Optimizing the Placement of Defibrillators in 3D Indoor Environment on Grid

Yuhong Zhou

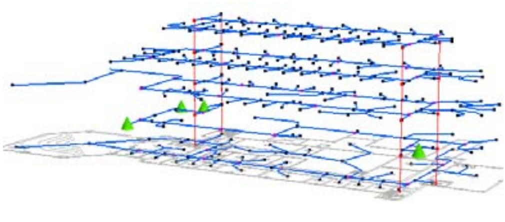
University of North Carolina at Charlotte

1. Problem Description

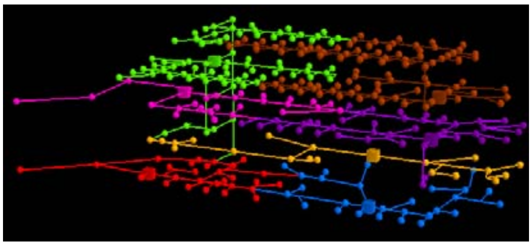
In *Operation Research* or *Spatial Optimization* domain, a common problem (namely *Location-Allocation* problem) researchers face with is how to locate and allocate resources/services/facilities (which are usually limited in their amount or/and in their spatial service extent) to the potential population of demands they are supposed to supply for in the most optimal way. The objective of optimization varies from problem to problem. Based on the objectives, different kinds of problems are widely acknowledged, such as Set-Covering Problem (SCP), Maximum Covering Location Problem (MCLP), P-Median Problem, etc, and are formulated into corresponding mathematical models in a generalized format. One example of the real world applications of MCLP is to locate certain number of fire stations within the city boundary (or any other administrative unit) so that they could serve and protect as many properties and lives as possible, given certain response time such as 5 minutes after a fire incident occurs.

So far, most of the applications of LA problem target a territory with a large extent in 2D space, however, the provision of emergency-services within buildings with multi-stories or/and complex layouts are as crucial as outdoor facilities to saving lives and properties. Our group has been working on a research project of optimizing the placement of Automated External Defibrillators (AEDs) (medical devices used to help the victims of Sudden cardiac arrest (SCA)) in 3D indoor space. The major innovations of our research are not only the integration of traditional LA models with an indoor 3D transportation network model, but also the incorporation of temporal component in the formulation of the mathematical model to capture discrete (although not continuous) dynamics of population distribution in reality. We have implemented and tested several types of models corresponding to SCP, MCLP and P-Median Problem with a case study, and the results are reasonable and roughly justify the correctness of the model formulation and their behaviors (Figure 1).

a. An AED at a Tokyo Railway Station



b. Indoor 3D Transportation Network for the chosen case study



c. One of MTW-P-Median optimized solutions

Figure 1: The case study of location-allocation modeling of AEDs (Dao *etc.,2010*)

However, we observed that, even if only one four-story academic building (316 potential demands, 43 candidate supplies) is tested, three time windows are considered and maximum 5 facilities will be located for MCLP problem, the running time on a single PC for preparing the input files from the raw data and computing the solution to one problem for one scenario is quite long (up to 4 hours in total when accumulating the time for all the steps). This computational constraint hindered us to do more tests by changing the layout of demand/supply distribution, the number of time windows, the design of scenarios, the time threshold to rerun the model, not to say, applying the model to all the buildings within the whole campus (totally 140) or combining the indoor and outdoor placement of AEDs together. Thus, there is an absolute need for employing a method of high-throughput or distributed computing for our research in order to make further investigation on the model behaviors given different types of buildings and compare the pros and cons of different models.

2. Requirements for Estimated Resources

The resource requirements are estimated based on the case study for MCLP problem (given 5 facilities will be located). The basic softwares we used to implement the models are ArcGIS™ Desktop and LINGO™, which are tightly coupled through Visual Basic Application, a VB scripting environment embedded/enabled in ArcMap. We run the code in an interactive ArcMap session (one package of ArcGIS), calling the LINGO runtime libraries during the run. The overhead of running ArcMap alone required 788 MB of virtual space and 55MB of RAM respectively. When the model is running, the use of RAM doubled and the extra 100MB-150MB of the page file for virtual memory is used. The total CPU usage of our PC's single core is less than 10% for the test. The raw input data (a personal database accommodating the spatial distribution of demands and supplies for one time window, 3D network) for the chosen building consumes around 5MB disk storage per time window, and the derived input files (a few distance matrices) are around 2MB per time window. A complete installation of ArcGIS Desktop and extensions requires 1433MB disk space and LINGO runtime libraries need 13 MB on disk. The storage for output files is 5MB.

3. Issues concerned for the implementation through OSG

3.1 About Jobs

One reason I picked up this project for the benefits from grid computing is that the task can be split into jobs at various levels.

(1) Assume the goal is to run the model for all the buildings separately. We could treat each building as an individual project (meaning an independent set of input files), and run the same model (with the same specification for time threshold, time windows) on many remote computers connected within the Grid and even local computers if they are joined into the Grid (we have only 5 PCs open for our research group use, not all of which are available all the time).

(2) If different models are applied (even to one building), we could run them simultaneously on different machines once we submit a DAG job which includes several parallel executable jobs corresponding to the different models (SCP, MCLP, P-Median). The same principle is applicable to running one model under the setting of different scenarios, in which case the executable files are the same for jobs, but the input data are different.

(3) Even if the focus is on one building, one model, one scenario, for MCLP and P-Median problems, we need to repeat the running procedure for multiple times according to the number of facilities we intend to check for the purpose of identifying the cut-off point through the trade-off curve. The repetitive runs can be split into separate jobs, too.

(4) When still one building/model/scenario is concerned, some amount of time is spent on generating some of the input files like distance matrix for each pair of demand and supply. Larger the numbers of demands and supplies, longer our code for data preparation creates those files. We

could split the pairs into several groups and run the code on different remote computers, and then combine the results returned from them to our own local machine into a complete piece.

3.2 Problems with HTC approach

Although arranging jobs is not a problem, I foresee many constraints or obstacles which create difficulties for making the project running on Grid. The first one is the software issue. An ArcGIS Desktop license and an LINGO license will be required on each work node and the ArcGIS Desktop software is also needed to be installed on. Since the ArcGIS Desktop software is a large package, it should not be transferred and install real-time. This means only the remote machines which has this software pre-installed or installed by their administrator when requested can accomplish the jobs. This also restricts the OS on the worknode of the Grid because ArcGIS Desktop requires a Windows system. Data transfer seems a minor issue. The code is embedded in a project configuration file of ArcMap, which should be transferred together with LINGO runtime libraries and input data files. The second major problem is that we need to change the way of coding so that it is executable since currently we run it interactively in the application of ArcMap. The possible steps could be: exporting our current modules, setting up a VB project and importing the modules, compiling the project and making it executable. Other modifications that should be made will be related to defining the relative path to retrieve the input data files and store the output files, and to accessing the physical data directly in the code rather than through the feature layer file in ArcMap.

Another way of making the grid computing feasible is to change the programming language to Python and use the Quantum GIS libraries to achieve some functionality from GIS. This means the implementation of the models needs to be redone. If we also want to make runs independent from LINGO, we could write our own algorithms to generate solutions to the mathematical model we formulated. I saw an example of parallelized generic algorithm which is implemented on the OSG and TeraGrid at the TeraGrid'10. This could be adjusted to our specific Location-Allocation problems.

4. Some thoughts on running the models through HPC

The decision on what kind of computer resources is suitable to the project will depend on the goal of our project. As I mentioned above, if the aim is to optimize the indoor and outdoor AEDs within the campus, using HPC will be better than relying on OSG. The main reason is that, the network (indoor and outdoor) connecting all the buildings will be huge and it becomes not easy to split jobs since all the computation should be done once on one database storing network and other spatial data. HPC could provide better performance in terms of RAMs, CPU time and disk storage. HPC solution also avoids the headache of installing licensed software on different remote computers. We probably need to change the code so that multi-cores on the HPC could be made use of.

References:

1. Thi Hong Diep Dao, Yuhong Zhou, Jean-Claude Thill, Eric Delmelle, 2010, *Spatial-Temporal Indoor 3D Coverage Modeling to Optimize Defibrillators' Placement* (Draft for publication)