

1. For each of the mathematical optimization models below, specify which of the following types of optimization models it is. In each model, x are the variables, all other letters refer to known data.

TYPES OF MODELS

- i. Linear program
- ii. Integer program
- iii. Quadratic program
- iv. Quadratic integer program
- v. Quadratically-constrained program
- vi. Quadratically-constrained integer program
- vii. Quadratic, quadratically-constrained program
- viii. Quadratic, quadratically-constrained integer program
- ix. None of the above

a. minimize
$$\sum_i c_i x_i$$

$$\begin{array}{ll} \text{subject to} & \sum_i a_{ij} x_i \leq b_j \text{ for all } j \\ & \text{all } x_i \geq 0 \end{array}$$

b. minimize
$$\sum_i c_i x_i$$

$$\begin{array}{ll} \text{subject to} & \sum_i a_{ij} x_i \leq b_j \text{ for all } j \\ & \text{all } x_i \in \{0,1\} \end{array}$$

c. minimize
$$\sum_i c_i x_i^2$$

subject to
$$\sum_i a_{ij} x_i \le b_j$$
 for all j all $x_i \ge 0$

d. minimize
$$\sum_i c_i x_i$$

subject to
$$\sum_i a_{ij} x_i^2 \le b_j$$
 for all j all $x_i \ge 0$

e. minimize
$$\sum_i c_i^2 x_i$$

subject to
$$\sum_i a_{ij} x_i \leq b_j$$
 for all j

all $x_i \ge 0$

f. minimize
$$\sum_i c_i x_i^2$$
 subject to $\sum_i a_{ij} x_i^2 \leq b_j$ for all j all $x_i \geq 0$

g. minimize
$$\sum_i c_i x_i$$
 subject to $\sum_i \sum_j a_{ij} x_i x_j \leq b$ all $x_i \in \{0,1\}$

h. minimize
$$\sum_i c_i x_i^2$$
 subject to $\sum_i a_{ij} x_i \leq b_j$ for all j all $x_i \geq 0$, integer

i. minimize
$$\sum_i c_i x_i^2$$
 subject to $\sum_i a_{ij} x_i^2 \leq b_j$ for all j all $x_i \in \{0,1\}$

j. minimize
$$\sum_i c_i^2 x_i$$
 subject to $\sum_i a_{ij}^{1.5} x_i \leq b_j^3$ for all j all $x_i \in \{0,1\}$

k. minimize
$$\sum_i c_i x_i^3$$
 subject to $\sum_i a_{ij} x_i \leq b_j$ for all j all $x_i \geq 0$

I. minimize
$$\sum_i c_i^2 x_i$$
 subject to $\sum_i \sum_k a_{ik} x_i x_k \leq b_j$ for all j all $x_i \geq 0$, integer



2. For each of the gurobipy optimization models below, specify which of the following types of optimization models it is. In each model, *x* are the variables, all other letters refer to known data.

TYPES OF MODELS

i. Linear program

ii. Integer program

iii. Quadratic program

iv. Quadratic integer program

v. Quadratically-constrained program

vi. Quadratically-constrained integer program

vii. Quadratic, quadratically-constrained program

viii. Quadratic, quadratically-constrained integer program

ix. None of the above

a.

	<pre>x = model.addVars(range(1,N+1), lb=0, vtype=gp.GRB.CONTINUOUS, name="x")</pre>
0.0,000.00	<pre>model.setObjective(sum(c[i]*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
	<pre>model.addConstrs((sum(a[i,j]*x[i] for i in range(1,N+1)) <= b[j]) for j in range(1,M+1))</pre>



b.

Variables	x = model.addVars(range(1,N+1), vtype=gp.GRB.BINARY, name="x")
Objective	<pre>model.setObjective(sum(c[i]*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
	<pre>model.addConstrs((sum(a[i,j]*x[i] for i in range(1,N+1)) <= b[j]) for j in range(1,M+1))</pre>

c.

Variables	<pre>x = model.addVars(range(1,N+1), 1b=0, vtype=gp.GRB. CONTINUOUS, name="x")</pre>
0.0,000.00	<pre>model.setObjective(sum(c[i]*x[i]*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
0000	<pre>model.addConstrs((sum(a[i,j]*x[i] for i in range(1,N+1)) <= b[j]) for j in range(1,M+1))</pre>

d.

	<pre>x = model.addVars(range(1,N+1), lb=0, vtype=gp.GRB. CONTINUOUS, name="x")</pre>
0.0,000.00	<pre>model.setObjective(sum(c[i]*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
0000	$ \begin{tabular}{ll} model.addConstrs((sum(a[i,j]*x[i]*x[i] for i in range(1,N+1)) <= \\ b[j]) for j in range(1,M+1)) \\ \end{tabular} $

e.

	<pre>x = model.addVars(range(1,N+1), lb=0, vtype=gp.GRB.CONTINUOUS, name="x")</pre>
0.0,000.00	<pre>model.setObjective(sum((c[i]**2)*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
	<pre>model.addConstrs((sum(a[i,j]*x[i] for i in range(1,N+1)) <= b[j]) for j in range(1,M+1))</pre>

f.



Variables	<pre>x = model.addVars(range(1,N+1), lb=0, vtype=gp.GRB. CONTINUOUS, name="x")</pre>
Objective	<pre>model.setObjective(sum(c[i]*x[i]*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
Constraints	<pre>model.addConstrs((sum(a[i,j]*x[i]*x[i] for i in range(1,N+1)) <= b[j]) for j in range(1,M+1))</pre>

g.

Variables	x = model.addVars(range(1,N+1), vtype=gp.GRB.BINARY, name="x")
0.0,000.00	<pre>model.setObjective(sum(c[i]*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
	<pre>model.addConstr(sum(sum(a[i,j]*x[i]*x[j] for i in range(1,N+1) for j in range(1,M+1)) <= b[j])</pre>

h.

Variables	<pre>x = model.addVars(range(1,N+1), lb=0, vtype=gp.GRB.INTEGER, name="x")</pre>
0.0,000.00	<pre>model.setObjective(sum(c[i]*x[i]*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
	<pre>model.addConstrs((sum(a[i,j]*x[i] for i in range(1,N+1)) <= b[j]) for j in range(1,M+1))</pre>

i.

Variables	x = model.addVars(range(1,N+1), vtype=gp.GRB.BINARY, name="x")
Objective	<pre>model.setObjective(sum(c[i]*x[i]*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
00.100.01.110	$\label{eq:model_addConstrs} \begin{subarray}{ll} $$\operatorname{model.addConstrs}((\operatorname{sum}(a[i,j]*x[i]*x[i]) & for i in range(1,N+1)) <= \\ b[j]) & for j in range(1,M+1)) \end{subarray}$

j.



	<pre>x = model.addVars(range(1,N+1), lb=0, vtype=gp.GRB.BINARY, name="x")</pre>
Objective	<pre>model.setObjective(sum((c[i]**2)*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
Constraints	$\label{eq:model_addConstrs} \begin{subarray}{ll} $model.addConstrs((sum(a[i,j])*math.sqrt(a[i,j])*x[i] & for i in \\ $range(1,N+1))$ &<= b[j]*b[j]*b[j]) & for j in $range(1,M+1))$ \\ \end{subarray}$



k.

Variables	<pre>x = model.addVars(range(1,N+1), lb=0, vtype=gp.GRB.CONTINUOUS, name="x")</pre>
,	<pre>model.setObjective(sum(c[i]*x[i]*x[i]*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
Constraints	<pre>model.addConstrs((sum(a[i,j]*x[i] for i in range(1,N+1)) <= b[j]) for j in range(1,M+1))</pre>

I.

	<pre>x = model.addVars(range(1,N+1), lb=0, vtype=gp.GRB.INTEGER, name="x")</pre>
0.0,000.00	<pre>model.setObjective(sum((c[i]**2)*x[i] for i in range(1,N+1)), GRB.MINIMIZE)</pre>
	model.addConstrs((sum(a[i,k]*x[i]*x[k] for i in range(1,N+1) for k in range(1,N+1)) $\leq b[j]$) for j in range(1,M+1))

NOTES:		

This is an open education course, and we encourage faculty and students to use the exercises and materials included to help meet educational and instructional goals. Gurobi Optimization, LLC ("Gurobi") is the owner of the content (all right, title, and interest) of this course ("Content"). Content is not designed or intended for commercial use or for the sale of courses. Gurobi makes this Content available at no cost for educational and instructional use via faculty and student download and use of Content in courses offered at your institution/organization; provided that such use does not scale or otherwise result in the use of Content for monetary gain including in any other online course. Content is offered as-is and all risk of use resides with you. Gurobi makes no warranties or guarantees of a particular outcome and accepts no liability, direct, consequential, or otherwise, arising from the access to or use of the course and Content contained in it.