Facility location via continuous optimisation with discontinuous objective functions

J. Ugon[*], S. Kouhbor, M. Mammadov, A. Rubinov, A. Kruger,
School of Information and Mathematical Sciences
University of Ballarat, Victoria, Australia, 3353

Abstract

Facility location problems are one of the commonest applications to optimisation. Traditionally these problems have been formulated as combinatorial problems, where the facilities can only be placed at a finite number of locations. However, many applications do not require this constraint, and in such a case, continuous optimisation formulations are more accurate. However, these formulations often result in very complex problems that cannot be solved using traditional optimisation methods. This paper looks at the use of a global optimisation method – AGOP – for solving location problems where the objective function is discontinuous. A real-world application is used for testing this approach numerically.

Keywords: global optimisation, discontinuous optimisation, location problems, AGOP

1 Introduction

Location problems represent an important part in optimisation, as they have a very broad area of practical applications: [4] lists over 3400 references on facility location and related problems.

In [17], the following definition is given for a location problem:

[*]Corresponding author; E-mail: j.ugon@ballarat.edu.au; Fax: 61 03 53279077

> "*Location problems* in the most general form can be stated as follows. A set of customers spatially distributed in a geographical area originates demands for some kind of goods or services. Customers demand must be supplied by one or more facilities (...). The decision process must establish where to locate the facilities in the territorial space taking into account users requirements and possible geographical restrictions. Each particular choice of facility site implies some set up cost for establishing the facility, and some operational costs for serving the customers. Issues like cost reduction, demand capture, equitable service supply, fast response time and so on, drive the selection of facility placement."

As a rule, location problems are generally tackled using combinatorial optimisation, the set of possible locations for facilities having to be finite. The paper [17] reviews this approach to facility location problems. Its complexity increases when the number of possible locations increases, leading to the development of approximation algorithms.

Many applications, however, do not require such a restriction on the placement of the facilities: these only have to be placed over a given area, not at special locations. This problem configuration occurs for example in telecommunications [16, 18], data analysis [2] or public transportation [3] problems. The classical approach to solve these is to transform them into combinatorial problems by discretizing the search space [1, 21]. Due to the NP-hardness of the combinatorial location problems, a solvable discretization may result in inaccurate results.

The alternative approach is to write the problem as a continuous optimisation problem. With such an approach, the complexity of the problem depends on the shape of the cost function, and of the search space. Often, the cost is represented by a min-type function, which prevents the problem to be convex or smooth.

The aim of this research is to develop and test an algorithm for solving continuous location problems with discontinuous functions directly. For that purpose we will select and modify an appropriate method: AGOP [9], whose efficacy will be tested on a real-world application.

In this paper we present a method to solve a particular type of continuous location problem, where the cost function is discontinuous. In section 2, the problem is presented in its general form. Possible approaches to solve are discussed , and a modification of AGOP is presented in Section 3. Section 4 is devoted to a case-study of a practical application, and sections 5 and 6 present respectively numerical results and the conclusion.

2 Special kind of Location Problem

The location problem considered in this paper can be stated as follows: given a set of customer, find the minimal number of facilities, and a satisfactory placement for covering their demand. For each customer, the demand is a binary function: either the customer is satisfied, or not.

This problem can be expressed as follows:

$$\text{minimise} \quad n \qquad \text{subject to:} \quad n \in \mathbb{N}; \tag{1}$$

$$\begin{aligned} \exists [x_i]_1^n : \forall j \in \{1, \ldots, J\}, \\ g_j(x_1, \cdots, x_n) \leq 0; \end{aligned} \tag{2}$$

where $(x_1, \ldots, x_n) \in \mathbb{R}^{n \times m}$, and $g_j : \mathbb{R}^{n \times m} \to \mathbb{R}$ for all $j \in \{1, \ldots, J\}$. J represents the number of customers, m the dimension of the geographical area. The inequality (2) is the constraint verifying that all the demand area is covered. The functions g_j represent the service provided to customer j. They may be discontinuous.

In the sequel it is assumed that the solution exists for a relatively small value of n, and therefore an enumeration method can be considered. On the other hand, the problem of finding a feasible solution satisfying (2) is a difficult one, due to the nature of the functions g_j.

The discontinuity of the functions g_j arise in many practical situations: everywhere the area is subject to obstacle, the service as a function of the distance is likely to be discontinuous. In the telecommunication problem considered further in this paper, each function g_j corresponds to one user, and measures the coverage of this user. Below a certain threshold, the user is not covered, and the constraint (2) is not satisfied. The functions g_j are composed of two parts: $g_j = g_j^1 + g_j^2$.

- g_j^1 is a continuous, nondifferentiable, and concave function.
- g_j^2 is a piecewise constant function.

This model has strong applications in telecommunications, but also in other areas such as data analysis. It can also be applied in many covering-type problems where obstacles have to be taken into account. In order to solve problem (1)-(2), a simple enumeration method is applied as follows:

Algorithm 1 ***- Algorithm for Solving Problem*** **(1)-(2)**

Step 0 *Set* $n = 1$.

Step 1 *Search for* $[x_i]_1^n$ *satisfying* (2).

If *the solution exists: stop.*

Else **Step 2** *Set* $n = n + 1$.

Step 3 *go to step 1.*

The main difficulty resides in step 1 of the algorithm 1. In order to solve this feasibility problem we reformulate the constraint as follows:

$$\max\left(0, g_j(x_1, \cdots, x_n)\right) = 0, \forall j \in \{1, \ldots, J\} \tag{3}$$

Finding a solution to (3) can be reformulated as an optimisation problem such as:

$$\text{minimise} \sum_{j=1}^{J} \max\left(0, g_j(x_1, \cdots, x_n)\right) \tag{4}$$

Problem (3) has a solution if and only if the optimal value of problems (4) is equal to zero. The objective functions of this optimisation problem is discontinuous, since functions g_j are. Therefore very few optimisation methods can be applied to solve these problems. On the other hand, due to the min-type nature of functions g_j, this function usually has a large number of local minima. As a result, most methods that are theoretically applicable to such problems will not be successful in solving them.

In our numerical experiments, problem (4) was preferred to other possible formulations, because considering the average error allows one to consider all constraints at the same time.

One evaluation of the objective function can be expensive, and therefore most optimisation methods will fail to provide a good solution within a reasonable time.

3 Solving the problem

3.1 Overview of Possible Solver

For solving a real-world optimisation problem, the choice of the solver is crucial. In the problem considered in the previous section, we can notice the following issues.

Although the problem (finding the minimum number of APs to be installed to cover an area) is really a nonlinear integer programming problem , for each of these numbers, the evaluation of the condition is a complex one. In this work, the solution of this problem will be considered small enough to allow one to use an enumeration method. On the other hand, for each evaluation of the condition, it is necessary to solve a continuous optimisation problem.

The objective functions of these problems are discontinuous. As a result, it is not possible to apply any of the classical methods based on local properties of the functions (such as Newton-based or bundle methods [8, 6] and their derivatives), nor even methods based on Lipschitz continuity (such as Branch and bounds methods [5]).

To confront the discontinuity of the functions met in telecommunications network designs due to obstacles, a few authors [20, 15, 19] have used genetic algorithms. Due to their heuristic nature, it is quite easy to adapt these methods for solving the type of problem under considerations. However, genetic algorithms are very dependant on the initial population. The complex structure of the functions (these functions have a large number of local minima) results in the necessity of having a large population size. Since even for a simple real-world problem, the evaluation of the objective function of the problem considered in this paper is computationally very demanding, it is very unpractical to use evolutionary algorithms for real-world examples.

Other heuristic approaches, such as simulated annealing or neural networks, present the same drawback: although these methods can be easily implemented, they would perform poorly or be too slow on the problem at hand.

Under all these considerations, the solver used for solving this problem is AGOP (A new Global Optimisation Algorithm - see [9, 10]). This solver finds a good solution using a relatively small number of function evaluations. Its operation is explained in the section 3.2.

3.2 Operation of AGOP

Consider the problem:

$$\text{minimise} \qquad f(x) : \mathbb{R}^n \to \mathbb{R}, \qquad \text{s.t} \quad x \in B, \tag{5}$$

where B is a given box constraints. AGOP must first be given a set of points, say $\Omega = x_1, ..., x_q \subset \mathbb{R}^n$. Generally, a suitable choice for an initial set of points is generated from the vertices of the box B.

Suppose that $x_\star \in \Omega$ has the smallest value of the objective function, that is, $f(x_\star) \leq f(x)$ for all $x \in \Omega$. A possible approach has been developed for finding possible descent direction v at the point $x_\star$ (see [9] for details). An inexact line search along this direction provides a new point $\hat{x}_{q+1}$. A local search about $\hat{x}_{q+1}$ is then carried out. This is done using the *local variation* method. This is an efficient local optimisation technique that does not explicitly use derivatives and can be applied to nonsmooth functions. A good survey of direct search methods can be found in [7]. Letting x_{q+1} denote the optimal solution of this local search, the set Ω is augmented to include x_{q+1}. Starting with this updated Ω, the whole process can be repeated. The process is terminated when v is approximately 0 or a prescribed bound on the number of iterations is reached. The solution returned is the current $x_\star$, that is, the point in Ω with the smallest cost.

The main part of the algorithm is to determine a possible descent direction v at each iteration. The method, used by AGOP for this aim, is based on dynamical systems described by non-functional relationships between two scalar variables (see [11] for details). These relationships are defined in terms of influences of the change (increase or decrease) of one variable on the change of the other. The forces acting from one variable on the change of the other variable are defined by the means of influences. This allows us to define a (non-standard) dynamical system, which provides the direction of changes of each variable at any given point. The algorithm AGOP uses this idea to determine a descent direction v. First, given set Ω, we define a dynamical system that describes the relationships between the objective function and a particular variable x^i, $i = 1, ..., n$. This provides a vector $\bar{v} = (\bar{v}^1, ..., \bar{v}^n)$, calculated at the point x_*, where the coordinate $\bar{v}^i$ is the force acting from x^i on the increase of f. Then the vector $v = -\bar{v}$ is taken as a possible descent direction at the point x_*.

3.3 Modification of AGOP

AGOP is a global solver, and therefore can be run out of the box for finding the solution to problem 4. However, since the calculation of the value of the objective function is very computationally intensive, it is necessary to modify the procedure of AGOP in order to run the program within a reasonable time.

The first modification to the algorithm makes use of the known lower bound to the problem: the value of the function is nonnegative. What is more, the function is zero when the solution is found. If there exists a coverage for a number n of APs, then in many cases the set of optimisers is quite large, and as a result a function value of zero may be found very early on by the algorithm. In such a case, it is not necessary to continue searching, and the algorithm can be exited.

The second modification is based on the sequence of problems that are being solved during the execution of Algorithm 1. It also takes into consideration the geographical nature of the problem: a solution x is structured as a set of geographical points $\bar{x} \in \mathbb{R}^m$. The solution reached at iteration $n-1$ can be used at iteration n. The set Ω of initial points is constructed as follows: $\Omega = \{x_1, \ldots, x_q\}$, where $x_i = (x_0^{*,n-1}, \ldots, x_{q-m}^{*,n-1}, y_1, \ldots, y_m)$, $x^{*,n-1}$ is the solution reached at iteration $n-1$, and $y \in \mathbb{R}^m$ is an initial point constructed from the boundaries of the geographical area.

This allows to reduce the initial size of the set Ω, and therefore to accelerate the execution of AGOP. Furthermore, it also generates initial points that are potentially closer to the set of optimisers, which is reached faster.

4 Case Study

Consider the following problem: Given a building where a Wireless Network needs to be installed, find the minimal number of antennas (Access Points - APs) which can cover the total area where users may move. Notice that the building can be divided into three types of areas:

- Areas where APs can be placed and users need to receive (for example offices);
- Areas where APs can be placed, but users need not to receive (for example stationary rooms);
- Areas where APs cannot be placed (for example elevators).

As a result, the area to cover may be quite complex, and the coverage cannot be computed easily. Therefore, this area is discretized: potential users are placed everywhere a user may need to access the network. Under such a configuration, this problem can be formulated using (1)-(2).

The signal emitted by an AP x_i deteriorates before reaching a potential user u_j. This can be measured by the so-called *pathloss*. Over a certain threshold, the pathloss is too large, and the user cannot receive the signal.

The pathloss can be written as follows [13, 14, 12]:

$$p(x_i, u_j) = p_1 + p_2, \tag{6}$$

where:

$$p_1(x_i, u_j) = p^0 + 20 \log \|x_i - u_j\|_2$$

represents the deterioration caused by the Euclidean distance $\|x_i - u_j\|_2$ between the AP and the user, and

$$\sum_{t=1}^{Q} \delta_t l_t$$

represents the deterioration caused by the obstacles. Here,

$$\delta_t = \begin{cases} 1 & \text{if the obstacle } t \text{ is crossed by the signal;} \\ 0 & \text{otherwise,} \end{cases}$$

and l_t represents the loss for crossing obstacle t.

In such a case, in the formulation (1)-(2), we have:

$$g_j(x_1, \ldots, x_n) = - \min_{1 \le i \le n} (p(x_i, u_j)),$$

and $g_{\max} = -p_{min}$.

This particular problem has the following characteristics:

- The number J of functions g_j is equal to the number of users. This number can be quite large, as the final coverage may depend on the density and the distribution of the users in the building.
- The function g_j is the minimum of functions g_j^i, where $1 \le i \le n$. The number of such functions to evaluate at each objective function evaluation is therefore $n \times J$.
- The functions g_j^i depends on the number of walls separating the user from the access point. When the number of walls in the building is large, evaluating g_j^i is computationally very expensive.

As a result, the evaluation of one objective function is computationally very demanding.

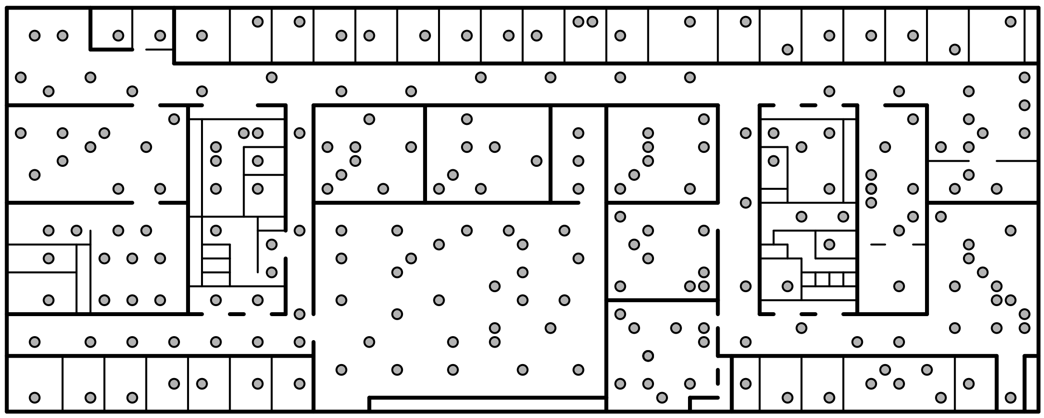

Figure 1: Layout of the design area

5 Testing and Results

5.1 Testing

In order to verify the effectiveness of the algorithm presented in section 3, we carried out some experiments to solve the problem (1)-(2), as formulated in the case study (Section 4). In the experiments, a real-world situation is implemented, which can be found in [19]. This building contains 129 walls (obstacles) of pathloss either 3 or 6. and measures 75 m x 30 m. In this building, 223 potential users are distributed over the area. Figure 1 shows the building specifications and the distributions of the users.

Another set of experiments with 183 potential users distributed over the area has been conducted. The effect of reducing the users is that some rooms do not contain any potential user anymore. This set of experiments is carried out in order to observe the effect of the distribution of users on the number of APs necessary to cover the area.

The experiments were carried on the VPAC supercomputer “Brecca” [22].

The threshold for the pathloss has been varied. Results are presented in tables 1 and 2.

5.2 Results

In [19], the same problem has been solved using a different approach, based on genetic algorithms. Our result show that the method presented here is superior

P_{th}	Num. APs	t_t(sec.)	Num. PL. eval.
115	1	2.74	5.6×10^5
114	2	12.48	2.4×10^6
105	2	12.78	2.4×10^6
100	2	14.51	2.7×10^6
95	2	20.48	3.5×10^6
92	3	48.57	7.1×10^6
90	3	48.55	6.9×10^6
80	4	125.87	1.3×10^7
75	6	407.27	3.3×10^7
70	8	701.24	5.1×10^7
65	14	2241.29	1.2×10^8

Table 1: Results of the experiments for various values of the threshold, for 223 potential users

P_{th}	Num. APs	t_t(sec.)	Num. PL. eval.
115	1	1.01	2.1×10^5
114	1	1.45	3.0×10^5
105	2	10.00	1.9×10^6
104	2	11.15	2.1×10^6
100	2	12.78	2.5×10^6
95	2	11.41	2.2×10^6
92	2	11.27	2.1×10^6
90	2	20.64	3.4×10^6
82	3	41.97	6.1×10^6
80	3	41.43	6.1×10^6
75	5	146.45	1.4×10^7
70	6	285.82	2.5×10^7
65	9	777.29	5.4×10^7

Table 2: Results of the experiments for various values of the threshold, for 183 potential users

to the approach from [19] in this particular building: for a threshold of 80dB, it found a coverage with 4 APs instead of 5, and for a threshold of 100dB, the same amount of 2 APs were found.

From the point of view of time complexity, the results are satisfactory: in most cases, the solution was found within reasonable delay. Only when the number of APs becomes larger, the time necessary to solve the problem increases. From that viewpoint, a few observations can be made:

- The threshold has an influence on the processing time: although for $P_{th} = 114$ and for $P_{th} = 95$ the same number of APs (meaning the same number of optimisation problems, of the same dimensions have been solved), the program needed nearly 1.5 more Path Loss evaluations to find the solution. This is due to the fact that a larger threshold means that more locations allow full coverage. This shows that it is very efficient to take into account the lower bound of the problem in AGOP, which stops as soon as the solution has been found.

- The time complexity is very dependent on the number of APs necessary to cover the area. This is due to the enumeration part of the method: the more APs are necessary, the more optimisation problems need to be solved. What is more, only the last problem resolution can be quickened by the technique described on the previous item.

- The number of users does influence both the result and the running time. However, the running time is only slightly increased when the number of APs is the same. This means that the potential users should well cover the area, otherwise there may be some inaccuracies. In an area covered with obstacles, the discontinuity of the objective function results in needing the users to be distributed densely enough, but also this distribution takes into account the possible effect of every obstacle. On the other hand, if the number of users is extremely large, the program may need more time to solve. This observation is particularly true when the threshold is low (many APs are needed): for $P_{th} = 65$, the number of APs varies from 14 to only 9 when we remove only 40 potential users.

6 Conclusion and Further Research

In this paper, we have presented a novel approach to solve a particular type of location problem. This problem consists of minimising the number of facilities necessary to cover a certain demand, where this coverage depends on their location (namely their distance from the customers). No assumption was made on the coverage of the demand, which can be discontinuous as a function of the distance.

To tackle this problem, a sequence of continuous optimisation problems with discontinuous objective functions are solved. The global optimisation software AGOP has been modified to take into account the particularities of these problems. In particular, the modifications have been devised in order to accelerate the search of a solution, using information obtained during the resolution of the previous problems.

A particular application to this type of problems arising in the design of wireless telecommunication networks has been presented, and numerical experiments have been carried out on a particular instance from a real-world situation.

These experiments have shown that in most cases, the algorithm outperforms other approaches, while solving within an acceptable amount of time. This is due to the very good performance of AGOP for solving the sequence of problems. It is also shown that the number of customers has little influence on the performance of the algorithm, which depends much more on the number of APs needed (that is on the demand threshold).

A number of points may be improved in the current algorithms:

- The enumeration method is not very efficient when the number of APs is large. Using approximation algorithms to estimate roughly the number of APs needed may accelerate the current method;

- Experiments have shown that a large portion of the computing time is spent on solving problems that are not very interesting: if the solution of the problem is a quite high number of APs, then much time is spent on solving cases for a lower number of APs. The use of the suggestion from the point above may reduce this effect, but it may also be of interest to analyse the problem more deeply, to improve the search method.

- Although the number of users does not have a strong influence on the efficiency of the method, this may still become an issue when the number of APs is larger. Experiments have shown, however, that if the users are not adequately distributed over the area, the results can be highly inaccurate. In order to accelerate the algorithm while still obtaining satisfactory results, it may be interesting to use a method similar to the ε-cleaning procedure presented in [2] for reducing the number of customers, while still reaching accurate results. Such a method would allow one to specify a density of potential users high enough to ensure that every obstacle is taken into account, while having an automatic tool that generates a smaller set of users representative of the problem. This is achieved because each potential user is within a reasonable range (that is distance, but also considering obstacles) of a representative user.

References

[1] M. D. Adickes, R. E. Billo, B. A. Norman, S. Banerjee, B. O. Nnaji, and J. Rajgopal, "Optimization of Indoor Wireless Communication Network Layouts," *IIE Transactions*, vol. 34, pp. 823 - 836, 2002.

[2] A.M. Bagirov, A.M. Rubinov, N. Soukhoroukova, and J. Yearwood. Supervised and unsupervised data classification via nonsmooth and global optimisation. *TOP: Spanish Operations Research Journal*, 11:1–93, 2003.

[3] M. J. Beckmann, Continuous models of transportation and location. *Sistemi Urbani*, 3, 403-413 (1981).

[4] , T. Hale http://www.ent.ohiou.edu/ thale/thlocation.html.

[5] E.R. Hansen, *Global Optimization Using Interval Analysis*, Dekker, New York, 1992.

[6] K.C. Kiwiel, " Methods of descent for nondifferentiable optimization," In *Lecture Notes in Mathematics*, volume 1133. Springer-Verlag, Berlin, 1985.

[7] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods," *SIAM Review*, Volume 45, Number 3, pp. 385-482.

[8] C. Lemar échal. "Nonsmooth Optimization and Descent Methods," *Technical Report RR-78-004, International Institute for Applied System Analysis*, Laxemburg, Austria, 1978.

[9] M. Mammadov, "A New Global Optimization Algorithm Based on a Dynamical Systems Approach," in *Proceedings International Conference on Optimization: Techniques and Applications - ICOTA6*, Ballarat, Australia, 2004. Also in *Research Report 04/04, University of Ballarat*, 2004. http://www.ballarat.edu.au/ard/itms/publications/researchPapers.shtml

[10] M. Mammadov, A. M. Rubinov, and J. Yearwood, "Dynamical Systems Described by Relational Elasticities with Applications," in *Continuous Optimization: Current Trends and Applications*, V.Jeyakumar and A.Rubinov (eds), Springer, 2005.

[11] M. Mammadov, J. Yearwood, and L. Aliyeva , "Multi Label Classification and Drug-reaction Associations Using Global Optimization Techniques," in *Proceedings of the 6th International Conference on Optimization: Techniques and Applications -ICOTA6*, Ballarat, Australia, December 2004.

[12] R. Morrow, *Wireless Network Coexistence*, McGraw-Hill, USA, 2004.

[13] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks: A Unified Approach*, Prentice - Hall, New Jersey-USA, 2002.

[14] M. A. Panjwani, A. L. Abbott, and T. S. Rappaport, "Interactive Computation of Coverage Regions for Wireless Communication in Multifloored Indoor Environments," *IEEE Journal Selected Areas in Communications*, vol. 14, pp. 420 - 430, 1996.

[15] B. S. Park, J. G. Yook, and H. K. Park, "The Determination of Base Station Placement and Transmit Power in an Inhomogeneous Traffic Distribution for Radio Network Planning," in *Proceedings of IEEE 56th Vehicular Technology Conference*, Vol.4, 2002, pp. 2051 - 2055.

[16] R. Rodrigues, G. Mateus, and A. Loureiro, "Optimal Base Station Placement and Fixed Channel Assignment Applied to Wireless Local Area Projects," in *Proceeding IEEE International Conference on Networks*, September 1999, pp. 186 - 192.

[17] M.P. Scaparra and M.G. Scutell'a, *Facilities, Locations, Customers: Building Blocks of Location Models. A Survey*. Technical Report TR-01-18, Universit'a Di Pisa.

[18] H. D. Sherali, C. M. Pendyala, and T. S. Rappaport, “Optimal Location of Transmitters for Micro-Cellular Radio Communication System Design,” *IEEE Journal Selected Areas in Communications*, vol. 14, pp. 662 - 673, 1996.

[19] K. S. Tang, K. F. Man, and K. T. Ko, “Wireless LAN Design using Hierarchical Genetic Algorithm,” in *Proceedings of the 7th International Conference on Genetic* Algorithm , 1997, pp. 629 - 635.

[20] K. S. Tang, K. F. Man, and S. Kwong, “Wireless Communication Network Design in IC Factory,” *IEEE Transaction on Industrial Electronics*, vol. 48, pp. 452 - 458, 2001.

[21] M. Unbehaun and M. Kamenetsky, “On the Deployment of Picocellular Wireless Infrastructure,” *IEEE Wireless Communication Magazine*, vol. 10, pp. 70 - 80, 2003.

[22] Victorian Partnership for Advanced Computing. http://www.vpac.org