Mixed and Integer linear programming

Types of Optimization

Linear Programming – objective function and constraints are linear.

Integer Linear Programming – objective function and constraints are linear but decision variables must be integers.

Mixed Integer Linear Programming – same as ILP with only some decision variables restricted to integers.

Non-linear Programming – objective function and constraints continuous but not all linear

Types of Optimization

~~Linear Programming – objective function and constraints are linear.~~

Integer Linear Programming – objective function and constraints are linear but decision variables must be integers.

Mixed Integer Linear Programming – same as ILP with only some decision variables restricted to integers.

Non-linear Programming – objective function and constraints continuous but not all linear

ILP

Sometimes linear programming is used to estimate ILP problems (if arrive with integers like chair example, this is optimal! Sometimes we can round)

However, this is not always best

- Can produce suboptimal solutions (see next slide….)

Example: Veerman Furniture Company (small change to Fab)

	Hours per unit			
Department	Chairs	Desks	Tables	Hours Avail
Fabrication	4	6	2	***1800***
Assembly	3	5	7	2400
Shipping	3	2	4	1500
Demand potential	360	300	100	
Profit	$15	$24	$18	

Chairs	Desks	Tables
0	266.67	100

Objective function is $8200

Note: if we "round", we will get an infeasible solution (4*0 + 6*267+2*100 = 1802)!!

ILP

Sometimes linear programming is used to estimate ILP problems (if arrive with integers like chair example, this is optimal! Sometimes we can round)

However, this is not always best

- Can produce suboptimal solutions

Moving from LP to ILP is a big step

- Need different algorithm (common algorithm: Branch and Cut….which starts with LP)
- More constraints (only integers)

Example: Veerman Furniture Company

	Hours per unit			
Department	Chairs	Desks	Tables	Hours Avail
Fabrication	4	6	2	1850
Assembly	3	5	7	2400
Shipping	3	2	4	1500
Demand potential	360	300	100	
Profit	$15	$24	$18	

```
c = m.addVar(vtype=GRB.INTEGER, name="Chair")
d = m.addVar(vtype=GRB.INTEGER, name="Desk")
t = m.addVar(vtype=GRB.INTEGER, name="Table")
m.setObjective(15*c+24*d+18*t, GRB.MAXIMIZE)
m.addConstr(4*c + 6*d + 2*t <= 1850, "Fabrication")
m.addConstr(3*c + 5*d + 7*t <= 2400, "Assembly")
m.addConstr(3*c + 2*d + 4*t <= 1500, "Shipping")
m.addConstr(c <= 360, "DemandC")
m.addConstr(d <= 300, "DemandD")
m.addConstr(t <= 100, "DemandT")
m.optimize()
for v in m.getVars():
    print('%s %g' % (v.varName, v.x)) #v.varName = Chair, Desk, Table
                                      #v.x = optimized value for that variable
print(m.status)
```

Binary Choice Models

Binary Choice Models are a form of ILP

Further restrict variables to be binary (0 or 1)

2 Common Binary Choice Models:

- Capital Budget Problem
- Set Covering Problem

Binary Choice Models

Binary Choice Models are a form of ILP

Further restrict variables to be binary (0 or 1)

2 Common Binary Choice Models:

- ***Capital Budget Problem***
- Set Covering Problem

Companies that want to have projects within a given year, but there is only a certain allocated budget to do a subset of these projects. How do we choose the most optimal subset of projects?

Example: Marr Corporation

Has ***$160 Million*** for capital projects

Five new projects to consider

1. New Information System
2. License New Technology from Other Firms
3. Build State-of-Art Recycling Facility
4. Install an Automated Machining Center in Production
5. Move Receiving Department to New Facility

Each Project has a cost and a projected Net Present Value (NPV) over the life of the project

Either a project is selected or not selected (no partial projects), therefore this is a binary choice model

Information

	Project 1	Project 2	Project 3	Project 4	Project 5
NPV	10	17	16	8	14
Expenditure	48	96	80	32	64

We want to maximize NPV.

Information

	Project 1	Project 2	Project 3	Project 4	Project 5
NPV	10	17	16	8	14
Expenditure	48	96	80	32	64

So, we are trying to Maximize NPV!

Maximize: $NPV = 10P_1 + 17P_2 + 16P_3 + 8P_4 + 14P_5$
Constraint: $48P_1 + 96P_2 + 80P_3 + 32P_4 + 64P_5 \leq 160$

$P_1,\ldots,P_5$ are binary --- 1 if project is done, 0 if it is not done

```python
m = Model("Marrs")

# Create variables
P1 = m.addVar(vtype=GRB.BINARY, name="Info Syst")
P2 = m.addVar(vtype=GRB.BINARY, name="New Tech")
P3 = m.addVar(vtype=GRB.BINARY, name="Recycle")
P4 = m.addVar(vtype=GRB.BINARY, name="Machine Center")
P5 = m.addVar(vtype=GRB.BINARY, name="Receiving")


# Set objective
m.setObjective(10*P1 + 17*P2 + 16*P3 + 8*P4 + 14*P5, GRB.MAXIMIZE)

# Add constraints
m.addConstr(48*P1 + 96*P2 + 80*P3 + 32*P4 + 64*P5 <= 160, "Money")

m.optimize()

for v in m.getVars():
   print('%s %g' % (v.varName, v.x))
print('Obj: %g' % m.objVal)
print(m.status)
```

Output

Careful with this!!

Solution count 2: 34 27

Optimal solution found (tolerance 1.00e-04)
Best objective 3.400000000000e+01, best
bound 3.400000000000e+01, gap 0.0000%
Info Syst 1
New Tech 0
Recycle 1
Machine Center 1
Receiving 0
Obj: 34
Convergence Status is: 2

Binary Decision Variables

Binary Choice Models

- ~~Capital Budget problem~~
- Set Covering problem

Need to make sure that an area is "covered" by available units. For example, how many EMS stations are needed to cover Metropolis City?

Set Covering Example

Emergency Coverage in Metropolis

- Metropolis city is divided into 9 districts
- 7 potential sites for emergency vehicles
- Sites can reach some districts, but not others, in the required 3 minutes response time
- Location of these sites MUST cover all districts (would like to have the least amount of sites that can accomplish this)

Set Covering Information

District	Site 1	Site 2	Site 3	Site 4	Site 5	Site 6	Site 7
1	0	1	0	1	0	0	1
2	1	0	0	0	0	1	1
3	0	1	0	0	0	1	1
4	0	1	1	0	1	1	0
5	1	0	1	0	1	0	0
6	1	0	0	1	0	1	0
7	1	0	0	0	0	0	1
8	0	0	1	1	1	0	0
9	1	0	0	0	1	0	0

Want to minimize the number of sites while making sure all districts are covered.

Set Covering example setup

Min: Sites = $S_1 + S_2 + S_3 + S_4 + S_5 + S_6 + S_7$

Subject to:

$S_2 + S_4 + S_7 \geq 1$

$S_1 + S_6 + S_7 \geq 1$

$S_2 + S_6 + S_7 \geq 1$

$S_2 + S_3 + S_5 + S_6 \geq 1$

$S_1 + S_3 + S_5 \geq 1$

$S_1 + S_4 + S_6 \geq 1$

$S_1 + S_7 \geq 1$

$S_3 + S_4 + S_5 \geq 1$

$S_1 + S_5 \geq 1$

```
m=Model('Set Covering')
S1=m.addVar(vtype=GRB.BINARY, name="S1")
S2=m.addVar(vtype=GRB.BINARY, name="S2")
S3=m.addVar(vtype=GRB.BINARY, name="S3")
S4=m.addVar(vtype=GRB.BINARY, name="S4")
S5=m.addVar(vtype=GRB.BINARY, name="S5")
S6=m.addVar(vtype=GRB.BINARY, name="S6")
S7=m.addVar(vtype=GRB.BINARY, name="S7")
m.setObjective(S1 +S2 +S3+S4+S5+S6+S7, GRB.MINIMIZE)
m.addConstr(S2 + S4 + S7 >=1, "District 1")
m.addConstr(S1 + S6 + S7 >=1, "District 2")
m.addConstr(S2 + S6 + S7 >=1, "District 3")
m.addConstr(S2 + S3 + S5 + S6 >=1, "District 4")
m.addConstr(S1 + S3 + S5 >=1 , "District 5")
m.addConstr(S1 + S4 + S6 >=1, "District 6")
m.addConstr(S1 + S7 >=1, "District 7")
m.addConstr(S3 + S4 + S5 >=1, "District 8")
m.addConstr(S1 + S5 >=1, "District 9")
m.optimize()
print('Convergence status is ',m.status)
for v in m.getVars():
   print('%s %g' % (v.varName, v.x))
print('Obj: %g' % m.objVal)
```

Optimal solution found (tolerance 1.00e-04)
Best objective 3.000000000000e+00, best bound 3.000000000000e+00, gap 0.0000%
Convergence status is 2
S1 0
S2 0
S3 0
S4 0
S5 1
S6 1
S7 1
Obj: 3

In Gurobi

```
m.setParam(GRB.Param.PoolSolutions, 1024)
m.setParam(GRB.Param.PoolGap, 0.1)
m.setParam(GRB.Param.PoolSearchMode, 2)
```

PoolSolutions sets the size of the number of solutions (I just put a REALLY big number)
PoolGap indicates the “gap” in the objective function (ensures you don’t go too far away…percent of the best solution)

Setting *PoolSearchMode* to 2 searches for more than one optimal solution

```
print('Number of solutions: ',m.SolCount)

print('Now showing all possible solutions and objective value')
for s in range(m.SolCount):
    m.setParam(GRB.Param.SolutionNumber, s)
    print('Objective value is:','%g ' % m.PoolObjVal)
    for q in m.getVars():
        print('%s %g' % (q.varName, q.Xn))
```

Objective value is: 3
S1 0
S2 0
S3 0
S4 0
S5 1
S6 1
S7 1
Objective value is: 3
S1 1
S2 1
S3 1
S4 0
S5 0
S6 0
S7 0
Objective value is: 3
S1 1
S2 0
S3 0
S4 0
S5 1
S6 0
S7 1

Logical Relationships

Binary variables can also be used to model if-then-else situations.

For example, here are 5 common relationships:

1. At least m items
2. At most n items
3. Exactly k items
4. Mutually exclusive items
5. Contingency based items

For example:

Marr Corporation

- Have \$160 million for capital projects over the year.
- Five new projects to consider.
- Each project has **cost** and projected **NPV** over life of project. Still trying to maximize NPV.
- Add 4 Restrictions:
  1. Must have at least one international project ($P2$ or $P5$).
  2. Only have the staff to support 2 projects total.
  3. Projects 4 & 5 have the same resources, so can't have both.
  4. Project 5 requires project 3 be selected.

Set up

MAX:

$$NPV = 10P_1 + 17P_2 + 16P_3 + 8P_4 + 14P_5$$

Subject to:

$48P_1 + 96P_2 + 80P_3 + 32P_4 + 64P_5 \leq 160$

$P_2 + P_5 \geq 1$

$P_1 + P_2 + P_3 + P_4 + P_5 = 2$

$P_4 + P_5 \leq 1$

$P_3 - P_5 \geq 0$

P_1, P_2, P_3, P_4, P_5 are all binary variables.

```
m = Model("Marrs")
P1 = m.addVar(vtype=GRB.BINARY, name="Info Syst")
P2 = m.addVar(vtype=GRB.BINARY, name="New Tech")
P3 = m.addVar(vtype=GRB.BINARY, name="Recycle")
P4 = m.addVar(vtype=GRB.BINARY, name="Machine Center")
P5 = m.addVar(vtype=GRB.BINARY, name="Receiving")
m.setObjective(10*P1 + 17*P2 + 16*P3 + 8*P4 + 14*P5, GRB.MAXIMIZE)
m.addConstr(48*P1 + 96*P2 + 80*P3 + 32*P4 + 64*P5 <= 160, "Money")
m.addConstr(P2 + P5 >= 1, "International")
m.addConstr(P1 + P2 + P3 + P4 + P5 == 2, "ShortStaff")
m.addConstr(P4 + P5 <= 1, "CommonResources")
m.addConstr(P3 - P5 >= 0, "ProjectDependent")
m.optimize()
print('Convergence status is ',m.status)
for v in m.getVars():
   print('%s %g' % (v.varName, v.x))
print('Obj: %g' % m.objVal)
```

Output

Optimal solution found (tolerance 1.00e-04)
Best objective 3.000000000000e+01, best bound 3.000000000000e+01, gap 0.0000%
Convergence status is 2
Info Syst 0
New Tech 0
Recycle 1
Machine Center 0
Receiving 1
Obj: 30

Linking Constraints & Fixed Costs

Up until this point we have discussed variable costs, but some decisions also have **fixed costs** that have to be accounted for.

Fixed Cost

- One time cost that is incurred when any amount of an item is made.
- Binary variables are used to account for fixed costs.
- The example we will be using will be a facility example:
  - ONLY incur a fixed cost if facility is being used (binary variable indicating if facility is used)
  - Can ONLY have “variable cost” coming from that facility if it is in use (think about producing products from that facility or shipping products to or from that facility)

Linking Constraints & Fixed Costs

Fixed Cost – binary variable

Variable Cost – continuous variable

x = level of production
$y = 0$ if $x = 0$
$y = 1$ if $x > 1$

$$Cost = \underbrace{F}_{\text{Fixed Cost}} * y + \underbrace{C}_{\text{Variable Cost}} * x$$

Linking Constraints & Fixed Costs

Fixed Cost – binary variable

Variable Cost – continuous variable

$$Cost = F * y + C * x$$

Binary Variable

Continuous Variable

Linking Constraints & Fixed Costs

- How do we invoke the restriction that if y = 1 then x > 0, but if y = 0 then x = 0?

same logic without the if statement

$$x \leq Max * y$$

if y=0, x<=0

x>=0 (x can only be 0)

Upper bound on x

Fixed Cost Example

Mayhugh Manufacturing Company

- Can produce up to 3 product families.
- Each product family requires production hours in 3 different departments.
- Each product family requires its own sales force **no matter how large the sales volume**.

Fixed Cost Example

Department	Product Family 1	Product Family 2	Product Family 3	Hours Available
A	3	4	8	2,000
B	3	5	6	2,000
C	2	3	9	2,000
Sales Force Cost	60	200	100	
Maximum Demand	400	300	50	
Profit	1.2	1.8	2.2	

Maximize profit. Note: the Sales force cost is NOT included in the profit (need to take this under consideration in optimization).

Fixed Cost Example

- MAX:

$$Profit = 1.2F_1 + 1.8F_2 + 2.2F_3 - 60FY_1 - 200FY_2 - 100FY_3$$

- Subject to:

$$3F_1 + 4F_2 + 8F_3 \leq 2000$$

$$3F_1 + 5F_2 + 6F_3 \leq 2000$$

$$2F_1 + 3F_2 + 9F_3 \leq 2000$$

$$F_1 - 400FY_1 \leq 0$$

$$F_2 - 300FY_2 \leq 0$$

$$F_3 - 50FY_3 \leq 0$$

- FY_1, FY_2, FY_3 are all binary variables.

In Gurobi

```
m=Model("Fixed Cost")
F1=m.addVar(vtype=GRB.CONTINUOUS,lb=0, name='F1')
F2=m.addVar(vtype=GRB.CONTINUOUS, lb=0,name='F2')
F3=m.addVar(vtype=GRB.CONTINUOUS, lb=0,name='F3')
y1=m.addVar(vtype=GRB.BINARY,name='y1')
y2=m.addVar(vtype=GRB.BINARY,name='y2')
y3=m.addVar(vtype=GRB.BINARY,name='y3')
m.setObjective(1.2*F1 + 1.8*F2 + 2.2*F3 - 60*y1 - 200*y2 - 100*y3,GRB.MAXIMIZE)
m.addConstr(3*F1 + 4*F2 + 8*F3 <= 2000,'Department A')
m.addConstr(3*F1 + 5*F2 + 6*F3 <= 2000,'Department B')
m.addConstr(2*F1 + 3*F2 + 9*F3 <= 2000,'Department C')
m.addConstr(F1 - 400*y1 <= 0,'Fixed Cost 1')
m.addConstr(F2 - 300*y2 <= 0,'Fixed Cost 2')
m.addConstr(F3 - 50*y3 <= 0,'Fixed Cost 3')
m.optimize()
print('Convergence status is ',m.status)
for v in m.getVars():
   print('%s %g' % (v.varName, v.x))
print('Obj: %g' % m.objVal)
```

Output

Optimal solution found (tolerance 1.00e-04)
Best objective 5.080000000000e+02, best bound 5.080000000000e+02, gap 0.0000%
Convergence status is 2
F1 400
F2 160
F3 0
y1 1
y2 1
y3 0
Obj: 508

```
coef_F=np.array([1.2,1.8,2.2])
coef_y=np.array([-60,-200,-100])
F_const=np.array([[3,4,8],[3,5,6],[2,3,9]])
y_const=np.array([400,300,50])
m=Model("Fixed Cost")
dec_len=range(3)
F,y = {}, {}
for j in dec_len:
    F[j] = m.addVar(obj=coef_F[j], vtype=GRB.CONTINUOUS, lb=0,name="F[%s]"%j)
for i in dec_len:
    y[i] = m.addVar(obj=coef_y[i], vtype=GRB.BINARY, name="y[%s]"%i)
m.ModelSense = GRB.MAXIMIZE
for i in range(F_const.shape[0]):
    m.addConstr((quicksum(F[j]*F_const[i][j] for j in dec_len)) <= 2000,name="Dept[%s]"%i )
    m.addConstr(F[i] - (y_const[i]*y[i]) <= 0, name="Fixed[%s]"%i )

m.optimize()
print('Convergence status is ',m.status)

for v in m.getVars():
    print('%s %g' % (v.varName, v.x))
print('Obj: %g' % m.objVal)
```

Output

Family1: 400
Family2: 160
Family3: 0
Family1-Binary: 1
Family2-Binary: 1
Family3-Binary: 0

Obj: 508

Another example: Facility Location Model

Goal is to set up an optimal number of supply locations and shipping schedules.

We are deciding supply locations as well as shipping schedules.

Conceivable to think these supply locations must remain for the foreseeable future.

Facility Location Model Example

Levinson Foods Company

- 10 distribution centers with monthly volumes.
- 6 of these locations are able to support warehouses.
- Warehouses have additional capacity, but monthly operating costs.
- Variable shipping costs between each location and potential warehouse is calculated.
- Want to develop the optimal shipping schedule as well as determine which locations to upgrade to warehouses.

Facility Location Model Example

Location	Alb	Boise	Dall	Denv	Hous	Okla	Phoe	Salt	SanA	Wich
Albuquerque	0	47	32	22	42.5	27	23	30	36.5	29.5
Dallas	32	79.5	0	39	12.5	10.5	50	63	13.5	17
Denver	21	42	39	0	51.5	31.5	40.5	24	47.5	26
Houston	42.5	91	12.5	51.5	0	23	58	72	10	31
Pheonix	23	49	50	40.5	58	49	0	32.5	50	52
San Antonio	36.5	83.5	13.5	47.5	10	24	50	66.5	0	32

Facility Location Model Example

Center / Warehouse	Volume	Capacity	Cost
Albuquerque (W)	3,200	16,000	$140,000
Boise	2,500		
Dallas (W)	6,800	20,000	$150,000
Denver (W)	4,000	10,000	$100,000
Houston (W)	9,600	10,000	$110,000
Oklahoma City	3,500		
Phoenix (W)	5,000	12,000	$125,000
Salt Lake City	1,800		
San Antonio (W)	7,400	10,000	$120,000
Wichita	2,700		

Facility Location Model Example

Decision variables: Fixed cost variables: Warehouse (build or not build…binary)

Variable cost variables: each warehouse/distribution center combination

Minimize: Cost = variable cost + fixed cost

Subject to:

- Warehouse capacity: can’t send more than it can store
- Distribution center volume: need to send each distribution center its pre-specified volume

Gurobi (Python)

Sorry, does not fit on one slide….refer to code