UNIVERSITÉ LIBRE DE BRUXELLES
Master in Engineer of Computer science (M-IRIFS)

Course : Combinational optimization

Project : P-Center 2
INFO-F424

Students: Keneth Ubeda Arriaza (000453317)
Yan Zhang (000461334)

Academic year 2017 - 2018

---

1 Two P Center Algorithms

The p-center problem is a well known combinatorial optimization problem that requires location of p-centers on a given network and allocation of the nodes to the selected centers, so that the maximum of distances between the nodes and their assigned centers is minimum. If the centers can be placed anywhere on the network (vertices as well as edges), the problem is called the absolute p-center problem while; if they can only be located on the vertices, it is called the vertex restricted p-center problem.

Several integer programming (IP) formulations for solving this problem are available in the literature. In this project, we studied two IP formulations and compared the result between them (the formulation proposed by Daskin (1995) and Elloumi et al (2004)) to find who has more optimized result.

1.1 Daskin Formulation

Daskin [17][1]'s formulation is for the vertex restricted case. Define a binary variable yj with $y_j = 1$ if a center is placed at vertex v_j and 0 otherwise. Define binary variables $x_{i,j}$ to be 1 if vi assigns to a center placed at v_j and 0 otherwise.

Denote the formulation of Daskin, given below:

Algorithm 1 Model and constraints

1: (P1): min Z

2: s.t $\sum\limits_{j \in N} d_{i,j} x_{i,j} \leq z \quad \forall i \in N$

3: $\sum\limits_{j \in N} x_{i,j} = 1 \quad \forall i \in N$

4: $x_{i,j} \leq y_j \quad \forall i, j \in N$

5: $\sum\limits_{j \in N} y_j \leq p$

6: $y_i \in \{0, 1\} \quad \forall j \in N$

7: $x_{i,j} \in \{0, 1\} \quad \forall i, j \in N$

Constraints at line 1 and 2 ensure that the objective value is no less than the maximum vertex-to-center distance. Line 3 assigns each vertex to exactly one center and line 4 ensures that no vertex assigns to y_i unless there is a center at y_i. Line 5 restricts the number of centers to p. Line 6 and 7 are binary restrictions, $y_i = 1$ iff $vertex_i$ is selected, $x_{i,j} = 1$ iff p_i is assigned to $vertex_i$, and Z is an upper bound on the radius of the feasible solution.

[1]Daskin MS.Network and discrete location : models,algorithms,and applications. NewYork:Wiley;1995

Below is the formulation implemented in Julia using the JuMP package:

```
function classicdaskinpcenter(selsolver, n, p, dist)

    m = Model(solver = selsolver)

    # Declaring variables
    @variable(m, y[ j=1:n ], category = :Bin)
    @variable(m, x[ i=1:n , j=1:n ], category = :Bin)
    @variable(m, z)

    # Declaring the objective
    @objective(m, Min, z)

    # Declaring the constraints (The order could change the results)
    @constraint(m, [ i=1:n ], sum( dist[i,j] * x[i,j] for j=1:n ) <= z)
    @constraint(m, [ i=1:n ] , sum( x[i,j] for j=1:n ) == 1 )
    @constraint(m, [ i=1:n, j=1:n ], x[i,j] <= y[j] )
    @constraint(m, sum( y[j] for j=1:n ) <= p )

    return m,x,y,z

end
```

1.2 Elloumi et al Formulation

The Elloumi et al formulation is for p-FC and is similar to a canonical representation. Let $p_1 < p_2 < \cdots < p_k$ be an ordering of the distinct distance values of the n by m matrix of distances $d_{i,j} \equiv d(v_i, f_j)$. Define y_j to be the same as in P1 and the additional binary variable u^k, k=2,...,K, with $u^k = 0$ only if all vertices can be covered within a radius value of p_{k-1} and $u^k = 1$ otherwise.

Denote the formulation of Elloumi et al, given below:

Algorithm 2 Model and constrains

1: (P2): $\min\, p1 + \sum_{k=2}^{K} (p_k - p_{k-1}) u^k$
2: s.t $\sum_{j \in M} y_j \geqslant 1$
3: $\sum_{j \in M} y_j \leq p$
4: $u^k + \sum_{j: d_{i,j} < p_k} y_j \geqslant 1 \quad \forall i \in N, k = 2, ..., K$
5: $y_i \in \{0, 1\} \quad \forall j \in M$
6: $u^k \in \{0, 1\}, \quad k = 2, ..., K$

Constraints at line 2 discards solutions with vertex which is out of the graph, line 3 is the same as algorithm 1, at line 5 restricts the number of centers to p.

Line 4 says that, for a given k, $u^k = 0$, iff all vertex can be served at a distance strictly lower than p_k. Therefore, when $u^k = 1$, a distance $(p_k - p_{k-1})$ is added to the radius in the objective at line 1.

Line 5 and 6 are binary restrictions, $y_i = 1$ iff $vertex_i$ is selected, and $u^k = 0$ if it is possible to locate p center and cover all the vertex within the radius p_{k-1}.

Below is the formulation implemented in Julia using the JuMP package:

```
function elloumipcenter(selsolver, n, p, dist)

    m = Model(solver = selsolver)

    # D = sort(unique(nonzeros(sparse(dist)))) # This can change the results
    D = sort(unique(dist))
    k = length(D)

    # Declaring variables
    @variable(m, y[ j=1:n ] , category = :Bin)
    @variable(m, uk[ ik=2:k ], category = :Bin)

    # Declaring the objective
    @objective(m, Min, D[1]  + sum( (D[ki]-D[ki-1]) * uk[ki] for ki = 2:k ))

    # Declaring the constraints (The order could change the results)
    @constraint(m, sum( y[j] for j=1:n ) >= 1 )
    @constraint(m, sum( y[j] for j=1:n ) <= p )
    @constraint(
        m,
        [ i=1:n, ik=2:k ],
        uk[ik] + sum( y[j] for j=1:n if dist[i,j] < D[ik]) >= 1
    )

    return m,y,uk

end
```

2 Daskin vs Elloumi et al formulation

In this project, we compare the formulation between Daskin and Elloumi et al for the general p-Center problem in two type of instances (easy and hard) with different number of vertex and p-Center selection.

2.1 Run environment specifications

Table 1: Computer specifications

Type	Compute optimized AWS EC2 mc5.xlarge
CPU	Intel Xeon Platinum 8124M 2 cores
VCPU	2
Clock speed	3 GHz
RAM	8GB
OS	TheAmazon Linux AMI OS
Julia	version 0.6.2

2.2 Results

We found that the Elloumi et al formulation is better than Daskin, both of formulations have the same optimal solution, but Elloumi et al formulation will pick up the p-Center with shortest distance between each vertex. Since Elloumi will calculate the distance of each vertex $(p_k - p_{k-1})$, so it will spend more computation time when calculate complexity instances.

2.2.1 Easy results

Table 2: n = 10 p = 1

				Daskin			**Elloumi**	
Instance	**n**	**p**	**t(s)**	**Optimal value**	**Solution**	**t(s)**	**Optimal value**	**Solution**
instance10_1_1	10	1	0.07	75	7	0.05	75	7
instance10_1_2	10	1	0.04	72	8	0.05	72	8
instance10_1_3	10	1	0.04	85	10	0.05	85	10
instance10_1_4	10	1	0.04	72	4	0.05	72	4
instance10_1_5	10	1	0.05	70	1	0.04	70	1

Table 3: n = 10 p = 2

				Daskin			**Elloumi**	
Instance	**n**	**p**	**t(s)**	**Optimal value**	**Solution**	**t(s)**	**Optimal value**	**Solution**
instance10_2_1	10	2	0.08	34	5, 6	0.08	34	5, 6
instance10_2_2	10	2	0.06	43	9, 10	0.97	43	9, 10
instance10_2_3	10	2	0.08	46	4, 6	0.21	46	4, 6
instance10_2_4	10	2	0.07	42	3, 4	1.32	42	1, 4
instance10_2_5	10	2	0.07	29	2, 3	0.07	29	2, 3

Table 4: n = 10 p = 5

				Daskin			**Elloumi**	
Instance	**n**	**p**	**t(s)**	**Optimal value**	**Solution**	**t(s)**	**Optimal value**	**Solution**
instance10_5_1	10	5	0.05	9	3, 5, 6, 7, 9	0.04	9	3, 5, 6, 7, 9
instance10_5_2	10	5	0.05	10	1, 5, 6, 10	0.20	10	1, 5, 6, 8, 10
instance10_5_3	10	5	0.04	9	3, 6, 8, 9, 10	1.00	9	3, 6, 8, 9, 10
instance10_5_4	10	5	0.04	9	2, 6, 8, 9, 10	0.58	9	2, 6, 8, 9, 10
instance10_5_5	10	5	0.05	11	1, 4, 5, 9	0.03	11	1, 4, 5, 8, 9

Table 5: n = 20 p = 2

			Daskin			Elloumi		
Instance	**n**	**p**	**t(s)**	**Optimal value**	**Solution**	**t(s)**	**Optimal value**	**Solution**
instance20_2_1	20	2	0.20	39	13, 19	9.57	39	13, 19
instance20_2_2	20	2	0.73	53	7, 18	19.9	53	7, 18
instance20_2_3	20	2	0.71	55	4, 5	18.06	55	4, 5
instance20_2_4	20	2	0.21	51	3, 16	16.07	51	3, 16
instance20_2_5	20	2	1.47	55	8, 14	17.23	55	8, 14

Table 6: n = 20 p = 4

			Daskin			Elloumi		
Instance	**n**	**p**	**t(s)**	**Optimal value**	**Solution**	**t(s)**	**Optimal value**	**Solution**
instance20_4_1	20	4	0.25	21	6, 15, 19, 20	5.56	21	15, 17, 19, 20
instance20_4_2	20	4	0.19	24	2, 11, 14, 19	0.63	24	2, 11, 14, 19
instance20_4_3	20	4	0.25	23	2, 12, 13, 19	6.95	23	2, 7, 11, 12
instance20_4_4	20	4	0.20	17	10, 13, 16, 20	5.04	17	10, 13, 16, 20
instance20_4_5	20	4	0.18	22	1, 4 , 5 , 8	3.35	22	1, 4, 5, 7

Table 7: n = 20 p = 10

			Daskin			Elloumi		
Instance	**n**	**p**	**t(s)**	**Optimal value**	**Solution**	**t(s)**	**Optimal value**	**Solution**
instance20_10_1	20	10	0.07	3	1, 3, 5, 6, 12, 13, 16, 17, 20	0.11	3	1, 5, 6, 10, 12, 13, 15, 16, 17, 20
instance20_10_2	20	10	0.08	4	2, 3, 5, 7, 9, 10, 14, 15, 16, 19	0.14	4	2, 3, 7, 9, 10, 15, 16, 18, 19
instance20_10_3	20	10	0.12	7	2, 3, 5, 6, 7, 8, 12, 18, 19, 20	0.21	7	1, 2, 3, 5, 6, 7, 8, 12, 18, 19
instance20_10_4	20	10	0.08	4	1, 3, 5, 11, 12, 13, 16, 18, 19	0.14	4	3, 5, 6, 8, 12, 13, 16, 18, 19, 20
instance20_10_5	20	10	0.10	7	1, 2, 4, 7, 10, 11, 12, 17, 18, 20	0.32	7	1, 2, 4, 7, 10, 11, 12, 13, 18, 20

2.2.2 Hard Result

Table 8: n = 100, 200 p = 5, 10, 20, 33

			Daskin			Elloumi		
Instance	**n**	**p**	**t(s)**	**Optimal value**	**Solution**	**t(s)**	**Optimal value**	**Solution**
instance1	100	5	80.25	127	7, 42, 78, 94, 99	2267.91	127	35, 65, 78, 85, 99
instance2	100	10	63.60	98	16, 33, 46, 60, 68, 73, 77, 96	1850.98	98	7, 11, 16, 32, 47, 60, 68, 77, 96, 98
instance3	100	10	53.45	93	18, 32, 36, 48, 49, 52, 77, 81, 82, 87	682.42	93	18, 26, 36, 48, 49, 52, 77, 81, 82, 87
instance4	100	20	10.44	74	5, 8, 11, 13, 25, 26, 35, 39, 40, 42, 48, 52, 65, 66, 72, 79, 81, 83, 90, 93	877.48	74	5, 11, 13, 25, 26, 36, 37, 40, 43, 49, 53, 64, 65, 66, 72, 79, 82, 83, 87, 93
instance5	100	33	6.09	48	4, 8, 9, 10, 12, 19, 22, 23, 26, 28, 37, 38, 41, 48, 53, 55, 58, 68, 70, 72, 80, 85, 91, 94, 95, 98	198.21	48	1, 3, 4, 8, 9, 10, 11, 12, 13, 19, 22, 23, 25, 26, 29, 32, 37, 38, 40, 45, 48, 53, 58, 68, 70, 73, 80, 82, 85, 91, 92, 95, 98
			Daskin			**Elloumi**		
instance6	200	5	805.05	84	32, 64, 110, 164	4057.92	84	32, 64, 110, 117, 169
instance7	200	10	397.85	64	13, 77, 83, 105, 131, 143, 164, 180, 183	2812.34	64	2, 10, 77, 95, 105, 164, 176, 180, 183, 186
instance8	200	20	431.22	55	10, 12, 46, 64, 80, 84, 99, 102, 133, 141, 145, 146, 162, 176, 182	2639.06	55	4, 40, 46, 84, 129, 133, 136, 138, 141, 145, 162, 164, 170, 176, 182, 185, 190, 194, 199, 200
instance9	200	40	85.10	37	2, 9, 23, 25, 29, 32, 36, 39, 41, 46, 51, 55, 59, 64, 67, 89, 98, 100, 101, 119, 129, 137, 156, 164, 166, 169, 172, 176, 180, 186, 189, 192, 194, 200	1660.15	37	1, 2, 6, 9, 11, 20, 24, 29, 32, 36, 41, 46, 48, 50, 55, 58, 60, 62, 65, 68, 88, 90, 98, 100, 101, 119, 124, 129, 136, 152, 156, 157, 166, 169, 172, 176, 181, 189, 192, 200
instance10	200	67	30.2	20	3, 11, 14, 15, 16, 19, 20, 33, 34, 39, 41, 42, 43, 48, 49, 50, 55, 58, 59, 62, 65, 68, 69, 70, 80, 81, 85, 90, 101, 117, 119, 121, 123, 125, 128, 129, 144, 145, 147, 148, 156, 159, 166, 168, 172, 173, 175, 176, 180, 183, 192, 193, 194, 196, 199, 200	304.16	20	2, 3, 4, 6, 11, 16, 19, 26, 28, 31, 33, 36, 41, 42, 43, 44, 45, 48, 50, 51, 55, 62, 64, 65, 67, 68, 69, 72, 75, 80, 81, 85, 90, 92, 97, 98, 101, 103, 106, 110, 115, 121, 123, 124, 125, 128, 129, 135, 137, 143, 144, 145, 146, 148, 156, 158, 159, 166, 168, 171, 180, 183, 190, 192, 193, 199, 200

Time (sec) to generate each model and pass it to the solver, a comparison between

Daskin and Elloumi formulation in easy and hard instances. The hard instances, all formulation take more time, in easy instances, there is no significant time computation difference, but we can find in general Daskin finds the optimal value faster than Elloumi, we look on other scientist paper, it may because the formulation given by Daskin which use bisection search, the Elloumi performs binary search accordingly.

So we take further investigation on a specific graphic instead of literature on facility location modeling.

2.3 Further Comparison

We took the additional data which is provided by Daskin for his formulation testing, and we are using the same data for Elloumi et al formulation to compare the optimal solution and the solution value (p-Center) selection.

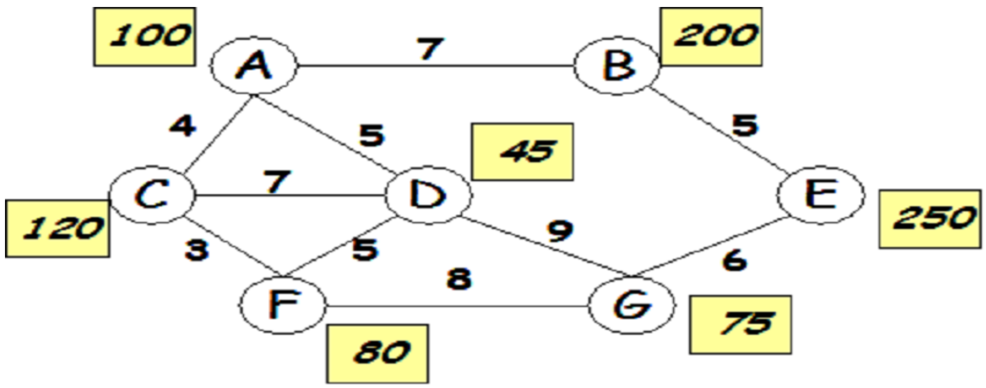

Figure 1: Network Graphic

Compared the result between Daskin and Elloumi et al formulation on above Network Graphic:

Table 9: Network test result

Daskin			Elloumi		
No Sites	**Max Distance**	**Sites**	**No Sites**	**Max Distance**	**Sites**
0			0		
1	14	A	1	14	G
2	7	D,E	2	7	E,F
3	6	D,E,F	3	6	A,E,F
4	5	B,D,F,G	4	5	C,D,E,G
5	4	B, C, D, E, G	5	4	B,C,D,E,G
6	3	A, B, D, E,F, G	6	3	A,B,C,D,E,G
7	0	A, B, C, D, E, F, G	7	0	A,B,C,D,E,F,G

In further comparison, the Daskin formulation will select a p-Center which has a maximum distance with a selected location, and the p-Center has the radius to coverage all vertex, the Elloumi et al will select p-Center which has the radium to coverage all vertex and select the better one in terms of optimal distance.

For example when selected sites ($p = 2$), Daskin formulation selects vertex D and E as p-Centers, the p-center directly connect to each vertex has max distance = 7. Elloumi et al selects vertex E and F as p-Centers, it has the same optimal solution ($MaxDistance$), but it will consider the point between each vertex, the path between vertex A to F, the max distance is 7, but since another vertex C between them, when implementation the network, the network will more flexible and effective.

3 Conclusion

In this paper, we compared the p-Center formulation between Daskin and Elloumi et al, we found that Elloumi et al formulation is better than Daskin because its linear program relaxation gives a better lower bound and because it takes better advantage from the knowledge of bounds on the optimal value. However Daskin formulation took significant less time for the most hard instance, this means if terms of computation time if there is a hard instances but you need a fast answer Daskin will reach the optimal objective value considerable faster than Elloumi et al.

4 Difficulties

The main difficulty that we met during this project was the implementation of the formulation 2, the ordered list of distances that are between in proposed lower and upper bounds, but difficult to validate a set of covering problem has been solved with correct value, there is few example of Cbc libary handle the objective setting in constrains + formula calculation. And when execute the formulation with hard instances, it takes long time consuming.

Finally we found a simple test data to validate both formulations in correctness and we are using Amazon cloud serve to execute all the instances to compare the results.

5 Annex

5.1 Code and implementation : julia-pcenter

This repository contains Daskin and Elloumi formulations to solve the p-center problem using Julia.

5.2 How to run it

To have more details you can read the README.md file.

```
Run from terminal

go to the runnable folder (`cd runnable`) and execute the following commands:

Set the julia home env variable: ( the path depends on where you installed Julia )

julia="/usr/local/julia/bin/julia"

Then you can use the following options to run it:


$julia pcentersolver.jl -f="Instances/xxxxx.dat"
                        -m=daskin|elloumi
                        -l=0|1|2|3
                        -r=bi|fi|vnd
                        -t=3600
                        -th=2

`-f` means the instance file
`-l` the log level you want to run on
`-r` the ratigo gap you want to accept
`-t` the timeout for the execution
`-th` the number of threads you want to run with

Tests

Easy

run-easy.sh [JULIA_HOME] [INSTANCES_HOME] [METHOD(daskin|elloumi)]

Daskin:  ./run-easy.sh /usr/local/julia/bin Instances/easy daskin

Elloumi: ./run-easy.sh /usr/local/julia/bin Instances/easy elloumi


## Hard

run-hard.sh [JULIA_HOME] [INSTANCES_HOME] [METHOD(daskin|elloumi)] [START(1-10)] [END
    (1-10)]

Daskin:  ./run-hard.sh /usr/local/julia/bin Instances/hard daskin 1 10

Elloumi: ./run-hard.sh /usr/local/julia/bin Instances/hard elloumi 1 10
```