Problem 1. Median Location Problem

One county needs to build two hospitals serving nine cities, where the hospitals can be built. The number of hospital visits made annually by the inhabitants of each city and the coordinates of each city are shown in the table.

City	x	y	Visits
1	10	0	5600
2	3	11	1400
3	11	5	2400
4	3	0	3600
5	10	9	800
6	10	13	2800
7	1	1	5100
8	13	5	1400
9	17	7	5400

Find location of hospitals so that the total distance patients must travel to hospitals will be minimized, and the assignment of cities to hospitals.

1) Mathematical model

Mathematical model	**AMPL names:**
Formulation:	
1) $min \sum_{i \epsilon I} \sum_{j \epsilon J} c_{ij} d_j x_{ij}$	Total_Sum
st	
2) $\sum_{i \epsilon I} x_{ij} = 1, \ \forall j \in J$	Users_Allocation {j in CITIES}
3) $x_{ij} \le y_j, \ \forall i \in I \ \forall j \in J$	User_Site_Linking {i in HOSP, j in CITY}
4) $\sum_{i \epsilon I} y_i = p$	Facility_Number
5) $y_j \in \{0; 1\}, \forall j \in J$	Locate {HOSP}
6) $x_{ij} \in \{0; 1\}, \ \forall i \in I \ \ \forall j \in J$	Assign {HOSP, CITY}
Notation:	
Sets:	
I – set of potential locations for hospitals	HOSPS
J – set of cities	CITIES
Parameters:	
d_j – number of visits in city j $\in J$	demand {CITIES}
p – number of hospitals to be located	number
c_{ij} – distance from city j to hospital i	distance {I in HOSPS, j in CITIES}
Variables:	
y_i – 1 if location i is selected, 0 otherwise	Locate {HOSPS}
X_{ij} – number of visitors assigned	Assign {HOSPS, CITIES}

Description

The objective function (1) expresses the total sum of weighted distances when choosing specific vendors for the products. Constraint (2) ensures that cities can be assigned to only one hospital. Constraint (3) is linking constraint to ensure that city can be assigned to a hospital only if hospital is located. Constraint (4) ensures that the number of hospitals is fixed.

Problem size

The resulting model has following dimensions:

90 variables

90 constraints

2) AMPL code

File task4.dat:

```
set HOSPS:= 1 2 3 4 5 6 7 8 9;
set CITIES:= 1 2 3 4 5 6 7 8 9;

param demand := 1 5600 2 1400 3 2400 4 3600 5 800 6 2800 7 5100 8 1400 9 5400;
param number:= 2;
param xcord:= 1 10 2 3 3 11 4 3 5 10 6 10 7 1 8 13 9 17;
param ycord:= 1 0 2 11 3 5 4 0 5 9 6 13 7 1 8 5 9 7;
```

File task4.mod:

```
set HOSPS; #potential hospital location sites
set CITIES; #user points (cities)

param demand{CITIES}; #demand of city j
param number; #number of hospitals to be located
param xcord {i in HOSPS};
param ycord {j in CITIES};
param distance {i in HOSPS, j in CITIES} := sqrt((xcord[i] - xcord[j])^2 + (ycord[i] - ycord[j])^2); #distance between site i and user j

var Locate{HOSPS} binary; #1 if site i is used for locating hospital, 0 otherwise
var Assign{HOSPS,CITIES} binary; #1 if city j is assigned to hospital i, 0 otherwise

minimize Total_Sum:
     sum{i in HOSPS, j in CITIES} distance[i,j] * demand[j] * Assign[i,j];
#Objective function: sum of weighted distances between hospitals and cities

subject to Users_Allocation {j in CITIES}:
     sum{i in HOSPS} Assign[i,j] = 1;
#Users allocation constraint: a city can be assigned to only one hospital

subject to User_Site_Linking {i in HOSPS, j in CITIES}:
     Assign[i,j] <= Locate[i];
#User/Site linking constraint: a user can be assigned to a facility only if facility is located
```

```
subject to Facility_Number:
     sum{i in HOSPS} Locate[i] = number;
#Facility number constraint: the number of facilities to be located is fixed
```

File task4.run:

```
option solver cplex;
option cplex_options 'sensitivity';
model task4.mod;
data task4.dat;
solve;
option omit_zero_rows 1;
display Total_Sum > task4.sol;
display Locate > task4.sol;
display Assign > task4.sol;
display distance > task4.sol;
option show_stats 1;
```

File task4.sol:

```
Total_Sum = 111853

Locate [*] :=
7  1
8  1
;

Assign :=
7 2   1
7 4   1
7 7   1
8 1   1
8 3   1
8 5   1
8 6   1
8 8   1
8 9   1
;

distance [*,*]
:      1          2           3          4           5          6          7   :=
1    0          13.0384     5.09902    7           9          13         9.05539
```

```
    2   13.0384      0            10           11            7.28011
7.28011   10.198
    3    5.09902   10            0            9.43398     4.12311
8.06226   10.7703
    4    7           11            9.43398     0            11.4018
14.7648     2.23607
    5    9           7.28011    4.12311   11.4018     0           4
12.0416
    6   13           7.28011    8.06226   14.7648     4           0
15
    7    9.05539   10.198      10.7703      2.23607   12.0416    15
0
    8    5.83095   11.6619     2          11.1803     5          8.544
12.6491
    9     9.89949    14.5602       6.32456    15.6525       7.28011
9.21954   17.088

    :       8          9         :=
    1    5.83095    9.89949
    2   11.6619    14.5602
    3    2          6.32456
    4   11.1803    15.6525
    5    5          7.28011
    6    8.544      9.21954
    7   12.6491    17.088
    8    0          4.47214
    9    4.47214    0
    ;
```

3) Solution

Minimum weighted distance between the cities and hospitals is 111853. Hospitals should be located in cities 7 and 8. Cities 2, 4 and 7 are assigned to the hospital in city 7, while cities 1, 3, 5, 6, 8, 9 are assigned to the hospital in city 8. Solution is different from the heuristic one.

Problem 2. Full Coverage Problem

An oil company needs to decide where to locate Search-and-Rescue (SAR) helicopters to serve seven offshore installations in emergency situations. Helicopters can be located on an onshore base or on any installation. Location of the base (B1) and the 7 installations (I1-I7) is shown on the picture below. The distances (in km) are given in the table.

	B1	I1	I2	I3	I4	I5	I6	I7
B1	0	123	243	130	402	335	307	417
I1	123	0	419	179	429	186	231	245
I2	243	419	0	344	324	293	492	409
I3	130	179	344	0	276	116	423	519
I4	402	429	324	276	0	183	184	279
I5	335	186	293	116	183	0	335	429
I6	307	231	492	423	184	335	0	405
I7	417	245	409	519	279	429	405	0

The assumed mobilization time for a helicopter is 10 minutes. The time to get to a destination is mobilization time plus travel time. The average travel speed of a helicopter is 260 km/h. The oil company needs to determine locations for helicopters to cover all the installations within 1 hour and 10 minutes with the minimal number of helicopters.

1) Mathematical model

Mathematical model	**AMPL names:**
Formulation:	
1) $min \sum_{j \epsilon j} y_j$	Total_Number
st	
2) $\sum_{j \epsilon J} a_{ij} y_j \geq 1, \ \forall\, i \in I$	Full_Coverage {i in I}
3) $y_j \in \{0; 1\}, \forall\, j \in J$	Use {J}
Notation:	
Sets:	
I – set of installations	I
J – set of possible locations for helicopters	J
Parameters:	
a_{ij} – covering parameter	cover {i in I, j in J}
d_{ij} – distance from installation i to helicopter's location j	distance {I, J}
1 – distance limit	D
Variables:	
y_j – 1 if helicopter is located on site j, 0 otherwise	Use {J}

Description

The objective function (1) expresses the total number of helicopters needed to cover all the installations. Constraint (2) ensures that helicopters are located within distance.

Problem size

The resulting model has following dimensions:

8 variables

8 constraints

2) AMPL code

```
File task5.dat:
set I:= B1 1 2 3 4 5 6 7;
set J:= B1 1 2 3 4 5 6 7;

param D:=260;

param distance : B1 1 2 3 4 5 6 7:=
```

```
B1 0 123 243 130 402 335 307 417
1 123 0 419 179 429 186 231 245
2 243 419 0 344 324 293 492 409
3 130 179 344 0 276 116 423 519
4 402 429 324 276 0 183 184 279
5 335 186 293 116 183 0 335 429
6 307 231 492 423 184 335 0 405
7 417 245 409 519 279 429 405 0;
```

File task5.mod:

```
set I; #installations
set J; #location

param distance {I, J}; #travel distance
param D; #distance limit
param cover {i in I,j in J} = if distance[i,j]>D then 0 else 1;

var Use{J} binary; #1 if helicopter is located in location j

minimize Total_Number: sum{j in J} Use[j];

subject to Full_Coverage{i in I}:sum{j in J} cover[i,j] * Use[j]
>= 1 ;
```

File task5.run:

```
option solver gurobi;
option cplex_options 'sensitivity';
model task5.mod;
data task5.dat;
solve;
option omit_zero_rows 1;
display Total_Number > task5.sol;
display Full_Coverage > task5.sol;
display Use > task5.sol;
display cover > task5.sol;
option show_stats 1;
```

File task5.sol:

```
Total_Number = 3

Full_Coverage [*] :=
;

Use [*] :=
 1  1
```

```
 6  1
B1  1
;

cover [*,*]
:     1   2   3   4   5   6   7  B1     :=
1     1   0   1   0   1   1   1   1
2     0   1   0   0   0   0   0   1
3     1   0   1   0   1   0   0   1
4     0   0   0   1   1   1   0   0
5     1   0   1   1   1   0   0   0
6     1   0   0   1   0   1   0   0
7     1   0   0   0   0   0   1   0
B1    1   1   1   0   0   0   0   1
;
```

3) Solution

Solution found is different from the heuristic solution: helicopters should be located on installations 1, 6 and on the base B1. Helicopter from B1 will serve B1 and installation 2. Helicopter from installation 1 will serve installations 1, 3, 5 and 7. Helicopter from installation 6 will serve installations 4 and 6.

Problem 3. Least-cost Coverage Problem

Locate emergency response and rescue vessels (ERRVs) to cover seven offshore locations with the minimum possible cost. Vessel maximum response time in case of emergency is set to 25 minutes. The average travelling speed of a rescue vessel is assumed to be 45 km/hour (25 knots). The minimum distances (in km) between offshore locations and the potential operating vessel costs (in thousands NOK per year) are given in the table below.

Offshore locations	1	2	3	4	5	6	7
1	0	28	24	28	29	25	12
2		0	46	30	32	28	11
3			0	28	14	48	29
4				0	39	11	50
5					0	30	29
6						0	7
7							0
Fixed cost	220	240	170	250	260	190	270

Find how many ERRVs are needed, and by which vessel(s) each offshore location will be covered.

1) Mathematical model

Mathematical model **AMPL names:**

Formulation:

1) $min \sum_{j \epsilon j} y_j f_j$ Total_Cost

st

2) $\sum_{j \epsilon J} a_{ij} y_j \geq 1, \ \forall i \in I$ Least_Cost {i in I}

3) $y_j \in \{0; 1\}, \forall j \in J$ Use {j in J}

Notation:

Sets:

I – set of locations	I
J – set of possible locations for ERRVs	J

Parameters:

a_{ij} – covering parameter	cover {i in I, j in J}
d_{ij} – distance from installation i to helicopter's location j	distance {I, J}
f_j – fixed cost for the operation of ERRV in location j	cost {j in J}
1 – time limit	T
v – speed	v

Variables:

y_j – 1 if ERRV is located at j, 0 otherwise	Use {j in J}

Description

The objective function (1) expresses the total cost for the operation of ERRVs. Constraint (2) ensures that each location is covered by at least one ERRV.

Problem size

The resulting model has following dimensions:

7 variables

7 constraints

2) AMPL code

```
File task6.dat:
set I:= 1 2 3 4 5 6 7;
set J:= 1 2 3 4 5 6 7;

param T:=0.4;
param v:=45;

param distance : 1   2  3  4  5  6  7:=
               1  0 28 24 28 29 25 12
               2 28  0 46 30 32 28 11
               3 24 46  0 28 14 48 29
               4 28 30 28  0 39 11 50
               5 29 32 14 39 0  30 29
               6 25 28 48 11 30  0  7
               7 12 11 29 50 29  7  0;

param cost:= 1 220 2 240 3 170 4 250 5 260 6 190 7 270;

File task6.mod:
set I; #locations
set J; #ERRVs locations
param T; #time limit
```

```
param v; #speed
param distance {I, J}; #travel distance
param D:=v*T; #distance limit
param cost{j in J}; #operation fixed cost
param cover {i in I,j in J} = if distance[i,j]>D then 0 else 1;

var Use{j in J} binary; #1 if ERRV is located in location j

minimize  Total_Cost:  sum{j  in  J}  Use[j]*cost[j];#Objective: minimize the total cost of ERRV locations

subject to Least_Cost{i in I}:sum{j in J} cover[i,j] * Use[j] >= 1 ; #coverage constraint: each installation i must be covered by at least one ERRV
```

File task6.run:

```
option solver gurobi;
option cplex_options 'sensitivity';
model task6.mod;
data task6.dat;
solve;
#option omit_zero_rows 1;
display Total_Cost > task6.sol;
display Use > task6.sol;
display cover > task6.sol;
```

File task6.sol:

```
Total_Cost = 630

Use [*] :=
1  0
2  0
3  1
4  0
5  0
6  1
7  1
;

cover [*,*]
:   1   2   3   4   5   6   7    :=
1   1   0   0   0   0   0   1
2   0   1   0   0   0   0   1
3   0   0   1   0   1   0   0
```

```
4    0    0    0    1    0    1    0
5    0    0    1    0    1    0    0
6    0    0    0    1    0    1    1
7    1    1    0    0    0    1    1
;
```

3) Solution

ERRVs should be located at offshores 3, 6, 7. ERRV 3 will cover locations 3 and 5, ERRV 6 will cover locations 4 and 6, ERRV 7 – locations 1, 2 and 7.

Total sum is 630000 NOK per year.

Problem 4. Maximum Coverage Problem

A new company making same-day deliveries of groceries to people's homes is launching its business in a large urban area. It has identified 8 neighbourhoods in the area where the company should concentrate its business and 6 locations where the company may locate grocery depots.

The table shows the average time (in minutes) required to between six potential depot locations and centres of each of the neighbourhoods, and target population (in thousands).

Neighbourhoods	Depots						
	1	2	3	4	5	6	Population
1	15	17	16	16	13	13	17
2	15	13	18	10	12	12	12
3	10	15	10	13	12	18	18
4	17	14	12	14	10	9	15
5	14	13	14	14	18	18	18
6	15	11	13	14	11	16	18
7	15	17	14	18	12	17	15
8	14	10	13	19	13	12	16

The company wishes to locate two depots so that they maximize the population served within 12 minutes of average travel time. Find with the optimization model the locations for the depots and show which neighbourhoods are served from each of these depots.

1) Mathematical model

Mathematical model	**AMPL names:**
Formulation:	
1) $max \sum_{i \epsilon I} d_i \, v_i$	Coverage
st	
2) $\sum_{j \epsilon J} a_{ij} y_j \geq v_i, \ \forall \, i \in I$	Cover {i in I}
3) $\sum_{j \epsilon J} y_j = p$	Depots
4) $v_i \in \{0; 1\}, \forall \, i \in I$	Serve {I in I}
5) $y_j \in \{0; 1\}, \forall \, j \in J$	Use {j in J}
Notation:	
Sets:	
I – set of locations	I
J – set of possible locations for ERRVs	J

Parameters:

a_{ij} – covering parameter	cover {i in I, j in J}
d_i – population of neighbourhood i	pop {i in I}
p – number of depots to locate	dep
t_{ij} – travel time from depot j to neighbourhood i	time {I, J}

Variables:

y_j – 1 if depot is located at j, 0 otherwise	Use {j in J}
v_i – 1 if neighbourhood i is served, 0 otherwise	Serve {i in I}

Description

The objective function (1) expresses the amount of people served by the depots. Constraint (2) ensures that each neighbourhood is served. Constraint (3) limits the number of possible depots.

Problem size

The resulting model has following dimensions:

14 variables
8 constraints

2) AMPL code

```
File task7.dat:
set I:= 1 2 3 4 5 6 7 8;
set J:= 1 2 3 4 5 6;

param T:=12;
param pop:= 1 17 2 12 3 18 4 15 5 18 6 18 7 15 8 16;
param dep:=2;
param time :1   2   3  4  5  6 :=
         1 15  17  16 16 13 13
         2 15  13  18 10 12 12
         3 10  15  10 13 12 18
         4 17  14  12 14 10  9
         5 14  13  14 14 18 18
         6 15  11  13 14 11 16
         7 15  17  14 18 12 17
         8 14  10  13 19 13 12;

File task7.mod:
set I; #neighbourhoods
set J; #locations
param T; #time limit
param time {I, J}; #travel distance
param pop {i in I};
param cover {i in I,j in J} = if time[i,j]>T then 0 else 1;
param dep; #number of the depots to locate

var Use{j in J} binary; #1 if depot is located in location j
var Serve{i in I} binary; #1 if population in i is served

maximize Coverage: sum{i in I} Serve[i]*pop[i];
```

```
subject to Cover{i in I}:
     sum{j in J} cover[i,j] * Use[j] >= Serve[i]; #coverage constraint: at least one depot must serve each area
subject to Depots:
     sum{j in J} Use[j] = dep; #constraint for the number of located depots
```

File task7.run:

```
option solver cplex;
option cplex_options 'sensitivity';
model task7.mod;
data task7.dat;
solve;
#option omit_zero_rows 1;
display Coverage > task7.sol;
display Use > task7.sol;
display Serve > task7.sol;
display cover > task7.sol;
```

File task7.sol:

```
Coverage = 94

Use [*] :=
1  0
2  1
3  0
4  0
5  1
6  0
;

Serve [*] :=
1  0
2  1
3  1
4  1
5  0
6  1
7  1
8  1
;

cover [*,*]
```

```
:   1   2   3   4   5   6     :=
1   0   0   0   0   0   0
2   0   0   0   1   1   1
3   1   0   1   0   1   0
4   0   0   1   0   1   1
5   0   0   0   0   0   0
6   0   1   0   0   1   0
7   0   0   0   0   1   0
8   0   1   0   0   0   1
;
```

3) Solution

The depots should be located at points 2 and 5. Depot 2 then covers neighbourhoods 6 and 8, depot 5 – neighbourhoods 2, 3, 4, 6 and 7. Neighbourhoods 1 and 5 won’t be covered.

In this case, maximum population cover is 94, leaving 35 people uncovered.