Appendix F: Multiple Traveling Salesman Problem

Two Trucks Solution

8 hour time limit- (5AM leave warehouse - 1 PM last delivery)

Route 1: St. Francis -> St. Joes -> St. James -> St. Elizabeth -> St. Paul -> St. Anthony -> St. Luke -> St. Francis

Route 2: St. Francis -> St. Mary -> Holy Family -> OSF Alton locations -> St. Francis

Total distance traveled: 755.3 miles

Last delivery occurs 8 hours and 6 minutes after departing the Peoria warehouse.

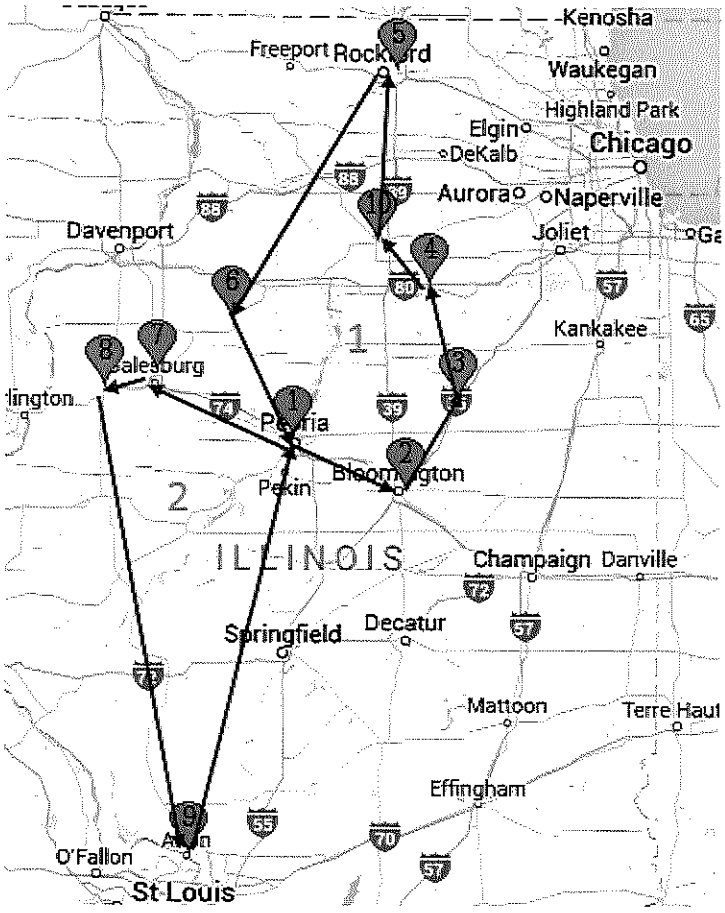

Figure F.1 Two Truck Solution

Three Trucks Solution

7 hour time limit- (5AM leave warehouse 9:30AM last delivery)

Route 1: St. Francis -> St. Joes -> St. James -> St. Elizabeth -> St. Paul -> St. Anthony Rockford ->St. Francis

Route 2: St. Francis -> OSF Alton locations-> St. Francis

Route 3: St. Francis -> St. Mary -> Holy Family -> St Luke-> St. Francis

Total distance traveled: 828 miles

Last delivery occurs 5 hours and 36 minutes after departing the Peoria warehouse.

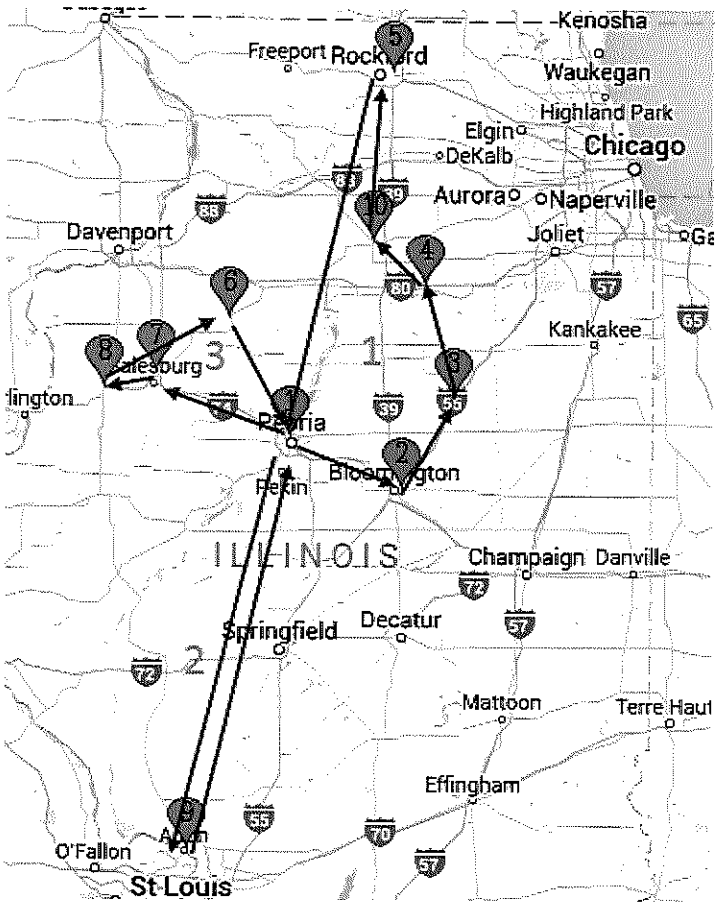

Figure F.2 Three Truck Solution

Four Trucks Solution

6 hour time limit- (5AM leave warehouse 9:30AM last delivery)

Route 1: St. Francis -> St. Joes -> St. James -> St. Elizabeth -> St. Paul -> St Anthony Rockford -> St. Francis

Route 2: St. Francis -> OSF Alton locations-> St. Francis

Route 3: St. Francis -> St. Mary -> Holy Family -> St. Francis

Route 4: St. Francis -> St. Luke-> St. Francis

Total distance traveled: 892.1 miles

Last delivery occurs 5 hours and 36 minutes after departing the Peoria warehouse.

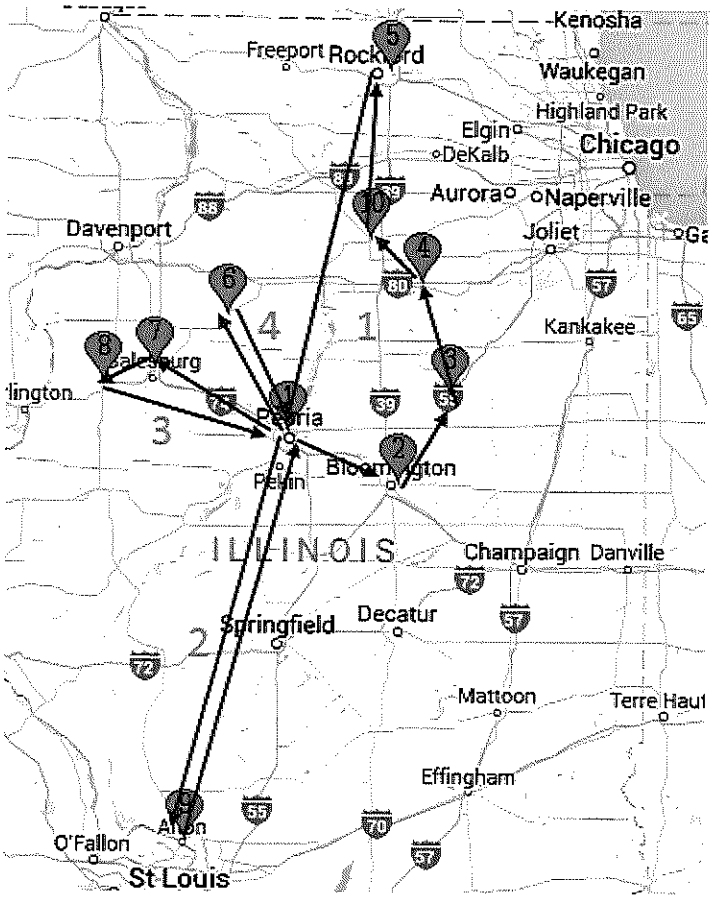

Figure F.3 Four Truck Solution

AMPL MOD FILE

```
param hospital;
param Trucks;
param CTruck;
param Tlim;
param unload;

set HOSP = 1..hospital;
set DEST = 2..hospital;

param D{HOSP,HOSP};
param T{HOSP,HOSP};

var x{i in HOSP, j in HOSP}, binary, default 0;
#var u{2..hospital} >= 0;

minimize distance_traveled:
sum {i in HOSP, j in HOSP} x[i,j]*D[i,j];

subject to depart:                  # T trucks leave peoria
sum{j in HOSP} x[1,j] = Trucks;

subject to return:                  # T trucks return to peoria
sum{j in HOSP} x[j,1] = Trucks;

subject to noreturn{i in HOSP}:     # trucks can't depart and arrive to same location
x[i,i] = 0;
```

```
subject to singleentrance{j in DEST}:      # all nodes are entered exactly once
sum{i in HOSP} x[i,j] = 1;

subject to singleexit{i in DEST}:  # all nodes are exited exactly once
sum{j in HOSP} x[i,j] = 1;

subject to notwocycles{i in DEST, j in DEST: i <> j}:
x[i,j] + x[j,i] <= 1;
                                        # no two node cycles that don't include 1

subject to nothreecycles{i in DEST, j in DEST, k in DEST: i <> j & j<>k & k<>i}:
x[i,j]+x[j,k]+x[k,i]<=2;
                                        # no three node cycles that don't include 1
subject to nothreecyclesrev{i in DEST, j in DEST, k in DEST: i <> j & j<>k & k<>i}:
x[j,i]+x[k,j]+x[i,k]<=2;
                                        # no three node cycles that don't include 1 reverse route

subject to nofourcycles{i in DEST, j in DEST, k in DEST, l in DEST: i <> j & i<>k & i<>l & j<>k & j <>l & k <>l}:
x[i,j]+x[i,k]+x[i,l]+x[j,k]+x[j,l]+x[k,l] <=3;
                                        # no four node cycles that don't include 1

subject to nofourcyclesrev{i in DEST, j in DEST, k in DEST, l in DEST: i <> j & i<>k & i<>l & j<>k & j <>l & k <>l}:
x[j,i]+x[k,i]+x[l,i]+x[k,j]+x[l,j]+x[l,k] <=3;
                                        # no four node cycles that don't include 1 reverse route

subject to nofivecycles{i in DEST, j in DEST, k in DEST, l in DEST, m in DEST: i <> j & i<>k & i<>l & j<>k & j <>l &
k <>l & m<>i & m<>j & m<>k & m<>l}:
x[i,j]+x[i,k]+x[i,l]+x[j,k]+x[j,l]+x[k,l]+x[m,i]+x[m,j]+x[m,k]+x[m,l] <=4;
                                        no five node cycles that don't include 1
```

```
subject to TforONEdrop{j in DEST}:

x[1,j]*T[1,j] <= Tlim;

                                        # Time limit constraint for a one destination cycle


subject to TforTWodrop{j in DEST, k in DEST: j<>k}:

x[1,j]*T[1,j] + x[j,k]*T[j,k] + unload <= Tlim;

                                        # Time limit constraint for a two destination cycle


subject to TforTHREEdrop{j in DEST, k in DEST, l in DEST: j<>k & j<>l & k<>l}:

x[1,j]*T[1,j] + x[j,k]*T[j,k] + x[k,l]*T[k,l] +unload*2 <=Tlim;

                                        # Time limit constraint for a three destination cycle


subject to TforFOURdrop{j in DEST, k in DEST, l in DEST, m in DEST: m <> j & m<>k & m<>l & j<>k & j <>l & k
<>l}:

x[1,j]*T[1,j] + x[j,k]*T[j,k] + x[k,l]*T[k,l]+x[l,m]+T[l,m] + unload*3 <=Tlim;

                                        # Time limit constraint for a four destination cycle


subject to TforFIVEdrop{j in DEST, k in DEST, l in DEST, m in DEST, n in DEST: m <> j & m<>k & m<>l & j<>k & j
<>l & k <>l & n<>m & n<>l & n<>k & n<>j}:

x[1,j]*T[1,j] + x[j,k]*T[j,k] + x[k,l]*T[k,l]+x[l,m]+T[l,m] + x[m,n]*T[m,n] + unload*4 <=Tlim;

                                        # Time limit constraint for a five destination cycle


subject to TforSIXdrop{j in DEST, k in DEST, l in DEST, m in DEST, n in DEST, o in DEST: m <> j & m<>k & m<>l
& j<>k & j <>l & k <>l & n<>m & n<>l & n<>k & n<>j & o<>j & o<>k & o<>l & o<>m & o<>n}:

x[1,j]*T[1,j] + x[j,k]*T[j,k] + x[k,l]*T[k,l] + x[l,m]+T[l,m] + x[m,n]*T[m,n] + x[n,o]*T[n,o] + unload*5 <=Tlim;

                                        # Time limit constraint for a six destination cycle
```

subject to TforSEVENdrop{j in DEST, k in DEST, l in DEST, m in DEST, n in DEST, o in DEST, p in DEST: m <> j & m<>k & m<>l & j<>k & j <>l & k <>l & n<>m & n<>l & n<>k & n<>j & o<>j & o<>k & o<>l & o<>m & o<>n & j<>p & k<>p & l<>p & m<>p & n<>p & o<>p}:

x[1,j]*T[1,j] + x[j,k]*T[j,k] + x[k,l]*T[k,l] + x[l,m]+T[l,m] + x[m,n]*T[m,n] + x[n,o]*T[n,o] + x[o,p]*T[o,p] + unload*6 <=Tlim;

Time limit constraint for a seven destination cycle

subject to TforEightdrop{j in DEST, k in DEST, l in DEST, m in DEST, n in DEST, o in DEST, p in DEST, q in DEST: m <> j & m<>k & m<>l & j<>k & j <>l & k <>l & n<>m & n<>l & n<>k & n<>j & o<>j & o<>k & o<>l & o<>m & o<>n & j<>p & k<>p & l<>p & m<>p & n<>p & o<>p &j<>q & k<>q & l<>q & m<>q & n<>q & o<>q & p<>q}:

x[1,j]*T[1,j] + x[j,k]*T[j,k] + x[k,l]*T[k,l] + x[l,m]+T[l,m] + x[m,n]*T[m,n] + x[n,o]*T[n,o] + x[o,p]*T[o,p] + x[p,q]*T[p,q] + unload*7 <=Tlim;

Time limit constraint for a eight destination cycle

AMPLE DATA FILE

```
param Trucks:=2;          #can change between 2-8
param Tlim:= 800;         #Time limit in minutes (leave warehouse to last delivery)
param unload:= 30;        #average unloading time

param CTruck:= 5000;      #DO NOT CHANGE
param hospital :=10;      #DO NOT CHANGE

param D:=
        #DO NOT CHANGE
1 1 0
1 2 43.2
1 3 69.3
1 4 84.4
1 5 133
1 6 48.3
1 7 49.2
1 8 67.5
1 9 162
1 10 84
2 1 43.2
2 2 0
2 3 34.6
2 4 84.7
2 5 134
2 6 88.7
2 7 89.5
2 8 108
2 9 151
2 10 82
3 1 69.3
3 2 34.6
3 3 0
3 4 42.1
3 5 119
3 6 91.7
3 7 118
3 8 136
3 9 187
3 10 69
4 1 84.4
4 2 84.7
4 3 42.1
4 4 0
4 5 84.2
4 6 70.1
4 7 116
4 8 125
4 9 231
4 10 29
5 1 133
5 2 134
5 3 119
5 4 84.2
5 5 0
5 6 112
5 7 146
5 8 154
5 9 281
5 10 54
6 1 48.3
6 2 88.7
6 3 91.7
6 4 70.1
6 5 112
6 6 0
6 7 34
6 8 51.7
```

6 9 205
6 10 71
7 1 49.2
7 2 89.5
7 3 118
7 4 116
7 5 146
7 6 34
7 7 0
7 8 18.9
7 9 206
7 10 117
8 1 67.5
8 2 108
8 3 136
8 4 125
8 5 154
8 6 51.7
8 7 18.9
8 8 0
8 9 162
8 10 135
9 1 162
9 2 151
9 3 187
9 4 231
9 5 281
9 6 205
9 7 206
9 8 163
9 9 0
9 10 228
10 1 84
10 2 82
10 3 69
10 4 29
10 5 54
10 6 71
10 7 117
10 8 135
10 9 228
10 10 0;

param T:=
1 1 0
1 2 45
1 3 64
1 4 91
1 5 139
1 6 50
1 7 45
1 8 63
1 9 155
1 10 89
2 1 45
2 2 0
2 3 36
2 4 80
2 5 128
2 6 95
2 7 89
2 8 108
2 9 146
2 10 78
3 1 64
3 2 36
3 3 0
3 4 51
3 5 112
3 6 103
3 7 106
3 8 125
3 9 172
3 10 67
4 1 91
4 2 80
4 3 51
4 4 0
4 5 83
4 6 76
4 7 105
4 8 138
4 9 211
4 10 30
5 1 139
5 2 128
5 3 112

5 4 83
5 5 0
5 6 120
5 7 134
5 8 154
5 9 257
5 10 54
6 1 50
6 2 95
6 3 103
6 4 76
6 5 120
6 6 0
6 7 45
6 8 61
6 9 196
6 10 75
7 1 45
7 2 89
7 3 106
7 4 105
7 5 134
7 6 45
7 7 0
7 8 20
7 9 190
7 10 106
8 1 63
8 2 108
8 3 125
8 4 138
8 5 154
8 6 61
8 7 20
8 8 0
8 9 179
8 10 123
9 1 155
9 2 146
9 3 172
9 4 211
9 5 257
9 6 196
9 7 190
9 8 179
9 9 0
9 10 215
10 1 89
10 2 78
10 3 67
10 4 30
10 5 54
10 6 75
10 7 106
10 8 123
10 9 215
10 10 0;