Globally Optimal Transmitter Placement for Indoor Wireless Communication Systems

J. He, A. Verstak, L. T. Watson *Fellow, IEEE*, C. A. Stinson, N. Ramakrishnan, and C. A. Shaffer
T. S. Rappaport, *Fellow, IEEE*, C. R. Anderson, *Student Member, IEEE*,
K. Bae, *Student Member, IEEE*, J. Jiang, *Student Member, IEEE*, and W. H. Tranter, *Fellow, IEEE*

***Abstract*—In this paper, a global optimization technique is applied to solve the optimal transmitter placement problem for indoor wireless systems. An efficient pattern search algorithm—DIRECT (DIviding RECTangles) of Jones, Perttunen, and Stuckman (1993)—has been connected to a parallel 3D radio propagation ray tracing modeler running on a 200-node Beowulf cluster of Linux workstations. Surrogate functions for a parallel WCDMA (wideband code division multiple access) simulator were used to estimate the system performance for the global optimization algorithm. Power coverage and BER (bit error rate) are considered as two different criteria for optimizing locations of a specified number of transmitters across the feasible region of the design space. This paper briefly describes the underlying radio propagation and WCDMA simulations and focuses on the design issues of the optimization loop.**

***Index Terms*—bit error rate, DIRECT algorithm, global optimization, power coverage, ray tracing, surrogate function, transmitter placement, wideband code division multiple access.**

I. Introduction

Optimal transmitter placement provides high spectral efficiency and system capacity while reducing network costs, which are the key criteria for wireless network planning [3]. As the complexity and popularity of modern wireless networks increases, automatic transmitter placement provides cost savings when compared to the traditional human process of site planning. Automatic design tools are being developed to offer efficient and optimal planning solutions. Besides [7], [13], and [19], S^4W (Site-Specific System Simulator for Wireless system design) is among the few known wireless system tools for in-building network design. It is being developed jointly by the Mobile & Portable Radio Research Group (MPRG) and the Problem Solving Environment (PSE) research group at Virginia Polytechnic Institute & State University. An optimization loop in S^4W is proposed to maximize the efficiency of simulated channel models and surrogate functions are proposed to reduce the cost of simulations. Transmitter placement optimization is one specific problem that can be solved by S^4W. An example of an S^4W model consisting of a propagation model, a wireless system model, and an optimizer is given in [24].

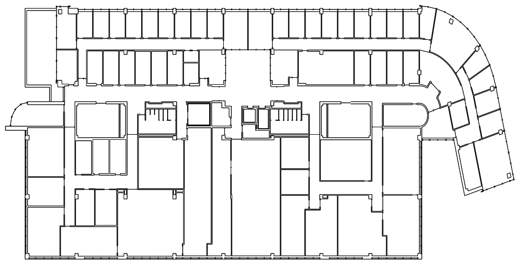


Fig. 1.1. Durham Hall, fourth floor.

In general, transmitter placement optimization is aimed at ensuring an acceptable level of wireless system performance within a geographical area of interest (Figure 1.1 shows an indoor environment for the present study) at a minimum cost. [7] considers the major performance factor to be the power coverage, defined as the ratio of the number of receiver locations with received power above an assumed threshold to the total number of receiver locations. This nonsmooth function leads to the rank based methods used by [7]. In [13] and [22], the objective function is based on several weighted

J. He, A. Verstak, L. T. Watson, C. A. Stinson, N. Ramakrishnan and C. A. Shaffer are with the Department of Computer Science, Virginia Polytechnic Institute, Blacksburg, VA 24061-0106.

T. S. Rappaport, C. R. Anderson, K. Bae, J. Jiang, and W. H. Tranter are with the Bradley Department of Electrical and Computer, Virginia Polytechnic Institute, Blacksburg, VA 24061-0111.

factors, such as covered area, interference area, and mean signal path loss. [4] proposes a QoS (Quality of Service)-based penalty function resulting in an unconstrained optimization problem. In the present work, two performance metrics form objective functions for optimal transmitter placement. The metrics are continuous penalty functions defined in terms of power levels (i.e., power coverage) and bit error rates at given receiver locations within the covered region. Both objective functions are devised to minimize the average shortfall of the estimated performance metric with respect to the corresponding threshold. 3D ray tracing is used as a deterministic propagation model to estimate power coverage levels and impulse responses within the region of interest for transmitter locations sampled by the optimization algorithm [20][21]. Surrogates for the Monte Carlo WCDMA simulation are used to estimate the BERs (bit error rates) for the second optimization criterion. Both the surrogates and the WCDMA simulation utilize the impulse responses estimated by the ray tracing model. Since 3D ray tracing and WCDMA simulation are computationally expensive, MPI-based parallel implementations are used in the present work.

The underlying optimization algorithm is known as DIRECT (DIviding RECTangles), a direct search algorithm proposed by Jones et al. [15]. It was proposed as an effective approach to solve global optimization problems subject to simple constraints. Jones et al. [15] named the algorithm after one of its key steps—dividing rectangles. DIRECT is a pattern search method that is categorized as a direct search technique by Lewis et al. [16]. Generally speaking, "pattern search methods are characterized by a series of exploratory moves that consider the behavior of the objective function at a pattern of points" [16], which are chosen as the centers of rectangles in the DIRECT algorithm. This center-sampling strategy reduces the computational complexity, especially for higher dimensional problems. Moreover, DIRECT adopts a strategy of balancing local and global search by selecting potentially optimal rectangles to be further explored. This strategy gives rise to fast convergence with reasonably broad space coverage. These features have motivated its successful application in modern large-scale multidisciplinary engineering problems [1], [2], and [25]. The present work extends that in [23], the second known application of DIRECT to wireless communication systems design other than the previous work in [12].

This paper is organized as follows. Section II presents the parallel 3D ray tracing model. Section III describes the parallel WCDMA simulator and the surrogate functions. An overview of the DIRECT algorithm is given in Section IV, followed by a description of dynamic data structures. Section V describes the formulations of two objective functions for optimizing transmitter locations in terms of both power coverage and bit error rate. In Section VI, optimization results are presented and analyzed. Finally, Section VII summarizes some key contributions of the present work and suggests directions for future research.

II. Ray Tracing Propagation Model

Received impulse responses are approximated with a 3D ray tracing propagation model that is based on geometrical optics. Electromagnetic waves are modeled as rays that are traced through reflections and transmissions through the walls. Beams [6] are shot from geodesic domes drawn around transmitters. Each beam is a triangular pyramid formed by the point location of the transmitter and one of the triangles on the surface of the dome. Essentially, the spherical wavefront is triangulated and the 3D sphere is split into pyramidal beams. Following the argument in [21], all such beams are disjoint and have nearly the same shape and angular separation. Only the central ray of each beam is traced to identify reflection locations. However, the whole beam is used for ray-receiver intersection tests. Once an intersection with a receiver location is detected, a ray will be traced back from the receiver to the transmitter through the sequence of reflections and transmissions (penetrations) encountered by the beam. The illustration of this process in 2D is given in Figure 2.1. Figure 2.2 depicts a fast intersection test of a beam with a grid of receiver locations. Neither diffraction nor scattering are modeled for computational complexity reasons, although these phenomena play an important role in propagation [18]. Octree space partitioning [11] and image parallelism with dynamic scheduling [8] are used to reduce simulation run time.

Although material parameters and incidence angles affect losses in a wireless channel, a constant 6 dB reflection loss (same as in [20]) and a constant 4.6 dB transmission (penetration) loss (the loss for plaster board in [5]) are assumed. The power contribution

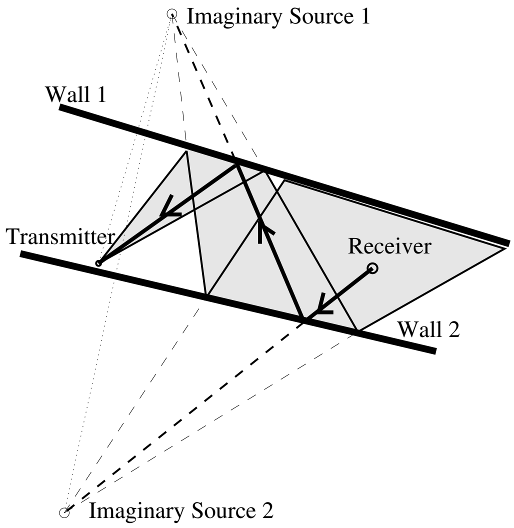


Fig. 2.1. 2D beam tracing: a beam (shadowed region) is traced from the transmitter location to the receiver location through two reflections, and then a ray (bold line) is traced back.

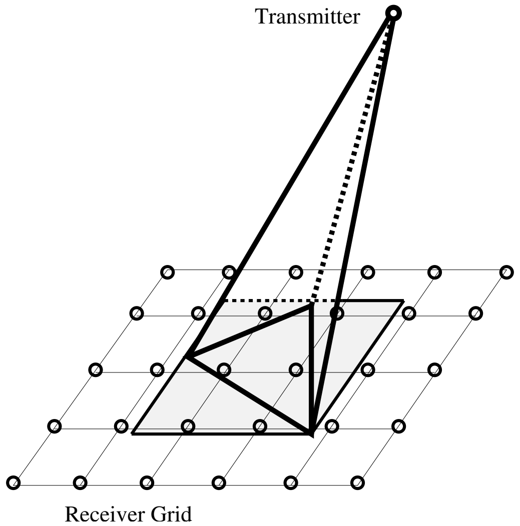


Fig. 2.2. Beam intersection with a receiver grid: only the locations inside of the bounding box of the projection of the beam onto the grid (shadowed region) are tested for intersection with the beam pyramid.

of each ray, in dBW, is calculated according to the model developed in [21]:

$$P_j = P(d_0) - 20\log_{10}(d/\lambda) - nL_r - mL_t, \qquad (2.1)$$

where P_j is the power of the j-th ray, d is the total distance traveled by the ray, $P(d_0)$ is the transmitter power at a reference distance d_0 from the transmitter, n and m are the numbers of reflections and transmissions, $L_r = 6$ dB and $L_t = 4.6$ dB are reflection and transmission losses, and λ is the wavelength.

The ray tracer has been validated and calibrated with a series of measurements in the corridor of the fourth floor of Durham Hall, Virginia Tech. An ultrawideband sliding correlator channel sounder [18] operating at 2.5 GHz and outfitted with omnidirectional antennas was used to record impulse responses at six separate locations. The sliding correlator utilized an 11-bit, 400 MHz pseudo-noise spreading code for a time domain multipath resolution of 2.5 nanoseconds and a dynamic range of 30 dB. Simulated power delay profiles were post-processed and compared to the measured ones location by location.

Comparing ray tracer output with a physical channel requires accounting for antennas and resampling the signal to match the sampling rate of the measurement system. The same conversion sequence was used for both validation against measurements and interfacing with the WCDMA simulation. The received E-field envelope of ray j (in V/m) that arrived at time t_j is $E_j = \sqrt{\eta 10^{0.1P_j}}$, where P_j is the output of the ray tracer (in dBW) and $\eta = 120\pi\ \Omega$ is the impedance of free space [18]. To account for antenna directivity, an omnidirectional antenna pattern must be applied to all E_js. The electric field that would be registered at time t_j by a hypothetical measurement system with infinite bandwidth resolution is

$$E'_j = E_j G_t G_r \cos\Theta_t \cos\Theta_r, \qquad (2.2)$$

where Θ_t and Θ_r are ray transmission and reception elevation angles relative to the horizon, and G_t and G_r are maximum transmitter and receiver antenna gains, respectively. Further, the discrete impulse response must be convolved with a Gaussian filter and sampled at uniform time intervals of width δ. The measurement system output samples with $\delta = 1$ ns while the WCDMA simulation used chip width $\delta \approx 260$ ns. The measured electric field E_k^m of bin k centered at time $k\delta$ is

$$E_k^m = C\sum_{j=1}^{Q} E'_j e^{i\phi_j} \int_{t_j-k\delta-\delta/2}^{t_j-k\delta+\delta/2} e^{-\tau^2/(2\sigma^2)} \mathrm{d}\tau, \qquad (2.3)$$

where Q is the number of rays, σ is the half-width of the Gaussian pulse (1.25 ns for measurements), and C is a scale factor that fits this generic equation to a particular system. Since most of the energy in the

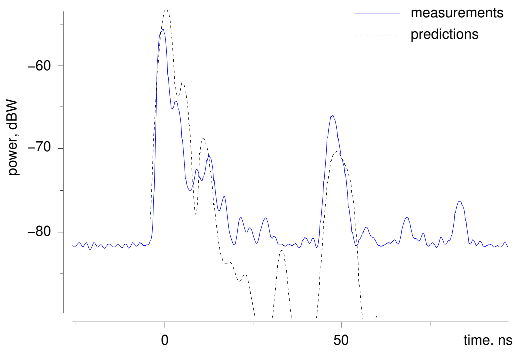


Fig. 2.3. Measurement vs. prediction of channel impulse response.

Gaussian pulse should fall into one time interval of width δ, assume that

$$C \int_{-\delta/2}^{\delta/2} e^{-\tau^2/(2\sigma^2)}\, d\tau = 1. \tag{2.4}$$

The complex factor $e^{i\phi_j}$ accounts for ray interference. Phase angles ϕ_j were determined from transmitter wavelength λ, total ray path length d_j, and number of reflections n (a 180 degree phase shift per reflection was assumed). Another interpretation of (2.3) is that every time bin registers a weighted average of the energies of all predicted rays, where the weight decreases exponentially as the time difference of the ray and the bin increases. Finally, $P_k^m = |E_k^m|^2/\eta$ gives the measured power of bin k, in watts.

Figure 2.3 shows measurements and predictions for one location with relatively strong multipath. As can be seen from the graph, the predictions are within 3–5 dB of the measurements, which is similar to the results achieved by earlier research [21]. The difference can be explained by device positioning errors (devices were positioned with ± 3 cm precision, which is crude given that the wavelength was 12 cm) and imprecise modeling of reflections. Additionally, small multipath components were missed by the ray tracer. These components are probably due to scattering and diffraction, which were not simulated. Geodesic tessellation frequency was 700 (9.8×10^6 beams) for calibration because the simulation results for frequencies above 700 were indistinguishable.

III. WCDMA Simulation

The ray tracing propagation model predicts a measured impulse response P_1^m, P_2^m, …, P_n^m of a wireless channel (see Section II). This propagation model does not directly predict the performance of any particular wireless system that operates in this channel. A meaningful performance metric is the bit error rate (BER) defined as the ratio of the number of incorrectly received bits to the total number of bits sent. The power level P_1^m at the receiver location maps directly to the BER of a narrowband system designed for $n = 1$. However, estimating the BER of a wideband system (designed for $n > 1$) in a mobile wireless environment usually involves analytically non-tractable problems [10]. This work uses simple least squares and multivariate adaptive regression splines (MARS [9]) to fit the results of a Monte Carlo simulation of a WCDMA system. The WCDMA simulation models channel variation due to changes in the environment as a random process [14]. Notice that channel variation due to receiver movement is modeled in both the ray tracing and the WCDMA simulations, but other kinds of variation are modeled only in the WCDMA simulation. This section outlines the WCDMA simulation and describes the surrogate functions used for optimization.

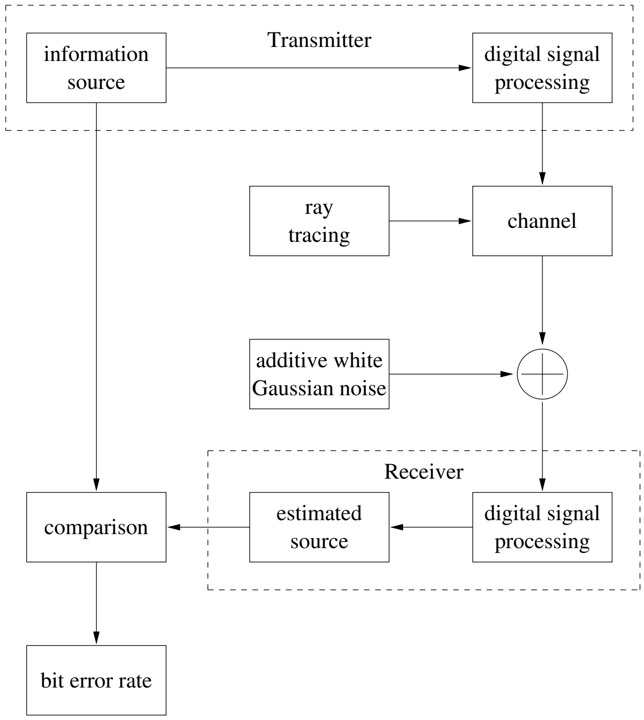


Fig. 3.1. Block diagram of the WCDMA simulator.

Figure 3.1 briefly describes the computational steps of the WCDMA simulator. The source module of the transmitter generates information data to be sent through a wireless channel. The generated information is processed with a series of digital signal processing algorithms to reduce the potential channel errors. The wireless channel is modeled as a linear time varying process in the present work. The channel is characterized by the impulse response predicted by the ray tracer. Before being sent to the receiver, the channel output is combined with Gaussian noise at the receiver front end. Similarly, the received distorted signal is processed with a series of digital signal processing algorithms by the receiver, which thereafter estimates the information bits to be compared with the original information bits for the BER.

The WCDMA simulation is computationally intensive since a satisfactory BER value ranges from 10^{-3} to 10^{-6}. The parallelized WCDMA simulator significantly speeds up the simulation process, but its run time is still far from practical for optimization problems. The BER depends on small-scale propagation effects that exhibit large variation with respect to receiver location. Practical coverage optimization problems involve wavelengths of less than a foot and areas of thousands of square feet. Four samples per wavelength should be taken to obtain meaningful aggregate results. Therefore, the BER results of the WCDMA simulation were approximated by simple models.

Consider a distribution of impulse responses in the environment shown in Figure 1.1, as measured by the receiver with the carrier frequency 900 HMz and the standard chip width $\delta \approx 260$ ns and a dynamic range (a ratio of the peak power to the noise level) of 12 dB. Empirically, 49% of the impulse responses have only one multipath component ($n = 1$), 42% have two multipath components where the first one is dominant ($n = 2,\ P_1^m \geq P_2^m$), 7% have two multipath components where the second one is dominant ($n = 2,\ P_1^m < P_2^m$), and the remaining 2% have three multipath components ($n = 3$). It turns out that simple models can approximate the BERs at the majority of the receiver locations. This work considers the first two cases that account for 91% of the data.

Given a measured impulse response P_1^m, P_2^m, …, P_n^m, define the relative strength of the first multipath component

$$p_1 = P_1^m / \sum_{1 \leq i \leq n} P_i^m \tag{3.1}$$

and the signal-to-noise ratio (SNR)

$$S = \max_{1 \leq i \leq n} \{10 \log_{10}(P_i^m / N_0)\} \tag{3.2}$$

(in dB), where N_0 is the noise power level (in watts).

The BER b_1 of a WCDMA system in the first case ($n = 1$, $p_1 = 1$) was approximated by

$$\log_e(b_1) = -0.251S - 2.258, \tag{3.3}$$

obtained by a linear least squares fit of the simulated BERs for $S = 0, 2, \ldots, 30$ in steps of 2 dB (16 points). In other words, the BER of a WCDMA system with a single path is a simple monotonically decreasing function of the SNR. This observation justifies the use of power levels to predict system performance when there is no multipath. However, using the strongest multipath component to predict the BER does not work when $n > 1$.

The second case ($n = 2,\ p_1 \geq 0.5$) was approximated using MARS models. The MARS models provided a more accurate fit to the data in comparison with the previously used linear least squares fit ([23]), reducing both the relative and absolute error. The MARS fit is a sum of products of univariate functions in the form

$$f(x) = a_0 + \sum_{n=1}^{M} a_n \prod_{k=1}^{K_n} B_{kn}\big(x_{v(k,n)}\big).$$

In this model, the multivariate spline basis functions are denoted by B, and their associated coefficients by a. This expansion of spline basis functions determines the number of basis functions, M, as well as product degree and knot locations (number of splits that gave rise to B_n is denoted by K_n) automatically from the data. In this model, the covariates are represented by x, where $v_{(k,n)}$ label the predictor variables. MARS models were developed for three different coding choices: no channel coding, rate 1/3 coding, and rate 1/2 coding. The latter two cases both use FECC (forward error correction code) to improve the BER performance. A soft decision Viterbi algorithm is used in decoding the convolutional FECC, because it produces a smaller BER than the hard decision Viterbi algorithm. The difference between these two cases lies in the convolutional code rate. Rate 1/3 coding provides a better error correction mechanism than the rate 1/2 coding.

The data used to build the no channel coding model consisted of 63 points from a Cartesian product of $S = 0, 1, \ldots, 20$ and $p_1 = 0.9, 0.7, 0.5$, and 48 points from a Cartesian product of $S = 0, 1, \ldots, 15$ and

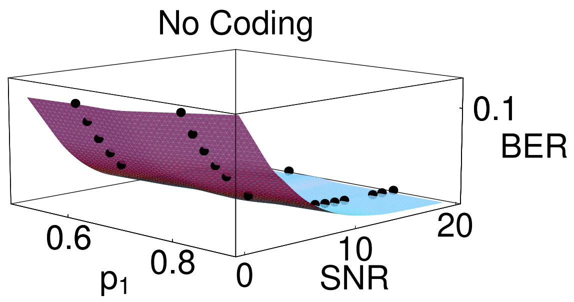


Fig. 3.2. The MARS surface plot of no channel coding model.

$p_1 = 0.9, 0.7, 0.5$ for the channel coding models. Plots of the fitted models reveal that the BER approaches zero as the SNR increases and that stronger multipath significantly improves performance for a fixed SNR. The latter needs some explanation because multipath is often thought of as an impairment that degrades the system performance. In this work, the SNR is defined in terms of the strongest component of the impulse response. When the SNRs of two channels that meet the criteria for this case are the same, the channel with a stronger second component contains more total power than the channel with a weaker second component. In this case, the benefits of more power outweigh the disadvantages of multipath.

Both surrogate models were validated with the simulated BER results. In the first case, the approximate values had an average relative error of 9.7% (0.9% minimum, 19.4% maximum) for the simulation output at $S = 1, 3, \ldots, 29$. In the second case, the approximate values had an average relative error of 12.8% and average absolute error of 0.0006 for the no channel coding model, 14.1% average relative error and 0.0005 average absolute error for the rate 1/3 coding model, and 19.9% average relative error and 0.0012 average absolute error for the rate 1/2 coding model. The validation sets for the second case consisted of 42 points for a Cartesian product of $S = 1, 2, \ldots, 20$ and $p_1 = 0.8, 0.6$ for the no channel coding model, and 32 points for a Cartesian product of $S = 1, 2, \ldots, 15$ and $p_1 = 0.8, 0.6$ for the channel coding models.

Finally, observe that the models for the two cases are not asymptotically matched. The simulated WCDMA receiver had two rake fingers, one of which was turned on or off depending on whether or not the second multipath component met the relative power threshold. Discontinuity can pose problems for the DIRECT optimization algorithm, which assumes Lipschitz continuity.

To summarize, this work considers two surrogate models for the BER of a WCDMA system. The first model was obtained using a least squares fit of the logarithm of the BER to a combination of channel characteristics. The second model was obtained using a MARS fit of the BER to a combination of channel characteristics. Empirically, both models cover 91% of the data with about 9.7% average relative error for the least squares model and 15.6% average relative error and 0.0008 average absolute error for the MARS models. The large average relative error is misleading since the larger errors occur where the BER is nearly zero, and the absolute error there is very small. The quality of the MARS approximation is apparent in Figure 3.2, which shows the MARS spline surface for the no channel coding model, the points used to construct it ($p_1 = 0.9,\ 0.7,\ 0.5$), and the points used to validate the approximation ($p_1 = 0.8,\ 0.6$). However, no confident claims can be made because the distribution of the fitted data is unknown. In particular, these models do not apply for $n > 2$. While the spline surrogate is quite good within the range of the fitted data, the surrogate function needs to be modified to give reasonable values for $S < 0$. The latter is crucial for using surrogates to solve the optimization problem with the DIRECT algorithm described in this paper.

IV. DIRECT

The multivariate DIRECT algorithm can be described by the following six steps [15].

Given an objective function f and the design space $D = \{x \in E^n \mid \ell \le x \le u\}$:

Step 1. Normalize the design space D to be the unit hypercube. Sample the center point c_i of this hypercube and evaluate $f(c_i)$. Initialize $f_{\min} = f(c_i)$, evaluation counter $m = 1$, and iteration counter $t = 0$.

Step 2. Identify the set S of potentially optimal boxes.

Step 3. Select any box $j \in S$.

Step 4. Divide the box j as follows:

- **(1)** Identify the set I of dimensions with the maximum side length. Let δ equal one-third of this maximum side length.
- **(2)** Sample the function at the points $c \pm \delta e_i$ for all $i \in I$, where c is the center of the box and e_i is the ith unit vector.

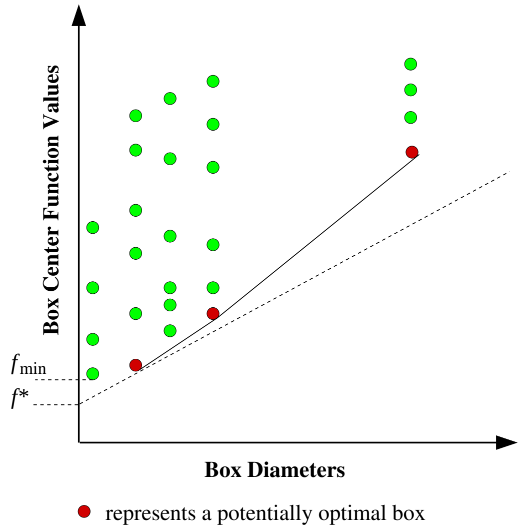


Fig. 4.1. Illustration of potentially optimal boxes on convex hull with ϵ test. Note that $f^* = f_{\min} - \epsilon|f_{\min}|$. Potentially optimal boxes are on the lower-right convex hull.

(3) Divide the box j containing c into thirds along the dimensions in I, starting with the dimension with the lowest value of $w_i = \min\{f(c+\delta e_i), f(c-\delta e_i)\}$, and continuing to the dimension with the highest w_i. Update $f_{\min}$ and m.

Step 5. Set $S = S - \{j\}$. If $S \neq \emptyset$ go to Step 3.

Step 6. Set $t = t+1$. If iteration limit or evaluation limit has been reached, stop. Otherwise, go to Step 2.

Steps 2 to 6 form a processing loop controlled by two stopping criteria— limits on iterations and function evaluations. Starting from the center of the initial hypercube, DIRECT makes exploratory moves across the design space by probing potentially optimal subsets. "Potentially optimal" is an important concept defined next [15].

Definition 4.1. Suppose that the unit hypercube has been partitioned into m (hyper) boxes. Let c_i denote the center point of the ith box, and let d_i denote the distance from the center point to the vertices. Let $\epsilon > 0$ be a positive constant. A box j is said to be *potentially optimal* if there exists some $\tilde{K} > 0$ such that for all $i = 1, \ldots, m$,

$$f(c_j) - \tilde{K}d_j \leq f(c_i) - \tilde{K}d_i, \tag{4.1}$$

$$f(c_j) - \tilde{K}d_j \leq f_{\min} - \epsilon|f_{\min}|. \tag{4.2}$$

Figure 4.1 represents the set of boxes as points in a plane. The first inequality (4.1) screens out the boxes that are not on the lower right of the convex hull of the plotted points, as shown in Figure 4.1. Note that $\tilde{K}$ plays the role of the (unknown) Lipschitz constant. The second inequality (4.2) prevents the search from becoming too local and ensures that a nontrivial improvement will (potentially) be found based on the current best solution. In Figure 4.1, $f_{\min}$ is the current best solution, but its associated box is screened out of the potentially optimal box set due to the second inequality (4.2). An example illustrating the behavior of DIRECT on a simple 2D function is given in [25].

Some modifications with respect to the stopping rules and box selection rules are proposed in the present implementation to offer more choices. Two new stopping criteria are (1) minimum diameter (terminate when the best potentially optimal box's diameter is less than this minimum diameter) and (2) objective function convergence tolerance (exit when the objective function does not decrease sufficiently between iterations). The minimum diameter of a hyperbox represents the degree of space partition, and therefore is a reasonable criterion for optimization problems requiring only some depth of design space exploration. The objective function convergence tolerance was inspired by some experimental observations in the later stages of running the DIRECT algorithm, when the objective function convergence tolerance test avoids wasting a great number of expensive function evaluations in pursuit of very small improvements. Figure 4.2 shows the history of $f_{\min}$ for the 20-dimensional Griewank function. The n-dimensional Griewank function

$$f(x) = 1 + \sum_{i=1}^{n} \frac{{x_i}^2}{d} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right), \tag{4.3}$$

where $d > 0$ is a constant to adjust the noise, has a unique global minimum at $x = 0$, and numerous local minima. DIRECT stops when the box holding the current $f_{\min}$ has reached the allowed minimum diameter, which is estimated to be at the round off level within the bounded design space. The trend, sharply decreasing at the beginning and leveling off at the end, is also observed in Section VI. The objective function convergence tolerance is defined as follows:

$$\tau_f = \frac{\tilde{f}_{\min} - f_{\min}}{1.0 + \tilde{f}_{\min}}, \tag{4.4}$$

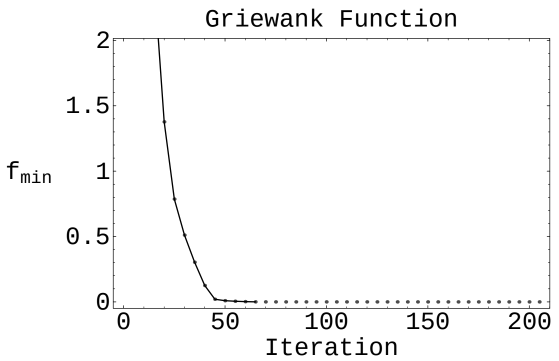


Fig. 4.2. Change in computed $f_{\min}$ as DIRECT progresses for the Griewank function with objective function tolerance $= 0$ (dotted) and 0.0001 (solid), for $n = 20$, $\epsilon = 0$.

where $\tilde{f}_{\min}$ represents the previous computed minimum. The algorithm stops when τ_f becomes less than a user specified value. It avoids wasting function evaluations for small improvements. The definition of objective function convergence tolerance (4.4) differs from the percent error in [15], which is based on the knowledge of the true global optimum of the objective function, while τ_f measures the convergence with the current best estimate of the optimum. This is a reasonable stopping criterion for large-scale engineering design problems.

The present implementation of the DIRECT algorithm addresses an efficiency issue involved in an unpredictable storage requirement in the phase of space partitioning. The main problem to be solved is how to store the large collection of boxes, typically viewed as a set of separate columns making up the points shown in Figure 4.1. The key operations are to find the element in a column with least value, to remove this least-valued element, and to add new elements to a column. Thus each column can be viewed abstractly as a priority queue.

Typical implementations for DIRECT simply allocate a large two-dimensional array to store the boxes as organized in Figure 4.1. Each column of the array corresponds to the set of boxes with a given diameter. This approach has the advantage of being simple, and matching well with the memory access patterns that work efficiently in parallel implementations. However, the actual performance for this implementation is poor for two reasons. First, there can be a large number of distinct box diameters at various times during the execution of the algorithm. This translates to a potential (but changing) need for many columns. Second, specific columns can get unusually large numbers of boxes at various times, translating into a potential (but changing) need for many rows. These behaviors are both transient and unpredictable. Thus, a dynamic data structure is needed.

In practice, only a few of the columns become large at any given time. The large memory requirements of the computations involved (of which the box processing is only a small part) argue against careless use of dynamic memory allocation, since, for example, a list implementation that spreads the contents of a column widely through virtual memory will result in poor use of the memory cache. The proposed implementation is a simple modification to the columns to provide flexibility in their length. Initially a two-dimensional array of fairly large size (depending on the dimension of the problem) is allocated in the usual way. Depending on the size and nature of the problem, this array might hold all boxes in the partitioning. Certainly, for most columns all elements in the column will remain in the array. However, the array is dynamic in that it can grow in either of two ways. First, if the array provides insufficient columns, new blocks of columns will be allocated as needed. Second, should a given column outgrow the space available in the array, a new chunk of space is allocated to that column.

Within a column, the points can be maintained in sorted order, removing the top (lowest) value as needed, and adding new values when needed. As necessary, a chunk of additional space is added or removed from the column. Within a column, shifting operations on boxes in time $O(m)$ are required to keep their function values sorted. A more efficient approach is to implement each column with a heap data structure, which is a typical priority queue that replaces shifting operations with sift down operations in time $O(\log_2 m)$.

Functionally, the Fortran 90 derived data type dynamic structures can be classified into two groups: box structures and linked list structures. The box structures (`BoxMatrix`, `BoxLink`, and `HyperBox`) are responsible for holding boxes. The linked lists (`setInd`, `setDia`, and `setFcol`) are built out of linked vectors (`real_vector` and `int_vector`), and manage the allocated memory for the box structures. Their use is illustrated by Figures 4.3 and 4.4.

Figure 4.3 shows the two dimensional chain structure of the box structures group. It contains all the necessary information about a hyperbox, namely,

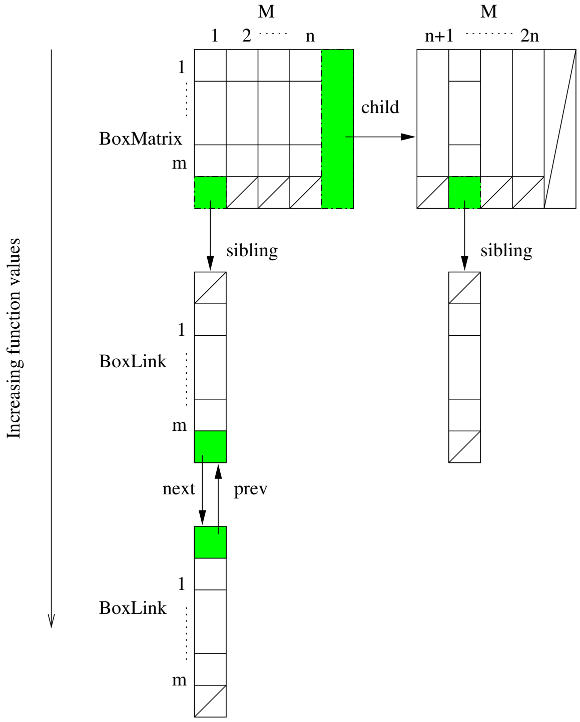


Fig. 4.3. Box structures comprised of HyperBoxes.

the objective function value at the box center, the coordinates of the center point, the side lengths in all dimensions, and the box size (diameter squared). Without further organizing the information listed above, some well-known methods for finding the convex hull can be applied. In [15], Graham's scan method is recommended for finding the convex hull of a set of m arbitrary points in time $O(m \log_2 m)$. Here, a different approach is taken to shrink the initial set with m points to a much smaller set of vertices exclusively around the low edge of the convex hull. With all the hyperboxes linked logically in the scatter plot pattern, Jarvis's march (or gift wrapping) method is applied starting from the box sequence with the biggest size, and eventually identifies all the potentially optimal boxes to be further subdivided for the next iteration.

As already described, all hyperboxes of a given diameter are sorted according to the center points' function values. The actual sorted list is made up of a column from a `BoxMatrix`, perhaps followed by some number of `BoxLink`s as shown in Figure 4.3. When a column in the initial `BoxMatrix` named `M` is full, a `BoxLink` is allocated and connected at the end of the column as a `sibling` link, which holds a one-dimensional array of `HyperBoxes` with the same number of `HyperBoxes` as a column in `M`. A `BoxLink` is extended in the same fashion when it becomes full. All boxes of the same size find their places in this box sequence, consisting of a column of `M` followed by an unlimited number of box links. Figure 4.3 illustrates the use of these box structures during execution of the DIRECT algorithm, when column one in `M` of the first `BoxMatrix` has become full, thereafter having been linked with two more `BoxLinks`, which are associated with each other by referencing their `next` and `prev` pointers. Notice that `M` has the same number of hyperboxes in a column as a `BoxLink` does, which unifies the procedures for box insertion both in `M` and `BoxLink`. Inserting a new box into a box sequence requires three steps. First, locate the segment of the sequence that the box's function value falls within, either the column in `M` or one of the `sibling` box links. Second, apply a binary search to the function values in the located segment to find the appropriate position at which to insert the new box. Third, shift the remaining elements in the column down by one position, possibly causing an additional `BoxLink` to be allocated.

While caching performance encourages maintaining adjacent elements of a column in adjacent memory locations, the same is not true of adjacent columns of the array. Further, during processing it may happen that a given column becomes empty (that is, all boxes of a given diameter may be split) and another column may need to be created (as boxes of new diameters are created by the splitting process). Because it would be costly to sort box sequences with respect to box sizes by rearranging the columns of `M`, columns are not kept sorted by box size. However, it is necessary to find the column (if any) that stores the boxes of a given size. A linked list structure (described in more detail in the next section) is used to maintain the box sizes in logical decreasing order. Physically, the columns in `M` are treated as independent cells that can be popped up for any boxes with a new size. In some sense, `M` acts as a memory pool of recyclable cells. When cells are used up, a new `BoxMatrix` is allocated and connected as the `child` link at the end of the chain of `BoxMatrices`, so that the memory pool can be filled up again using new cells from `M` in the newly allocated `BoxMatrix`. As an instance, Figure 4.3 shows a chain of two `BoxMatrices`. In this specific example, a `BoxMatrix` allocates `M` with m rows and n columns of `Hyperboxes`. The column indices of the second `BoxMatrix` begin with $n+1$ to be distinguished from indices in the first `BoxMatrix`. Cell recycling is handled by the linked list data structures, discussed below.

The linked list data structures (as shown in Figure 4.4) play an important role in maintaining the logical scatter plot pattern and recycling memory cells. They are doubly linked lists constructed with two derived data types. In some sense, `M` (the two-dimensional array defined in `BoxMatrix`) acts as a memory pool of recyclable cells for holding boxes. When cells are used up, a new `BoxMatrix` is allocated and connected as the `child` link at the end of the chain of `BoxMatrices`, so that the memory pool can be filled up again using new cells from `M` in the newly allocated `BoxMatrix`.

For faster execution, sorting is not involved in the strategy for maintaining a logical scatter plot pattern of hyperboxes. Instead, binary search is used in locating the insertion positions in sorted box size sequences. Some heap sifting operations are needed for inserting/deleting boxes in a particular column of boxes in `M` and its box links, if any, while shifting boxes among columns is avoided by keeping column indices sorted (by decreasing box sizes) in `setInd`.

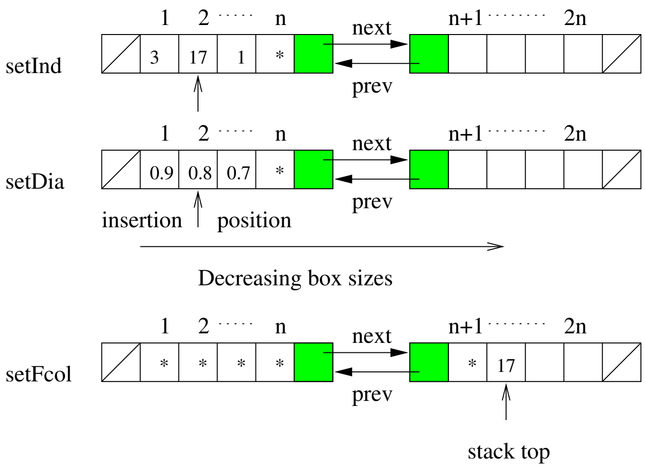


Fig. 4.4.. Linked list structures. Insertion of a new box size (0.8) has four steps: (a) request a free column index (17) from the stack top of `setFcol` by popping the stack, (b) locate the insertion position (2) in `setDia` and insert the new diameter (`setDia` is shown after the insertion), (c) add the column index to `setInd` at the insertion position (shown), and (d) add the box at the beginning of the requested column (17) in `M`.

V. Objective Formulation

As [15] proved, the DIRECT algorithm is guaranteed to converge globally if the objective function is Lipschitz continuous. However, the original definitions for both performance criteria—power coverage and BER (bit error rate) do not satisfy this condition. Similar to the power coverage criterion introduced in Section I, BER is the ratio of bits that have errors relative to the total number of bits received in a transmission. Reformulation is required to eliminate the discontinuity. The following two subsections describe the single transmitter case and the multiple transmitters case, respectively.

A. Single Transmitter Case

Consider the placement of a single transmitter in a coverage-limited indoor environment, in which outage is a result of excessively low local mean wanted signal power and fading of the wanted signal [4]. The ray tracing technique serves as a deterministic way to calculate the local mean signal power propagating from the transmitter to each receiver on the reception grid. To optimize the location of the transmitter, the decision variables are the x and y coordinates (transmitter height $z = z_0$ is fixed, which is a reasonable assumption in indoor environments). The objective function for the power coverage is the average shortfall of the received power $P_i(x, y, z_0)$ from the threshold T_p:

$$f(x, y, z_0) = \frac{1}{m}\sum_{i=1}^{m}\left(T_p - P_i(x, y, z_0)\right)_+, \qquad (5.1)$$

where T_p is the given power threshold (in dBm), $P_i(x, y, z_0)$ is the power received at the ith receiver (in dBm) with a single transmitter located at (x, y, z_0), and m is the total number of receivers. The penalty $(T_p - P_i(x, y, z_0))_+$ implies that receivers with power above the threshold do not contribute to f, while receivers with power below the threshold contribute the difference of the power and the threshold. The goal of the optimization is to minimize f. The best possible value for f is zero, which corresponds to perfect coverage.

Similarly, the objective function for BER optimization is the average shortfall of the continuous surrogate BERs $b_i(x, y, z_0)$ from the specified threshold T_b:

$$f(x, y, z_0) = \frac{1}{m}\sum_{i=1}^{m}\left(b_i(x, y, z_0) - T_b\right)_+, \qquad (5.3)$$

where $b_i(x, y, z_0)$ is the surrogate estimate of BER at the ith receiver from a single transmitter located at (x, y, z_0). The penalty here is for a high bit error rate.

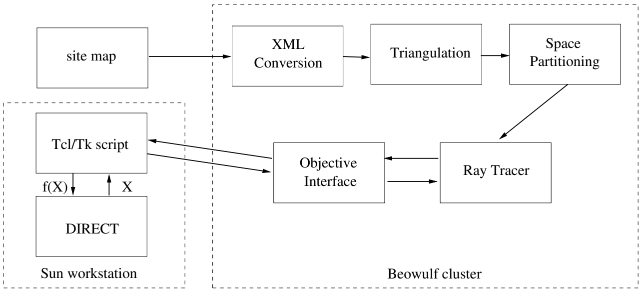


Fig. 6.1. Problem solving environment for transmitter placement optimization.

B. Multiple Transmitter Case

To extend the problem to the placement of multiple transmitters, an assumption was made to validate the objective function reformulation. Transmitters are assumed to operate at sufficiently different frequencies so that receivers can pick up the strongest signal. In other words, the indoor environment is only considered as a coverage-limited one, which is different from the interference-limited environment, where the outage is a result of co-channel signals dominating or interfering with the wanted signals [4]. Such an environment presents more complexities and challenges for implementing the WCDMA channel model. Section VII will discuss this issue in the context of future work.

As in [7], the design variables are the transmitter coordinates

$$X = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_n, y_n, z_n),$$

where all $z_j = z_0$ are assumed to be fixed, the same as in the single transmitter case. Permuted coordinates will occur during optimization, since the DIRECT algorithm treats the function as a black box and has no knowledge of any symmetry relationships. When system settings are a permutation of an earlier setting, clearly the objective should not be reevaluated by using expensive ray tracing. A solution is to simply sort coordinates on each dimension, and buffer current pairs of sorted coordinates and the corresponding function values. When the permuted coordinates are detected by the optimizer, the function value will be taken directly from the buffer instead of calling the ray tracer to reevaluate the function.

With the stated assumption and the design variables above, the single transmitter location problem is a special case of n transmitter problem with $n = 1$. The goal is to minimize the average shortfall (power coverage or bit error rate) of the n transmitters over

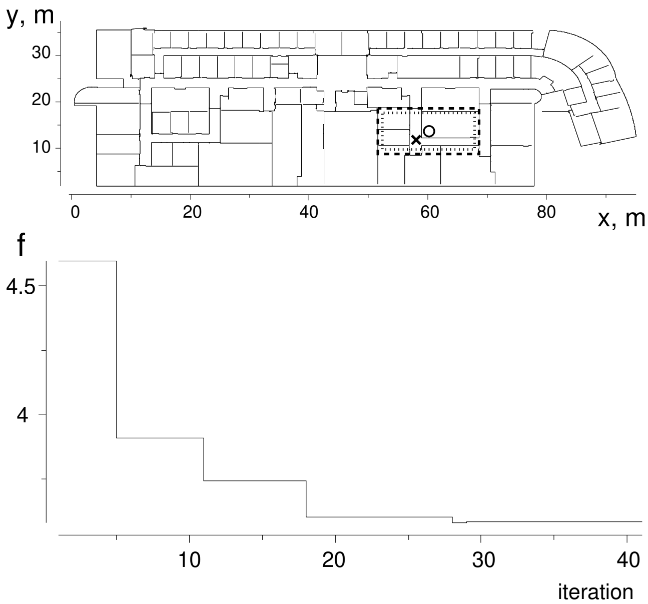


Fig. 6.2. Power coverage optimization results for one transmitter. Bounds on transmitter placement are drawn with dotted lines and the initial (final) position is marked with a circle (cross) in the center.

m receiver locations. Let transmitter (k, i), located at (x_k, y_k, z_0), $1 \le k \le n$, generate the highest peak power level $P_{ki}(x_k, y_k, z_0) \ge P_{ji}(x_j, y_j, z_0)$, $1 \le j \le n$, at the receiver location i, $1 \le i \le m$. The objective function is the average shortfall of the estimated performance metric from the given threshold T, given by

$$f(X) = \begin{cases} \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}(T - p_{ki})_+, & \text{coverage,} \\ \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}(p_{ki} - T)_+, & \text{BER,} \end{cases} \tag{5.4}$$

where p_{ki} is the performance metric of transmitter (k, i) evaluated at the ith receiver location. For power coverage optimization, p_{ki} is $P_{ki}(x_k, y_k, z_0)$ and $(T - p_{ki})_+$ is the penalty for a low power level. For BER optimization, p_{ki} is BER_{ki} and $(p_{ki} - T)_+$ is the penalty for a high bit error rate.

VI. Optimization Results

Optimization was done inside a problem solving environment (PSE) as shown in Figure 6.1. Ray tracing was performed on a 200-node Athlon 650 Beowulf cluster of Linux workstations. The DIRECT optimizer and the user interface ran on a Sun workstation outside the cluster. Tcl/Tk scripts glued

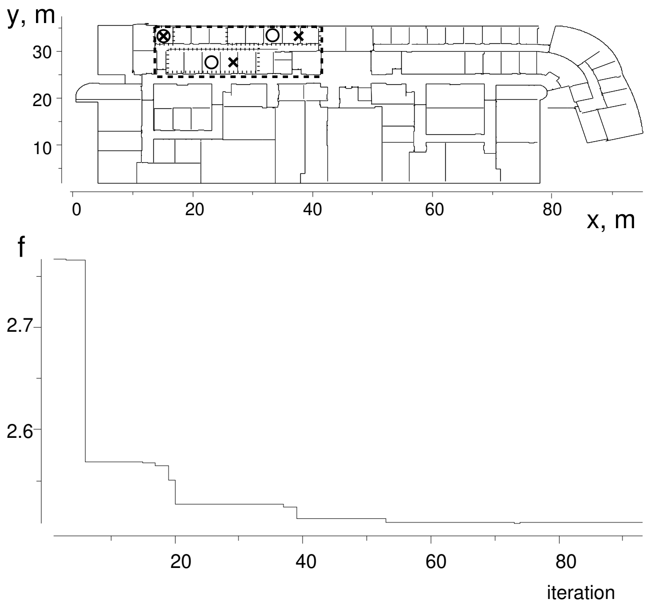
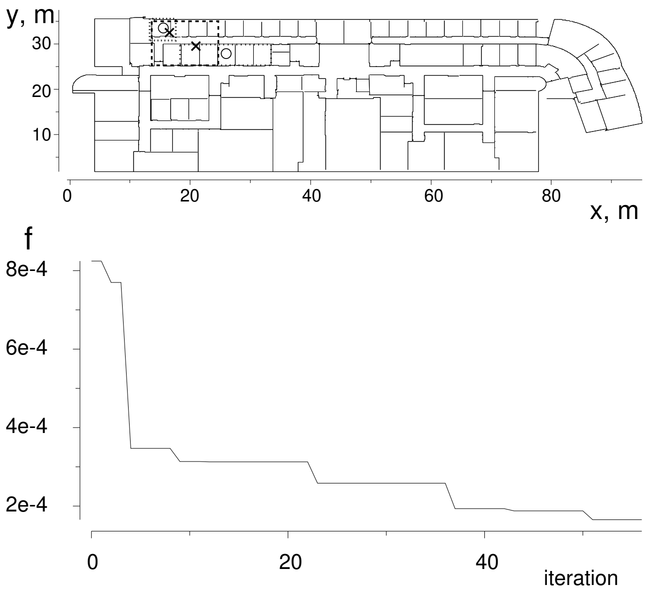
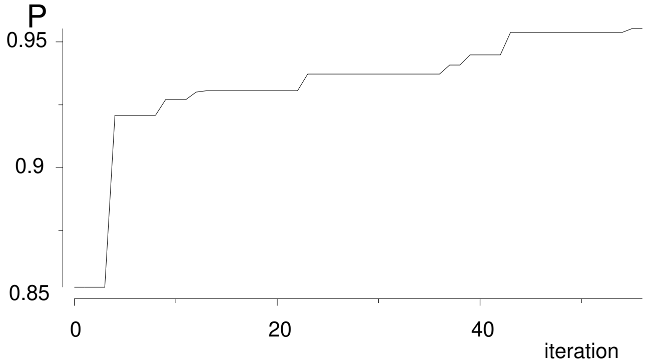


Fig. 6.3. Power coverage optimization results for three transmitters. Bounds on transmitter placement are drawn with dotted lines and their initial (final) positions are marked with circles (crosses). The dashed line delimits the region to be covered.



(a) Objective function.



(b) Percentage of receivers with satisfied BER values.

Fig. 6.4. BER optimization results for two transmitters. The region to be covered is half of that in Figure 6.3

the pieces together and provided a graphical user interface. Similar to [17], users could select regions for transmitter placement (to be optimized) and regions to be covered.

Runs for optimizing transmitter placement were executed with respect to the two performance criteria—coverage and BER. The ray tracer's tessellation frequency was 100 for coverage and 300 for BER. The former was sufficient to match the peak powers against measurements, while the latter was required to match the whole impulse responses. Two different indoor environments were chosen to demonstrate the effectiveness of the optimization. The first environment is located on the fourth floor of Durham Hall at Virginia Tech. It was the first case study for the global optimization technique. Simulations have been verified with measurement data (see Section II). The simulation verification with measurement on the second environment—the second floor of Whittemore Hall—is in progress. The same environment had been used in both raytracing simulations and measurements in [21], which considers signal diffractions in the propagation model so that it can match well the measured and predicted propagation in a variety of indoor environments. (The propagation model code of [21], while having better physics than the present ray tracing code, is orders of magnitude slower because of inefficient data structures.)

A. Durham Hall, fourth floor

The results of optimizing a single transmitter location in the case of coverage are shown in Figure 6.2. It took 41 evaluations (3 minutes, 45 seconds) to reduce the objective function by 22.2% (from 4.60 dB to 3.58 dB) demonstrating the fast convergence of the DIRECT algorithm.

Figure 6.3 illustrates power coverage optimization of the locations of three transmitters to cover eighteen rooms and a corridor bounded by the box in the upper-left corner. 93 function evaluations reduced the objective from 2.77 dB to 2.51 dB, or by 9.4%, in

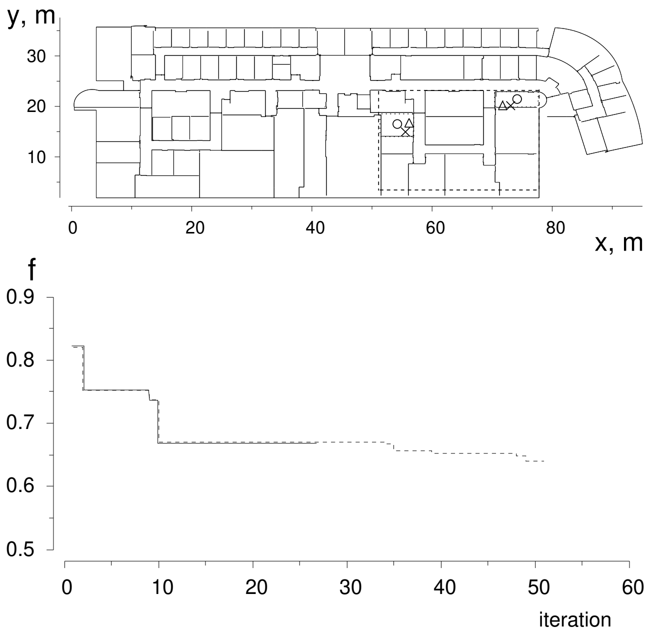
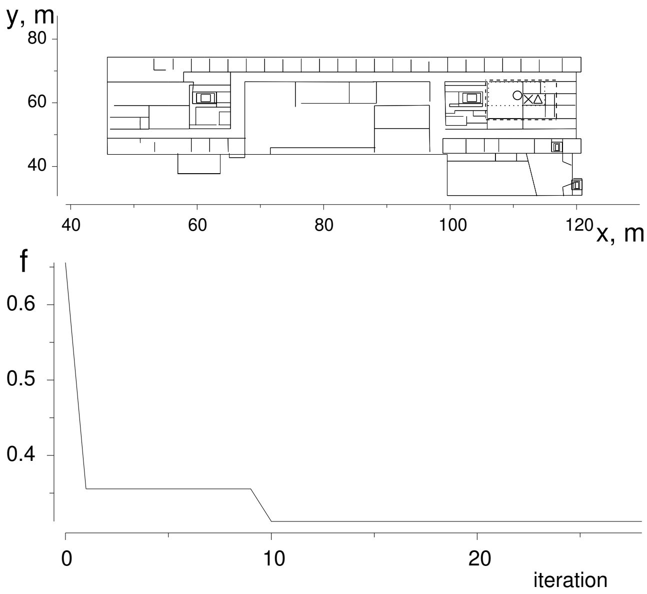


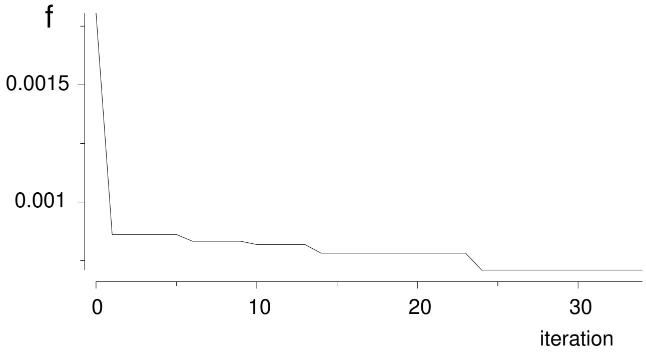
Fig. 6.5. Power coverage optimization results for two transmitters with objective function convergence tolerance = 0 (dashed) and 0.001 (solid).

38 minutes on 40 machines. Figure 6.4 depicts BER optimization of the locations of two transmitters to cover half of the former region. In Figure 6.4 (a), 56 iterations reduced the objective function from 8.24e-4 to 1.65e-4 in 3 hours and 26 minutes on 40 machines. The BER threshold was 10^{-3}, so this improvement corresponds to a 79.9% reduction in the average BER. From Figure 6.4 (b), it's observed that the percentage of the receivers with satisfied BER is growing as the objective function decreases. In both cases, the optimization loop stops with the minimum diameter required by the problem. System performance was significantly improved by DIRECT with a reasonable number of evaluations.

Figure 6.5 demonstrates the effectiveness of the new stopping criterion—objective function convergence tolerance. This figure shows the power coverage optimization results for two transmitters in the region delimited by dashed lines. The initial locations are marked as circles. Bounds on transmitter placement are drawn with dotted lines. Two simulations were done with different objective function convergence tolerances—0 and 0.001. In the former case, it took 52 iterations to reach the final locations (marked as crosses). In the latter case, the final locations (marked as triangles) were found after 27 iterations. Using a nonzero objective function convergence tolerance saved 25 expensive ray tracing iterations.



(a) Power coverage



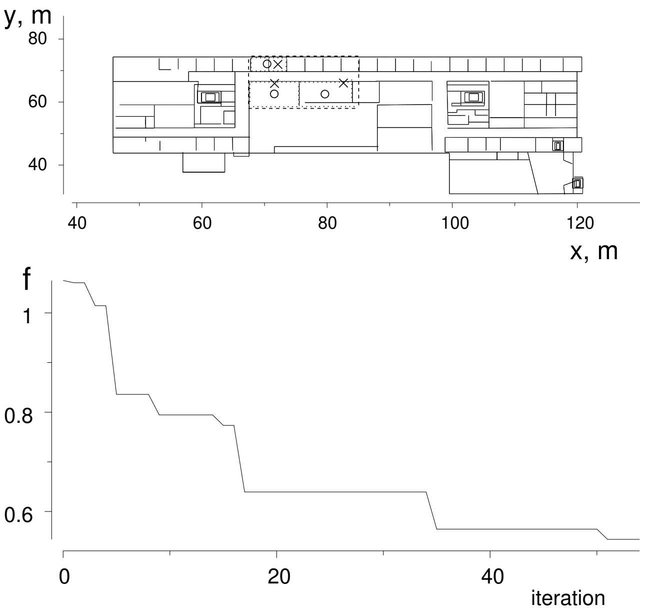
(b) BER

Fig. 6.6. Power coverage and BER optimization results for a single transmitter. Bounds on transmitter placement are drawn with dotted lines and the initial position is marked with a circle. Final position found by the power coverage (BER) optimization is marked with a triangle (cross). The dashed line delimits the region to be covered.

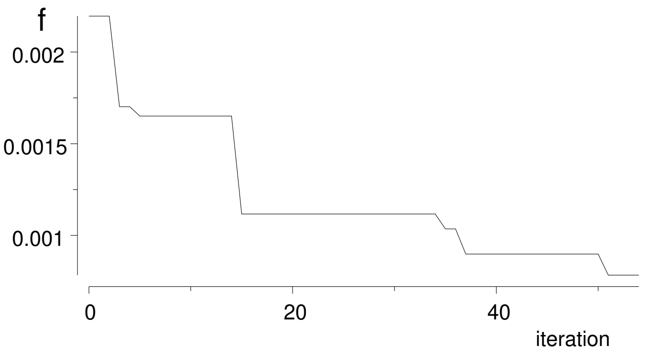
B. Whittemore Hall, second floor

Similar numerical experiments were conducted for the second environment to compare the optimization results and performance in terms of power coverage and BER. Figure 6.6 and Figure 6.7 show the results for optimizing the placement for a single transmitter and three transmitters respectively.

To optimize the single transmitter location, the minimum diameter stopping criterion was used in both power coverage and BER optimizations. It took 6 more evaluations for the BER optimization to finish than the power coverage optimization. Since



(a) Power coverage



(b) BER

Fig. 6.7. Power coverage and BER optimization results for three transmitters. Bounds on transmitter placement are drawn with dotted lines and their initial (final) positions are marked with circles (crosses). The dashed line delimits the region to be covered.

BER simulation is affected by numerous system and channel parameters such as signal-to-noise ratio, data rate, modulation type, etc. [10], it is very sensitive to parameter changes caused by changing transmitter locations. The objective function for BER exhibits more complexity (both multipath components are involved when there are two resolvable paths.) than the one for power coverage (only the dominant multipath component is considered), therefore it takes more evaluations to approach the global optimum. It's also observed in Figure 6.6 that the final locations are different. Generally, BER optimization results are preferred, since BER is considered a better performance criterion in the design of mobile communication systems as pointed out in Section III.

Table 6.1. Cost (number of ray tracing iterations) and improvement (relative function value reduction) comparison for power coverage and BER optimization experiments. (Second environment.)

	COST	IMPROVEMENT
Power coverage	28	37.7%
BER	34	60.7%

(a) single transmitter

	COST	IMPROVEMENT
Power coverage	54	48.9%
BER	54	64.2%

(b) three transmitters

In the case of optimizing three transmitter locations, the stopping criterion was the maximum number of evaluations. Both BER and coverage optimization stopped at the 54th iteration. The exact same final transmitter locations were reached at the 51st iteration. Interestingly, the final locations are exactly the same (marked as crosses at the top of Figure 6.7). This indicates a reasonable connection between these two different performance metrics—power coverage and BER.

Table 6.1 compares the cost and improvement for these four optimization experiments. The computational cost of ray tracing iterations is the metric. Improvement is defined as the ratio of reduction in the function value to the initial function value. In both cases, the BER optimization achieved a better improvement than the power coverage optimization with almost the same cost. For the single transmitter, the objective function value of the BER optimization was reduced by 60.7% while the power coverage optimization improved only by 37.7%. In the case of three transmitters, the objective function was reduced by 48.9% for the power coverage optimization and by 64.2% for the BER optimization. From this comparison, the DIRECT algorithm works very cost-effectively for BER optimization problems. On the other hand, the center-sampling strategy of DIRECT benefits the power coverage optimizations by starting at the centers of bounded areas, so that the well-distributed initial locations only need a little adjustment. This can also explain why the power coverage optimization gave less improvement.

VII. Conclusion

The main contribution of the present work is the design of an optimization loop that takes feedback from a sophisticated wireless system model. The model involves parallel 3D ray tracing and two surrogate functions that estimate the BER of a WCDMA system. DIRECT has demonstrated its effectiveness in solving the global optimization problem of transmitter placement in wireless communication systems design.

Several extensions to the present work are envisioned. First, the surrogate functions for the BER can be extended to channels with relatively strong multipath and interference. Second, wireless systems with data quality bit error rates (10^{-6}) can be considered. Third, different ways of combining the deterministic propagation model with the stochastic wireless system model can be explored.

Acknowledgments

This work was supported in part by NSF Grant DMI-9979711, and NSF Grant EIA-9974956.

References

[1] C. A. Baker, L. T. Watson, B. Grossman, R. T. Haftka, and W. H. Mason, "Parallel global aircraft configuration design space exploration," in *High Performance Computing Symposium 2000, A. Tentner (Ed.), Soc. for Computer Simulation Internat*, San Diego, CA, 2000, pp. 101–106.

[2] M. C. Bartholomew-Biggs, S. C. Parkhurst, and S. P. Wilson, "Global optimization approaches to an aircraft routing problem," *Engrg. Comput.*, To appear.

[3] K. S. Butterworth, K. W. Sowerby, and A. G. Williamson, "Base station placement for in-building mobile communication systems to yield high capacity and efficiency," *IEEE Transactions on Communications*, vol. 48, pp. 658–669, 2000.

[4] K. W. Cheung and R. D. Murch, "Optimizing Indoor Base Station Locations in Coverage- and Interference-Limited Indoor Environments," in *IEE Proceedings-Communications*, vol. 143(6), 1998, pp. 445–450.

[5] G. D. Durgin, T. S. Rappaport, and Hao Xu, "Measurements and models for radio path loss and penetration loss in and around homes and trees at 5.85 GHz," *IEEE Transactions on Communications*, vol. 46(11), pp. 1484–1496, 1998.

[6] S. J. Fortune, "A beam-tracing algorithm for prediction of indoor radio propagation," in *WACG: 1st Workshop on Applied Computational Geometry: Towards Geometric Engineering, Lecture Notes in Computer Science*, vol. 1148, pp. 157–166, 1996.

[7] S. J. Fortune, D. M. Gay, B. W. Kernighan, O. Landron, R. A. Valenzuela, and M. H. Wright, "WISE design of indoor wireless systems: practical computation and optimization," *IEEE Computational Science & Engineering*, vol. 2(1), pp. 58–68, Spring, 1995.

[8] B. Freisleben and D. Hartmann and T. Kielmann, "Parallel raytracing: a case study on partitioning and scheduling on workstation clusters," in *Proc. Thirtieth International Conference on System Sciences, Hawaii*, vol. 1, pp. 596–605, 1997.

[9] J. H. Friedman, "Multivariate adaptive regression splines," *Annals of Statistics*, vol. 19(1), pp. 1–67, 1991.

[10] Victor Fung, Theodore S. Rappaport, and Berthold Thoma, "Bit error simulation for $\frac{\pi}{4}$ DQPSK mobile radio communications using two-ray and measurement-based impulse response models," *IEEE Journal on Selected Areas in Communications*, vol. 11, No. 3, April, 1993.

[11] A. Glassner, "Space subdivision for fast ray tracing," *IEEE Computer Graphics and Applications*, vol. 4(10), pp. 15–22, October, 1984.

[12] J. He, A. Verstak, L. T. Watson, T. S. Rappaport, C. R. Anderson, N. Ramakrishnan, C. A. Shaffer, W. H. Tranter, K. Bae, and J. Jiang, "Global optimization of transmitter placement in wireless communication systems," in *Proc. High Performance Computing Symposium 2002*, A. Tentner (ed.), Soc. for Modeling and Simulation International, San Diego, CA, pp. 328–333, 2002.

[13] X. Huang, U. Behr, and W. Wiesbeck, "Automatic base station placement and dimensioning for mobile network planning," in *Proc. of Vehicular Technology Conference*, IEEE VTS Fall VTC 2000. 52nd, vol. 4, pp. 1544–1549, 2000.

[14] M. C. Jeruchim, Philip Balaban, and K. S. Shanmugan, *Simulation of Communication Systems*, Plenum Press, New York, 1992.

[15] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79(1), pp. 157–181, 1993.

[16] R. M. Lewis, V. Torczon, and M. W. Trosset, "Direct search methods: then and now," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 191–207, 2000.

[17] M. A. Panjwani, A. L. Abbott, and T. S. Rappaport, "Interactive computation of coverage regions for wireless communication in multifloored indoor environments," *IEEE Journal on Selected Areas in Communications*, vol. 14(3), pp. 420–430, 1996.

[18] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, New Jersey, 1996.

[19] T. S. Rappaport and R. R. Skidmore, Wireless Valley Communications, Inc., *Method and System for Automated Optimization of Antenna Positioning in 3-D*, US Patent 6,317,599, November, 2001.

[20] K. R. Schaubach, N. J. Davis IV, and T. Rappaport, "A ray tracing method for predicting path loss and delay spread in microcellular environments," in *Proc. IEEE Vehicular Technology Conference*, vol. 2, pp. 932–935, 1992.

[21] S. Y. Seidel and T. S. Rappaport, "Site-specific propagation prediction for wireless in-building personal communication system design," *IEEE Transactions on Vehicular Technology*, vol. 43(4), pp. 879–891, 1994.

[22] H. D. Sherali, C. M. Pendynala, and T. S. Rappaport, "Optimal location of transmitters for micro-cellular radio communication system design," *IEEE Journal on Selected Areas in Communications*, vol. 14(4), pp. 662–673, 1996.

[23] A. Verstak, J. He, L. T. Watson, N. Ramakrishnan, C. A. Shaffer, T. S. Rappaport, C. R. Anderson, K. Bae, J. Jiang, and W. H. Tranter, "S^4W: globally optimized design of wireless communication systems," in *16th Internat. Parallel & Distributed Processing Symp.*, CD-ROM, IEEE Computer Soc., Los Alamitos, CA, 2002, 8 pages.

[24] A. Verstak, M. Vass, N. Ramakrishnan, C. Shaffer, L. T. Watson, K. K. Bae, J. Jiang, W. H. Tranter, and T. S. Rappaport, "Lightweight data management for compositional modeling in problem solving environments," in *Proc. High Performance Computing Symposium 2001*, A. Tentner (ed.), Soc. for Modeling and Simulation Internat., San Diego, CA, pp. 148–153, 2001.

[25] L. T. Watson and C. A. Baker, "A fully-distributed parallel global search algorithm," *Engineering Computations*, vol. 18(1/2), pp. 155–169, 2001.