Geometric Chemotaxis: A Biologically-Inspired Framework for a Class of Wireless Coverage Problems

H. Ozgur Sanli, Rahul Simha and Bhagi Narahari

Abstract— **We present a new, biologically-inspired algorithm for the problem of covering a given region with wireless "units" (sensors or base-stations). The general problem, framed mathematically as the problem of covering a polygon with a minimum number of circles, is applicable both to sensor networks in which the units are sensors with known sensing range and to wireless networks in which the units are base stations that have a known transmission range. While past work has considered the problem of locating a *given* number of units, we consider the joint problem of both determining the optimal number as well as locating them. Our approach to solving the problem invokes a new biological metaphor in algorithm design, *chemotaxis*, that is different from other biology-inspired algorithms such as neural networks or genetic algorithms. In this metaphor, the wireless units are treated as food-seeking organisms that coalesce around nutrient sources and thereby cover a region; by carefully selecting the geometry of these nutrient sources and their distribution, we show that it is possible to control the chemotactic process to efficiently provide overall coverage. We also consider the additional problem of coverage in the presence of obstacles such as indoor walls that attenuate transmission.**

Keywords— **Indoor wireless,circle covering,base station placement**

I. Introduction

The area of wireless networks has grown to encompass various types of networks such as sensor networks and mobile ad hoc networks, all of which are designed to exploit the transmission of signals through space. However, freedom from wiring has also created the expectation among users that adequate transmission quality will be provided all over the region of service. This expectation of "full coverage" gives rise to the problem of providing coverage using as few resources as possible. We term these resources "units." In a sensor network application, the unit is a sensor and its range is the maximum distance at which a sensor reliably detects its target. In a base-station application, the unit is a base-station whose range is a radius of coverage within which signals of a given strength can be reliably decoded. In both applications, a region is to be covered by units that each have a fixed coverage range. The resource optimization goal is to use as few units as possible in ensuring that every part of the region is covered by at least one unit.

We use a standard two-dimensional model in which a unit's range is a circle and the given region is a polygon. Thus, the problem can be stated as follows: given a polygon P and circles of radius r, what is the minimum number of circles needed to cover P, such that each point in P lies in some circle? A secondary, but just as important, question is: where should the circle centers be located? We also consider the additional problems of accounting for the signal attenuation caused by indoor obstacles such as walls and different capacity requirements among the region. The main contribution of this paper is a relatively fast algorithm that constructs an efficient covering of the given polygon.

Department of Computer Science, The George Washington University, 20052 Washington DC ,USA. {sanli,simha,narahari}@gwu.edu

A. Prior Work

The coverage problem we consider and some of its variations are known to be NP-hard [10]. Computational geometers have studied the related point-covering problem of finding the minimum number of circles to cover a discrete set of points in the plane that do not necessarily form a polygon. For this problem Hochbaum et al [17], [18] give a metholodogy in which a "shifting lemma" provides upper bounds on solutions obtained by global application of an algorithm as compared to local application using divide and conquer. In [8] an ϵ-net approach is given for the point covering problem. Although algorithms for the point covering problem can be used as approximations to the polygon covering problem by selecting a set of discrete points in the polygon, this approach can result in highly sub-optimal solutions if the polygon is not regular.

Another related literature is the analysis of spherical packings in two dimensions. In these problems the goal is to derive or approximate fundamental constants such as the minimal radii required of n circles to cover the unit square [25] or to cover the plane [9]. The particular constraints introduced by an irregular polygon fundamentally change the nature of the covering problem; nonetheless, some constants from the standard planar packing results will prove useful in our approach.

Coverage and infrastructure planning has received attention of many researchers in the area of wireless networks. Much of the research has concentrated on the problem of finding the best placement of a fixed number of transmitters over an area and not on the problem of minimizing the number of such units required for total coverage. Various objectives have been considered including minimizing interference, providing best coverage, maximizing average signal strength or a combination of them. Fortune et al [12] use the Nelder-Mead simplex method [28] for optimization while Ephremides et al [32] use neural networks and downhill simplex with omni-directional antennas and adaptive antenna arrays at base stations. Wu et al [36] propose a site

prediction tool which makes use of genetic algorithms and neural networks to find suitable places for access points. In [35], a variant of Nelder Mead simplex method is proposed that can be used for many optimizations in the wireless research besides placement. The work in [3] on outdoor planning, is an example of applying simulated annealing to placement problems. There has also been some research on outdoor cellular base site placement [34], [26],[19].

The initial number of units used in most of the previous efforts is determined by the *regular installation* method in which strips of equal width are first projected on the plane where the width of each strip is equal to the side of the maximum enclosed square inside a circle. Circles are then placed sequentially by sweeping row-wise over the polygon to be covered. Figure-1 shows an example of this methodology in which the strip boundaries are indicated by the solid black lines. Since a circle covers each inscribed square, each strip can be covered by adjacent squares, the centers of which are also the centers of the circles. However, as we show by comparison with our algorithm, this approach may yield inefficient results.

Some research efforts [23], [24] examined the problem of coverage in sensor networks in which, instead of complete coverage, the goal is to reduce the width of the maximal breach path from a start point to a destination. Another variation of the coverage problem involves capacity requirements of the region. Some portions of the region demand more data rate (for example, a conference room where most users bring their laptops) and therefore, the coverage problem can be extended to provide additional capacity for high-bandwidth areas. In [20], a greedy approach is introduced in which the initial number of units is approximated with respect to aggregate data rate need of the area. Placement of the units is done on the basis that if a unit cannot meet the demand, it shares some of its clients with neighbors to minimize the demand on each access point.

To our knowledge, there are only two papers which addressed the problem of determining the minimal number of units for coverage. Frukwirth et al [14] have used a constraint-based optimization method by finding for each grid point the places at which a unit can be placed to cover the point and then applying a "branch and bound" based methodology to minimize the number of these candidate positions. In [33] a hierarchical genetic algorithm is presented in which one gene controls the number of others representing the base station positions. However in both these papers, no comparison with an existing optimization method is given. In terms of approach, Howard et al [16] use the notion of magnetic fields to spread a given number of mobile sensor nodes in an indoor area but do not consider optimized coverage.

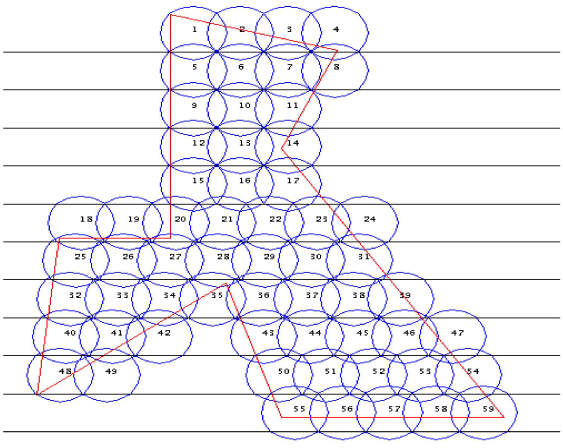

Fig. 1. Regular installation method on a nonconvex indoor area

Our work is distinguished from these efforts in several ways. First, we present a new and fast heuristic for the coverage problem. Second, we demonstrate the effectiveness of our method by comparing it to two heuristics, including simulated annealing. Lastly we show how our method can be applied to the non-uniform bandwidth coverage problem.

Finally, we note that the metaphor of chemotaxis has been independently developed in [27]. However, their interpretation of chemotaxis for optimization is quite different, and is aimed at very different kinds of problems. Their goal is to search for the minimum (or maximum) of a general non-linear function and to perform a local search by considering the value of the function in a neighborhood of the current iteration. In contrast, our problem has elements of both a continuous and discrete nature and the chemotactic metaphor is applied to geometric constraints.

II. Algorithm Description

In this paper, we explore a new biological paradigm in designing an algorithm for the coverage problem. For comparison, consider that other biological paradigms in algorithm design have enjoyed measured success: genetic algorithms [15], evolutionary algorithms [11] and neural networks [7]. Unlike the selectivity theme of genetic and evolutionary algorithms, or the function-approximation theme of neural networks, our approach is inspired by the phenomenon of *chemotaxis* in which small organisms (bacteria) move in response to chemical cues [2],[4]. In chemotaxis, bacteria regulate movement to move along concentration gradients of chemical attractants (towards higher concentration) or repellants (towards lower concentration) and in so moving can form clusters whose shape is determined by the concentrations.

We treat the circles (sensors, transmitters or wireless units) in our problem as the organisms and we arrange for nutrients (attractants) and toxins (repellants) to be judiciously placed in the polygon to encourage the organisms (circles) to properly cover the polygon. Each circle is added to the system one by one and when added finds its optimal location through chemotaxis. The successive location of such "selfish" circles eventually results in a covering of the polygon that is reported as the solution by the algorithm.

Unfortunately, as other algorithms inspired by biological metaphors have shown, a naive implementation does not necessarily produce good results. For example, a simple

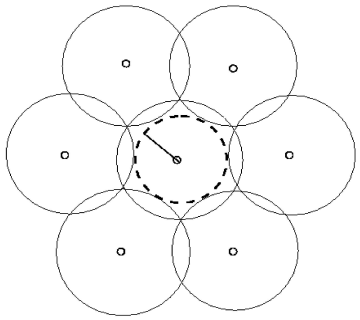

Fig. 2. Optimal planar (lattice) arrangement

initial attempt to place nutrients in the interior of the polygon and toxins in the exterior does not work because it is necessary to allow limited overlap of circles. Furthermore, for good coverage it is necessary to exploit the geometry of overlapping circles and their complicated interaction with the boundary of the polygon. Interestingly, as we show, it is possible to incorporate these geometric concepts into the chemical gradients laid out in the polygon.

To incorporate the local geometry, we introduce *local adaptations* that are made after the final location is found for a particular circle. The general idea is that coverage starts by attracting circles to cover the periphery of the polygon and to gradually encourage clustering towards the interior. At the same time, the nutrient and toxin concentrations are altered in the vicinity of the most recently placed circle to force future circles into near-optimal coverage patterns. We discuss these adaptations in some detail below. The overall structure of the algorithm is:

1. **while** coverage not achieved
2. Add a new circle
3. Find optimal location for circle
4. Adjust nutrient and toxin concentrations
5. **endwhile**
6. **return** circle locations

One issue that surfaced early in our experimentation with this approach was the proper representation of space: we discovered that movement in free space (i.e., with real-valued positions) results in a system that is slow to stabilize and can also result in cyclic behavior, as in many continuous dynamical systems [6]. Accordingly, we place a discrete grid over the polygon using a grid spacing that is a small enough fraction (3-5%) of the circles' common radius. Then, each grid position is evaluated by each circle as a potential location for its center thereby limiting the computation time for each iteration. The grid contains both the polygon and a set of points outside the polygon which form a band around the polygon boundary of thickness equal to the diameter of the units.

Finally, we use a single number $v(x, y)$ at each grid point (x, y) to indicate the *level* of nutrient or toxin: a positive value indicates the concentration of a nutrient and a negative value indicates a toxin concentration. Define the *value* of placing a circle's center at (x, y) as:

$$V(x,y) = \sum_{x'=x-r}^{x+r} \sum_{y'=y-r}^{y+r} v(x', y') \tag{1}$$

Both the level and value of a grid point may change as the algorithm proceeds.

Then, a slightly more refined description of the algorithm can be given as:

1. Initialize nutrient and toxin concentrations
2. $i = 0$
3. **while** coverage not achieved
4. $i = i + 1$
5. Add circle i
6. Find grid point (x_i, y_i) that maximizes $V(x_i, y_i)$
7. Adjust concentrations $v(x, y)$ in vicinity of (x_i, y_i)
8. **endwhile**
9. Adjust for obstacles, if applicable
10. **return** circle centers (x_i, y_i)

As line 9 shows, we address the issue of propagation attenuation from obstacles after an initial placement of circles via chemotaxis. This adjustment uses a variant of simulated annealing [1] that we describe later in Section III.

We next describe the initial concentration of nutrients and toxins and the concept of local adaptations since these are collectively important to the success of the algorithm.

A. Nucleal structure and Initial Environment Density

We incorporate geometric information in the chemotaxis metaphor in two ways. The first involves *structural* elements such as defining a "nucleus" for each organism (circle) and a "cell wall" around the perimeter of the polygon. The second is *chemical* and involves defining concentration gradients within these structural elements.

To explain why we use a nucleus, consider covering the plane with circles. It is well known that the best arrangement is hexagonal with six neighbors to each circle [9] as shown in Figure-2, with thickness $\frac{2\pi}{3\sqrt{3}}$. In this covering, there is some degree of overlap (as there must be in all circle coverings). Our use of a nucleus is motivated by this question: how should chemical concentrations be defined to permit an optimal packing in the plane (i.e., without the constraints of a polygon)? Our solution is to define for each circle (of radius r) a nucleus of radius $r_{nuc} = r * (\sqrt{3} - 1)$ and to set the toxin level high inside the nucleus while setting a nutrient level outside the nucleus. In this way, a new circle will not overlap existing circles to a degree beyond what is optimal (because if it did, the overall "value" $V(x, y)$ would decrease from the negativity of the toxins).

The planar-optimal nucleal radius is sufficient for covering the plane, but is far from optimal for the unusual constraints imposed within an arbitrary polygon. To see why, consider the placement of circles with respect to polygon's edges as shown in Figure-3. The two shaded areas, A1 and

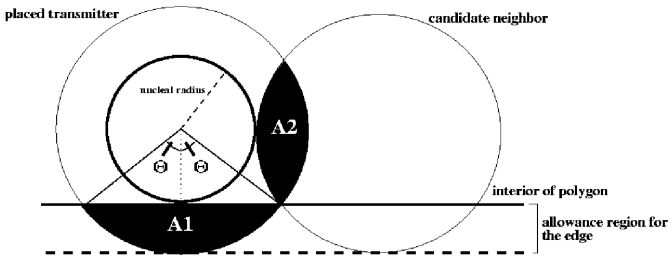

Fig. 3. Optimal geometry along polygon boundary

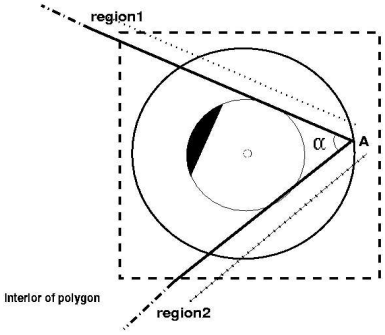

Fig. 4. Adaptation with respect to boundary

A2, represent "wasted" coverage in some sense because A1 is outside the polygon and A2 is covered by two circles. Consider what happens to each of these shaded areas as the circles are moved. If the left circle is moved upwards (into the interior of the polygon) to decrease A1, then A2 will need to increase because the right circle will have to be closer to the left one (to cover the polygon boundary). Similarly to decrease A2, we will have to increase A1. Thus, there is some value of A1 and A2 that minimizes their sum (and therefore minimizes wasted coverage). More precisely, based on the angle 2Θ subtended by the chord above A1, we wish to minimize

$$\begin{aligned} f(\Theta) &= A_1 + A_2 \\ &= \pi r^2 - 3/2 r^2 sin(2\Theta) - \Theta r^2. \end{aligned}$$

The minimum is found to occur at $\Theta = 55°$, which is then used to define the optimal nucleal radius for circles that lie along a polygon's edge. Note that the dashed line at the bottom of Figure-3 indicates the maximum extent to which circles overlap with the exterior of the polygon. We refer to this band around the polygon's edge as the *extension region* for that edge. as shown in the figure.

The computation of this extension region bears some explanation since it is key to determining the nucleal radius. We first start by placing circles to cover the end-points of each edge, as shown in Figure-5. Now, it would seem that the above definition of nucleal radius might suffice for all circles. However, three factors conspire to force differing nucleal radii on circles. The first is that polygon edges are of arbitrary length and so an integral number of circles may not cover an edge. Accordingly, we use the optimal length of A1 from above to compute the (possibly fractional) number of circles needed for a particular polygon edge. This number is rounded off to the nearest integer to decide how many circles will be used to cover the polygon edge. The circles are then equally spaced so that their A1 chord lengths are now different from the optimal value of A1 as computed above. This results in smaller or larger extension regions for each polygon edge.

The second factor arises from considering the covering of polygon corners as shown in Figure-4. To avoid requiring two circles in covering a corner with acute interior angle (α, in the figure) and to minimize wasted space, the circle perimeter should pass through the corner. This results in defining a different nucleal radius of the corner-covering circle as $r \sin(\frac{\alpha}{2})$.

The third factor complicating the nucleal structure arises from the optimal packing wrought by differing nucleal radii in the vicinity of a circle. For example, in Figure-4 the extension regions of the two polygon edges meeting at the corner may have different widths. Because of this and nucleal radii difference for corner-covering circles, the degree of circle overlap along one edge (corresponding to area A2 in Figure-3) may be different from the overlap along the other edge. This overlap can be computed exactly as described earlier. In the example of Figure-4, the overlap will need to be higher along the top edge than along the other one because its extension region is narrower. To enable this packing to occur, the effective nucleus of the corner circle needs to be reduced along the top edge, but not along the lower edge. This results in an irregular or "truncated" nucleus as shown in the figure. Again, the width of the shaded area can be computed exactly.

B. Initial distribution of nutrients and toxins

Because we wish to exploit the above geometric analysis of polygon corners and edges, our algorithm initially places circles to cover corners and then places circles along polygon edges. To orchestrate such a bias towards the polygon perimeter, we assign nutrient values in higher concentrations along the polygon boundary, with the highest concen-

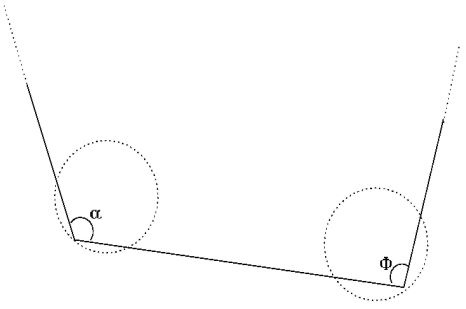

Fig. 5. Extension length calculation

tration at the corners and gradually decrease the level of nutrient as a function of distance from the boundary. This graded nutrient distribution is the analogue of the chemical gradient in chemotaxis. To prevent circles from intruding too deep into the exterior of the polygon, high toxin concentrations are placed on grid points in the exterior of the polygon outside extension regions.

C. Local adaptations

Local adaptations form a key component of our algorithm and are motivated by the observation that nucleal radii and shape are alone not sufficient to ensure efficient coverage. In some cases, a definition of the nucleal radius must be accompanied with a *re-distribution* of toxins or nutrients. The re-distribution is intended to preserve the total level of toxin inside the nucleus even if the shape was made irregular to accommodate local geometry.

Consider the irregular nucleus in Figure-4. As mentioned earlier, part of the nucleus needs to be truncated to accommodate a closer packing of circles along one or the other edge at the corner. Recall also that we place a high level of toxin inside the nucleus to control the degree of overlap of circles. Now, if we simply reduced the size of the nucleus without maintaining the same level of toxin inside the nucleus, it is possible that a third circle will overlap the affected region of the nucleus and compute a higher than normal "value" using the convolution in equation (1). This affects the optimality of circle placements that do not need the accommodation provided by the truncation. Accordingly, we have discovered that better solutions are obtained by maintaining the same level of toxin, via re-distribution of the total toxin with the nucleus.

We refer to the toxin re-distribution above as a local adaptation. Another adaptation arises in the interior of the polygon as circle placements get "squeezed" into arrangements that deviate from the hexagonal planar-optimal arrangement. Consider the two circles A and B in Figure-6. Assume they have been already placed and that their nuclei have been computed and toxins assigned inside the nuclei. Because these two circles can come (towards each other) from incremental growth initiated at different polygon edges, their overlap may not be planar-optimal. This means that if a potential third circle is located at point C and covers the space in between, including the point P, then the extent of overlap that C has with A will be different from the overlap C has with B. To accommodate these differences, we need to re-structure the nuclei of circles A and B so that no toxins are present in the shaded regions.

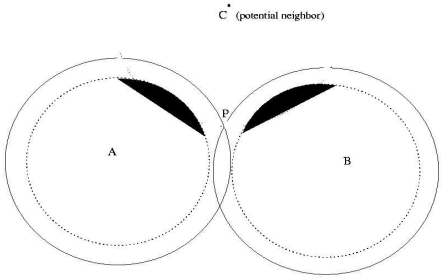

Fig. 6. Adaptation with respect to neighbors

Note that both these shaded areas and the optimal location of C can be computed knowing the centers of A and B. Once these are computed, the toxins in the shaded areas are distributed amongst the grid points in the truncated nucleus. The two adaptations above are the two main modifications needed to accommodate the irregular geometry imposed by an irregular polygon.

D. Complete Algorithm

A formal description of the algorithm is given below following the definition of some terms:

- P: the polygon, including its boundary.
- ∂P: the polygon's boundary
- $P^\circ = P - \partial P$: interior of the polygon
- P': the exterior of the polygon
- τ: user-supplied coverage threshold value
- r: transmission radius of each unit
- G: the set of grid points
- $e_{j,ext}$: the extension region of edge $e_j \in \partial P$
- K_{max}: maximum positive value that can be assigned to a grid point
- $|a, b|$: distance function where the first parameter a is a point and the second parameter b is either a point or line segment
- T : set of currently assigned units
- U : set of units in the neighborhood of the recently placed unit
- E : set of boundary edges in the neighborhood of the recently placed unit

ALGORITHM_COVERAGE (P, τ, r)

1. **begin algorithm**
2. $T = \emptyset$
3. $coverage = 0\%$
4.
5. *// Find the extension region for each boundary edge*
6. **for each** $e_j \in \partial P$
7. Compute extension region $e_{j,ext}$ for each edge e_j
8. **endfor**
9.
10. **for each** $(x, y) \in G$ *//For every point in grid*
11.
12. **if** $(x, y) \in P^\circ$ //Grid point inside polygon
13. *//Add value depending on distance from edge*
14. **if** $(\exists e_i \in \partial P : |(x, y), e_i| \leq e_{i,ext})$
15. **for each** $e_j \in \partial P : |(x, y), e_j| \leq e_{j,ext}$
16. $v(x, y) = v(x, y) + \frac{r}{|(x,y),e_j|}$
17. **endfor**
18. **else**
19. $v(x, y) = 1$
20. **else if** $(x, y) \in P'$ *//Grid point outside*
21. *//Assign unit value if it belongs to an extension region*
22. **if** $(\exists e_j \in \partial P : |(x, y), e_j| \leq e_{j,ext})$
23. $v(x, y) = 1$
24. **else**
25. $v(x, y) = -K_{max}$
26. **endfor**
27.
28. **while**$(coverage < \tau)$
29.
30. Choose (x^*, y^*) to maximize $V(x, y)$

31. Place a new unit t at (x^*, y^*)
32. T = T $\cup$ $\{t\}$
33. Compute neighborhood E, U of t
34. Assign $r_{nuc}(t)$ with respect to E and U
35.
36. *//First compute default nutrient/toxin distribution after placement*
37. **for each** (x, y) : $(|(x, y), (x^*, y^*)| < r$ **and**
38. $\nexists t' \in (T - \{t\}) : |(x, y), (t'.x, t'.y)| < r_{nuc}(t'))$
39. **if** $(|(x, y), (x^*, y^*)| < r_{nuc}(t))$
40. $v(x, y) = \frac{-K_{max}}{|(x,y),(x^*,y^*)|}$
41. **else**
42. $v(x, y) = 1$
43. **endfor**
44.
45. *//Apply adaptations with respect to neighborhood*
46. **for each** $t' \in U$
47. Apply unit-unit adaptations to t and t'
48. **endfor**
49. **for each** $e_j \in E$
50. Apply unit-boundary adaptations to t with respect to e_j
51. **endfor**
52.
53. *//Modify values of grid points on cluster of units'*
54. *//inner boundary*
55. **for each** (x, y) : $\exists \hat{t} \in T, (|(x, y), (\hat{t}.x, \hat{t}.y)| \cong r)$ **and**
56. $\nexists \bar{t} \in (T - \{\hat{t}\})(|(x, y), (\bar{t}.x, \bar{t}.y)| < r)$
57. $v(x, y) = v(x, y) + 1$
58. **endfor**
59.
60. Update *coverage*
61. **endwhile**
62. **return** T
63. **end algorithm**

The algorithm description above is intended for completeness. Note that lines [53-58] describe how the algorithm proceeds once the regions close to the boundary are covered. By favoring neighborhood of the units according to their placement order, a continuous cluster of units is built up from boundary to center which yields complete coverage of the polygon.

E. Discussion

After seeing the incorporation of geometric constraints, the reader familiar with biological systems might wonder if *phyllotaxis* might not be the more appropriate metaphor for our approach. After all, phyllotaxis is concerned with regular arrangements of biological units (leaves, cells or organs) in patterns. Indeed, phyllotaxis has been mathematically modeled as a dynamical system [5] and models have been used in generating realistic patterns for use in computer graphics [13]. Note however that phyllotaxis is concerned with the generation of patterns with known mathematical structure (such as the commonly occurring plant spiral). Thus, while it is true that our algorithm models local geometry, its use of graded nutrient and toxin quantities suggests that chemotaxis is the closer metaphor.

III. Extension for obstacles and signal attenuation

The propagation of a wireless signal indoors is affected by many factors including indoor obstacles such as walls. To incorporate such effects in coverage, we extend our framework to a two-phase approach. The first phase uses chemotaxis to assign coverage without consideration of obstacles. The second phase consists of modeling signal quality in the presence of obstacles and making an iterative adjustment of the initial chemotactic assignment. This section focuses on the second phase, which itself consists of two parts.

We begin by describing the signal propagation model used in the presence of obstacles. Propagation models are either based on the direct path between sender and receiver or use ray-tracing for a more precise determination of path loss [22]. A comparison of these two types of models is given in [30]. We use the statistical model of [31] that incorporates indoor obstacles in to the free space propagation model. The mean path loss $P_L(d)$ in dB at a distance d is given by

$$\begin{aligned} P_L(d)[dB] &= 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) \\ &+ \mathrm{n_p} * (\text{partition attenuation})[dB] \\ &+ \mathrm{n_w} * (\text{wall attenuation})[dB] \end{aligned} \tag{2}$$

where n_p, n_w are the number of soft partitions and concrete walls in the direct line of sight between sender and receiver. A polygon point at a distance d is said to be covered in this case if $P_L(d)$ is less than some given threshold T, a property of the transmitter units. From the path loss equation above, one can find the effective range of a unit $ER((p, q), (x, y))$ located at (p, q) at a receiving point (x, y). This can be calculated from equation (2) after eliminating the path loss caused by walls and partitions.

The first part of the second phase of the algorithm consists of appropriately modeling and evaluating signal quality. Here our approach is standard [31]: a point is said to be covered by a unit if the signal quality predicted at the point from the unit is sufficiently high. Thus, an obstacle might degrade the signal, whose strength at the point of interest can be computed using equation (2). Hence, for any grid point one can compute the level of coverage from neighboring circles. In the second part of the adjustment phase, we apply a modified version of simulated annealing to adjust circle positions so that as many grid points as possible are covered.

We refer the reader to [1] for a comprehensive description of simulated annealing and only briefly describe the main ideas here. Simulated annealing explores the state space of candidate solutions by defining for each state a collection of "neighboring" states via a neighborhood function. Because a greedy search of the state space may result in getting stuck at a local minimum, simulated annealing provides an opportunity to accept occasional, probabilistic jumps to higher-cost states. The idea is that such jumps help lead the process out of local minima to facilitate further searching of the state space. This acceptance of jumps to higher-cost states is itself gradually lowered by decreasing a parameter referred to as the *temperature*.

Most annealing implementations leave the neighborhood function intact while gradually decreasing the temperature. However, in some geometric optimization problems, it has been found valuable to simultaneously also adjust the neighborhood function [21],[29], We take this approach

here and reduce the neighborhood as the algorithm proceeds.

To complete describing the algorithm, we need to define the neighborhood, the temperature schedule and the adjustments made to the neighborhood. For the temperature schedule, we have experimented with both linear and polynomial cooling schedules (see [1]).

We now describe the neighborhood function. Recall that the first part of this phase identifies all uncovered grid points. More formally, let $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ be the set of uncovered points and $\{(p_1, q_1), \ldots, (p_m, q_m)\}$ be the set of circle centers computed in the first phase. For each uncovered grid point (x_i, y_i), we identify the least distance $\gamma(x_i, y_i)$ a neighboring circle needs to be moved toward the grid point in order to cover it using the effective range ER equation found in Section-**??**:

$$\gamma(x_i, y_i) \;=\; \min_{1 \le k \le m} \{|(x_i, y_i), (p_k, q_k)| - ER\,((p_k, q_k), (x_i, y_i))\}$$

The initial neighborhood is defined as the average of such values over all uncovered points. Since the distance $\gamma(x_i, y_i)$ changes as the circle placements are adjusted, the neighborhood changes as the algorithm proceeds.

IV. Extension for Capacity Based Planning

We have assumed in the previous sections that the data rate demand is uniformly distributed (unit demand per grid point) among the indoor region to be covered and total demand from coverage area is within the capacity of each transmitter (unit). The problem with this *infinite capacity* assumption is that the wireless infrastructure may not meet the quality-of-service users are expecting during high load.

One approach to increasing capacity is to add more units so that each grid point is covered by multiple units. However, such an approach might require additional hardware or software to distinguish between the units during operation and to balance the load. Our approach, instead, is to use smaller radii to cover regions with high demand. The idea is to let the range r be inversely proportional to the demand. In this manner, high-demand sub-regions will be assigned more units.

Suppose a high demand region has M units of demand per grid point. The new range is taken to be $r' = \frac{r}{\sqrt{M}}$. Thus, an algorithm for the variable-demand case can use the previous algorithm as follows: If $P = \bigcup_{j=1}^{n} p_j$ and capacity requirement for each of the subpolygon is $p_{j,cap}$, the extended algorithm is

Algorithm_CapacityBasedCoverage (P, τ, r)
1. **begin algorithm**
2. $T = \emptyset$
3. **for each** $p_j \in P$
4. $\quad T = T \cup$ Algorithm_Coverage $(p, \tau, \frac{r}{\sqrt{p_{j,cap}}})$
5. **end for**
6. **return** T
7. **end algorithm**

V. Algorithm Evaluation

We have evaluated our algorithm using randomly generated polygons. For concreteness, we have used transmission parameters arising from the 914MHz band in equation (2). The grid spacing was taken to be 1m and circles with radii 30m and 50m were used in two sets of simulations. Figure-7 shows the corners and edges of a seven-sided irregular polygon covered by circles and the placement obtained at the end of the chemotaxis phase of the algorithm. Three attenuating walls are shown in light gray; the longer one has 6dB attenuation whereas the smaller two have 5dB attenuation. Figure-8 and Figure-9 shows the solution after the second phase(extension for obstacles) for units with 30m and 50m radii for the same region. All the figures also display the Voronoi regions arising from the point set consisting of the circle centers. The Voronoi regions are used in determining the extent of coverage.

We compare our algorithm with a standard implementation of simulated annealing for the same problem. Note that in comparison with our algorithm, this simulated annealing implementation directly solves the whole problem of both allocating circles and placing them. We experimented with various temperature schedules and neighborhood functions for simulated annealing; in particular, the neighborhood had to be defined to allow the number of circles to change. We also compare our algorithm with the *regular installation* heuristic described in Section I-A. The two existing techniques, simulated annealing and regular installation, delineate the extremes of a spectrum of performance: regular installation is simple and fast but suboptimal whereas simulated annealing is time-consuming yet produces significantly better solutions.

The overall qualitative results can be summarized as follows:

- Our chemotaxis algorithm almost always produces better solutions than either simulated annealing or regular installation – about 10-15% fewer circles.
- With the densest grid, the chemotaxis algorithm (the first phase) runs orders of magnitude faster than simulated annealing, as would be expected of a technique that directly constructs a solution instead of searching the state space with an average ratio close to 1:700. Due to the high grid density and optimizations, an average ratio of 10^3:1 exists with respect regular installation. Our algorithm therefore lies in the middle of the computational spectrum as compared to these two classical approaches.
- The difference between chemotaxis and the other methods is greater when the polygon is irregular, as measured by the Brinkhoff complexity [37] of the polygons.

Note that we use simulated annealing as a base-line comparison. Our goal here is to explore a new paradigm in directly constructing a solution. Hence, in some ways, our approach may serve as a valuable *starting solution* for state-space algorithms such as annealing or tabu search.

Table-I and Table-II depict the results of five sample experiments that compare coverage (as a percentage of the polygon area) and number of circles required for the three approaches with units of range 30m and 50m. The first column shows the *complexity* of the polygon.

We have used the formal measure of the [37] for polygon complexity whose results are multiplied by 100 to show a

	Complexity		*Chemotaxis*	*Annealing*	*Regular Install.*
Expt. 1	12.69	Coverage	99.91%	99.09%	100%
		Num. circles	46	64	59
Expt. 2	11.17	Coverage	99.79%	99.32%	100%
		Num. circles	48	65	58
Expt. 3	7.49	Coverage	99.92%	98.57%	100%
		Num. circles	63	72	68
Expt. 4	0	Coverage	99.93%	99.53%	100%
		Num. circles	58	73	64
Expt. 5	1.52	Coverage	99.91%	98.92%	100%
		Num. circles	57	69	60

TABLE I

Coverage yield vs number of units, 30m range

	Complexity		*Chemotaxis*	*Annealing*	*Regular Install.*
Expt. 1	12.69	Coverage	99.80%	99.46%	100%
		Num. circles	20	23	23
Expt. 2	11.17	Coverage	99.83%	98.86%	100%
		Num. circles	19	22	24
Expt. 3	7.49	Coverage	99.99%	98.83%	100%
		Num. circles	27	27	28
Expt. 4	0	Coverage	99.95%	99.16%	100%
		Num. circles	21	25	24
Expt. 5	1.52	Coverage	99.93%	99.11%	100%
		Num. circles	22	23	23

TABLE II

Coverage yield vs number of units, 50m range

percentage scale. The model emphasizes the global shape of the object (how it deviates from its convex hull) and local vibration of its boundary by considering the following factors

• *frequency of vibration* : Notches define the nonconvex corners of the polygon. The normalized number of notches in a polygon *pol* is

$$notches_{norm}(pol) = \frac{notches(pol)}{vertices(pol) - 3}$$

where the denominator is the upper limit on number of notches. The frequency of vibration is

$$freq(pol) = 16(notches_{norm}(pol) - 0.5)^4 - 8(notches_{norm}(pol) - 0.5)^2 + 1$$

• *amplitude of vibration* : As the objects get more complex, their boundary increases. This boundary amplitude increase is referred as

$$ampl(pol) = \frac{boundary(pol) - boundary(convexhull(pol))}{boundary(pol)}$$

• *deviation from the convex hull* : For the global shape of the object the following convexity measure is defined

$$conv(pol) = \frac{area(convexhull(pol)) - area(pol)}{area(convexhull(pol))}$$

The complexity of a polygon $compl(pol)$ is defined as follows

$$compl(poly) = 100 * ((0.8) * ampl(pol) * freq(pol) + (0.2) * conv(pol))$$

A convex polygon has measure close to 0 and values larger than 40 are for very complex polygons. Values smaller than 10 are typical for polygons representing indoor areas. All these experiments have emphasized the chemotaxis phase of the algorithm which constructs the direct solution and did not include extension models. User supplied tolerance values are selected randomly from the [99.5 - 100]% interval.

A. Results for Variable-Demand Case

We have tested effectiveness of our variable-demand version, using experiments 2 and 3 from tables I and II, corresponding to the indoor area of Figure-7. For the obstacles and signal attenuation, when walls are taken into account with 30m range experiment, the coverage value reduces to 97.89% from 99.79%. Application of the second phase has improved the coverage in case of obstacles to 98.57%. Figure-10 shows the centers of the units for the result of demand-based coverage extension when the same indoor area to be covered is divided into three regions A, B, C with demands 5,3,2 for each grid point respectively.

VI. Summary

In this paper, we proposed a new biologically-inspired algorithm for a class of indoor wireless coverage problems.

Initial experimentation with the algorithm appears to show promising results: the algorithm results in fewer wireless units being allocated than two existing algorithms for the problem: simulated annealing and the regular installation method.

References

[1] E.Aarts and J.Korst. Simulated Annealing and Boltzmann Machines, A Stochastic Approach to Combinatorial Optimization and Neural Computing, *John Wiley and Sons* ,1989.

[2] J.P. Adler. How Motile Bacteria Sense and Respond to Chemicals. *Olfaction and Taste IX*, Vol. 510, pp.95-97, 1987.

[3] H.R. Anderson and J. P. McGeehan. Optimizing Microcell Base Station Locations Using Simulated Annealing Techniques, *IEEE 44th Vehicular Technology Conference*, pp. 84-89, March 1991.

[4] J.P. Armitage. Bacterial Motility and Chemotaxis. *Science Progress*, Vol. 76, pp. 451-477, 1992.

[5] P. Atela, C. Gole and S. Hotton. A Dynamical System for Plant Pattern Formation: A Rigorous Analysis. Smith College, 2001.

[6] E. Beltrami. Mathematics for Dynamic Modeling. *Academic Press*, 1987.

[7] C.Bishop. Neural Networks for Pattern Recognition, *Oxford University Press*, 1996.

[8] H.Bronnimann and M.Goodrich. Almost optimal set covers in finite vc-dimension, *Discr. Comput. Geom.*, Vol. 14, Pages 463-479, 1995.

[9] J.H. Conway, N.J.A. Sloane. Sphere Packings,Lattices and Groups, *Springer-Verlag*, New York, 1988.

[10] J.C. Culberson and Robert A. Reckhow. Covering Polygons is Hard, *Journal of Algorithms*, Vol. 17,2-44, Pages 745-770, 1994.

[11] K. Deb. Multi-Objective Optimization Using Evolutionary Algorithms, *John Wiley and Sons*, 2001.

[12] S. Fortune, D.M. Gay, B. Kernighan, O. Landron, R. Valenzuela and M.H. Wright. WISE Design of Indoor Wireless Systems, *IEEE Computational Science and Engineering*, Vol.2, No.1, pp. 58-68, 1995.

[13] D.R.Fowler, P.Prusinkiewicz and J.Battjes. A Collision-Based Model of Spiral Phyllotaxis. *ACM SIGGRAPH*, pp.361-368, 1992.

[14] T. Frukwirth, P. Brisset. Optimal Placement of Base Stations in Wireless Indoor Telecommunication, Special Issue on Practical Applications of Constraint Technology, *IEEE Intelligent Systems Magazine*, Vol.15, No. 1, pp. 49-59, January 2000.

[15] D.Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning, *Addison-Wesley*, 1989.

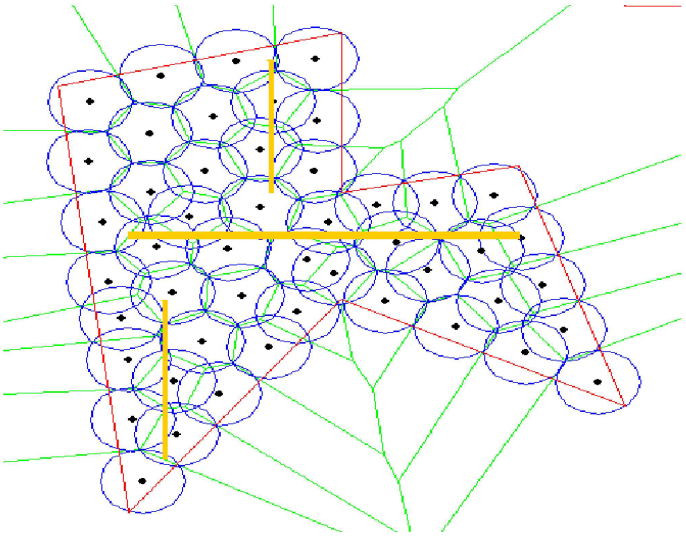

Fig. 8. Result of second phase with 30m-range units

[16] A. Howard, M. J. Matari and S. Sukhatme, Mobile Sensor Network Deployment using Potential Fields: A Distributed Scalable Solution to the Area Coverage Problem,*Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems(DARS02) Fukuoka, Japan, June 25-27 2002*

[17] D.S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI, *Journal of ACM*, Vol. 32, No. 1, pp.130-136, January 1985.

[18] D.S. Hochbaum and W.Maass. Fast approximation algorithms for a nonconvex covering problem, *Journal of Algorithms*, Vol. 8, No. 3, pp. 305-323, September 1987.

[19] L.J. Ibbetson and L.B. Lopes. An Automatic Base Site Placement Algorithm, *IEEE 47th Vehicular Technology Conference*, Vol. 2, pp. 760-764, 1997.

[20] J. Kabara,P. Krishnamurthy and D. Tipper. Capacity Based Network Planning for Wireless Data Networks, *Proc. Information Society Technologies(IST) Mobile Summit*, Barcelona,Spain, September 2001.

[21] S.A. Kravitz and R. Rutenbar. Placement by simulated annealing on a multiprocessor, *IEEE Trans. on Computer-Aided Design*, Vol. 6, pp. 534-549, 1987.

[22] P. Kreuzgruber, T. Brundl, W. Kuran and R. Gahleitner. Prediction of Indoor Radio Propagation with the Ray Splitting

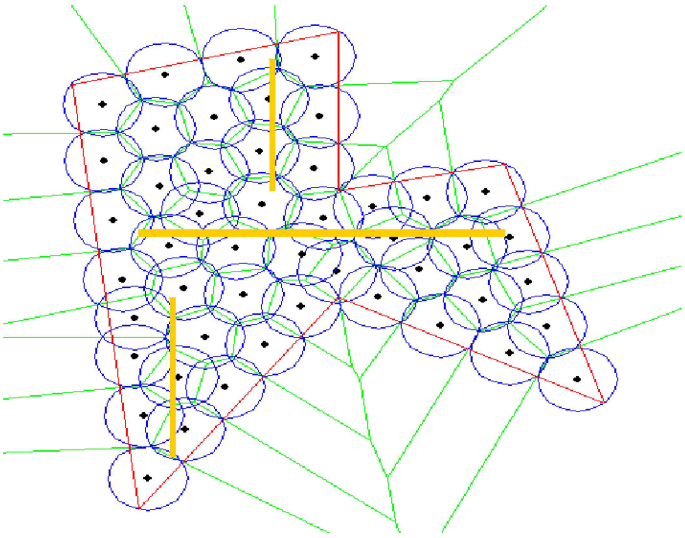

Fig. 7. Result of initial placement without taking walls into account

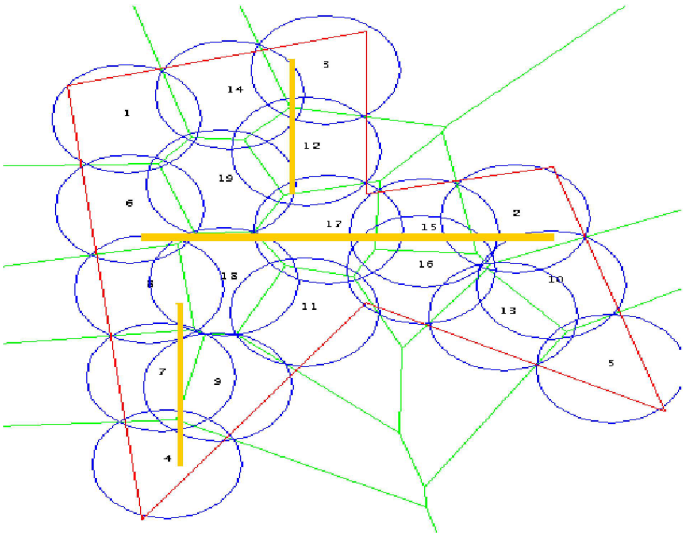

Fig. 9. Final placement with 50m-range units

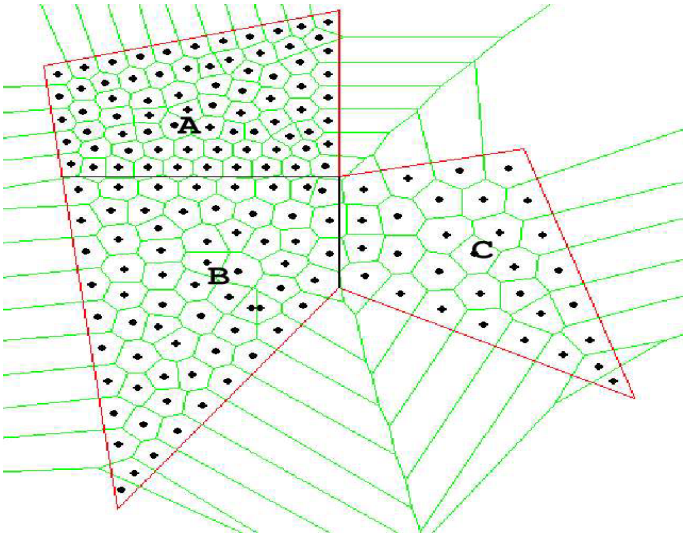

Fig. 10. Result for demand based coverage problem

Model Including Edge Diffraction and Rough Surfaces, *IEEE 44th Vehicular Technology Conference*, Vol. 2, pp. 878-882, June 1994.

[23] S. Meguerdichian,F. Koushanfar,M. Potkonjak,M.B. Srivastava. Coverage Problems in Wireless Ad-Hoc Sensor Networks, *IEEE Infocom*, Vol. 3, pp. 1380-1387, April 2001.

[24] S. Meguerdichian, F. Koushanfar, M. Potkonjak, G. Qu. Exposure in wireless ad-hoc sensor networks, *Seventh international conference on mobile computing and networking*, pp. 139-150, July 2001.

[25] J.B.M. Melissen and P.C. Schuur. Improved Coverings of a Square with Six and Eight Equal Circles, *Electronic Journal of Combinatorics*, Vol. 3, No. 1, 1996.

[26] A. Molina, G.E. Athanasiadou and A.R. Nix. The Automatic Location of Base Stations for Optimised Cellular Coverage: A New Combinatorial Approach, *IEEE 49th Vehicular Technology Conference*, pp. 606-610, 1999.

[27] S.D.Muller, J.Marchetto, S.Airaghi and P.Koutmoutsakos. Optimization Based on Bacterial Chemotaxis. *IEEE Trans. Evolutionary Computation*, Vol. 6, No. 1, Feb 2002.

[28] J.A. Nelder and R. Mead. A simplex method for function minimization, *Computer Journal*, Vol. 7, pp. 308-313, 1964.

[29] R.H.J.M. Otten and L.P.P.P Van Ginneken. Floorplan Design Using Simulated Annealing, *Proc. IEEE Int. Conf. on Computer-Aided Design*, Santa Clara, pp. 96-98, 1984.

[30] V. Sampath, C. Despins, B. Sultana, W. Lippler and G.Y. Delisle. Comparison of statistical and deterministic indoor propagation prediction techniques with field measurements, *IEEE 47th Vehicular Technology Conference*, Vol. 2, pp. 1138-1142, 1997.

[31] S.Y. Seidel and T.S. Rappaport. 914 MHz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings, *IEEE Transactions on Antennas and Propagation*, Vol. 40, No. 2, pp. 207-217, February 1992.

[32] D. Stamatelos and A. Ephremides. Spectral Efficiency and Optimal Base Placement for Indoor Wireless Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 14, No. 4, pp. 651-661, May 1996.

[33] K.S. Tang, K.F. Man and K.T. Ko. Wireless LAN Design using Hierarchical Genetic Algorithm, *Proc. 7th International Conference on Genetic Algorithms*, East Lansing, MI, pp. 629-635, July 1997.

[34] M. Vasquez and J-K. Hao. A Heuristic Approach for Antenna Positioning in Cellular Networks, *Journal of Heuristics*, Vol. 7, No.5, 2001.

[35] M.H. Wright. Optimization Methods for Base Station Placement in Wireless Applications, *48th IEEE Vehicular Technology Conference*, Vol. 1, pp. 387-391, 1998.

[36] R-H. Wu, Y-H. Lee and S-A. Chen. Planning System for Indoor Wireless Network, *IEEE Transactions on Consumer Electronics*, Vol. 47, No. 1, pp. 73-79, February 2001.

[37] T. Brinkhoff,H. P. Kriegel, R. Schneider,A. Braun, Measuring the Complexity of Polygonal Objects, *Proceedings 3rd ACM International Workshop on Advances in Geographic Information Systems*, Baltimore, MD, 1995, 109-118.