Using a Math Programming Modeling Language (LINGO)

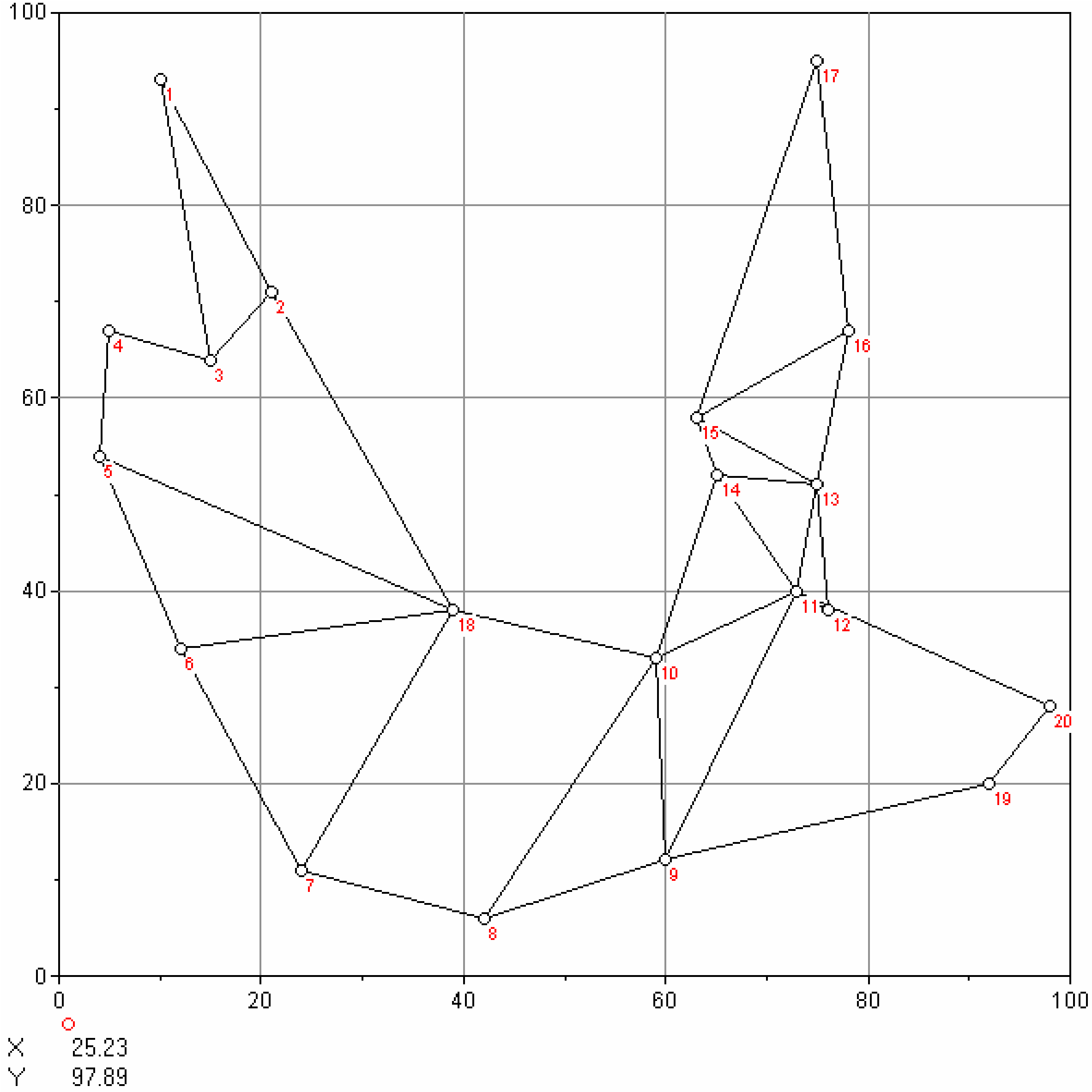
for Set Covering Problems

© Dennis Bricker
Dept of Mechanical & Industrial Engineering
The University of Iowa

A map of the main streets in an urban area are shown on the right.

The public library has decided to put local drop-off boxes for patrons wanting to return books, so that no intersection is more than 30 meters from the nearest drop-off box.

What is the smallest number of boxes which can satisfy this requirement, and where should they be placed?



The following shortest path lengths (in meters) between every pair of nodes in the network were computed (using Floyd's algorithm) and are shown below:

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	0	25	29	39	52	74	94	113	105	84	100	104	111	104	110	127	149	63	138	128
2	25	0	9	19	32	54	69	88	80	59	75	79	86	79	85	102	124	38	113	103
3	29	9	0	10	23	45	71	90	89	68	84	88	95	88	94	111	133	47	122	112
4	39	19	10	0	13	35	61	80	93	72	88	92	99	92	98	115	137	51	126	116
5	52	32	23	13	0	22	48	67	80	59	75	79	86	79	85	102	124	38	113	103
6	74	54	45	35	22	0	26	45	64	48	64	68	75	68	74	91	113	27	97	92
7	94	69	71	61	48	26	0	19	38	51	67	71	78	71	77	94	116	31	71	81
8	113	88	90	80	67	45	19	0	19	32	48	52	59	52	58	75	97	50	52	62
9	105	80	89	93	80	64	38	19	0	21	31	35	42	41	47	58	86	42	33	43
10	84	59	68	72	59	48	51	32	21	0	16	20	27	20	26	43	65	21	54	44
11	100	75	84	88	75	64	67	48	31	16	0	4	11	14	20	27	55	37	38	28
12	104	79	88	92	79	68	71	52	35	20	4	0	13	18	24	29	57	41	42	32
13	111	86	95	99	86	75	78	59	42	27	11	13	0	10	14	16	44	48	49	39
14	104	79	88	92	79	68	71	52	41	20	14	18	10	0	6	23	45	41	52	42
15	110	85	94	98	85	74	77	58	47	26	20	24	14	6	0	17	39	47	58	48
16	127	102	111	115	102	91	94	75	58	43	27	29	16	23	17	0	28	64	65	55
17	149	124	133	137	124	113	116	97	86	65	55	57	44	45	39	28	0	86	93	83
18	63	38	47	51	38	27	31	50	42	21	37	41	48	41	47	64	86	0	75	65
19	138	113	122	126	113	97	71	52	33	54	38	42	49	52	58	65	93	75	0	10
20	128	103	112	116	103	92	81	62	43	44	28	32	39	42	48	55	83	65	10	0

The coverage matrix, using a cover distance of S=30, is

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
7	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0
11	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	1
12	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
13	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
14	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
15	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
18	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
20	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1

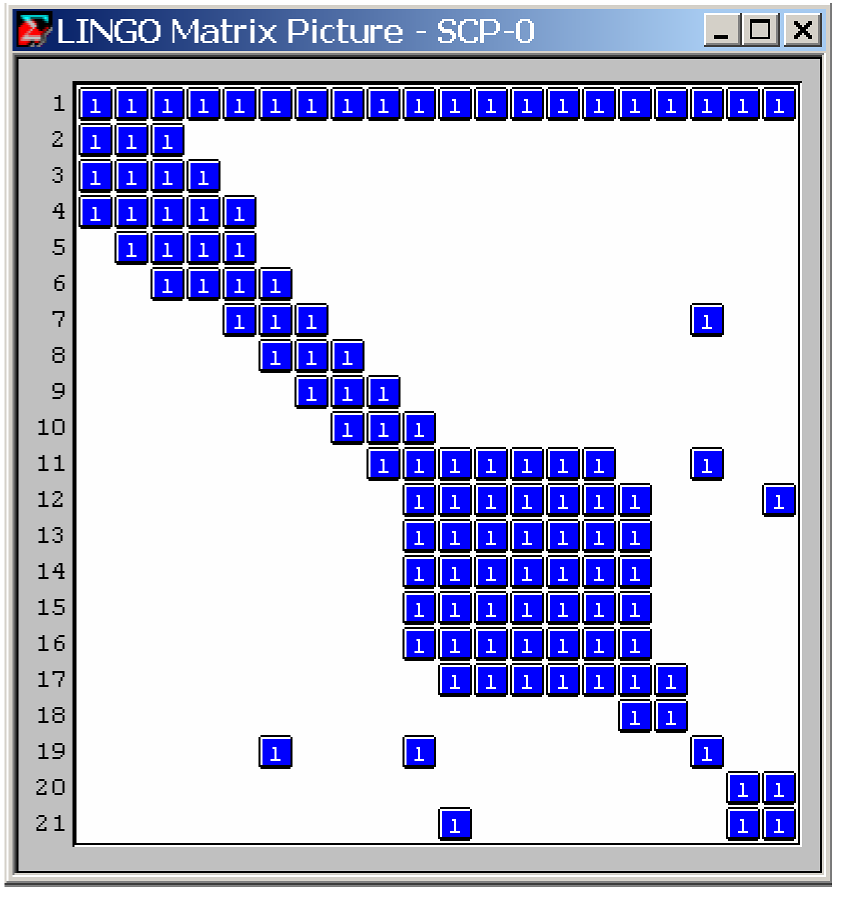
```
MODEL:  ! Set covering problem;

MIN = X_1 +  X_2 +  X_3 +  X_4 +  X_5 +  X_6 +  X_7 +  X_8 +  X_9 +
 X_10 + X_11 + X_12 + X_13 + X_14 + X_15 + X_16 + X_17 + X_18 +
 X_19 + X_20;

X_1 +  X_2 +  X_3 >= 1;
X_1 +  X_2 +  X_3 +  X_4 >= 1;
X_1 +  X_2 +  X_3 +  X_4 +  X_5 >=1;
X_2 +  X_3 +  X_4 +  X_5 >=1;
X_3 +  X_4 +  X_5 +  X_6 >=1;
X_5 +  X_6 +  X_7 + X_18 >=1;
X_6 +  X_7 +  X_8 >=1;
X_7 +  X_8 +  X_9 >=1;
X_8 +  X_9 + X_10 >=1;
X_9 + X_10 + X_11 + X_12 + X_13 + X_14 + X_15 + X_18 >=1;
X_10 + X_11 + X_12 + X_13 + X_14 + X_15 + X_16 + X_20 >=1;
X_10 + X_11 + X_12 + X_13 + X_14 + X_15 + X_16 >=1;
X_10 + X_11 + X_12 + X_13 + X_14 + X_15 + X_16 >=1;
X_10 + X_11 + X_12 + X_13 + X_14 + X_15 + X_16 >=1;
X_10 + X_11 + X_12 + X_13 + X_14 + X_15 + X_16 >=1;
X_11 + X_12 + X_13 + X_14 + X_15 + X_16 + X_17 >=1;
X_16 + X_17 >=1;
X_6 + X_10 + X_18 >=1;
X_19 + X_20 >=1;
X_11 + X_19 + X_20 >=1;
END
```

LINGO

This direct method of specifying the model is

- very prone to errors which are difficult to detect
- difficult to update or modify

Using the “PICTURE” command of LINGO shows the location of the nonzero elements of the coverage matrix, and helps in detecting errors.



Using LINGO's **SETS** simplifies the specification of the model:

The SETS section

```
MODEL:   ! Set covering problem;

SETS:
   INTERSECTION/1..20/:X;          ! Communities to be covered;
   COVER(INTERSECTION,INTERSECTION):A;  ! Coverage matrix;
ENDSETS
```

(Other math programming modeling languages have similar capabilities!)

The DATA section

```
DATA:
   A =  1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ,
        1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ,
        1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ,
        0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ,
        0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ,
        0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 ,
        0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 ,
        0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 ,
        0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 ,
        0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1 0 0 ,
        0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 1 ,
        0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 ,
        0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 ,
        0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 ,
        0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 ,
        0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 ,
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 ,
        0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 ,
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 ,
        0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 ;
ENDDATA
```

The objective & constraints

```
MIN = @SUM(INTERSECTION(J):  X(J));

@FOR(INTERSECTION(I):
    @SUM(INTERSECTION(J):  A(I,J)*X(J) ) >= 1;
);

@FOR(INTERSECTION(J):   @BIN(X(J));     !  Specify that X(J) is binary;
);

END
```

The Solution

```
Global optimal solution found at step:            0
 Objective value:                           5.000000
 Branch count:                                     0
             Variable           Value        Reduced Cost
                 X( 3)        1.000000            1.000000
                 X( 6)        1.000000            1.000000
                 X( 9)        1.000000            1.000000
                X( 16)        1.000000            1.000000
                X( 20)        1.000000            1.000000
```

Nonzero slacks:

```
         Row    Slack or Surplus      Dual Price
           6          1.000000         0.0000000
          12          1.000000         0.0000000
```

i.e., two intersections (#6 & 12) are doubly-covered.

Another way to input the coverage matrix, by indicating which pairs exist:

```
SETS:
   INTERSECTION/1..20/:X;          ! Communities to be covered;
   COVER(INTERSECTION,INTERSECTION)/   ! List of pairs in cover matrix;
   1,1    1,2    1,3
   2,1    2,2    2,3    2,4
   3,1    3,2    3,3    3,4    3,5
   4,2    4,3    4,4    4,5
   5,3    5,4    5,5    5,6
   6,5    6,6    6,7    6,18
   7,6    7,7    7,8
   8,7    8,8    8,9
   9,8    9,9    9,10
  10,9   10,10  10,11  10,12  10,13  10,14  10,15  10,18
  11,10  11,11  11,12  11,13  11,14  11,15  11,16  11,20
  12,10  12,11  12,12  12,13  12,14  12,15  12,16
  13,10  13,11  13,12  13,13  13,14  13,15  13,16
  14,10  14,11  14,12  14,13  14,14  14,15  14,16
  15,10  15,11  15,12  15,13  15,14  15,15  15,16
  16,11  16,12  16,13  16,14  16,15  16,16  16,17
  17,16  17,17  18,6   18,10  18,18  19,19  19,20
  20,11 20,19   20,20/ ;
ENDSETS
```

LINGO

The constraints are stated somewhat differently:

```
MIN = @SUM(INTERSECTION(J): X(J) );

@FOR(INTERSECTION(I):
    @SUM(COVER(I,J): X(J) ) >= 1;
     );

@FOR(INTERSECTION(J): @BIN(X(J));     !  Specify that X(J) is binary;
     );

END
```