A method of the people-serving police station location problem

Peng Lu, Jijin Wang, Weixun Ning and Xiaoqiang Xiao*

Department of Computer Science, National University of Defence and Technology, Changsha, Hunan, 410000, China

*E-mail: xqxiaonudt.edu.cn

Abstract. The people-serving police station location problem targets to optimize the location of the police stations and minimize total costs. However, it is very difficult to directly give the solution to the problem. To solve the problem, a method of extending demand points is proposed, which can convert the original problem into a binary integer programming problem. By randomly generating maps, the GNU linear programming kit (GLPK) package is used to solve and verify the converted method.

1. Introduction
Location problem refers to finding a suitable location for a particular function or activity [1], which is essentially an optimization problem. How to properly allocate different spatial resources is difficult, and reasonable location and decision-making will bring huge economic and social benefits [2]. A variety of solving methods have been formed for the location problem. In the early days, the simplex method, the relaxation algorithm and the branch and bound method were used to obtain the optimal solution. However, the location model system structure is more and more complex, and such as set covering problem (SCP), location set covering problem (LSCP), minimum vertex covering (MVC), P-median problem [4,5], K-vertex covering problem have proved to be NP-hard [3]. Traditional optimization methods are difficult to solve these problems with the increasing size. Thus, various intelligent algorithms such as genetic algorithm (GA), particle swarm algorithm, simulated annealing algorithm [6-8], etc., are used to solve complex large-size location problems.

The people-serving police station is a typical public service facility. According to the regulations, the police force is to completely cover the urban area, ensuring that the police can reach any site in the city within a limited time. In the past, the establishment of a station was mainly based on a greedy strategy. Therefore, the location of the convenience police station is often not able to be set up at a minimum cost, resulting in waste of financial expenditure and human resources.

Sections 2 present the initial model, method of conversion and converted model. In Section 3, the feasibility of the method of conversion is verified and the algorithm is compared with the genetic algorithm. A summary of our study is provided in Section 4.

2. Models and Method

2.1. Initial model
The location problem is considered in urban scene, we use grids that fits more closely with the urban scene. And to simplify the calculations, we assume that the police are moving at a constant speed. Then the problem can be abstracted as that in a weighted connected grid, there is a weight upper limit LIMIT, the vertices on the grid are candidate points of the police station, and each candidate point corresponds
to a construction cost. The optimization goal is to minimize the construction cost, so that at any point on the edges, it can reach at least one selected candidate point within the weight LIMIT range. As shown, when LIMIT=30, the points on the black thick line can be covered by point 5.

Figure 1. An example of the problem.

**Definition 1** Grid $G = (V, E, w, c)$, where $V$ is the set of candidate points, $E$ is the set of edges, when $(v_i, v_j) \in E$, $v_i, v_j$ are neighbours and the weight of the corresponding edge is represented as $w_{ij}$, $c_i$ is the cost of candidate point $v_i$, and the upper limit is LIMIT.

**Definition 2** $\text{Dist}(v_i, v_j)$ means the least weight of the path from $v_i$ to $v_j$, and if $i = j$, $\text{dist}(v_i, v_j) = 0$.

**Definition 3** $\text{Extra}(v_i, v_j)$ means the extra weight from $v_i$ to $v_j$, shown in Formula 1.

$$\text{extra}(v_i, v_j) = \begin{cases} \text{LIMIT} - \text{dist}(v_i, v_j) & \text{if dist}(v_i, v_j) < \text{LIMIT} \\ 0 & \text{else} \end{cases}$$  \hspace{1cm} (1)

Figure 2 shows the example that $\text{extra}(v_5, v_6)=10$ when $\text{LIMIT} = 30$.

Figure 2. An example of $\text{extra}(v_i, v_j)$

The mathematical representation of a covered edge: $(v_i, v_j)$ belongs to $E$, there is already selected point $v_i$ and $v_k$, so that:

$$\max(\text{extra}(v_t, v_i)) + \max(\text{extra}(v_k, v_j)) \geq \text{LIMIT}$$  \hspace{1cm} (2)

Then the problem can be considered as selecting a set with the least total cost under the condition that each edge is covered. As show in Formula 3:

$$\min \ c^T x$$

s.t. $\forall (v_i, v_j) \in E, \exists x_t = 1, x_k = 1$ and

$$\max(\text{extra}(v_t, v_i)) + \max(\text{extra}(v_k, v_j)) \geq \text{LIMIT}$$  \hspace{1cm} (3)

$$x_t = 0 \text{ or } 1, i = 1,2,3 \ldots n$$

2.2. Method of conversion
Since the problem model is complex, we consider whether we can convert the problem model. In this problem, the goal is minimizing costs where the selected points can cover all the edges. If the farthest position that a candidate point can reach is set to one point, the problem can be converted by expanding demand points. As figure 3 shows:

![Figure 3. Expanding point v_i.](image)

However, this will lead to an error of the covering of the edge, as shown in Figure v_i v_9 is not covered, so we will extend two points v_i- and v_i+ in the minimal field of v_i instead.

![Figure 4. Expanding point v_i and v_i+.](image)

v_5 can meet the demand point v_7 but can’t meet v_8, so we can use demand points instead of definitions of dist() and extra(). Still taking the edge v_6 v_9 as an example, the new points v_a~v_d is added as shown:

![Figure 5. Expand points at edge v_6 v_9.](image)

The association matrix D can be listed for newly added points in Table 1:

|     | V1  | V2  | V3  | V4  | V5  | V6  | V7  | V8  | V9  |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| va  | 0   | 0   | 1   | 0   | 1   | 1   | 0   | 1   | 1   |
| vb  | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 1   | 1   |
| vc  | 0   | 0   | 1   | 0   | 1   | 1   | 0   | 0   | 1   |
| vd  | 0   | 0   | 1   | 0   | 1   | 1   | 0   | 1   | 1   |

The row represents added demand points, the column represents the candidate points, and D_{ij}=1 indicates that the demand point i can be covered by the candidate point j. Therefore, if we do such a transformation for all edges, we can get the association matrix D_{m×n}, where m is the number of demand points and n is the number of candidate points, we can convert the problem model into a binary integer programming model.

### 2.3. Binary integer programming model

Binary integer programming is a special form of integer programming, which is an important model in operations research [9]. The decision variable x_i of this kind of planning can only take the value 0 or 1, so x_i is called 0-1 variable or binary variable, which can indicate the status of on or off, selected or unselected, presence or absence, etc. There are a large number of applications in the location problem. Using mentioned method of conversion, we can get a binary integer programming model as follows:

\[
\begin{align*}
\text{s. t. } & D \mathbf{x} \geq 1^T \\
\min & \quad c^T \mathbf{x} \\
& x_i = 0 \text{ or } 1, \; i = 1, 2, \ldots, n
\end{align*}
\]

(4)

By randomly generating grids, we focus on the relationship between the number of demand points m and the number of candidate points n. As shown in Fig. 6, we randomly generate the edge lengths obeying the normal distribution of N ([10, 20, 30, 40, 50, 60], [5, 10, 15, 20, 25, 30]) with bound of [0,60], 20 times at sizes of 5 * 5, 10 * 10, 15 * 15, 20 * 20 grids within LIMIT = 30. It can be seen that m and n belong to the same order of magnitude. Unfortunately, this issue is still NP-hard.
3. Experiment

3.1. Experimental algorithm

The GNU linear programming kit is a GNU-developed and cross-platform software package for large-scale linear programming (LP), mixed integer programming (MIP) and other related issues. The one used to solve the integer programming problem is the $ILP(c, G, h, A, b, I, B)$ function, which is defined as follows:

$$
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Gx \leq h^T \\
& \quad Ax = b^T \\
& \quad x_i \in \{0, 1\}, \quad i \in I \\
& \quad x_i \in \{0, 1\}, \quad i \in B
\end{align*}
$$

Therefore, the algorithm with converting method for the location of the people-serving police station is as follows:

Algorithm 1

\begin{verbatim}
V, E, w, c = readmap()
dist = floyd(E)
D = \emptyset
\text{foreach} (v_i, v_j) \text{ in } E:
   \text{foreach } v_k \text{ in } V:
      \text{if } extra(v_k, v_i) > 0 \text{ and } extra(v_k, v_j) < weight(v_i, v_j) :
         \text{expand_node}(D, v_k, v_i, v_j)
      \text{endif}
solution = glpk.ilp(c, -D, {-1}^n, 0, 0, 0, \{1,2,3...n\})
\end{verbatim}

| Parameter      | Detail                        |
|----------------|-------------------------------|
| CPU            | Intel I5 6500@3.2GHz          |
| RAM            | 4GB                           |
| SYSTEM         | Windows 7 X64                 |
| GLPK           | Package based on C program    |
| GENETIC ALGORITHM | Based on Java program         |

3.2. Experimental settings
Experimental scenario: A grid within size of 20*20 and edge lengths obeys a normal distribution of $N(10, 5)$.

Experimental platform: Shown in Table 2.

3.3. Experimental results

![Figure 7. Time costs and solutions by 2 algorithms.](image)

The results shows that the algorithm can quickly find a solution 39 within 60 seconds and got a global optimum solution 33 after 10030 seconds compared with GA solution 167 within 17 seconds and solution 44 after 30,000 generations and 98080 seconds.

4. Conclusions

The people-serving police station location problem is analysed mathematically, and converted into a binary integer programming problem by the method of expanding demand points. The feasibility of the location optimization model is verified by randomly generating grids and using the optimization software GLPK to solve the model. Compared with GA, our method has a better effect.

Acknowledgments

This research was supported by No.61872372 from the National Natural Science Foundation of China. We wish to thank the anonymous referees for their insightful and constructive comments.

References

[1] Susan, H.O., Mark, S.D. (1998) Strategic facility location: A review. European Journal of Operational Research. Volume 111, Issue 3, Pages 423-447, ISSN 0377-2217.

[2] García, S., Marín, A. (2015) The covering location problem. In: Laporte G, Nickel S, Saldanha da Gama F (eds) Location science. Springer, Berlin.

[3] Farahani, R.Z., Asgari, N., Heidari, N., Hosseininia, M., Goh, M. (2012) Covering problems in facility location: a review. Computers & Industrial Engineering, 62(1), 368-407.

[4] Daskin, M.S., Maass, K.L. (2015) The p-median problem. In: Laporte G, Nickel S, Saldanha da Gama F (eds) Location science. Springer, Berlin.

[5] García, S., Labbé, M., Marín, A. (2011) Solving large p-median problems with a radius formulation. INFO-RMS J Comput 23(4):546–556.

[6] Alnahhal, M., Noche, B. (2015) A genetic algorithm for supermarket location problem. Assembly Automation, 35(1), 122–127. doi:10.1108/aa-02-2014-018.

[7] Adeh, I.M., Baroughi, F., Alizadeh, B. (2017) A modified particle swarm optimization algorithm for general inverse ordered p-median location problem on networks. Facta Univ. Ser. Math. Inform. 32, no. 4, 447–468.
[8] Qin, J., Ni, L., Shi, F. (2012) Combined Simulated Annealing Algorithm for the Discrete Facility Location Problem. The Scientific World Journal, 1–7. doi:10.1100/2012/576392.

[9] Fujito T. (2004) On combinatorial approximation of covering 0–1 integer programs and partial set cover. J. Comb. Optim. 8 (4), pp. 439-452.