Optimization (Formulation) Examples:

1. Suppose that we wish to enclose a rectangular equipment yard using at most 80 meters of fencing. Formulate an optimization model to find the design of maximum area.

Decision variables:

X1 = length(m)

X2 = width (m)

Objective: maximize area = x1 \* x2 (non-linear programming)

s.t.

2x1 + 2x2 <= 80 (perimeter of fence)

X1, x2 >= 0 (non-negativity)

1. A large manufacturer of corn seed, Hybrids Unlimited, operates p = 10 facilities producing seeds of q = 20 hybrid corn varieties and distributes them to customers in t = 30 sales regions. After considerable effort, a variety of parameters have been estimated:

* the cost per bag of producing each hybrid at each facility
* the corn processing capacity of each facility in bushels
* the number of bushels of corn that must be processed to make a bag of each hybrid
* the number of bags of each hybrid demanded in each customer region
* the cost per bag of shipping each hybrid from each facility to each customer region

They want to know how to carry out production and distribution operations at minimum cost.

Indexed set:

i = facility (1, …, p)

j = hybrid corn varieties (1, …, q)

k = regions (1, …, t)

Data:

Cij = cost/bag of producing hybrid j at facility i

Ui = capacity (bushels) of facility i

Bj = Bushels of corn required to make a bag of hybrid j

Djk = demand (bags) of hybrid j in region k

Sijk = cost/bag to ship hybrid j from facility i to region k

Decision variables:

Xij = bags of hybrid j produced at facility i

Yijk = bags of hybrid j produced at facility i and shipped to region k

Objective Function: Minimize costs = production costs – shipping costs

Min cost =

s.t.

(demand for each hybrid/region combination)

(capacity for each facility)

(flow conservation – must produce in order to ship)

1. An agricultural mill manufactures feed for chickens. This is done by mixing several ingredients, such as corn, limestone, or alfalfa. The mixing is done in such a way that the feed meets certain levels for different types of nutrients, such as protein, calcium, carbohydrates, and vitamins. There are *m* ingredients and *n* nutrients, and the unit cost of ingredient *j* is *cj*. The amount of final product needed is *b* and *aij* is the amount of nutrient I present in a unit of ingredient *j*. The acceptable lower and upper limits of nutrient *i* in a unit of chicken feed are *li* and *ui*, respectively. Also, the mill cannot acquire more than *tj* units of ingredient *j*. Determine how to mix the ingredients so that cost is minimized.

Indexed Sets:

i = nutrient number (i = 1, …, n)

j = ingredient number (j=1, …, n)

Data:

cj = unit cost of ingredient j

aij = amount of nutrient i in the final product (per unit)

b = final product needed (amount)

li = lower limit of nutrient i in the final product (per unit)

ui = upper limit of nutrient i in the final product (per unit)

tj = upper limit on amount of ingredient j

Variables:

Xj = number of units of ingredient j used in the final product

Objective: min cost =

Constraints:

(Lower and upper limits on Nutrients)

(Upper limit on ingredients; supply constraint, Explicit constraint)

(Final product amount)

(Non-negativity)

1. A typical large university picks one of *n = 30* final exam periods for each of *m = 2000* class units on its main campus. Most exams involve one class section, but there are a substantial number of “unit exams” held at a single time for multiple sections. The main issue in exam scheduling is “conflicts,” instances where a student has more than one exam scheduled during the same time period. Conflicts burden both students and instructors because a makeup exam will be required in at least one of the conflicting courses. The university scheduling procedure begins by processing enrollment records to determine how many students are jointly enrolled in each pair of course units. The objective is to **minimize total conflicts** as it selects time periods for all class units.

Indexed sets:

j = exam period number (j = 1, …, n)

i = class unit number (i = 1, …, m)

Data:

Cii’ = number of students in both class unit i and class unit i’

Variables:

Xij = 1 if class unit i is assigned to exam period j; 0 otherwise

Objective: minimize conflicts between class unit I and class unit i’

Constraints:

(each class unit i must be assigned to exactly 1 exam period j)

Xij is binary