Technical report DIT-03-052, Università di Trento, October 2003

Optimal Wireless Access Point Placement for Location-Dependent Services

Roberto Battiti, Mauro Brunato, Andrea Delai

Università di Trento,
Dipartimento di Informatica e Telecomunicazioni,
via Sommarive 14, I-38050 Trento — ITALY

***Abstract*— In mobile computing scenarios, context-aware applications are more effective in relieving from the mobile user the burden of introducing information that can be automatically derived from the environment. In particular, the physical position of the mobile system (and hence of the user) is fundamental for many types of applications.**

User position estimation methods based on strength of the radio signals received from multiple wireless access points have been recently proposed and implemented by several independent research groups.

In this paper a new approach to wireless access point placement is proposed. While previous proposals focus on optimal coverage aimed at connectivity, the proposed method integrates coverage requirements with the reduction of the error of the user position estimate.

In particular, this paper proposes a mathematical model of user localization error based on the variability of signal strength measurements. This model has been designed to be independent from the actual localization technique, therefore it is only based on generic assumptions on the behavior of the localization algorithm employed.

The proposed error model is used by local search heuristic techniques, such as local search, a prohibition-based variation and simulated annealing. Near-optimal access point placements are computed for various kinds of optimization criteria: localization error minimization, signal coverage maximization, a mixture of the two.

The different criteria are not expected to be compatible: maximizing signal coverage alone can lead to degradation of the average positioning error, and vice versa. Some experiments have been dedicated to quantify this phenomenon and to introduce possible trade-offs.

I. Introduction

The area of context-aware computing is in constant development. Context-aware services and applications gather information about the context they operate in, in order to augment the system's knowledge about the user's needs, and thus its ability to meet them, while partially relieving the burden of entering data (by means of traditional input devices) from the user. This is particularly true for mobile systems, that can be used in a wide variety of conditions. An application designed to adapt its operating parameters to the environment could, for example, provide traffic and weather information when used outdoors, while automatically switching to more complex activities such as email management when it recognizes a suitable environment. An important parameter that can be used to determine system behavior in this case is location.

Most wireless networks are based on the cellular paradigm, where several radio *access points* (APs) are placed throughout the networked area and act as relays between the radio network and the fixed network.

User position estimation methods based on strength of the radio signals received from multiple wireless access points have been recently proposed and implemented by several independent research groups [1], [2], [3], [4], [5], [6].

In this paper a new approach to wireless access point placement techniques is proposed. While previous proposals focus on optimal coverage aimed at connectivity in hotspots, the proposed method integrates coverage requirements with the reduction of the error of the user position estimate.

The rest of this paper is organized as follows: section II contains a brief overview of related works, mostly in the field of wireless LAN base station placement. Section III describes the statistical framework for the analysis of position estimation algorithms and introduces the error estimates that shall be used by the optimization techniques. Section IV describes the signal coverage area estimates. Section V explains the basic principles of Local Search techniques and explains the Reactive Search algorithm used in the experiments. Section VI reports experimental results on a real-world setting.

II. Related works

Recent scientific literature contains various proposals of optimization-driven placement algorithms for WLAN access points. Section II-A contains an overview of base station placement techniques (mainly for coverage purposes), while Section II-B is dedicated to research on localization of mobile hosts.

To our knowledge, the only paper that relates signal strength detection errors and localization errors, hence suggesting localization-aware AP placement, is [7], where linear and multiple regression methods to estimate the signal strength model of an indoor wireless AP by experimental data are presented. Some results are obtained by analyzing the relationship between signal strength error and localization error, and the relationship between the standard deviation of localization error and signal strength error is analyzed for a few AP configurations. However, the analysis assumes that the localization system operates by triangulation based on a given signal propagation model.

A. Access point placement methods

This section describes the related work about optimal cellular and Wireless LAN base station placement. Many techniques have been proposed in the recent literature to analyze the indoor and outdoor signal coverage problem.

A wireless AP placement technique for optimal signal coverage inside buildings is analyzed in [8]. The signal propagation model used is the empirical Motley-Keenan indoor wave propagation model where the path loss depends on the type of walls, ceilings and floors. The cost function is the ratio of the points on the floor with higher path loss than a specific threshold; the function is minimized by using genetic algorithms.

A method to deploy a 3-dimensional large-scale indoor IEEE 802.11b WLAN or other micro-cellular networks is proposed in [9], where placement is constrained so that there are no coverage gaps, and no overlaps between APs operating on the same channel are allowed. After placing the APs for the first time by maximizing the signal coverage and minimizing the gaps between the APs, a series of signal strength measurements are collected to determine the average coverage of the APs. By using these measurements, the signal coverage of an AP is modeled as a cylinder into which the coverage is optimal. Then, AP positions are adjusted on the floor by using geometrical schemes to fill the area with the cylinders without leaving gaps. Then other measurements are taken to repeat the procedure until the solution is acceptable. Finally, frequency assignment is done by minimizing co-channel coverage overlap.

The DIRECT (DIviding RECTangles) search method is proposed in [10] aimed at maximizing signal coverage. This algorithm is a version of the Nelder-Mead simplex method, and it implements a pattern search algorithm that considers the minimization of the cost function (the ratio of covered points in a mesh). In particular, this algorithm is useful when the cost function is non-differentiable because the gradient cannot be calculated.

The method described in [11] investigates how to deploy a Wireless LAN to have a good transmission rate over a 2-dimensional outdoors area (a campus environment with streets and buildings). The purpose is to evaluate an objective function that measures the capability of the AP configuration to give a good signal coverage and uses a ray-tracing model for signal propagation. It refers to maximization of the average signal coverage and to the minimization of the position with the lowest signal strength. The optimal AP configuration is found by first applying a preprocessing technique called *pruning*, then by neighborhood search and simulated annealing. Pruning is used either alone or to calculate an initial AP configuration for the other techniques. The pruning algorithm starts with a number of APs equal to the possible AP positions. Iteratively, the AP whose removal causes the smallest increase in the cost function is removed, until the number of AP has been reduced to the desired one.

A combination of a greedy preprocessing algorithm followed by a genetic and a combinatorial scheme is described in [12] to maximize coverage in outdoors cellular systems. The Combination Algorithm for Total optimization (CAT) finds the optimal combinations of a specific group of APs that can be obtained given all the possible AP positions. Unfortunately the number of combinations is too large and therefore they are divided into small groups where the algorithm tries all possible base station combinations by finding the best one for every group. Then these partial solutions are put together and the process is repeated until the number of solutions cannot be reduced. If the final group cannot become smaller and is not too large, the algorithm tries all the possible AP combinations. If the final group is too large, a good solution is randomly chosen for each group in order to reduce the number of possible combinations.

Paper [13] considers the problem of placing multiple transmitters in order to have a good transmission rate for a given distribution of receivers in a 3-dimensional area. In this case the estimate of the object function uses the Seidel and Rappaport path loss model and is composed of a term for the maximization of the average signal coverage and a term for the minimization of the position with the lowest signal strength. The algorithms used to minimize it are the Hooke and Jeeves' method, quasi-Newton and conjugate gradient. In particular, a procedure is proposed to establish a starting point to the search techniques. The area of interest is initially divided into two 3D rectangles by partitioning its longest dimension at the center of gravity point with respect to the weights found by the squared Euclidean distance minimization problem. This procedure is repeated with the 3D rectangle that has the highest cumulative weights. The APs are initially placed in the center of gravity of the partitioned 3D rectangles. Furthermore, the grid used to represent the receiver positions is changed during the search to maintain a trade-off between the accuracy of the solution and the computational complexity of the algorithm.

In [14] several aspects of mobile cellular network selection and configuration are considered, such as area coverage, cost, traffic, interference, capacity and handover capabilities. The optimal cellular planning is found by minimizing the ad-hoc cost functions for all of these aspects by using a simulated annealing technique.

B. Localization Methods

Due to the blooming of wireless network technologies, much work has been done on localization in mobile networks. Systems based on common radio communications technologies such as Wi-Fi and cellular telephony are being actively studied and developed by various research groups. Unlike purpose-specific systems such as GPS, infrared tags or ultrasounds, the proposed Wi-Fi-based localization methods exploit the properties of the communications medium itself.

For instance, the RADAR location system by Microsoft Research [1] calculates the position of a Wi-Fi device either by similarity with previous measurements, or by modeling signal propagation. Bayesian inference models are employed both in Wi-Fi [2], [3], [6] and for GSM telephone networks [15]. Previous work of our research group is focused on the use of multi-layered perceptrons [4], on radio propagation estimates and on various statistical models [16].

III. Error in estimating user position

In this paper a planar environment is considered, with a local two-coordinate system. Valid coordinates are constrained inside a rectangular region A: $\boldsymbol{x} \equiv (x, y) \in A = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \subset \mathbb{R}^2$.

If the user's position is $\hat{\boldsymbol{x}}$, a positioning algorithm will return an estimate $\boldsymbol{x}$ with a conditional probability distribution $P(\boldsymbol{x}|\hat{\boldsymbol{x}})$, which shall be considered normalized on A:

$$\forall\hat{\boldsymbol{x}} \in A \quad \int_A P(\boldsymbol{x}|\hat{\boldsymbol{x}})\,\mathrm{d}\boldsymbol{x} = 1,$$

$$\forall\hat{\boldsymbol{x}}, \boldsymbol{x} \in A \quad 0 \leq P(\boldsymbol{x}|\hat{\boldsymbol{x}}) \leq 1.$$

This probability distribution depends on many factors, such as radio propagation, interference with other ISM equipment, signal strength discretization. After defining $d(\boldsymbol{x}, \hat{\boldsymbol{x}})$ as a convenient distance function between locations $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ (e.g., Euclidean distance in space), the average error can be calculated as

$$E(\hat{\boldsymbol{x}}) = \int_A d(\boldsymbol{x}, \hat{\boldsymbol{x}}) P(\boldsymbol{x}|\hat{\boldsymbol{x}})\,\mathrm{d}\boldsymbol{x}. \qquad (1)$$

The overall expected error is obtained by averaging (1) on the whole area A

$$E = \frac{1}{|A|} \int_A E(\hat{\boldsymbol{x}})\,\mathrm{d}\hat{\boldsymbol{x}}. \qquad (2)$$

A more useful error estimate can be obtained by considering a weighting factor $w(\boldsymbol{x})$ that can be used whenever a higher precision is desired at certain locations at the expense of other less important places. For instance, a computer room — where facilities such as printers and scanners must be located with good precision — may be more important than a lesson room, and deserve a lower localization error. Hence, this weighting function can be obtained from *a priori* considerations.

If $w(\cdot)$ is determined by such preliminary considerations, then (2) becomes, after substitution of (1) and renormalization:

$$E = \frac{\displaystyle\int\int_{A\times A} d(\boldsymbol{x}, \hat{\boldsymbol{x}}) P(\boldsymbol{x}|\hat{\boldsymbol{x}}) w(\hat{\boldsymbol{x}})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\hat{\boldsymbol{x}}}{\displaystyle\frac{1}{A}\int_A w(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}. \qquad (3)$$

While $w(\cdot)$ is given by preliminary considerations, a reasonable estimation of the conditional probability $P(\boldsymbol{x}|\hat{\boldsymbol{x}})$ is more difficult, and shall be discussed in the following.

Suppose that n access points have been positioned. Let us consider a possible signal propagation model, the *logarithmic loss* model. Let $d_{\mathrm{AP}_i}(\boldsymbol{x})$ be the distance of point $\boldsymbol{x}$ from the i-th access point ($i = 1, \dots, n$), and let $w_{\mathrm{AP}_i}(\boldsymbol{x})$ the sum of the widths of all walls crossed by the segment joining the i-th access point to point $\boldsymbol{x}$. Then the average strength of the signal received at point $\boldsymbol{x}$ from the i-th access point is

$$\mu_i(\boldsymbol{x}) = \beta_0 + \beta_1 \log d_{\mathrm{AP}_i}(\boldsymbol{x}) + \beta_2 w_{\mathrm{AP}_i}(\boldsymbol{x}). \qquad (4)$$

Coefficients β_0, β_1 and β_2 can be determined by a least-squares fit on experimental data. Only systems where all transmitters are identical will be considered, so that the same coefficients can be applied to signal strength measures from all access points.

Let $S(s|\mu)$ be the probability density of detecting signal strength s when μ is expected as average. It shall be modeled as a Gaussian, where standard deviation σ shall be determined by empirical observations:

$$S(s|\mu) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}\left(\frac{s-\mu}{\sigma}\right)^2}. \qquad (5)$$

Consider now the probability that, while being located in $\hat{\boldsymbol{x}}$, measured signals induce the system into believing that it is located in $\boldsymbol{x}$. This can happen if the received signal from access point i is $\mu_i(\boldsymbol{x})$, and the density of probability for this to happen is $S[\mu_i(\boldsymbol{x})|\mu_i(\hat{\boldsymbol{x}})]$. The probability that all measurements in $\hat{\boldsymbol{x}}$ are equal to the average value for position $\boldsymbol{x}$ is the product of all such densities when i varies from 1 to n, because measurement errors from different access points can be considered as roughly independent. By normalizing the product, we get the following estimate for the conditional probability:

$$P(\boldsymbol{x}|\hat{\boldsymbol{x}}) \approx \frac{\displaystyle\prod_{i=1}^{n} S[\mu_i(\boldsymbol{x})|\mu_i(\hat{\boldsymbol{x}})]}{\displaystyle\int_A \prod_{i=1}^{n} S[\mu_i(\boldsymbol{\xi})|\mu_i(\hat{\boldsymbol{x}})]\,\mathrm{d}\boldsymbol{\xi}}. \qquad (6)$$

This may be substituted into (3) to calculate an estimate of the average error, given the position of the access points and the parameters of the model.

Given a signal strength n-tuple $(s_1, \dots, s_n)$, equation (6) will take it into account as a possible measurement outcome if and only if its components are simultaneous expected values for the same physical position:

$$\exists\boldsymbol{x} \in A \qquad \forall i = 1, \dots, n \qquad s_i = \mu_i(\boldsymbol{x}) \qquad (7)$$

All n-tuples that don't obey constraint (7) are not considered when computing the average error. However, these n-tuples are handled in different ways by different algorithms, and taking into consideration all n-tuples has an excessive computational cost. Two hypotheses on the localization algorithm help handle this problem:

1) *Consistency*: if an n-tuple that obeys constraint (7) is fed to the algorithm, then the outcome is "near" $\boldsymbol{x}$ (in a sense that will be clear in the following lines);
2) *Continuity*: small variations in the algorithm outcome are caused by small variations in signal strength.

These hypotheses match many practical algorithms, such as the k-nearest-neighbors technique, multi-layer perceptrons and Bayesian techniques. Signal strength space (see Figure 1) can be divided into "domains", each containing points obeying (7).

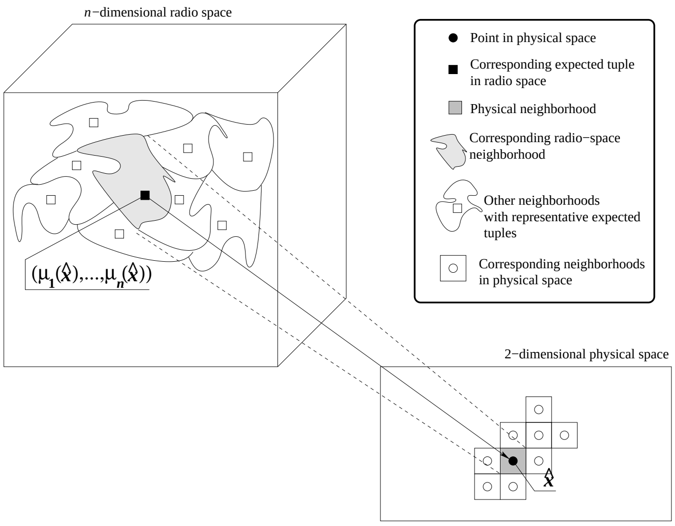

Fig. 1. Correspondence between radio space (coordinates given by received radio strength from all APs) and 2-dimensional physical space. If the correspondence is continuous, then neighborhoods in the physical space map into neighborhoods in the radio space, and the expected signal strength n-tuple at a physical point can be considered as a representative for the whole neighborhood.

The above hypotheses ensure that signal strength domains are mapped by the localization algorithm into physical space neighborhoods, so that n-tuples obeying constraint (7) are "representative" for all nearby tuples (i.e., belonging to the same domain), and the normalized product (6) can be retained as a reasonable approximation of the conditional probability distribution.

Estimation (6) can be refined by also considering incomplete measures. In the logarithmic loss model (4), signal strength distribution is continuously distributed in all the real numbers range, without lower or upper bounds. However, radio equipment is limited both in *resolution* and in *sensitivity*. This means that a wireless network cards usually returns signal strength data in a discrete and limited range.

A. Signal strength discretization

A wireless network card returns signal strength data either in an arbitrary (e.g. $0 \div 100$) scale or in dBm (decibels over one milliwatt). In both cases, values are integer. In this case, we may assume that signal strength values are rounded to the nearest integer, so that value $s \in \mathbb{N}$ will be returned if the actual signal strength lies in the interval $[s - .5, s + .5)$. As a result, a discrete probability measure should be introduced. Let μ be a (continuous) average signal strength value, as determined by (4). Then the probability of the received signal being reported as $s \in \mathbb{N}$ by the radio card is the overall probability of the actual signal falling into the range $[s - .5, s + .5)$:

$$S'(s|\mu) = \int_{s-.5}^{s+.5} S(t|\mu)\,\mathrm{d}t.$$

In other words, $S' : \mathbb{N} \times \mathbb{R} \to [0,1]$ defines the probability that the card reports a given integer given the average (expected) signal strength. A better approximation for $P(\boldsymbol{x}|\hat{\boldsymbol{x}})$ is thus obtained by replacing the density function $S(\cdot|\cdot)$ with the probability function $S'(\cdot|\cdot)$ and by rounding its arguments — $\mu_i(\boldsymbol{x})$ and $\mu_i(\boldsymbol{\xi})$ — to the nearest integer. However, this discretization correction is very marginal: when $\mu = -50$ and $\sigma = 6$ (acceptable values for a signal strength distribution in dBm), in the $\mu \pm 3\sigma$ region the two values never differ by more than 1.2% (the Gaussian probability distribution is smooth and all integration intervals are one unit wide), so that $S'(s|\mu)$ can be approximated by the central value $S(s|\mu)$, and the discretization error (whose treatment would add significant complexity to the algorithm) need not be taken into account.

B. Signal strength below detection

The major issue to be handled is missed detection due to finite hardware sensitivity: hardware will just notify absence of signal if its level falls below a certain threshold, usually set at the noise level (signal to noise ratio falling below 0dBm). In practical terms, this means that a wireless card is unable to report whether it is one or ten kilometers far from an access point: the two cases are indistinguishable because the signal strength is well below the noise level. To take this phenomenon into account, we need to consider a cumulated probability for a measurement to fall below the chosen threshold. Let $\eta \in \mathbb{N}$ be the lowest signal level that can be reported by the card. Then a version of the probability distribution $S(\cdot|\cdot)$ aware of limited hardware sensibility can be defined by cumulating the probability of the lower tail of the distribution and assigning it to η:

$$\bar{S}(s|\mu) = \begin{cases} S'(s|\mu) & \text{if } s > \eta \\ \displaystyle\int_{-\infty}^{\eta} S(t|\mu)\,\mathrm{d}t & \text{if } s = \eta \\ 0 & \text{if } s < \eta \end{cases}$$

The value of $\bar{S}(\cdot|\cdot)$ in $s \in \mathbb{N}$ is the probability for the card to return value s, if s is over the threshold, and the probability of missing the signal if s is below the threshold. These probabilities depend on the expected signal strength μ, so that a network card that is out of an access point's range has a high probability of missing its signal and only a marginal probability of detecting it at an acceptable level.

By replacing $S(\cdot|\cdot)$ by $\bar{S}(\cdot|\cdot)$ in (6), the fact that an access point is not received is treated as useful information in order to minimize the localization error.

IV. Estimating signal coverage

In a wireless network, attention must be paid to signal coverage of the networked area. Location error estimation, considered in the previous Section, has no direct relationship with the actual network availability, which should be the primary purpose of every networking infrastructure.

Many coverage measures can be devised. Among them, we consider *lowest signal strength* and *coverage area*.

A. Maximizing the lowest signal strength

The suitability of a configuration (defined as the access point coordinate values) is given by finding the lowest measured signal in the area. In the notation of the previous Section, the average signal strengths detected at position $\boldsymbol{x}$ from the access points are $\mu_1(\boldsymbol{x}), \dots, \mu_n(\boldsymbol{x})$. The network card will usually associate to the strongest signal, so that an intensity measure for that point will be

$$\bar{\mu}(\boldsymbol{x}) = \max_{i=1,\dots,n} \mu_i(\boldsymbol{x}).$$

Finally, the lowest intensity value throughout the are is a measure of suitability for the current configuration:

$$S = \min_{\boldsymbol{x}\in A} \bar{\mu}(\boldsymbol{x}).$$

Clearly, S depends on the access points positions, through the average signal strength functions $\mu_i(\cdot)$. The goal of the algorithm will be that of maximizing the value of S by optimizing the positions of the access points.

B. Maximizing the coverage area

Another possible measure of suitability is obtained by defining a signal strength threshold τ, below which the signal is considered unacceptable. According to the previous notation, we define the set of all points in the networked area that receive sufficient coverage by at least one access point:

$$A_C = \{\boldsymbol{x} \in A : \bar{\mu}(\boldsymbol{x}) \geq \tau\}.$$

Depending on the chosen threshold, connectivity may be marginal at points outside A_C, or be unavailable at all.

In the experimental section, the size of A_C

$$C = |A_C| = |\{\boldsymbol{x} \in A : \bar{\mu}(\boldsymbol{x}) \geq \tau\}| \tag{8}$$

is considered as a measure of coverage (the objective function for the optimization algorithms).

Function (8) can be extended to a weighted version by considering the same weighting factor $w(\boldsymbol{x})$ used in Section III and with a similar normalization, so that $C = A$ in case of total coverage:

$$C = \frac{\displaystyle\int_{A_C} w(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\displaystyle\frac{1}{A}\int_A w(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}. \tag{9}$$

Of course, the weighted function reduces to the unweighted one if $w(\boldsymbol{x}) = 1$ throughout the area.

V. Error minimization techniques

In this Section the proposed optimization techniques are introduced. Their purpose is to minimize the average positioning error E given by (3), the coverage area (8), or a combination of both, by an appropriate access point placement, thus changing the conditional probability distribution $P(\boldsymbol{x}|\hat{\boldsymbol{x}})$ and the average signal strength values.

All considered techniques fall within the range of *local search* heuristics.

In the following, the position of access points is referred to as *configuration*; a *move* is a change in configuration; a move is *local* if it belongs to a given set of allowed changes. For example, in our case a configuration is encoded in the program as a binary string, and local moves are those that modify a configuration by changing a single bit. A *cost function* associates a real-valued cost to each configuration. The purpose of the algorithms is that of finding the configuration associated with the (unknown) minimum cost.

A. Local search heuristics

Local Search techniques are hill-climbing search strategies where the configuration space (in our case, access point coordinates) is searched for the optimal configuration by starting from an initial position, that is either random or generated by a preprocessing step, and then by changing the configuration by means of *local moves* until some local minimum is found.

In order to apply the algorithm, a suitable discretization of the search space must be operated. The physical space shown in Figure 4 is discretized by a grid, so that only positions on mesh points are considered. Next, the meaning of *local move*, i.e., transforms in the configuration space, must be adequately defined. For example, it is possible to take into account all transforms that move one access point from its current position to one of four nearby grid points. In this case, for each configuration a maximum of $4n$ moves are allowed (they may be less, if some access points are placed at the border of the area).

Another option is to consider the binary expression of the coordinates and concatenate them together, so that the configuration is expressed as a long binary string; in this case, a local move can be changing the value of a single bit. Suppose, for sake of simplicity, that the discretization grid has 2^k intervals. Then every coordinate can be expressed as a k-bit string, and each configuration is expressed by $2nk$ bits, so that $2nk$ possible moves are allowed at each configuration, each of them leading to another valid one. Unfortunately, configurations that are physically very close may differ by an arbitrarily large number of bits in their binary representations (for instance, changing a coordinate from 2^j-1 to 2^j requires changing j bits). To overcome this drawback, *Gray encoding* of coordinates is used. Let $n \in \mathbb{N} \setminus \{0\}$; its binary representation is $l = 1 + \lfloor \log_2 n \rfloor$ bits long. Let $(b_i)_{i=0,\dots,l-1}$ the binary representation of n:

$$n = \sum_{i=0}^{l-1} b_i 2^i, \qquad \forall i = 0, \dots, l-1 \quad b_i \in \{0, 1\}.$$

The Gray encoding $(b'_i)_{i=0,\dots,l-1}$ of n is obtained from its binary representation by leaving the most significant bit unchanged and XORing consecutive pairs of bits to obtain the other digits:

$$b'_{l-1} = b_{l-1}$$

$$\forall i = 0, \dots, l-2 \quad b'_i = \begin{cases} 0 & \text{if } b_{i+1} = b_i \\ 1 & \text{otherwise.} \end{cases}$$

The Gray encoding has the property that adjacent numbers always differ by exactly one bit, therefore it more suitable to implement a locality scheme.

Figure 2 outlines a local search algorithm. Variables *bestcfg* and *bestobj* store the best configuration ever found and its cost (objective value). If *cfg* is a configuration and *mv* is a local

```
1.  iter ← 0
2.  bestobj ← +∞
3.  bestcfg ← none
4.  repeat
5.      cfg ← random configuration
6.      currentobj ← f(cfg)
7.      repeat
8.          bestmove ← none
9.          moveobj ← currentobj
10.         for mv in moves
11.             obj ← f(cfg•mv)
12.             if obj < moveobj
13.                 moveobj ← obj
14.                 bestmove ← mv
15.         if bestmove ≠ none
16.             cfg ← cfg • bestmove
17.             currentobj ← moveobj
18.         iter ← iter + 1
19.     until bestmove = none or iter ≥ maxiter
20.     if currentobj < bestobj
21.         bestobj ← currentobj
22.         bestcfg ← cfg
23. until iter ≥ maxiter
```

Fig. 2. The basic local search algorithm

```
1.  iter ← 0
2.  bestobj ← +∞
3.  bestcfg ← none
4.  for mv in moves
5.      lastperformed[mv] ← −∞
6.  cfg ← random configuration
7.  currentobj ← f(cfg)
8.  repeat
9.      bestmove ← none
10.     moveobj ← +∞
11.     for mv in moves
12.         if lastperformed[mv] < iter - T
13.             obj ← f(cfg•mv)
14.             if obj < moveobj
15.                 moveobj ← obj
16.                 bestmove ← mv
17.     cfg ← cfg • bestmove
18.     currentobj ← moveobj
19.     lastperformed[bestmove] ← iter
20.     if currentobj < bestobj
21.         bestobj ← currentobj
22.         bestcfg ← cfg
23.     iter ← iter + 1
24. until iter ≥ maxiter
```

Fig. 3. The Tabu Search algorithm

move, then *cfg•mv* is the configuration obtained by applying *mv* to *cfg*, while f(*cfg*) is the cost of *cfg*. The basic local search strategy works by generating a random configuration (line 5), then it repeatedly performs local moves (inner loop, lines 7–19) by selecting the configuration that best improves the cost function value (lines 8–14) and applying it to the configuration (lines 15–17). When no move is found to improve the cost, then a new random configuration is generated. The best configuration ever found is stored into *bestcfg* (lines 20–22). Iterations are counted, so that the algorithm terminates after *maxiter* steps.

B. Simulated Annealing

The main drawback of the basic local search scheme is its inability to escape local minima of the cost function. Once a minimum is found, the search is restarted elsewhere. However, the internal structure of the configuration space can be better exploited by exploring the nearby zone, rather than abandoning it. Therefore the problem of escaping local minima without going too far must be faced.

The Simulated Annealing technique [17] is based on the local search scheme discussed in Section V-A. However, a probabilistic scheme allows the acceptance of a move, even if the objective function gets worse. Let f be the objective function evaluated at the current configuration, and let f' be its new value if a move were performed. Then the move will be accepted with a probability that is proportional to

$$\mathrm{e}^{-\frac{f'-f}{T}}, \tag{10}$$

where T is a "temperature" parameter whose value decreases during the execution of the algorithm, according to the physical analogy to a system exploiting some randomization, but converging to a stable system as soon as the temperature converges to zero.

C. Tabu and Reactive Search

The *Reactive Search* technique [18], [19] is a history-sensitive generalization of the local search heuristic algorithm for discrete optimization. It is able to search for the global minimum of the cost function through a memory-based feedback scheme for the on-line determination of free parameters. Reactive Search is based on the Tabu Search technique [], a prohibition-based heuristic.

The important modification introduced by Tabu Search with respect to the basic local search scheme is the choice of performing a move even when no improvement is obtained over the current cost, so that when the system falls inside a local minimum it can move uphill. To avoid undoing the uphill move at the very next step, a *prohibition* scheme is introduced where a move cannot be undone for the next T steps of the search. For example, if the configuration is stored as a binary string and a local move consists of flipping a single bit (from 0 to 1 or from 1 to 0), after a bit has been changed it remains "frozen" in its new state for T moves.

Figure 3 outlines the basic steps of a Tabu Search algorithm. An integer array, *lastperformed*, associates to each move the last iteration in which it was chosen. In lines 4–5 its entries, one per move, are set to the lowest possible value (which we symbolize by $-\infty$), so that all moves are initially allowed. In order to accept the best move even when it increases the cost function value, the best cost in the move search loop is set to a very high value ($+\infty$, compare line 10 of Figure 3 with line 9 of Figure 2). If a move is forbidden (line 12) it shall not be considered. After the best move is applied, the current iteration number is stored in the corresponding array entry (line 19), so that it is prohibited for the next T iterations. The outer loop, with the random restart, has been omitted for simplicity, and because the prohibition mechanism should provide the necessary diversification when local minima are reached. More complex schemata are possible, of course, with random restart mechanisms and other modifications.

While simple Tabu Search allows exploration of the neighborhood of a local minimum, it depends on a critical parameter, T (the prohibition period), whose optimal value is determined by the structure of the configuration space, and can hardly be set a priori. In fact, an excessively low value of T prevents escape from deep local minima, while a high value reduces the algorithm's freedom of choice, so that a new minimum would be skipped because no moves are allowed toward it.

The Reactive Search scheme proposes a simple mechanism in order to adapt the prohibition period T. Visited configurations are stored, so that repetitions can be detected. Once configurations are repeated too often, T is increased. If the number of repeated configurations is below a given threshold, T is decreased. It turns out that the dependency of system behavior on the actual increase and decrease rates is low, so the system does not rely on critical parameters. Rather than storing configurations, which would amount to huge memory requirements and search times, small "fingerprints", usually 64-bit long, are computed and stored into a hash table for fast retrieval.

The prohibition period is obviously limited by the number of bits used to encode the configuration (once all bits are frozen, no more moves are possible), so in some cases a suitable differentiation cannot be enforced. If such situation is detected, for example by noticing that configuration repetitions are not avoided by increasing T, then an *escape* mechanism may be invoked: prohibitions are reset, T is initialized and a new random configuration is generated.

Preliminary experiments have shown that the Reactive Search algorithm is more effective if search of the best move is truncated as soon as an improving move is found, provided that moves are scanned in random order. This permits a higher iteration rate at the expense of the improvement obtained from each iteration.

VI. Experiments

In order to check the optimization method on a real-world case, the map shown in Figure 4 was used, representing a floor of a building situated in the Faculty of Science of the University of Trento. All measures are expressed in meters, with an approximate area of 750m^2. The floor is composed of two classrooms, a computer room, an open space with graduate student desks and common areas. The map also reports the weight values attributed to each room, to be used in the average calculation (3).

A series of simulated tests has been performed to compare reactive tabu search (RTS), local search (LS) and simulated annealing (SA). For each technique, thirty one-hour runs with different initial AP configurations have been performed for each of three problems: positioning error minimization, signal coverage maximization and a combination of both.

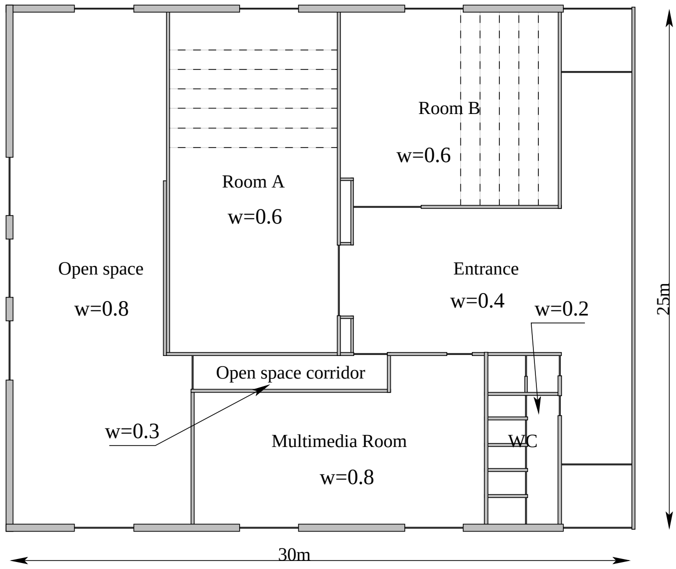

Fig. 4. Test map with error weights for each room

TABLE I
PARAMETERS FOR SIMULATED ANNEALING

	α
localization error	0.9995
coverage	0.9998
combination	0.99

In order to calculate the positioning error (3), user positions and the estimated positions are discretized on a 10×10-point mesh. Calculation of the covered area (8) are performed on a 50×50-point mesh. The threshold that gives the optimal transmission rate is set to 40dBm. Furthermore, the observations with a signal strength value below 5dBm are considered as missing in order to simulate a real situation in a building where an AP cannot cover the whole area of interest. The two cost functions use different discretization steps because the positioning error function has a higher computational complexity (it computes two nested two-dimensional integrals), and a finer granularity would result in unacceptable times. The experimental section VI reports execution times.

Simulated Annealing uses a probability distribution proportional to (10) where the "temperature" T decreases by a constant factor α after each iteration, starting from an initial value T_0:

$$T_k = \begin{cases} T_0 & \text{if } k = 0 \\ \alpha T_{k-1} & \text{otherwise.} \end{cases}$$

Preliminary tests with different parameters suggest that the best convergence rate and probability of finding a good local optimum is found for $T_0 = 100$ as the initial temperature value, while the temperature coefficient α differs among the three problems, as seen on Table I.

TABLE II

LOCALIZATION ERROR —RESULTS

	best fi tness (m)	occurrencies of best	average (m)	90th (m)	95th (m)
LS	5.857280	1	5.97214	6.03029	6.03085
RTS	5.857280	17	5.89551	6.01243	6.01375
SA	5.857280	15	5.88532	5.93771	5.95139

In order to place the APs by minimizing the average error and by maximizing the signal coverage, a cost function for the combination problem has been used

$$C_{\text{err,cov}} = E + \gamma \frac{1}{C} \quad (11)$$

where E and C are the cost function for error minimization (3) and signal coverage (8). The constant γ is chosen in order to obtain a linear combination of the two errors so that the two terms have approximately the same weight (a value of $\gamma = 2500$ has been experimentally determined).

VII. RESULTS

The following subsections report and compare the results of the three search techniques when applied to the three problems.

Note that the expression “global optimum” refers to the best value of the fitness function found by any of the three heuristics. There are good reasons to suppose that this is the actual optimum. In fact, the value is found many times in independent runs by all three algorithms, and it always corresponds, within the same objective function, to the same access point configuration.

Also note that the actual user localization error values are not significant in their values, because they refer to general assumptions about the localization algorithms, and do not take into account any specific technique. These values are only meaningful when used in comparison among themselves.

A. Localization error minimization

When applied to the average localization error problem, the RTS and SA techniques clearly outperform simple LS. In fact, they find the global minimum (5.85728 m) respectively 17 and 15 times of the 30 one-hour runs (see table II).

Figure 5 shows a typical trace of the execution of each algorithm. CPU time is on the horizontal axis, while the vertical axis reports the best value found up to that moment. The Figure depicts a behavior that is quite common in local search techniques: initially, the simple Local Search algorithm tends to have the same (or even better) performance as the two more complex techniques. In fact, a typical downhill run of LS lasts only a few steps, after which a local minimum or a plateau is found, and a new random start is generated, so that many local minima are explored. On the other hand, RTS and SA tend to remain for a longer time around a local minimum, trying to escape it, and generating a new random solution only in particular conditions. This strategy tends to pay in the long run: in the example, the first three minutes

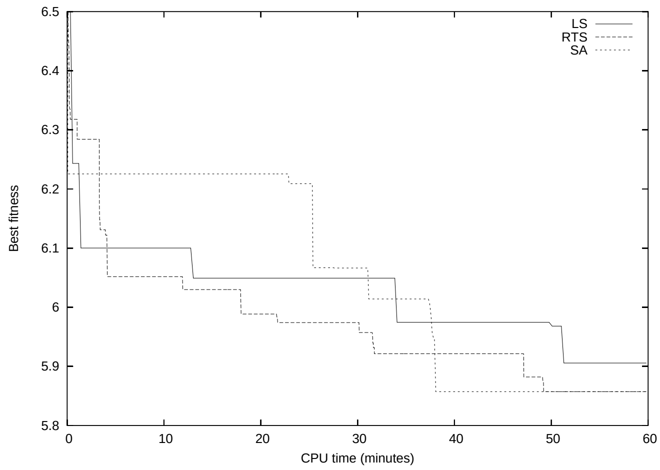

Fig. 5. Localization error —Typical traces for the three algorithms: best solution found against CPU time.

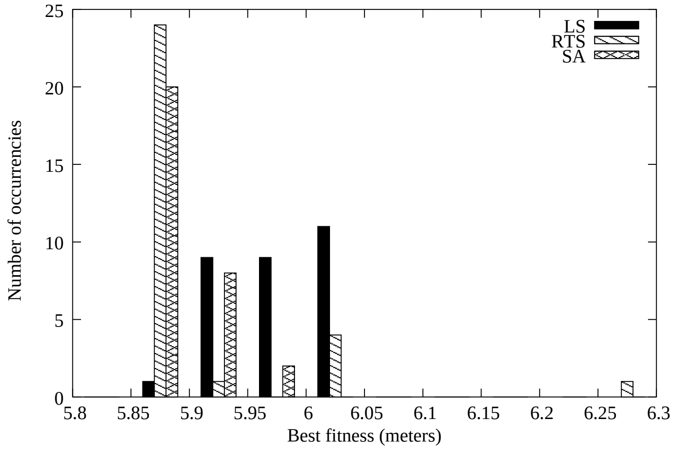

Fig. 6. Localization error —Distribution of the best fi tness reported by the algorithms. Each bar counts the number of one-hour runs reporting a minimum localization error value in the corresponding range.

show a neat prevalence of simple Local Search, but due to the random restarts, improvements tend to concentrate at the beginning and to slow down after the most probable minima have been found.

Figure 6 shows the distribution of the best fitness values found by each run of the three algorithms.

Figure 7 shows the CPU time required to get the optimum value (only runs yielding the optimum value are considered). The RTS search algorithm seems to have a slight advantage over the SA, in that it often finds the best fitness value within a shorter time.

Figure 8 shows the expected average user localization error over a 50×50 point-mesh, given the optimum AP configuration as found by the local search algorithms. Note that localization error is rather low in the neighborhood of the access point positions, while it may go up to 10 meters in some locations.

In Figure 9 the AP positions corresponding to the minimum average error is shown. Note that the AP placement is quite unpredictable: all APs are placed on the south side of the building.

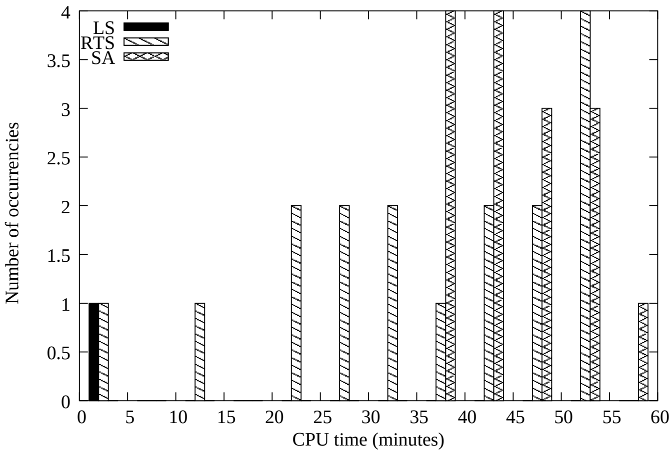

Fig. 7. Localization error — Distribution of elapsed times to best fitness. Only runs yielding the global minimum within one hour are counted.

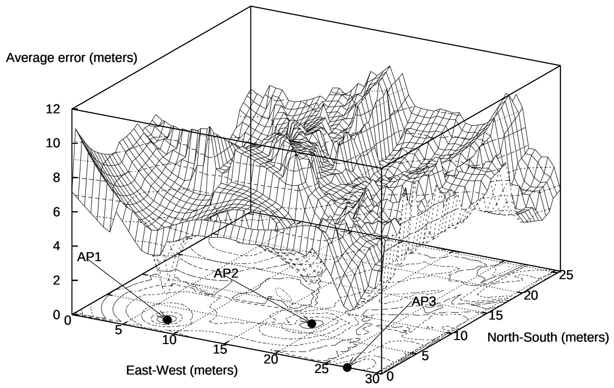

Fig. 8. Localization error — Average for optimal AP configuration

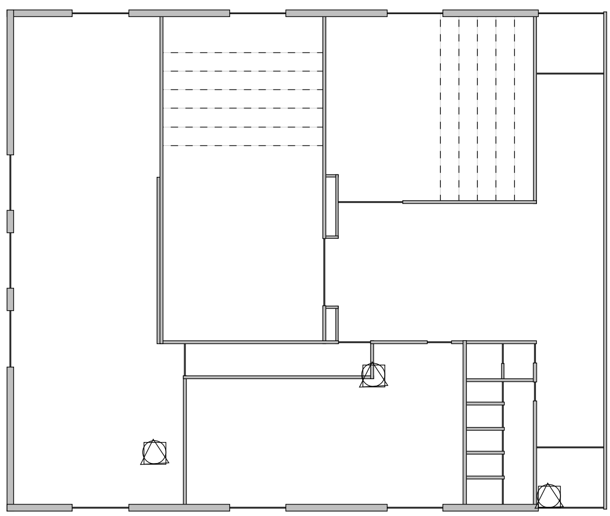

Fig. 9. Localization error — Optimal AP configuration obtained by RTS (triangles), LS (squares) and SA (circles).

TABLE III

SIGNAL COVERAGE — RESULTS

	best fitness (m^2)	occurrencies of best	average	90th	95th
LS	419.454	15	419.002	418.164	418.002
RTS	419.454	29	419.443	419.454	419.454
SA	419.454	30	419.454	419.454	419.454

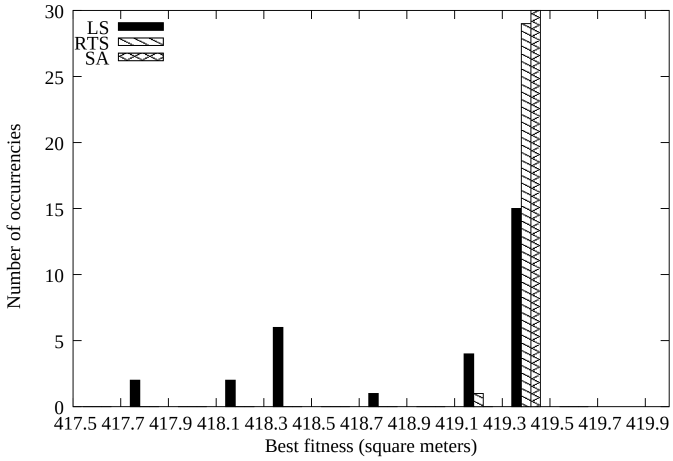

Fig. 10. Signal coverage — Distribution of the best fitness reported by the algorithms.

B. Signal coverage

As Table III reports, the basic Local Search algorithm finds the global optimum within the hour only 50% of the times. RTS and SA always find the global optimum (with the exception of one run of the RTS heuristic, yielding a slightly lower value, as showm in Figure 10).

On the average, RTS can find the best fitness more quickly than SA: the global maximum is found within the first 5 minutes in 11 runs out of 30. As a consequence, if search is truncated after a short time (less than an hour), RTS seems to have the highest probability of reporting the optimal value.

Figure 12 shows the signal strength distribution given by the optimal AP configuration, where the horizontal plane represents the signal threshold for an optimal wireless connection.

The optimal AP configuration by maximizing the signal coverage for the coverage is rather predictable because they are placed in the interior of the building, far from the outer walls (Figure 13).

C. Combined problem

Table IV shows the results of the search algorithms when applied to the combined fitness function (11). As in the previous cases, a common best fitness is found by all techniques. In particular, RTS finds the best value 90% of the thirty one-hour runs (SA just 70%).

Figure 14 shows that all thirty runs of the RTS algorithm report values that fall in the immediate neighborhood of the minimum.

However, unlike the previous cases, Figure 15 shows that there is no prevalence of RTS over SA when considering the

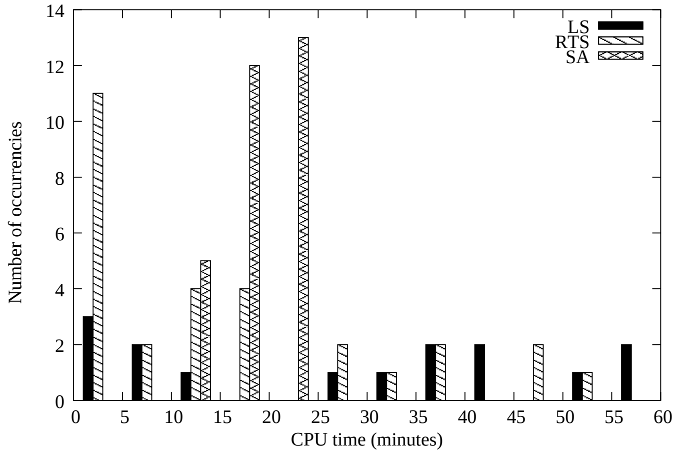

Fig. 11. Signal coverage —Distribution of elapsed times to best fitness.

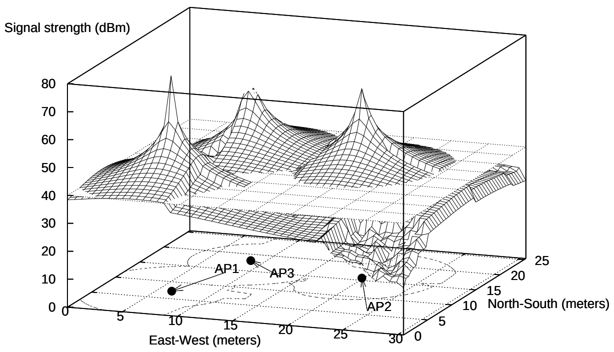

Fig. 12. Signal coverage —Average for optimal AP configuration

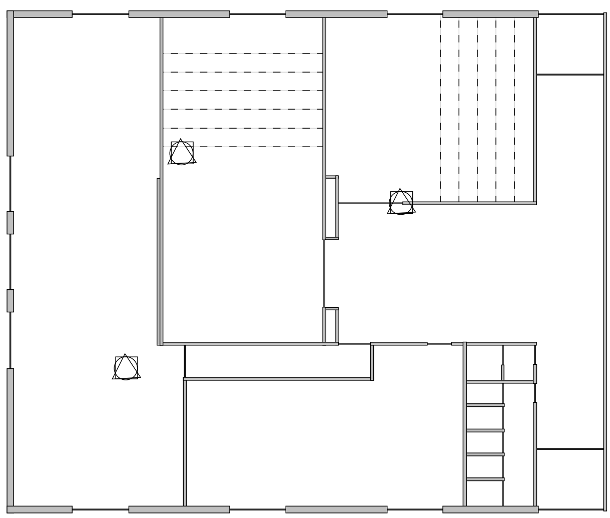

Fig. 13. Signal coverage —Optimal AP configuration obtained by RTS (triangles), LS (squares) and SA (circles).

TABLE IV

COMBINED PROBLEM —RESULTS

	best fitness	occurrencies of best	average	90th	95th
LS	13.0872	3	13.1111	13.1375	13.1541
RTS	13.0872	27	13.0875	13.0872	13.0875
SA	13.0872	21	13.1224	13.2128	13.2607

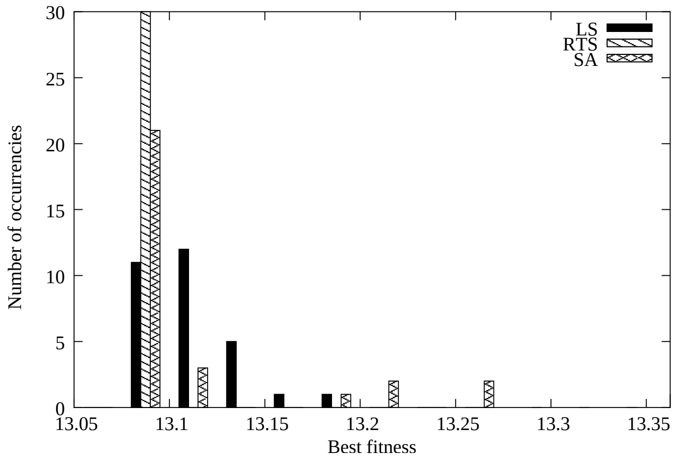

Fig. 14. Combined problem —Distribution of the best fitness reported by the algorithms.

time to discover the best fitness.

Figure 16 shows the optimal AP configuration for the combined problem. It is apparently a midway configuration between that obtained by the localization error minimization and the signal coverage maximization heuristics.

D. *Cross-Compatibility*

The best configurations calculated in the previous subsectons are very different. In particular, the configuration aimed at minimizing the localization error, Figure 9, looks too asymmetric to be effective in signal coverage. Vice versa, the outcome of signal coverage maximization, shown in Figure 13, may not work as well in localizing users, because APs are too near to each other, and so many places with similar signal strength values may be encountered.

Moreover, the outcomes of the local search techniques are rather different from what may have been planned by hand. In particular, Figure 17 shows two AP placements that have been determined by "rule of thumb": the first (white circles) is meant to maximize signal coverage; the second (black circles) to minimize localization error.

Table V compares the outcomes of all configurations for all kinds of objective function. Obviously, each "best" configuration (in the first three lines) is the best for the problem it is meant for. Best values for each criterion are shown in bold, percentage values show the relative difference with the best value in the column. The table shows that localization error tends to increase by 40% when the configuration for coverage is used, while applying the configuration meant for localization error yields a proportional reduction in covered area.

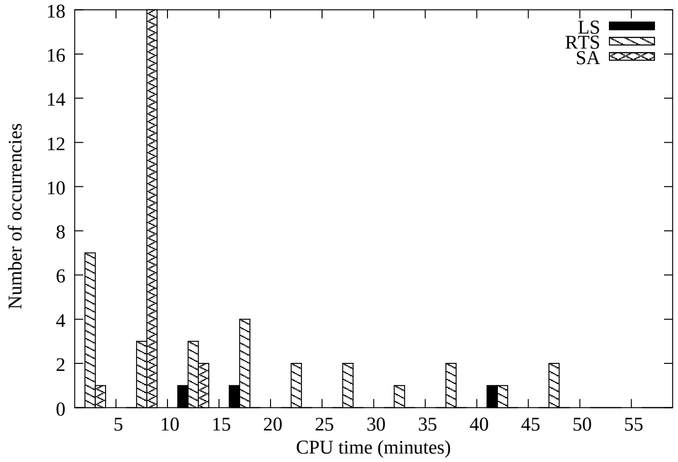

Fig. 15. Combined problem —Distribution of elapsed times to best fitness.

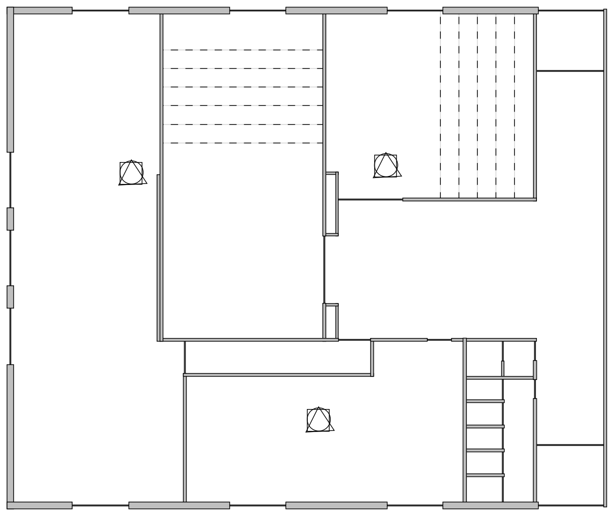

Fig. 16. Combined problem —Optimal AP configuration obtained by RTS (triangles), LS (squares) and SA (circles).

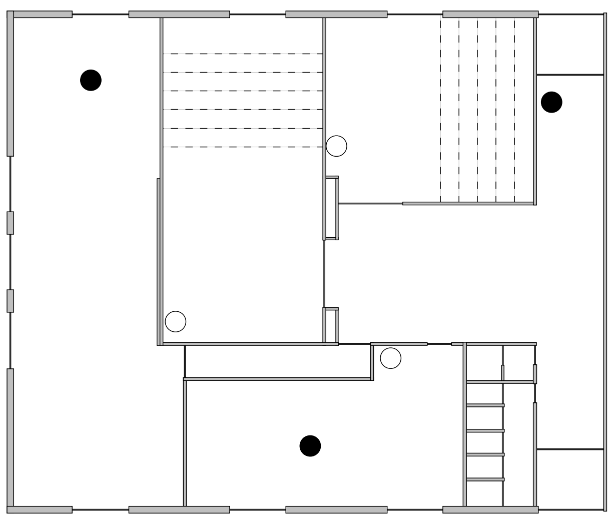

Fig. 17. Two AP configurations placed by hand: the white circles show a configuration aimed at maximizing coverage, while the black circles are meant to minimize the localization error.

TABLE V

CROSS-COMPATIBILITY OF CONFIGURATIONS FOUND BY HEURISTIC SEARCH AND HAND PLACEMENT (FIGURE 17)

	Error (m)	Coverage (m^2)	Combined
Localization best	**5.857280**	249.737 (40%)	15.8638 (21%)
Coverage best	8.22589 (40%)	**419.454**	14.1836 (8%)
Combined best	6.58422 (12%)	384.285 (8%)	**13.0872**
By hand (white)	8.75573 (49%)	310.719 (26%)	16.7984 (28%)
By hand (black)	6.81586 (16%)	284.261 (32%)	15.6071 (19%)

The combined optimization yields a configuration that actually stands in the middle of the two: by achieving an average localization error that is only 12% worse than the optimum, and a coverage area that is just 8% smaller, the combined optimization solution seems to be a good trade-off between the two main criteria.

The two configurations designed by hand also stand in the middle of the two criteria, the first configuration having a slight preference towards improved localization, the second for a better coverage.

VIII. CONCLUSIONS

A mathematical model of localization error for algorithms based on signal strength measurements has been proposed by making generic assumptions on the nature of positioning estimate. This model has been implemented in three local search techniques in order to optimize wireless access point placement with respect to user localization error. The techniques aim at reducing the error before the application of postprocessing algorithms such as moving averages, Kalman filters or path-based corrections.

ACKNOWLEDGMENTS

This research is funded by the Province of Trento (Italy) in the framework of the WILMA (Wireless Internet and Location Management) project[1].

REFERENCES

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of IEEE INFOCOM 2000*, Mar. 2000, pp. 775–784.

[2] A. M. Ladd, K. E. Bekris, G. Marceau, A. Rudys, L. E. Kavraki, and D. S. Wallach, "Robotics-based location sensing using wireless ethernet," Department of Computer Science, Rice University, Tech. Rep. TR02-393, 2002.

[3] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A probabilistic approach to WLAN user location estimation," *International Journal of Wireless Information Networks*, vol. 9, no. 3, July 2002.

[4] R. Battiti, A. Villani, and T. Le Nhat, "Neural network models for intelligent networks: deriving the location from signal patterns," in *Proceedings of AINS2002*, UCLA, May 2002.

[5] M. Brunato and Csaba Kiss Kalló, "Transparent location fingerprinting for wireless services," in *Proceedings of Med-Hoc-Net 2002*, Cagliari, Italy, Aug. 2002.

[6] M. A. Youssef, A. Agrawala, A. U. Shankar, and S. H. Noh, "A probabilistic clustering-based indoor location determination system," University of Maryland Computer Science Department, Tech. Rep. CS-TR-4350, Mar. 2002. [Online]. Available: http://www.cs.umd.edu/Library/TRs/CS-TR-4350/CS-TR-4350.ps.Z

[1] `http://www.wilmaproject.org/`

[7] Y. Chen and H. Kobayashi, "Signal strength based indoor geolocation," in *Proceedings of IEEE ICC 2002*, vol. 1, 2002, pp. 436–439.

[8] L. Nagy and L. Farkas, "Indoor base station location optimization using genetic algorithms," in *IEEE PIMRC 2000*, vol. 2, 2000, pp. 843–846.

[9] A. Hills, "Large-scale wireless LAN design," *IEEE Communications Magazine*, vol. 39, pp. 98–107, 2001.

[10] M. H. Wright, "Optimization methods for base station placement in wireless applications," in *Proceedings of IEEE Vehicular Technology Conference*, vol. 89, 1998, pp. 11 513–11 517.

[11] M. Kamenetsky and M. Unbehaun, "Coverage planning for outdoor wireless LAN systems," in *International Zurich Seminar on Broadband Communications, Access, Transmission, Networking*, vol. 49, 2002, pp. 1–6.

[12] A. Molina, G. E. Athanasiadou, and A. R. Nix, "The automatic location of base-stations for optimised cellular coverage: a new combinatorial approach," in *Proceedings of IEEE 49th Vehicular Technology Conference*, vol. 1, 1999, pp. 606–610.

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

[14] S. Hurley, "Automatic base station selection and configuration in mobile networks," in *Proceedings of IEEE 52nd Vehicular Technology Conference*, vol. 6, 2000, pp. 2585–2592.

[15] T. Roos, P. Myllymäki, and H. Tirri, "A statistical modeling approach to location estimation," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, Jan. 2002.

[16] R. Battiti, M. Brunato, and A. Villani, "Statistical learning theory for location fingerprinting in wireless LANs," Dipartimento di Informatica e Telecomunicazioni, Università di Trento, Tech. Rep. DIT-02-0086, Oct. 2002. [Online]. Available: http://eprints.biblio.unitn.it/archive/00000238/

[17] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[18] R. Battiti, "Reactive search: Toward self-tuning heuristics," in *Modern Heuristic Search Methods*, V. J. Rayward-Smith, Ed. John Wiley and Sons Ltd, 1996, ch. 4, pp. 61–83.

[19] R. Battiti and G. Tecchiolli, "The reactive tabu search," *ORSA Journal on Computing*, vol. 6, no. 2, pp. 126–140, 1994.