Linear programming modeling examples: Feed Mix Problem

Feed Mix Problem

An agricultural mill produces a different feed for cattle, sheep, and chickens by mixing the following raw ingredients: corn, limestone, soybeans, and fish meal.

These ingredients contain the following nutrients: vitamins, protein, calcium, and crude fat in the following quantities:

	Nutrient, *k*			
Ingredient, *i*	**Vitamins**	**Protein**	**Calcium**	**Crude Fat**
Corn	8	10	6	8
Limestone	6	5	10	6
Soybeans	10	12	6	6
Fish Meal	4	18	6	9

Let a_{ik} = quantity of nutrient *k* per kg of ingredient *i*

constraints

The mill has (firm) contracts for the following demands:

Demand (kg) d_j	Cattle	Sheep	Chicken
	10,000	6,000	8,000

There are limited availabilities of the raw ingredients:

Supply (kg) s_i	Corn	Limestone	Soybeans	Fish Meal
	6,000	10,000	4,000	5,000

The different feeds have "quality" bounds per kilogram:

	Vitamins		Protein		Calcium		Crude fat	
	min	max	min	max	min	max	min	max
Cattle	6	--	6	--	7	--	4	8
Sheep	6	--	6	--	6	--	4	8
Chicken	4	6	6	--	6	--	4	8

The above values represent bounds: l_{jk} and u_{jk}

costs and objective

Cost per kg of the raw ingredients is as follows:

	Corn	Limestone	Soybeans	Fish meal
cost/kg, c_i	20¢	12¢	24¢	12¢

Formulate problem as a linear program whose solution yields desired feed production levels at minimum cost.

Conceptually:

1. **What decisions are we making?**
2. **What is our objective?**
3. **Uncontrollable inputs?**
4. **Types of constraints?**
5. **Important assumptions?**

Mathematically, define:

1. **Sets, indices**
2. **Decision variables**
3. **Parameters**
4. **Objective**
5. **Constraints**

PAUSE THE VIDEO

sets and indices

$i \in I$ ingredients { corn, limestone, soybeans, fish meal }

$j \in J$ products { cattle, sheep, chicken feeds }

$k \in K$ nutrients { vitamins, protein, calcium, crude fat }

parameters

d_j demand for product j (kg)

s_i supply of ingredient i (kg)

l_{jk} lower bound on number of nutrients of type k per kg of product j

u_{jk} upper bound on number of nutrients of type k per kg of product j

c_i cost per kg of ingredient i

a_{ik} number of nutrients k per kg of ingredient i

decision variables

x_{ij} amount (kg) of ingredient i used to produce product j

$$\min \sum_{i \in I} \sum_{j \in J} c_i x_{ij}$$

minimize cost

$$\text{s.t.} \sum_{i \in I} x_{ij} = d_j \quad \forall j \in J$$

demand constraint of product *j*

$$\sum_{j \in J} x_{ij} \leq s_i \quad \forall i \in I$$

supply constraints for ingredient *i*

$$\left\{ \begin{array}{l} \sum_{i \in I} a_{ik} x_{ij} \geq l_{jk} d_j \quad \forall j \in J, k \in K \\ \sum_{i \in I} a_{ik} x_{ij} \leq u_{jk} d_j \quad \forall j \in J, k \in K \end{array} \right.$$

for every product and every nutrient, meet minimum and maximum requirements

$$x_{ij} \geq 0 \quad \forall i \in I, j \in J$$

non-negativity

generalization of feed mix problem gives *blending* problems

Raw Materials	Qualities	Blended commodities
corn, limestone, soybeans, fish meal	protein, vitamins, calcium, crude fat	feed
butane, catalytic reformate, heavy naphtha	octane, volatility, vapor pressure	gasoline
pig iron, ferro-silicon, carbide, various alloys	carbon, manganese, chrome content	metals
≥ 2 raw ingredients	≥ 1 quality	≥ 1 commodity

AMPL code for the feed mix blending problem

```
reset;

options solver cplex;

set I;    # ingredients
set J;    # products
set K;    # nutrients
```

```
#demand for product j (kg)
param d{J};

#supply of ingredient i (kg)
param s{I};

#lower/upper bound on nutrient k for product j (per kg)
param l{J,K};
param u{J,K} default Infinity;

#cost of ingredient (per kg in cents)
param c{I};

#qty of nutrients k per ingredient (per kg)
param a{I,K};
```

```
#amt of ingredient i to produce product j
var x{I,J} >= 0;

#load data file
data models/feedMix.dat;

#objective
minimize cost:
    sum {i in I, j in J} x[i,j]*c[i];
```

```
subject to demand {j in J}:
   sum {i in I} x[i,j] = d[j];

subject to supply {i in I}:
    sum {j in J} x[i,j] <= s[i];

subject to lowerB {j in J, k in K}:
  sum {i in I} a[i,k]*x[i,j] >= l[j,k]*d[j];

subject to upperB {j in J, k in K}:
  sum {i in I} a[i,k]*x[i,j] <= u[j,k]*d[j];
```

```
data;

set I := corn limestone soybeans fishmeal;
set J := cattle sheep chicken;
set K := vitamins protein calcium fat;

#supply and cost of each ingredient
param:          s       c    :=
   corn        6000    .20
   limestone  10000    .12
   soybeans    4000    .24
   fishmeal    5000    .12;
```

```
#demand for the products
param  d :=  cattle 10000
             sheep   6000
             chicken 8000;


param a:
          vitamins protein calcium fat :=
 corn         8       10       6     8
 limestone    6        5      10     6
 soybeans    10       12       6     6
 fishmeal     4       18       6     9;
```

```
#lower and upper bounds on nutrients
param:                  l   u   :=
   [cattle,*]  vitamins 6   .
               protein  6   .
               calcium  7   .
               fat      4   8
   [sheep,*]   vitamins 6   .
               protein  6   .
               calcium  6   .
               fat      4   8
   [chicken,*] vitamins 4   6
               protein  6   .
               calcium  6   .
               fat      4   8;
```