Solving Maximal Covering Location Problem with Dynamic Programming

Akshay Joshi

North Carolina State University

Solving Maximal Covering Location Problem with Dynamic Programming

Maximal covering location problem (MCLP) was developed by Church and ReVelle (1974) and extended the basic coverage model. MCLP deals with situation in which the number of vehicles is less than the number needed to cover all demand zones. In a classical MCLP, one seeks the location of several facilities on a network in such a way that the covered population is maximized. A facility covers a demand node, if it is established in a distance less than the threshold to the demand node. This pre-defined threshold is often called the coverage radius, which directly affects the solution of the problem (Zarandi, Davari, & Sisakht, 2011). Figure 1 shows a sample solution to the problem with 20 facilities. In this figure, bold nodes represent the facilities, and circles are the coverage area of each facility. Moreover, smaller dots are the demand centers to be covered.

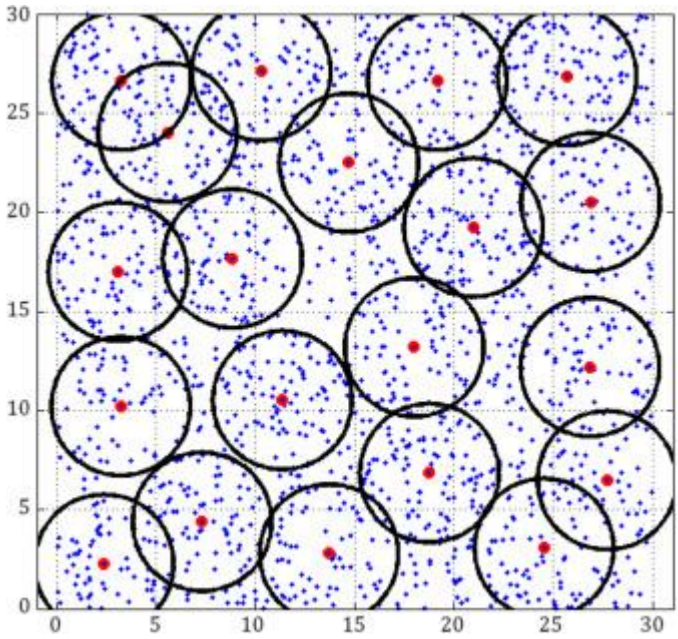

Figure 1. Sample 20 Facility Coverage Setup

In this report, MCLP model is applied to solve EMS location problem by finding the optimal location for available EMS vehicles to maximize the number of calls in the demand zones. The

problem is set up as an Integer Programming Problem in project description. The report will define the problem, conceptually and formally, as a dynamic programming model. Separate solution approaches for both the small and large-size problems will be provided. Finally, an analysis and discussion about the problem complexity and computational efficiency of this model will be carried out.

Setting up the DP Model

The basic idea of dynamic programming (DP) is to take a problem, split it into subproblems, solve those subproblems, and reuse the solutions to those subproblems. It's like a lesson in recycling (MIT OpenCourseWare, 2013).

Since the number of vehicles are less than the total stations, it will be simplistic to think of the setup as a problem of allocating *p* vehicles to *m* stations. We can consider the *stage* as the current vehicle. Thus, as opposed to the original set up, the point of view of this model will be focused on vehicles. Therefore, all the model constraints need to be rewritten considering their effect on vehicles. Consider a vehicle k. We can allocate this vehicle to any of the m available stations. Since the objective is to maximize the number of calls, we must consider the effect of allocating the vehicle (i.e. Reward) in terms of number of calls. According to the problem set up, each station serves multiple demand zones within its coverage radius. Thus, allocating vehicle k to station j will lead to covering calls from these demand zones.

In the original set up, J_i is a set of all stations within the coverage radius of a demand zone. Analogously, we can define M_j as a set of all demand zones within the coverage radius of a station j. For each station j, M_j can be defined as

$M_j \triangleq$ {Set of all demand zones i within coverage radius of station j}

$M_j \triangleq \{ i \mid d_{ij} < D \}$

where D = Coverage Radius

Note that, $\bigcap M_j \neq \phi$ i.e. there is overlap between coverage areas of different stations. To avoid over-counting, we must instead consider the *increase* in number of calls covered because of allocating vehicle k to station j as our reward. Thus, the allocation of each vehicle to a station is considered as the subproblem of the dynamic programming model. In the next section, the DP model is formally defined.

Formulation of the DP Model

Stage: Current Vehicle v = 1…p

State: Set of available Stations {S}

Decision: Which station to allocate the current vehicle j?

Optimality Equation:

Let $f_v(\{S\})$ be maximum calls which can be handled by the system by allocating vehicle v to one of available stations {S} given vehicle (v+1)..p already allocated.

$$f_v(\{S\}) = \max_{j \in \{S\}} \left\{ C_j + f_{v+1}(\{S\} \setminus \{j\}) \right\}$$

where

$C_j \triangleq$ Calls added by allocating a vehicle at station j

$$C_j \triangleq \sum_{i \in N_j} h_i$$

where $N_j = M_j \setminus \left\{ \bigcup_{k \in \{S\}^C} M_k \right\}$

N_j ensures that we don't over-count the number of calls due to overlap of coverage radius of stations.

Boundary Condition: $f_{p+1}(\{S\}) = 0$

Mathematically, {S} is enough to define the state. However, {D} = $\{S\}^C$, a set of stations which have been already allocated a vehicle, is defined for computational ease in R.In the next section, a small size problem will be solved. The details of this problem can be found in the project description.

Case Study: A Small-size Problem

For this problem,

Number of vehicles v = 2

Number of stations j = 4

Number of zones n = 5

Based on the distance matrix, we can define M_j for each station j. Recall, M_j is set of all demand zones within the coverage radius (i.e. 4 miles) of any station j.

Station j	M_j Set of all demand zones within D = 4 miles
1	{1, 4, 5}
2	{2}
3	{3}
4	{1, 4}

Boundary condition $f_3\left(\{S\}\right)=0$. Starting with Stage 2, we can recursively compute calls added to the system based on allocation of vehicles.

Stage v = 2

$\{S\}=\{1,2,3,4\}$

$f_2\left(\{S\}\right)=\max_{j\in\{S\}}\left\{C_j+f_3\left(\{S\}\setminus j\right)\right\}$

$f_2\left(\{S\}\right)=\max_{j\in\{S\}}\left\{C_j\right\}$

Now

$C_j \triangleq \sum_{i\in N_j} h_i$

For stage 2, $\{S\}^C=\phi\Rightarrow\left\{\bigcup_{k\in\{S\}^C} M_k\right\}=\phi\Rightarrow N_j=M_j$

$C_1=h_1+h_4+h_5=12$

$C_2=h_2=10$

$C_3=h_3=6$

$C_4=h_1+h_4=7$

$\therefore f_2\left(\{S\}\right)=12$ where j* = 1

Stage v = 1

$\{S\} = \{2,3,4\}$

$f_1(\{S\}) = \max_{j \in \{S\}} \{C_j + f_2(\{S\} \setminus j)\}$

Now

$C_j \triangleq \sum_{i \in N_j} h_i$

For stage 1, $N_j = M_j \setminus \left\{ \bigcup_{k \in \{S\}^C} M_k \right\}$

$C_2 = h_2 = 10$

$C_3 = h_3 = 6$

$C_4 = 0$

$$f_1(\{S\}) = \max \begin{Bmatrix} 10+12 \\ 6+12 \\ 12 \end{Bmatrix} = 22$$

$\therefore f_1(\{S\}) = 22$ where j = 2

Thus, the maximum number of calls handled by the system will be 22 with EMS vehicles at station 1 and 2.

Case Study: A Large-size Problem

The MCLP for a large-size problem is based on EMS department in Hanover County, Virginia. Two cases are considered with 5 and 10 available vehicles to cover the 16 existing stations. The DP model is coded in RStudio (Version 1.0.136) using {base} package to run this model. The rewards for all available stations at each stage are computed using a function in R. Loops primarily handle all the relevant computations. The input data is in the same pattern as the small size problem. The computational flow and variables are defined in detail in the code package. The summary solutions are discussed below. In both cases, computational time was under 1 second. Coverage rate of 91% & 98% was obtained for 5 and 10 available vehicles respectively.

Case I

Number of vehicles v = 5. Boundary condition $f_6\left(\{S\}\right) = 0$

Number of stations j = 12

Number of zones n = 122

Solution can be summarized in the table below:

Vehicle	Allocated Station	Calls Added
1	4	67
2	15	77
3	14	146
4	1	407
5	6	862
Total Calls		**1559**

Case II

Number of vehicles v = 10. Boundary condition $f_{11}\left(\{S\}\right) = 0$

Number of stations j = 12

Number of zones n = 122

Solution can be summarized as follows:

Vehicle	Allocated Station	Calls Added
1	8	11
2	12	14
3	9	21
4	5	32
5	11	45
6	4	67
7	15	77
8	14	146
9	1	407
10	6	862
Total Calls		**1682**

It's evident that the DP algorithm performs very well. In the last section, a preliminary analysis will be carried out to understand the effect of number of vehicles on runtime and coverage rate.

Analysis and Discussion

Facility location/allocation represents a large and important class of problems that is highly visible both in the optimization and public-sector literatures. These problems are typically NP-complete with a solution space of n^m (Saydam, and Aytuğ, 2003). If MCLP problems are solved using IP, the number of decision variables depends on the zone, which is typically high. Instead, if a Dynamic Programming approach is implemented, the problem is much simpler to solve, since we allocate only p available resources among the solution space. Since there are m stations among which p vehicles are to be allocated, the size of the solution space will be

Solutions to be checked: $C_1^m + C_2^m \dots C_p^m = \sum_{k=1}^{p} C_k^m$

For large-size problem (Case I)

Solution space for DP: $\sum_{k=1}^{5} C_k^{16} = 6884$

Solution space for IP: $n^m = 122^{16} = 2.4 * 10^{33}$

The solution space for DP is further reduced by using functions in R. For the large-size problem (Case I), the reward computing function is called 16+15+14+13=48 times, which is acceptable.

Runtime Analysis

The computational runtime in the DP algorithm depends on the number of times the reward computing function is called. In general, the function will be called $\sum_{k=0}^{p-1}(m-k)$ times, where m is number of stations and p is the number of vehicles. Using the large-scale problem set-up, the model was run for p =1…16. The summary result can be found in figure 2 below.

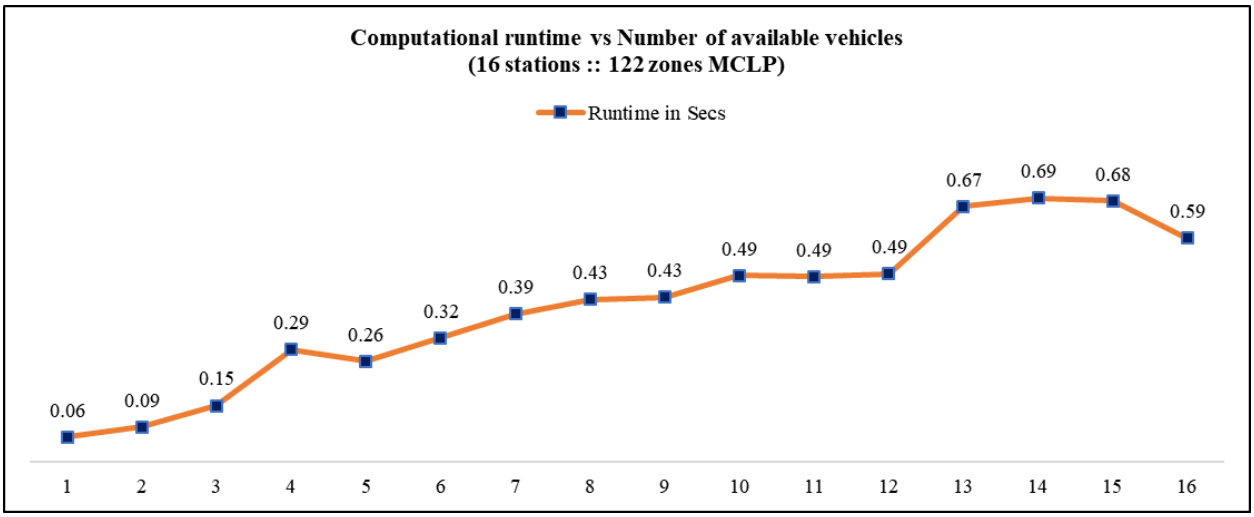

Figure 2. Runtime Analysis

The overall runtime increases as the number of available vehicles increases. However, the rate of increase in runtime falls with the number of vehicles. This can be attributed to the recursive nature of DP algorithm, since the function is called less times for later vehicles. This is corroborated by the reduction in slope for higher values of '*v*' (number of vehicles), as is visible from figure 2.

Coverage % Analysis

A coverage analysis was carried to understand the total calls covered by the system given '*v*' number of vehicles. Coverage % can be defined as

$$\text{Coverage \%} = \frac{\text{Number of calls covered by } v \text{ vehicles}}{\text{Total call demand in the area}}$$

Using the large-scale problem set-up, the model was run for $v = 1 \ldots 16$. The summary result can be found in figure 3 below. Though Coverage % primarily depends on the nature of the system and pattern in which zones and stations are set up, the DP algorithm also plays a significant role. Since available vehicles are allocated to stations based on the maximum *increase* in number of calls, there is rapid increase in coverage % as *v* increases. The rate of increase in coverage % plateaus as $v \to m$, as is evident in figure 2.

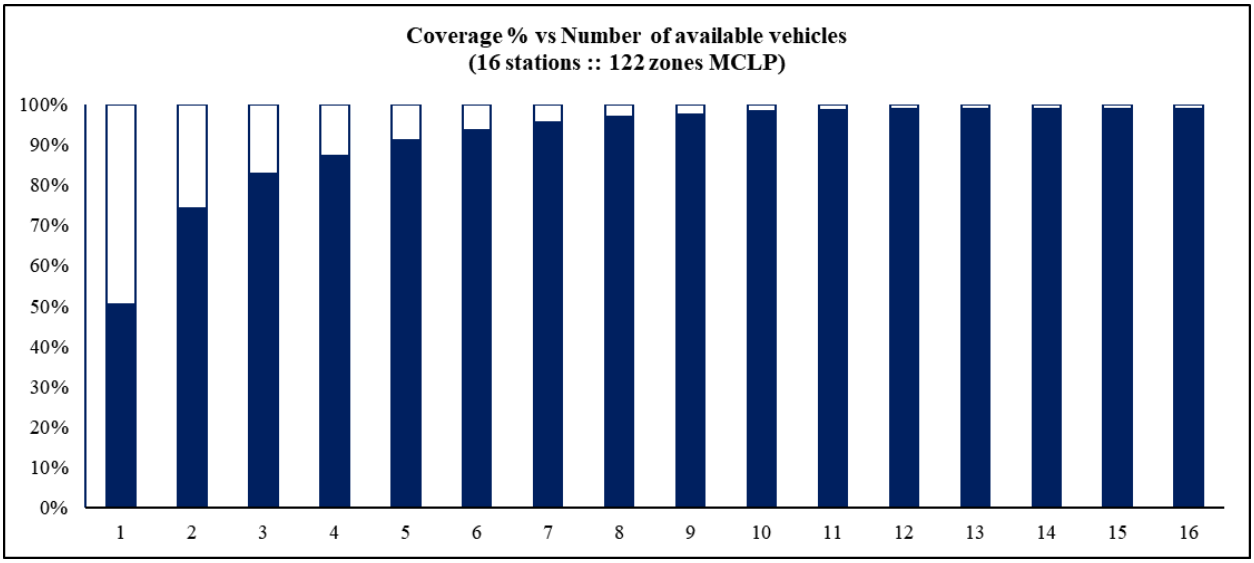

Figure 3. Coverage % Analysis

MCLP in conjunction with DP can also be used to understand minimum number of vehicles needed to attain a specific coverage %. Unlike IP, the DP model must be run only *once* with v=16 to understand the coverage %, total calls and the allocated stations of the entire system. Specific parameters of the system any value of v can be extrapolated from the same solution. For example, from the solution of v = 5 (Case I Problem), we can extrapolate the number of calls covered if v=4 (1559-67=1492 calls). In this regard, the DP model is highly effective.

Coverage Radius Analysis

In the original problem set up, the response time was set to 9 mins or coverage radius of 4 miles. A deep dive into number of calls handled by the system for different coverage radii was carried out. The results suggest that improving coverage radius of stations drastically improves their coverage % for same number of vehicles. Thus, relaxing the expected response time or increasing the response speed can be an alternate way to improve coverage radius without increasing the number of vehicles.

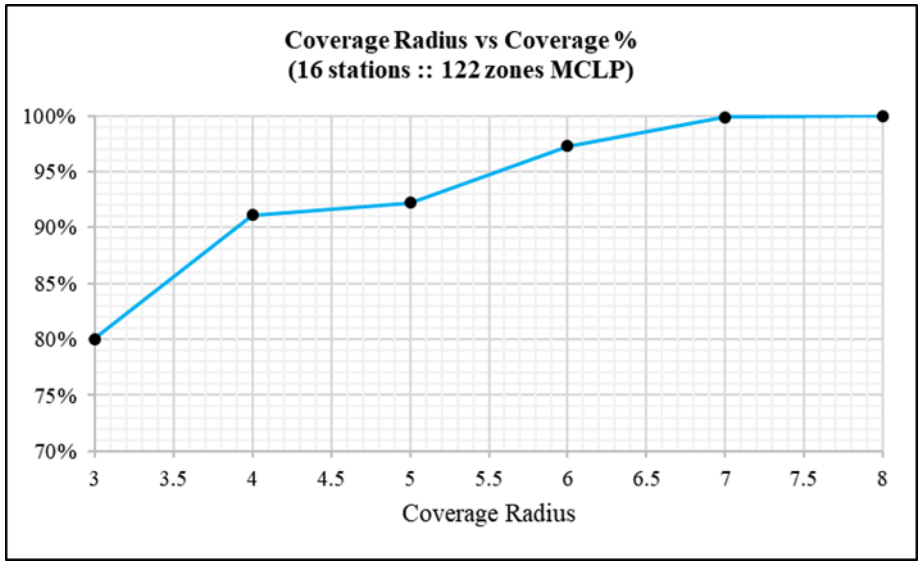

Figure 4. Coverage Radius Analysis

Conclusion

The report presented a dynamic programming approach to solve maximal coverage location problem. The DP was set up formally and sample solutions to small and large-size problems were provided. Finally, preliminary analysis and discussion was carried out to understand the effect of parameters on the coverage of the system. Another avenue of interest might be comparison of this DP model with existing meta-heuristics search methods such as tabu search, simulated annealing and genetic algorithm.

Notes

The author is grateful to the subject teacher for her constructive feedback. This project was submitted towards completion of OR 709 Dynamic Programming (Spring 2018). Implementation R code for the DP algorithm can be found in the author's GitHub repository. (Joshi, 2018)

References

Church, R. L. and ReVelle, C., 1974, “The maximal covering location problem,” Papers of Regional Science Association, 32, 101-118.

Joshi A., 2018, *Solving MCLP with Dynamic Programming* [GitHub]. Retrieved from https://github.com/akshayjoshi942/Solving-MCLP-with-Dynamic-Programming.git

MIT OpenCourseWare. (2013, January 14). *19. Dynamic Programming I: Fibonacci, Shortest Paths* [Video file]. Retrieved from https://youtu.be/OQ5jsbhAv_M

Saydam, C., & Aytuğ, H. (2003). Accurate estimation of expected coverage: revisited. *Socio-Economic Planning Sciences, 37*(1), 69-80. doi:10.1016/s0038-0121(02)00004-6

Zarandi, M. F., Davari, S., & Sisakht, S. H. (2011). The large scale maximal covering location problem. *Scientia Iranica, 18*(6), 1564-1570. doi:10.1016/j.scient.2011.11.008