual optimal solution:

| Menu Item | Quantity |
|---|---|
| Hamburger | 1 |
| Big Mac | 0 |
| McChicken | 0 |
| Caesar Salad | 0 |
| French Fries | 2 |

# MILPs I

Mixed Integer Linear Programs:

- Affine constraints.
- Linear objective function.
- Restrictions that some (or all) variables are integer.
- NP-Hard in general
- Currently, algorithms require exponential amount of time in worst-case.
- Very good commercial software
- Many instances tractable in practice.

Yes/no decisions can be represented by binary variables.

- Binary variables take values of 0 or 1

# MILPs II

Wide applications:

- Production scheduling
- Airline crew scheduling
- Sports scheduling
- Portfolio selection
- Telecommunication network design
- Design of radiation treatments
- Molecular biology

# MILPs III

In Gurobi:

- Creating an MILPs is almost the same as creating an LP; only difference is variable type.

```
my_var = my_model.addVar(lb=0.0,
  ub=gurobipy.GRB.INFINITY,
  vtype=gurobipy.GRB.INTEGER,
  name="myvariable")
```

- If variable should take values 0 or 1, then use BINARY type:

```
my_var = my_model.addVar(name="myvar",
  vtype=gurobipy.GRB.BINARY)
```

# Valid Formulations

- In an integer linear program, constraints describe a polyhedron.
- However, we are only interested in the integer points.
- A formulation describing a polyhedron $P$ is *valid* for a set of integral points $S$ if the integral points of $P$ are exactly equal to $S$
- There may be multiple valid formulations

# Selecting from a set I

Setting:

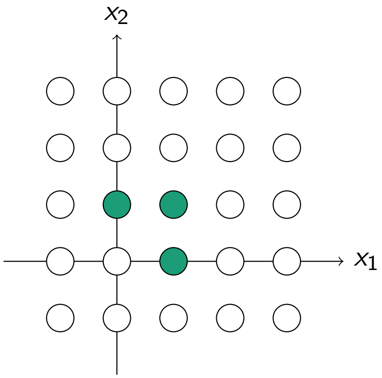
- There is a set $S$
- We want to choose at least one item from this set

Variables:

$$x_i = \begin{cases} 1 & \text{if element } i \text{ is chosen} \\ 0 & \text{otherwise.} \end{cases}$$

# Selecting from a set II

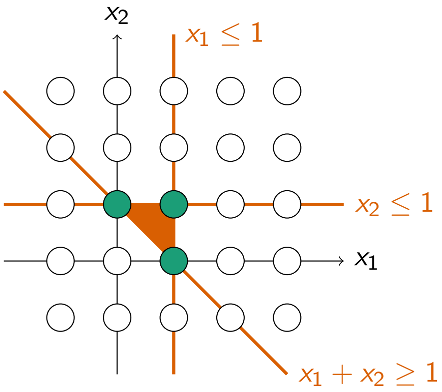
Feasible points (in 2 dimensions): $(0, 1)$, $(1, 0)$ and $(1, 1)$.



# Selecting from a set III

A possible formulation:

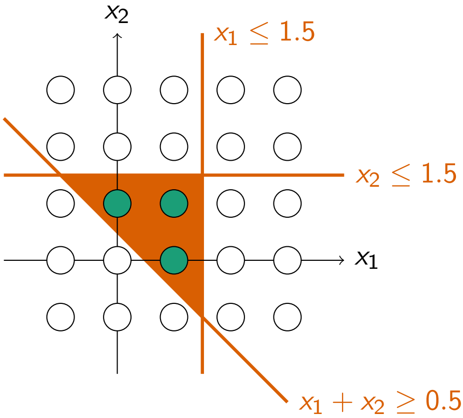
$$
\begin{aligned}
x_1 + x_2 &\geq 1 \\
x_1 &\leq 1 \\
x_2 &\leq 1 \\
x_1, x_2 &\text{ integral}
\end{aligned}
$$



# Selecting from a set IV

Another possible formulation:

$$\begin{aligned} x_1 + x_2 &\geq 0.5 \\ x_1 &\leq 1.5 \\ x_2 &\leq 1.5 \\ x_1, x_2 &\text{ integral} \end{aligned}$$



# Selecting from a set V

Usually, we don't list every feasible point. However, we do need to check:

- Every desired integral solution satisfies the constraints
- Every undesired integral solution does not satisfy the constraints

# Selecting from a set VI

If we have a set $S$ of size $k$:

$$\sum_{i=1}^{k} x_i \geq 1$$
$$x_i \leq 1 \text{ for } i \in \{1, \ldots k\}$$
$$x \text{ integral.}$$

is a formulation for selecting at least one element of $S$. Similar constructions can be used to select at most one element of $S$ or exactly one element of $S$.

# Selecting from a set VII

Notation: we would usually write this formulation as

$$\sum_{i=1}^{k} x_i \geq 1$$
$$x \text{ binary.}$$

or

$$\sum_{i=1}^{k} x_i \geq 1$$
$$x \in \{0,1\}^k.$$

# Assignment Problem I

Problem:

- There are $n$ people.
- There are $m$ jobs ($n \geq m$)
- Each job must be done by exactly one person
- Each person can do at most one job.
- Cost of person $i$ to do job $j$ is $c_{ij}$.

Goal: complete all jobs with minimal total cost.

# Assignment Problem II

Decision variables:

$$x_{ij} = \begin{cases} 1 & \text{if person } i \text{ assigned to job } j \\ 0 & \text{otherwise.} \end{cases}$$

Objective:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij}$$

# Assignment Problem III

Constraints:

- Every job is assigned to some person:

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \text{for } j = 1, \ldots, m$$

- Every person is assigned to at most one job:

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

- Variables are binary:

$$x_i \in \{0, 1\}$$

# Logical conditions.

Binary variables can enforce "if-then" relations between decisions.

- Suppose $x_1$ and $x_2$ are binary variables, with

$$x_i = \begin{cases} 1 & \text{if action } i \text{ is taken} \\ 0 & \text{otherwise.} \end{cases}$$

- The constraint $x_1 \leq x_2$ can be interpreted as "if action 1 is taken then action 2 must also be taken."
- Note that $(1 - x)$ takes a value of 1 if and only if action $x$ *not* taken.
- So, $(1 - x) \leq y$ enforces "if action $x$ is not taken, then action $y$ is taken".

# Portfolio design I

Stockco is considering four investments:

| Investment | Cost | NPV |
|---|---|---|
| 1 | \$5000 | \$16000 |
| 2 | \$7000 | \$22000 |
| 3 | \$4000 | \$12000 |
| 4 | \$3000 | \$8000 |

# Portfolio design II

There are some restrictions:

- Stockco only wants to invest in at most two opportunities.
- If Stockco invests in investment 2, they must also invest in investment 1.
- If Stockco invests in investment 2, they cannot invest in investment 4.

Stockco has $14000 to invest. How should they invest this money?

# Portfolio design III

Variables:

$$x_i = \begin{cases} 1 & \text{if Stockco invests in } i. \\ 0 & \text{otherwise.} \end{cases}$$

Objective:

$$\max 16000x_1 + 22000x_2 + 12000x_3 + 8000x_4$$

# Portfolio design IV

Constraints:

- At most two opportunities:

$$\sum_{j=1}^{4} x_j \leq 2$$

- If Stockco invests in investment 2, they must also invest in investment 1:

$$x_2 \leq x_1$$

- If Stockco invests in investment 2, they cannot invest in investment 4:

$$x_2 \leq 1 - x_4$$

# Portfolio design V

- Stockco has \$14000 to invest:

$$5000x_1 + 7000x_2 + 4000 + x_3 + 3000x_4 \leq 14000$$

- Variables are binary:

$$x_i \in \{0, 1\}$$

# Modeling fixed costs.

Actions may have fixed costs and variable costs. Binary variables can be used to model fixed costs.

# Facility Location I

Problem:

- There are $n$ possible facility locations
- There are $m$ customers
- Cost $c_j$ associated with opening facility at location $j$
- Cost $d_{ij}$ to serve customer $i$ from facility $j$.
- Facility must be open to serve customer.

Goal: serve all customers at minimum cost.
Variables:

$$y_j = \begin{cases} 1 & \text{if facility } j \text{ is opened.} \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if customer } i \text{ served by facility } j. \\ 0 & \text{otherwise.} \end{cases}$$

# Facility Location II

Objective:

$$\sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij}$$

# Facility Location III

Constraints:

- All customers served by exactly one facility:

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ for all } i, j$$

- A customer can only be served by an open facility:

$$x_{ij} \leq y_j \qquad \text{for all } i, j$$

Alternatively:

$$\sum_{i=1}^{m} x_{ij} \leq m y_j \qquad \text{for all } j$$

# Facility Location IV

- Variables are non-negative and integer:

$$
\begin{aligned}
x, y &\geq 0 \\
x_{ij} &\in \mathbb{Z} \\
y_j &\in \mathbb{Z}
\end{aligned}
$$

# Disjunctions I

Suppose we have $n$ variables, each with some constant upper bound:

$$0 \leq x_i \leq u_i \quad i \in \{1, \ldots, n\}$$

Suppose that we have $k$ sets of constraints (Note: superscripts are indices, not exponents).

$$A^1 x \leq b^1, \quad A^2 x \leq b^2, \quad \ldots, \quad A^k x \leq b^k$$

Suppose at least one of these sets of constraints must be satisfied:

$$A^1 x \leq b^1 \text{ or } A^2 x \leq b^2 \text{ or } \ldots \text{ or } A^k x \leq b^k$$

In other words, we want a feasible region that is the union

$$\cup_{i=1}^{k} \{x \mid A^i x \leq b^i\}$$

# Disjunctions II

We will add $k$ binary variables $y_1, \ldots, y_k$. We will want:

$$y_i = \begin{cases} 1 & \text{if we select constraint set } i \\ 0 & \text{otherwise} \end{cases}$$

We will also add $k$ vectors $x^1, x^2, \ldots, x^k$ each containing $n$ variables. We will want:

- $x = x^i$ and $A^i x^i \leq b$ if we select constraint set $i$
- $x^i = 0$ otherwise.

# Disjunctions III

The formulation:

$$
\begin{array}{rl}
\displaystyle\sum_{i=1}^{k} y_i = 1 & \text{select exactly one constraint set} \\
0 \le x_j^i \le u_j y_i & \text{if set } i \text{ is not selected, then } x^i = 0 \\
A^i x^i \le b^i y_i & \text{if set } i \text{ is selected, then } A^i x^i \le b^i \\
\displaystyle\sum_{i=1}^{k} x^i = x & \text{if set } i \text{ is selected, then } x = x^i
\end{array}
$$

# Example: production planning I

Consider a variant of the production planning problem:

- We make three different products using two materials: plastic and metal.
- It requires the following quantities of each material to make one unit each product:

| Product | Material 1 | Material 2 |
|---|---|---|
| 1 | 3 | 2 |
| 2 | 4 | 5 |
| 3 | 2 | 2 |

- Suppose that the per-unit revenues for products 1, 2, and 3 are \$12, \$25 and \$10 respectively.
- Material 1 costs \$1 per unit and Material 2 costs \$2 per unit.
- Our current budget is \$10000

How can we maximize profit?

# Example: production planning II

Decision variables:

- $x_i$: number of product $i$ produced
- $q_i$: quantity of material $i$ purchased.

Objective:

$$\max_{x,q} 12x_1 + 25x_2 + 10x_3 - q_1 - 2q_2$$

Constraints:

$$\begin{aligned} 3x_1 + 4x_2 + 2x_3 &\leq q_1 \\ 2x_1 + 5x_2 + 2x_3 &\leq q_2 \\ q_1 + 2q_2 &\leq 10000 \\ x, q &\geq 0 \end{aligned}$$

# Example: production planning III

Now suppose that we can upgrade our factory.

- This costs \$5000
- After the upgrade, the products require the following material quantities:

| Product | Material 1 | Material 2 |
|---|---|---|
| 1 | 2 | 1 |
| 2 | 3 | 3 |
| 3 | 1 | 2 |

- How can we formulate an MILP that choose both whether to upgrade or not and how much of each product to produce?

# Example: production planning IV

We can write the feasible region as a disjunction:

$$\begin{aligned} 3x_1 + 4x_2 + 2x_3 &\le q_1 \\ 2x_1 + 5x_2 + 2x_3 &\le q_2 \\ q_1 + 2q_2 &\le 10000 \\ x, y &\ge 0 \end{aligned} \quad \text{or} \quad \begin{aligned} 2x_1 + 3x_2 + 1x_3 &\le q_1 \\ 1x_1 + 3x_2 + 2x_3 &\le q_2 \\ q_1 + 2q_2 &\le 5000 \\ x, q &\ge 0 \end{aligned}$$

Hang on, don't we need upper bounds for $x$ and $q$?

# Example: production planning V

Upper bounds for $x$ and $q$:

- Many formulations do not include explicit upper bounds for variables involved in a disjunction.
- We can often find an upper bound from the other constraints.
- E.g. in this problem

$$\text{Either } q_1 + 2q_2 \leq 10000 \text{ or } q_1 + 2q_2 \leq 5000$$

- So, $q_1 \leq 10000$ and $q_2 \leq 5000$
- Using similar reasoning, we can figure out that $x_1 \leq 5000$, $x_2 \leq 5000/3$ and $x_3 \leq 2500$
- It's not necessary to find the tightest possible upper bounds, but tighter upper bounds usually lead to faster solve times.

# Example: production planning VI

Applying our recipe, we will add variables:

$$y_1 : \text{no upgrade}$$
$$y_2 : \text{upgrade}$$

and the variables:

$$x_i^1 : \text{quantity of product } i \text{ produced if no upgrade}$$
$$x_i^2 : \text{quantity of product } i \text{ produced if upgrade}$$
$$q_i^1 : \text{quantity of material } i \text{ purchased if no upgrade}$$
$$q_i^2 : \text{quantity of material } i \text{ purchased if upgrade}$$

# Example: production planning VII

Constraint: choose exactly one of the two options.

$$y_1 + y_2 = 1$$

Constraint: If we don't choose option, then corresponding solution is exactly zero:

$$\begin{aligned} 0 &\leq x_i^1 \leq My_1 \\ 0 &\leq x_i^2 \leq My_2 \\ 0 &\leq q_i^1 \leq My_1 \\ 0 &\leq q_i^2 \leq My_2 \end{aligned}$$

The "Big-$M$"s are a shorthand for an constant or upper bound that needs to be calculated.

# Example: production planning VIII

Constraints: the solutions correspond to each option must satisfy the corresponding constraints:

$$\begin{aligned} 3x_1^1 + 4x_2^1 + 2x_3^1 &\leq q_1^1 \\ 2x_1^1 + 5x_2^1 + 2x_3^1 &\leq q_2^1 \\ q_1^1 + 2q_2^1 &\leq 10000 \\ x, y &\geq 0 \end{aligned} \qquad \text{and} \qquad \begin{aligned} 2x_1^2 + 3x_2^2 + 1x_3^2 &\leq q_1^2 \\ 1x_1^2 + 3x_2^2 + 2x_3^2 &\leq q_2^2 \\ q_1^2 + 2q_2^2 &\leq 5000 \\ x, q &\geq 0 \end{aligned}$$

Constraint: the production and purchase quantities must match one of the options:

$$\begin{aligned} x &= x^1 + x^2 \\ q &= q^1 + q^2 \end{aligned}$$

# Example: production planning IX

Objective:

$$\max_{x,q,y} 12x_1 + 25x_2 + 10x_3 - q_1 - 2q_2 - 5000y_2$$

# Example: production planning X

Note: we could have written two separate MILPs and solved them separately.

- This is always an option with disjunctions.
- If you have only one disjunction, it is probably better to treat each option as its own IP.
- If you have many disjunctions, the total number of options is the product of the number of options in each disjunction.
- So, with many disjunctions, it is usually better to have one integrated MILP.

# Example: production planning XI

Note: these formulations introduce a lot of variables.

- Try to formulate constraints using only the original variables first.
- Only use a disjunction formulation when necessary.
- E.g. consider the condition that at least one element in a set is chosen. This can be written as:

$$x_1 \geq 1 \text{ or } x_2 \geq 1 \text{ or } \ldots \text{ or } x_n \geq 1$$

  but it is better to simply use the constraint

$$\sum_{i=1}^{n} x_i \geq 1$$