Diversified Local Search for the Optimal Layout of Beacons in an Indoor Positioning System

MANUEL LAGUNA
Leeds School of Business, University of Colorado at Boulder, USA, laguna@colorado.edu

JAVIER O. ROA, ANTONIO R. JIMÉNEZ AND FERNANDO SECO
Instituto de Automática Industrial – CSIC, Ctra. Campo Real km. 0.2, 28500 La Poveda, Arganda del Rey (Madrid), Spain, {javieroa, arjimenez, fseco}@iai.csic.es

ABSTRACT

The navigation of autonomous guided vehicles (AGV's) in industrial environments is often controlled by positioning systems based on landmarks or artificial beacons. In these systems, the position of an AGV navigating in an interior space is determined by the calculation of its relative distance to beacons, whose location is known in advance. A fundamental design problem associated with landmark navigation systems consists of determining the optimal location of the minimum number of beacons necessary to achieve a desired level of accuracy and reliability. A local search procedure coupled with a diversification strategy is developed for this problem. We provide comparisons with an earlier solution method based on genetic algorithms and show that our proposed procedure finds better designs in a fraction of the computational time employed by the genetic algorithm.

Latest Revision: July 24, 2007

1 Introduction

An autonomous guided vehicle (AGV) must continuously determine its position within a manufacturing facility in order to find a path to a desired destination. While global positioning systems (GPS) have provided a reliable and cost-effective solution for location outdoors, no equivalent solution exists for interior environments. One of the most common Local Positioning Systems (LPS) for indoors is the so-called *beacon or landmark navigation*. The approach is similar to GPS in that the position of the AGV is determined by a process known as 3-D multilateration. With this technique, the position of the vehicles is computed as the intersection point of several spheres that have their centers at the beacons' positions and radii equal to the distance between beacons and the mobile element. The reliability and robustness of these positioning systems depend on the correct location of the beacons. An ineffective beacon configuration results in some areas that are either not covered at all or where the position cannot be calculated (e.g., when the beacons within the sensorial range are in line with the vehicle).

Complete coverage and absence of singularities in an area does not guarantee accuracy in the position estimation. Geometric dilution of precision or *GDOP* is a measure of position accuracy (R. Yarlagadda, 2000). The *DOP* acts as an amplification factor that translates errors in the range estimation to errors in the position estimation. Hence, even starting from fairly accurate range measurements, the position estimation may contain large errors if the *DOP* at the measurement point is high. This may be caused, for example, by the configuration geometry of the beacons in sight.

The problem of finding a beacon configuration that meets a desired level of accuracy and reliability is relatively simple in small regular areas (e.g., a square). In practice, however, the areas to be covered may be irregular (e.g., those with several rooms with different shapes, passage ways and corridors), forcing the use of a large number of beacons that must be located in such a way that the resulting design is not only usable but also cost-effective and flexible. Fundamentally, this engineering design problem is multiobjective in nature. Specifically, we seek a beacon configuration that is optimal or near to optimal with respect to three objectives:

> Minimize the number of beacons. While the cost of the beacons is relatively low compared to other costs in a manufacturing facility, minimizing the number of beacons results in time savings and translates into higher system flexibility, scalability and capacity to cover large areas (Sinriech and Shoval, 2000). Flexibility is important because short product life cycles force manufacturing facilities (particularly those in the high-tech and electronic industries) to change process configurations that result in new vehicle paths.

> Maximize the coverage of the workspace. The covered workspace is defined as the area where the required number of beacons is available and their configuration results in an acceptably low *DOP*. The uncovered area consists of singular points with high (even infinite when the beacons are positioned in a straight line) *DOP* values, and points where the beacons in Line Of Sight (LOS) are not sufficient for trilateration. In the current context, we require 3 beacons because we use spherical positioning. However, it is possible for other applications to require more beacons. In practice, less than 100% coverage might be acceptable, if the number of beacons needed to reach difficult points (e.g., corners) is unreasonably large.
>
> Maximize the percentage of the area that is covered with admissible *DOP* values. Lower *DOP* values mean that the estimations of the positioning system are more accurate. Admissible *DOP* values depend on the manufacturing context and therefore the setting of a threshold is an a priori design decision. Maximizing the covered area that falls below the chosen threshold becomes an objective of the optimization process.

As described in Roa, et al. (2005), a number of studies have been reported in the literature concerning the problem of optimally locating beacons, for instance the work by Oh and No (1994) in a nuclear reactor facility and the articles by Kang, et al (1996) and Kang, Park and Agrawal (1998) that address optimal placement of sensors for vibration control of laminated beams and plates, respectively. The work that is closest to our current development is the article by Sinriech and Shoval (2000). They developed a nonlinear mixed-integer programming model to minimize the number of beacons subject to covering a set of critical points in a work area. The model calculates Euclidean distances between the selected beacons and the critical points and forces at least three beacons to be within specified range to each of the critical points in the entire area. The three-beacon requirement is the same that we impose to be able to calculate the position of a vehicle using triangulation. The given range (specified by a minimum and maximum distance to a critical point) is related to the physical limitation of the laser navigation system used in the specific environment being studied. Sinriech and Shoval (2000) do not solve the proposed model due to its complexity. Instead, they relaxed the model by removing all nonlinearities and solved the resulting set covering problem in order to find a lower bound in the number of beacons. The set covering formulation assumes that beacons can only be placed in the intersections of a grid that is overlaid on the work area. There is a constraint for each critical point that forces the point to be covered by at least three beacons. The constraint coefficients are zero or one to indicate whether or not a beacon placed in a particular grid intersection covers a critical point. A greedy heuristic is then used to find a feasible solution to the problem. The heuristic considers, in descending order, the locations in the grid that will cover the most number of critical points. As beacons are added to these locations, the convexity constraints (which are relaxed in the set covering problem) are checked. The convexity constraints ensure that each critical point is located within the convex hull of the set of beacons serving the critical point.

Although the work of Sinriech and Shoval (2000) addresses certain aspects of the problem that are of interest to us (e.g., the minimization of the number of beacons), their procedure focuses on finding a beacon configuration that covers critical points (such as pick up/ delivery points for material-handling vehicles) as opposed to one that covers an entire area. To the best of our knowledge, the only exiting procedure for the optimization problem that we propose to tackle is the genetic algorithm (GA) described by Roa, et al (2005). This basic GA follows the traditional literature (see e.g., Holland 1975) and starts with a population of solutions consisting of beacons deployed at random positions. Solutions were represented as sets of two-dimensional coordinates indicating the location of each beacon. Experiments showed that a large computational effort was required to obtain designs of a reasonable quality. It was also observed that occasionally the GA was unable to escape inferior local minima even when allowed to run for an extended amount of time. Our current goal is to overcome these deficiencies with a different heuristic approach and compare our results to the GA implementation.

2 Problem Description

In interior spaces, such as industrial bays or office spaces, where a navigation system for robots or autonomous vehicles is being considered, it is necessary to first determine the characteristics of the work area that are relevant to the optimization process. These characteristic include the navigation area, beacon area and possible obstacles (as described below). Since beacon capabilities are considered known, the problem consists of finding the placement of a minimum number of beacons that maximizes coverage of the navigation area while guaranteeing reliability and achieving a desired level of accuracy in the position estimations (i.e., a low *DOP*).

We define three types of areas within the indoor facility: 1) navigation area (*Av*), 2) beacon area (*Ab*) and 3) obstacles (*Ao*). Figure 1 shows a schematic representation of these areas. The navigation area is shown in yellow and represents the space on the floor where the vehicle moves and localization services must be provided. The grid represents the area on the ceiling of the facility where beacons could be located (i.e., the beacon area). Finally, the gray areas are obstacles that the vehicle must avoid. Some obstacles, such as walls, reach the ceiling of the facility and interrupt the signals used for positioning, acting as discontinuities in the beacon area. Other obstacles, however, do not reach the ceiling and therefore only affect the navigation area. In general, the navigation area does not have to be the same as the beacon area.

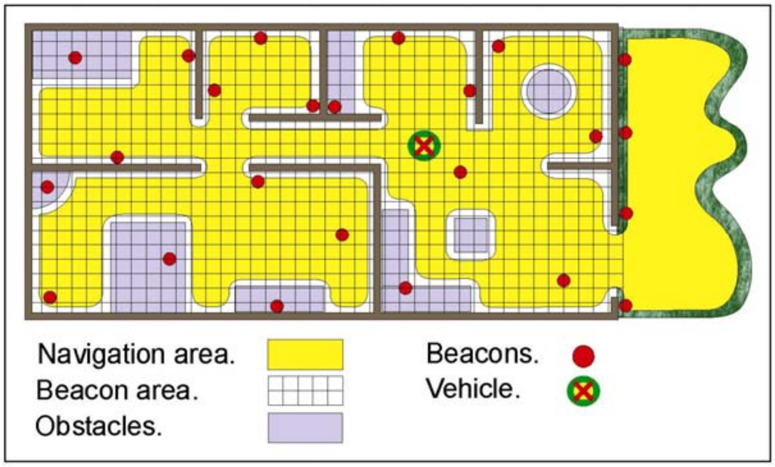


Figure 1. Schematic representation of a facility with a navigation system

This problem is general in the sense that it can be applied to multilateration processes regardless of the technology used to estimate the range between the beacons and the mobile device (e.g., RF, UF, IR, UWB or Laser) (Hightower, 2001). Different technologies can be considered within this framework by a precise characterization of the transducers used for emission and reception (in terms of patterns of emission, maximum range, etc). In current setting, we consider a sub-centimeter exact ultrasonic positioning system where broadband transducers and Golay codes are used to codify the emitted signals, allowing increased noise immunity, capability of simultaneous measurements and increased precision (Prieto, 2007). In this system, ranges to the mobile vehicle are estimated by measuring times-of-flight (TOFs) of RF-synchronized ultrasonic signals traveling from emitters to receivers. This is a common scheme used in the Robotics community for navigation of AGV's (Hazas and Ward, 2003; YI, 2003).

In our problem, we consider that the beacons will be placed at a fixed height, determined by the ceiling of the interior space under consideration. Therefore, the search for the best placement of beacons is restricted to a two-dimensional space, where the beacon's height, z_b, takes on a fixed value. We also assume that the height of the vehicle (or robot) is fixed. The fixed value, z_v, is the maximum height that the vehicle could reach in the navigation area, as shown in Figure 2.

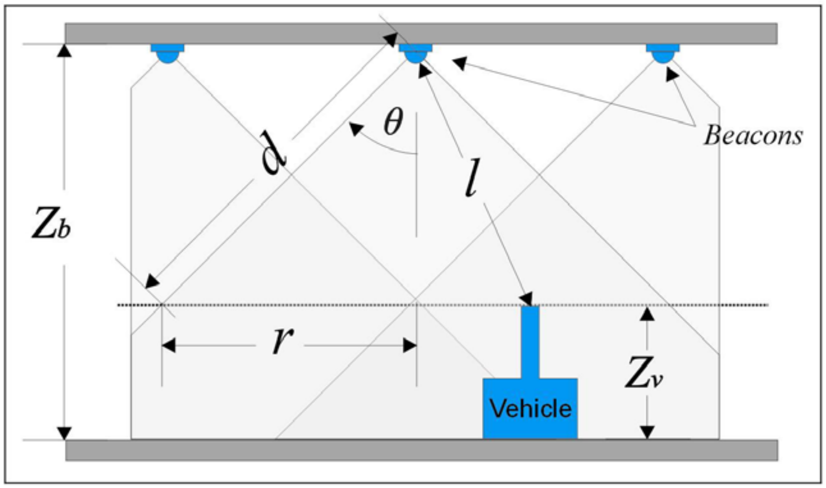


Figure 2. Horizontal range (r) as a function of z_b, z_v and θ

The horizontal range (r) covered by a beacon is a critical parameter for the optimization process. The horizontal range depends on the minimum sound pressure level (*SPL*) of the ultrasonic signal that is required to produce a reliable estimation of the TOF, and consequently, of the range (Minkoff, 1992). In the situation shown in Figure 2, the *SPL* is given by:

$$SPL(r) = SPL_{ref} - 20\log_{10}\left(\frac{d(r)}{d_{ref}}\right) + At_{abs} + At_e(\theta)\cdot At_r(\theta)$$

where SPL_{ref} is the *SPL* (in dB) at an axial distance d_{ref} from the transducer, as given in the datasheet, the second term corresponds to the attenuation of the ultrasonic signal caused by spreading of the wavefront during propagation, and At_{abs} represents the attenuation due to air absorption. Following Manthey (1992), this attenuation value (in dB) may be calculated as:

$$At_{abs} = \left(-1.6^{-10} f^2\right)\cdot\left(d(r) - d_{ref}\right)$$

This term dominates for high frequency ultrasonic signals and severely restricts the measuring range. This is the reason that frequencies (f) for indoor localization applications are commonly between 20 and 100 kHz. The term $At_e(\theta)\cdot At_r(\theta)$ is the product of the directivities of the emitter and receiver patterns, respectively. These values are either given by the manufacturer or are experimentally measured. The relationship between the distance traveled by the ultrasonic wave $d(r)$ and the viewing angle θ is given by:

$$d(r) = \sqrt{r^2 + \left(z_b - z_v\right)^2}$$

$$\theta = \operatorname{atan}\left(\frac{r}{z_b - z_v}\right)$$

In practical ultrasonic ranging systems, a minimum operating SPL_{min} is chosen and the operating range r_{min} is determined by solving $SPL(r_{min}) = SPL_{min}$ numerically. Thus, the beacon coverage is the circle of radius r_{min} around its position, assuming that no blocking obstacle exists in that range.

The horizontal range *r* is used to calculate the number of beacons available for evaluating the *DOP* at each point of the navigation area. This calculation is possible as long as the point of interest is inside the radius coverage of at least three beacons at the same time; otherwise such a point is considered uncovered. In general, if a given vehicle position (x_v, y_v, z_v) is covered by *n* beacons, we build an $n\times 3$ matrix *A* such that its ith-row is the normalized direction vector from the vehicle position to the ith-beacon:

$$\left(a_{i,1} = \frac{x_{b_i} - x_v}{l_i}, a_{i,2} = \frac{y_{b_i} - y_v}{l_i}, a_{i,3} = \frac{z_{b_i} - z_v}{l_i}\right)$$

where $(x_{b_i}, y_{b_i}, z_{b_i})$ are the coordinates of the i^{th} beacon and l_i is the distance between the i^{th} beacon and the vehicle, i.e., $l_i = \left\|(x_{b_i}, y_{b_i}, z_{b_i}) - (x_v, y_v, z_v)\right\|$, as shown in Figure 2. Then the *DOP* is calculated as follows (R. Yarlagadda, 2000):

$$DOP = \sqrt{m_{11} + m_{22} + m_{33}}$$

where $M = \{m_{ij}\} = A^{-1}$ for $n = 3$ and $M = \left(A^T A\right)^{-1}$ for $n > 3$.

A solution to the beacon layout problem is given by the set $\Omega = \{x_{b_i}, y_{b_i}, z_{b_i}; i = 1,\ldots,n\}$ of the coordinates of all beacons included in the design. The three objectives described in the previous section are aggregated in the objective function $f(\Omega)$, to be minimized:

$$\min f(\Omega) = \min \sum_{t=1}^{3} k_t f_t \,,$$

where

$$f_1 = \overline{DOP} \qquad f_2 = \left(\frac{sPoints}{nGrid}\right) \qquad f_3 = \left(\frac{n}{nGrid}\right)$$

To evaluate $f(\Omega)$ efficiently, we consider a grid of *nGrid* discrete points overlaid on *Av*. In our application, the points in the grid have a separation of 10 centimeters (*Grid density* = 100 points/m^2). Then, *nGrid* is given by:

$$nGrid = (Grid\ density)(Av) = 100\ Av$$

Let V_{ij} be the visibility between beacon i (b_i) at location (x_i, y_i) for $i = 1,2,...,n$ and position j (p_j) at location (x_j, y_j) for $j = 1,2,...,nGrid$. V_{ij} is a binary variable determined as follows:

$$V_{ij} = \begin{cases} 1 & if\ \left(\left\|b_i - p_j\right\| \le r_{\min}\right) \wedge LOS\left(b_i, p_j, Ao\right) \\ 0 & Otherwise. \end{cases}$$

Let C_j be the availability matrix at position *j*, which identifies points in the discretized *Av* $(p_j, j = 1,2,...nGrid)$ with at least three visible beacons and acceptable *DOP* value. C_j is also binary and is given by:

$$C_j = \begin{cases} 1 & if\ \left(\sum_{i=1}^{n}\left[V_{ij}\right] \ge 3\right) \wedge \left(DOP_j \le 10\right) \\ 0 & Otherwise. \end{cases}$$

Then, the average *DOP* and the *sPoints* can be expressed as functions of the *C* values as follows:

$$\overline{DOP} = \frac{\sum_{j=1}^{nGrid}\left[DOP_j C_j\right]}{\sum_{j=1}^{nGrid}\left[C_j\right]}$$

$$sPoints = \sum_{j=1}^{nGrid}\left[1 - C_j\right]$$

The first term in the objective function attempts to minimize $\overline{DOP}$, which is the average of the *DOP* values at all the points of the grid that are covered by the current configuration. The second term minimizes the percentage of *sPoints*, which are those positions in the grid where it is not possible to localize a moving vehicle (either due to a lack of enough beacons in LOS or *DOP* values above the desired threshold). The final term minimizes the number of beacons (*n*) used in the design. The weights k_1, k_2 and k_3 are set by the designer and measure the relative importance of the three components of the objective function, which can be described qualitatively as: *inaccuracy* of the position estimation, its *unavailability* and its *cost*, respectively. The objective of the optimization process is finding a design that minimizes the weighted sum of these three components.

Since the objective function is based in a known weighted aggregation of objectives, in our work, we considered the Conventional Weight Aggregation (CWA) and the Dynamic Weight Aggregation (DWA) methods in order to select the values of k_1 and k_2, according with the desired trade-off between *inaccuracy* and *unavailability* (Jin, 2001).

CWA assumes a priori knowledge in order to specify an appropriate set of weights. We considered only values of *DOP* smaller than 10 because larger values diminish notably the accuracy of position estimations, resulting in singular areas. Moreover, we assumed that the positioning system has odometric and inertial sensors allowing beaconless navigation in small areas. Therefore, a maximum of 20% of unavailable area was allowed. These considerations allowed us to establish the following relationship between k_1 and k_2 and the maximum inaccuracy and unavailability:

$$k_1 \cdot inaccuracy_{\max} = k_2 \cdot unavailability_{\max}$$
$$k_1 \cdot 10 = k_2 \cdot 0.2$$

Therefore, for $k_1 = 1 \Rightarrow k_2 = {}^{10}\!/_{0.2} = 50$.

Then, we used the DWA method to assess the performance of our relationship. In DWA, the weights in the objective function are changing periodically during the optimization process in order to find several solutions on the *Pareto Frontier* (*PF*); set of solutions to a multi-objective minimization problem where no improvement can be achieved by decreasing one objective without increasing another (Steuer, 1986). Then, the designer selects the efficient solutions according to a desired tradeoff of the objective functions.

Over a square area of 16.81 m^2 using 12 beacons, the objective function was minimized using the optimization process described in Section 3.2, changing the values of k_1 and k_2, in each search, from $k_2 >> k_1$ and vice versa. In this way, an approximation of the Pareto front was found (Figure 3). We confirmed that desired solutions — i.e., those with small unavailable areas and within the *DOP* threshold — were found when the $k_2/k_1 = 50$ relationship was used; see the red dot on Figure 3. This solution achieves an average DOP of 3.5 (f_1 value) and a 16% of unavailability (f_2 value). Ratios of k_2/k_1>100 tend to produce solutions with maximum coverage but poor accuracy, while ratios of k_2/k_1<20 produce the opposite effect.

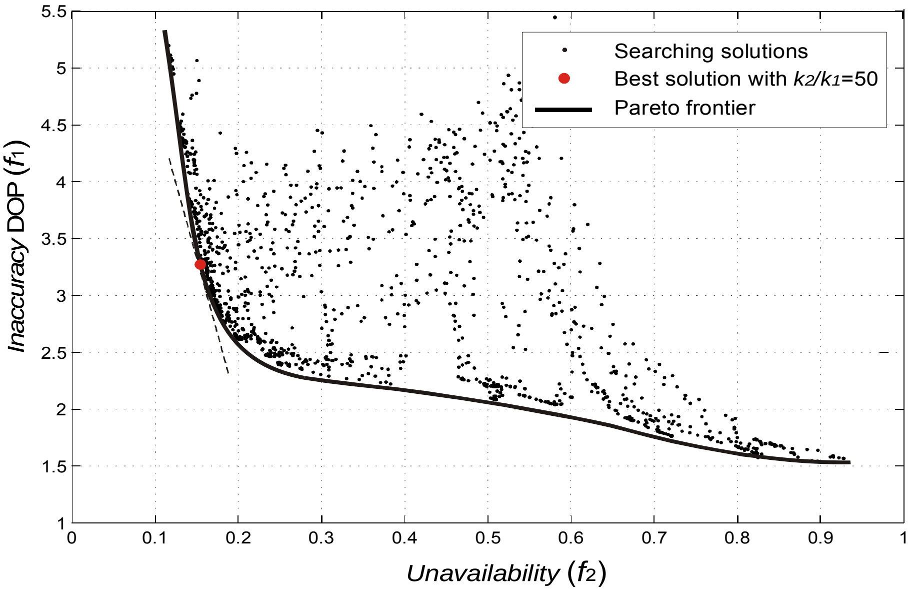


Figure 3. Pareto frontier found with DLS optimization process (Section 3.2) by changing k_1 and k_2 according to DWA method and the best solution (red point) with k_2/k_1=50.

In order to select a reasonable value for k_3, the optimization process was first performed using only the first two components of the objective function with the preferred weights for k_1 and k_2 (i.e., 1 and 50, respectively). Starting with 12 beacons, we performed searches eliminating one beacon at a time until only four beacons remained. Then, the value of k_3 was selected in such a way that the recalculated lower value in the objective function, using the three components, corresponded to the best deployment found with a desired 98% of coverage. This is how we arrived to the value of $k_3 = 20$. However, if we want $f(\Omega)$ to be interpreted as the cost in dollars for square meter, this value for k_3 is not a realistic cost of a beacon, as those used in Prieto (2007). To make it more realistic, we changed k_3 from 20 to 200 [$/beacon] and k_1 and k_2 were multiplied by 10 in order to maintain the desired balance in the function. We will show later that the value of k_3 is not relevant during the optimization process and we only use it to compare solutions found by the methods described below.

3 Optimization Processes

In this section, we describe the two solution methods that are available for the beacon layout design problem discussed above. The first method is based on genetic algorithms (GA) and was developed prior to this study. We provide a brief description of this method because we use it for comparison purposes. We then describe the implementation of our diversified local search.

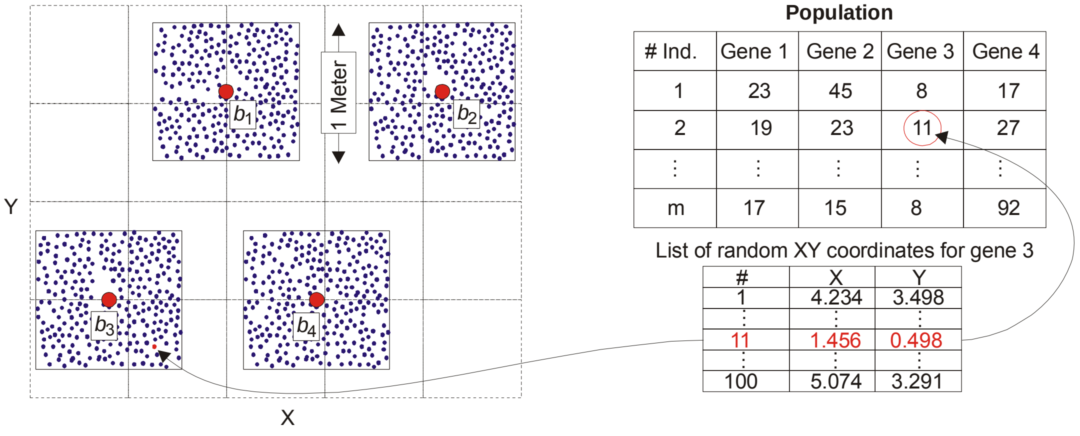
3.1 Genetic Algorithm

To the best of our knowledge, the GA by Roa, et al (2005) is the only existing procedure for the problem on hand. The optimization process starts from an initial solution based on locating beacons employing a regular pattern (see Section 3.2.1). Then, an initial population (group of individuals) is formed by randomly perturbing the positions of the beacons in the initial solution. Each individual in the initial population is evaluated, where fitness is given by the value of $f(\Omega)$. Selection and crossover operators are applied in order to obtain two new solutions, which are evaluated and added to the population. Then, individuals are sorted according to their fitness value and the two worst individuals are eliminated from the population. This process is repeated until a termination criterion is satisfied. Figure 4 shows a sketch of the GA process.

```
Construct initial solution
Initialize population randomly around initial solution.
Evaluate the population with the objective function $f(\Omega)$.
While (termination criterion not satisfied)
{
    Select two individuals with the selection operator
    Apply crossover operator in order to obtain two new individuals
    Evaluate and add the new individuals to the population
    Sort the individuals of population in ascending $f(\Omega)$ order
    Eliminate the two worst individuals from the population.
}
```

Figure 4. Outline of the GA by Roa, et al (2005)

Individuals in the context of the GA outlined in Figure 4 are beacon layouts. An individual consists of a vector of size n+1; where the first element is a number representing the index of individual. The remaining elements (genes) contain numbers that correspond to positions of beacons in *Ab*. Figure 5 shows a schematic representation of the individuals in a population.



Population

# Ind.	Gene 1	Gene 2	Gene 3	Gene 4
1	23	45	8	17
2	19	23	11	27
⋮	⋮	⋮	⋮	⋮
m	17	15	8	92

List of random XY coordinates for gene 3

#	X	Y
1	4.234	3.498
⋮	⋮	⋮
11	1.456	0.498
⋮	⋮	⋮
100	5.074	3.291

a) Initial layout with four beacons (red circles) and their perturbed locations (blue points).

b) Population with *m* individuals and four genes.

Figure 5. Schematic representation of a population of individuals

The population includes all individuals in the search process and is represented by a $m\times(n+1)$ matrix, where m is the size of the population and n is the number of beacons. The initial solution is represented by the red dots in Figure 5a. The initial population is created by randomly selecting 100 layouts, where each beacon is randomly located in a square around the initial location. These locations are shown as blue dots in Figure 5a. The 100 possible positions for each beacon are then coded as shown in Figure 5b. Individuals are represented by their index and the numbers representing the XY positions of the beacons, as shown in the "Population" table in Figure 5b. Note that when these individuals are combined — via crossover operations —in order to create new individuals, the positions of the beacons are limited to those coded in the XY list for each gene.

3.2 Diversified Local Search Method

Our new proposal to search for optimal beacon deployments in LPS is based on a neighborhood search that includes intensification and diversification phases. It is important to clarify that although we search for optimal layouts, our procedure is heuristic in nature and therefore the best layouts found are not known to be optimal in a global sense. The designs represent the best local optimal solutions that our search encountered. The intensification phase is a local search procedure while the diversification phase employs memory structures from the tabu search methodology (Glover and Laguna, 1997). The phases are alternated as shown in Figure 6. This design follows the strategies suggested by Kelly, Glover and Laguna (1994) in the context of the quadratic assignment problem. Once the intensification phase reaches a local optimal point, the diversification phase is triggered to encourage the search to move to a different region of the design space. The number of steps in the intensification phase varies according to the descend trajectory from the solution where the phase initiates. The diversification phase consists of a fixed number of iterations (*dSteps*). The process stops after a termination criterion is satisfied. The termination criterion may be given by a fixed number of searches or by a time limit. It is also possible to stop the optimization at the current *n*-level after a design of a specified quality has been found. In our experiments, we perform the inner while-loop of Figure 6 *nSearch* times.

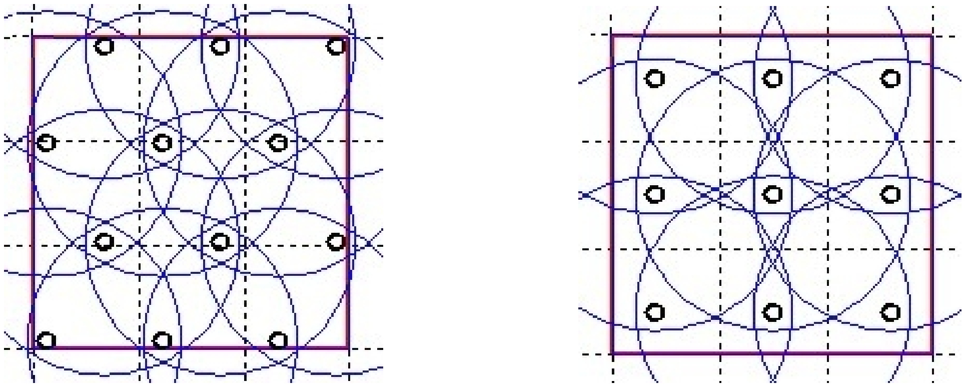
```
Construct initial solution
Let $n$ = number of beacons in the initial solution
while ($n \geq n_{\min}$)
{
    while (termination criterion not satisfied)
    {
        Perform search intensification
        Perform dSteps diversification iterations
    }
    $n = n - 1$
}
```

Figure 6. Outline of the optimization process

Figure 6 also shows that the third component of the objective function is handled explicitly as a hard constraint during the search. In other words, every time the termination criterion in the inner loop is satisfied, the maximum number of beacons is reduced and the search continues. This process makes the third component of the objective function a constant and therefore only the first two components are relevant during the search. The reduction of the number of beacons is achieved by eliminating one beacon from the current best solution. Let Ω_n^* be the best solution found with n beacons. Then, the initial solution for the next application of the intensification phase is the result of eliminating the beacon that causes the smallest increase in the objective function value of the current best solution. Mathematically, we look for the beacon that minimizes $f(\Omega_{n-1}) - f(\Omega_n^*)$ when setting k_3 = 0 (i.e., when ignoring the cost of the beacons). The outcome of this procedure is a set of solutions that are the best designs found for a given number of beacons. The maximum number of beacons is determined by the initial solution and the minimum number is given by $n_{\min}$.

3.2.1 Initial Solution

The initial solution is obtained by applying a systematic pattern for placing beacons in the beacon area. The construction procedure considers two patterns, square and triangular, as shown in Figure 7. The separation of the beacons in both patterns is determined by a minimum availability value (*aMin*). That is, the initial design should be such that $\left(1 - \frac{sPoints}{nGrid}\right) \geq aMin$.



a) Triangular

b) Square

Figure 7. Layout patterns for initial solutions

Two solutions are constructed, one with each pattern. The design with the fewer number of beacons is chosen. A second phase of the procedure attempts to eliminate redundant beacons from the chosen design. The beacons are sequentially considered and eliminations occur as long as the design meets the minimum availability value. The objective function value is calculated and the search is initiated from the resulting beacon configuration.

3.2.2 Intensification Phase

This phase consists of a local search based on the neighborhood depicted in Figure 8. The black circle represents a beacon under consideration and the gray circles represent the trial positions. We consider 8 possible directions for moving the current beacon. The directions are equally space at 45-degree angles from 0 to 315 degrees. Five different positions are tried in each direction. The positions are 10 centimeters apart, resulting in a maximum move of 50 centimeters in any of the eight directions.

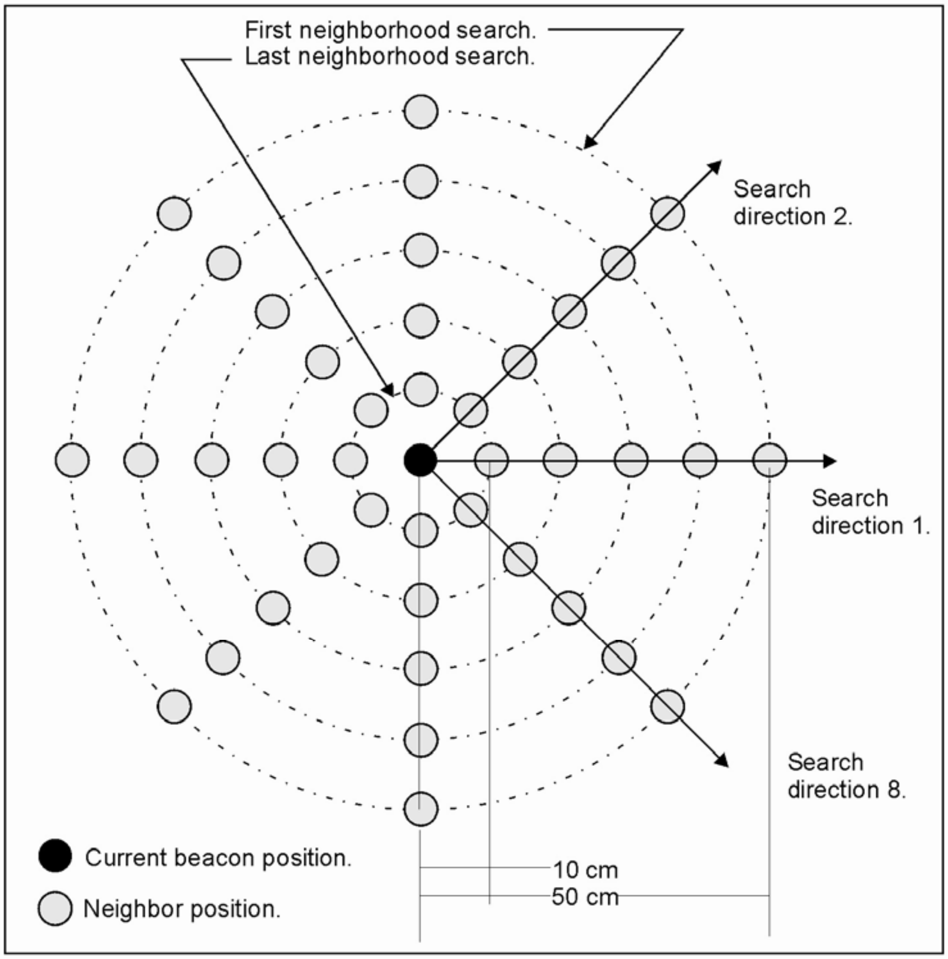


Figure 8. Neighborhood of a given beacon

The neighborhood is explored from the outside in. That is, the positions that are 50 centimeters away from the current position are explored first, followed by the positions that are 40 centimeters away, and so on. After the eight trial positions at a given distance are explored, the beacon is immediately moved if an improving position is found. Otherwise, the neighborhood search moves to the next (closer) set of eight positions. This means that a maximum of 40 positions are explored when either no improving neighbor position is found or the improving position is in the set of the eight closest positions. The beacons in the current design are considered one by one in a random order. This order changes ever time all the beacons have been considered. If all beacons are considered and none is moved, the maximum moving distance of 50 centimeters is reduced by 10 centimeters (and the spacing between the five concentric circles shown in Figure 8 is adjusted accordingly). The process finishes when the maximum distance reaches zero. At this point, all the beacons are locked in their positions and the solution is considered a local optimum with respect to our neighborhood.

During preliminary experimentation, we tested the idea of increasing the number of search directions when the search stalls. In particular, once the local search was unable to find better positions for any of the beacons, the maximum move was set to 30 centimeters and the number of directions was increased. This strategy was not as effective as the one described above and therefore we fixed the number of directions to eight and systematically reduced the length of the moves in an attempt to find the (locally) best positions for the beacons.

3.2.3 Diversification Phase

This phase starts from the last local optimal design found in the intensification phase. The main goal of this phase is to move the beacons away from their current positions. Instead of applying a totally random perturbation to the current design, this phase employs a more systematic approach to escape the current local optimum in search of a global optimal beacon layout. This phase is also based on the neighborhood depicted in Figure 8. The process works as shown in Figure 9.

```
while (number of steps ≤ dSteps)
{
    Order beacons randomly, where $b_i$ is the $i^{th}$ beacon in the list
    for ($i = 1, \ldots, n$)
    {
        Evaluate the neighbor positions for beacon $b_i$
        Move beacon $b_i$ to its best neighbor position
        Update the current and best objective function value
        Update the tabu memory structure
    }
}
```

Figure 9. Diversification phase

Each iteration of the diversification phase consists of moving all the beacons to a new position. The procedure uses a short-term memory structure that classifies certain moves as tabu. The structure can be thought of as consisting of n tabu matrices of size t×4, one for each beacon, where t is the *tabu tenure*. A row in the tabu matrix contains four values, consisting of the two pairs of *xy* coordinates that correspond to the forbidden move. Initially, there are no forbidden (i.e., tabu) moves and therefore the matrices are blank. Suppose that at iteration 1 the i^{th} beacon is moved from (2.3, 5.7) to (2.38, 5.78) then the first row of the i^{th} matrix will be updated to (2.38, 5.78, 2.3, 5.7) indicating that the move from (2.38, 5.78) to (2.3, 5.7) is forbidden. Each tabu matrix is a circular list and therefore after t iterations all rows are updated. This means that a move is forbidden for only t iterations.

The while-loop in Figure 9 first creates a random ordering of the beacons. Then, each beacon is considered following the current random order. The entire neighborhood of the beacon under consideration is evaluated and the best neighbor position is identified. The best neighbor position must meet one of the two following conditions:

1. The move from the current position to the neighbor position must be non-tabu.

2. If the move from the current position to the neighbor position is tabu then the objective function value of the beacon layout after the move should be better than the best objective function value found during the entire search.

The best position from all those meeting one of the two previous conditions is selected and the beacon is moved to it. The move may result in a deterioration of the objective function value of the current solution. In fact, such deterioration is guaranteed in the first iteration of the diversification phase given that the initial beacon layout is a local optimum with respect to the neighborhood that is used in both phases. After a few iterations of the diversification phase, however, it is possible to find improving search trajectories. Although we are mainly concern with inducing diversification, this phase has an element of intensification given by the rule that chooses the “best” available move at each step.

After the move, the objective function value of the current solution must be updated. If the move is to a new best design then the best solution and its corresponding objective function value must be updated as well. Finally, the tabu memory structures are updated and the process is repeated until all beacons are considered. The diversification phase stops after *dSteps*. No adjustment of the neighborhood is performed during this phase and the moving distance for the beacons is set to 30 centimeters.

We also experimented with a variant of the diversification phase that delayed the movement of beacons until all of the neighborhoods were explored. In this variant, a move is chosen only after all of the beacons are considered and therefore each iteration is considerably more computationally expensive than the procedure outlined in Figure 9. In our preliminary experimentation we determined that extra effort related to find the best overall move did not result in improved outcomes. Hence, the experiments reported in the next section focus on the performance of the procedure shown in Figure 9.

4 Experimental Results

In this section, we describe the experiments that we performed to test the proposed optimization procedure. We created three initial test cases, whose characteristics are summarized in Table 1.

Test	Shape	Area (Ab=Av) [m^2]	Radius ($r_{\min}$)[m]	Beacons $n_{\min}$
1	Square	16.81	2	4
2	L	14.91	1.4	10
3	Horseshoe	14.01	1.28	10

Table 1. Characteristics of test cases

The height of the vehicle (z_v) is set at 2 meters and the beacons are placed at 4 meters (i.e., $z_b = 4$). In each test case, we consider that the beacon area (*Ab*) and navigation area (*Av*) have the same shape. We use the following values for the search parameter:

$k_1 = 10$
$k_2 = 500$
$k_3 = 200$
aMin = 100%
dSteps = 12
nSearch = 3
$t = 8$

Tables 2 to 4 show the values of each component of the objective function found for the designs with number of beacons ranging from the one in the initial design to the specified $n_{\min}$. The initial designs with 12, 18 and 17 beacons for cases 1, 2 and 3, respectively, guarantee 100% of availability (as specified by the *aMin* value). Since the cost of the design is fixed for a given number of beacons, the third component of objective function does not play a role during the optimization process. In other words, the search is completely driven by the tradeoff between accuracy and availability, as specified by the values of k_1 and k_2. The value of k_3 is therefore used only to obtain an overall evaluation of the best designs found for each value of n. In Tables 2-4, GA refers to the genetic algorithm in Roa, et al. (2005) and DLS to our diversified local search, both implemented in Matlab. We ran DLS with the parameters specified above and recorded the total time used during the optimization process. We then ran the GA for the same amount of time (approximately 30 CPU minutes on a PC with an Intel processor at 1.5GHz and 512 of RAM).

	Number of beacons (*n*)								
	12	11	10	9	8	7	6	5	4
GA: $f(\Omega)$	166.24	153.97	148.54	148.45	160.53	181.68	202.68	241.79	309.76
Inaccuracy f_1 (*DOP*)	2.31	2.25	2.42	2.70	3.35	3.38	4.83	4.34	7.21
Unavailability $f_2.Av$ [m^2]	0.01	0.02	0.18	0.48	1.07	2.17	2.79	4.67	6.39
DLS: $f(\Omega)$	163.36	153.13	144.94	144.12	155.52	172.37	198.97	238.91	296.18
Inaccuracy f_1 (*DOP*)	2.05	2.19	2.41	2.84	3.89	3.85	4.75	4.46	7.90
Unavailability $f_2.Av$ [m^2]	0.00	0.01	0.06	0.29	0.72	1.70	2.69	4.53	5.70
Cost [\$/m^2]	142.77	130.87	118.97	107.07	95.18	83.28	71.38	59.48	47.59

Table 2. Objective function values of designs for test case 1

	Number of beacons (n)								
	18	17	16	15	14	13	12	11	10
GA: $f(\Omega)$	271.79	258.54	247.19	237.91	229.99	231.23	236.41	233.92	264.67
Inaccuracy f_1 (*DOP*)	3.03	3.05	3.22	3.43	3.78	4.17	4.45	4.88	5.91
Unavailability $f_2.Av$ [m^2]	0.00	0.00	0.01	0.07	0.13	0.45	0.92	1.12	2.13
DLS: $f(\Omega)$	269.26	257.18	245.47	234.44	226.52	223.20	223.57	228.40	245.14
Inaccuracy f_1 (*DOP*)	2.78	2.91	3.08	3.28	3.70	4.44	5.12	5.60	6.47
Unavailability $f_2.Av$ [m^2]	0.00	0.00	0.00	0.01	0.05	0.13	0.34	0.74	1.38
Cost [$\$/\mathrm{m}^2$]	241.44	228.03	214.62	201.20	187.79	174.37	160.96	147.55	134.13

Table 3. Objective function values of designs for test case 2

	Number of beacons (n)							
	17	16	15	14	13	12	11	10
GA: $f(\Omega)$	284.24	273.31	272.28	278.73	297.79	302.68	329.47	340.11
Inaccuracy f_1 (*DOP*)	3.87	3.84	4.24	4.63	4.72	5.50	5.32	6.24
Unavailability $f_2.Av$ [m^2]	0.08	0.19	0.46	0.96	1.93	2.27	3.55	4.02
DLS: $f(\Omega)$	278.33	269.82	263.99	268.20	275.43	292.11	313.52	333.89
Inaccuracy f_1 (*DOP*)	3.56	3.89	4.16	5.26	5.63	5.97	5.47	6.40
Unavailability $f_2.Av$ [m^2]	0.00	0.07	0.23	0.44	0.94	1.71	2.85	3.56
Cost [$\$/\mathrm{m}^2$]	242.68	228.40	214.13	199.85	185.58	171.30	157.03	142.75

Table 4. Objective function values of designs for test case 3

Tables 2, 3 and 4 show the merit of the proposed procedure in terms of finding designs that in all cases provide higher levels of accuracy and availability. Furthermore, DLS converges to high-quality designs at a much faster rate than GA, as illustrated in Figure 10. This figure shows the value of the objective function corresponding to the best solution found by each procedure for the Horseshoe case with 15 beacons. In this run, we allowed both procedures to go beyond the 30-minute mark in order to observe their long-term behavior. DLS was ran with *nSearch* = 6, resulting in 80 minutes of computational time. The GA was then run for the same amount of time.

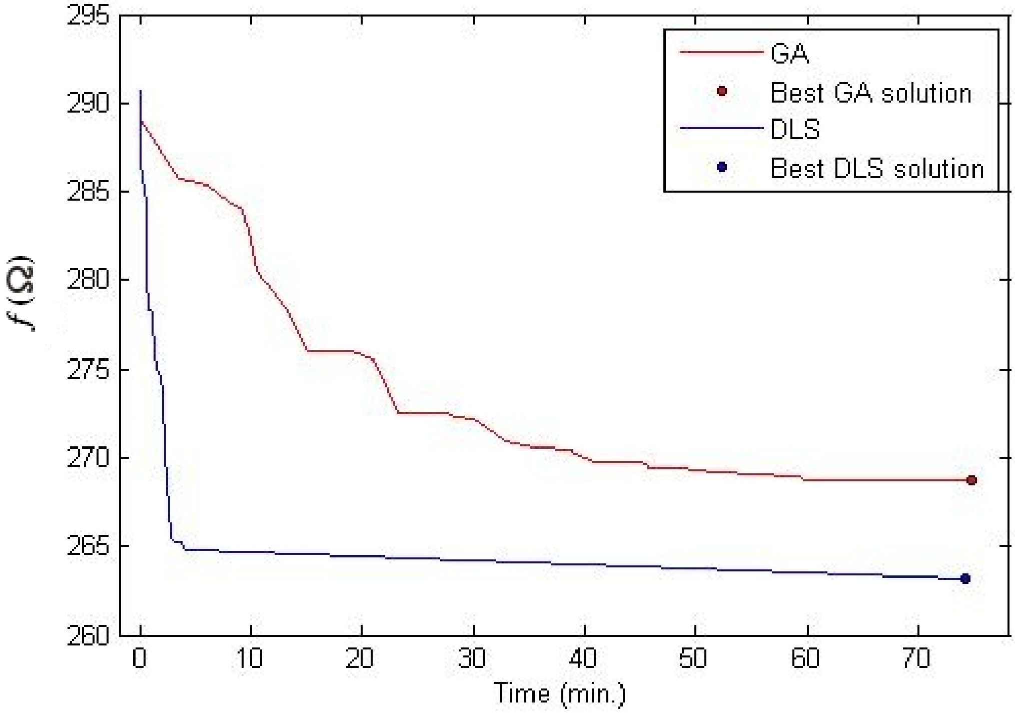
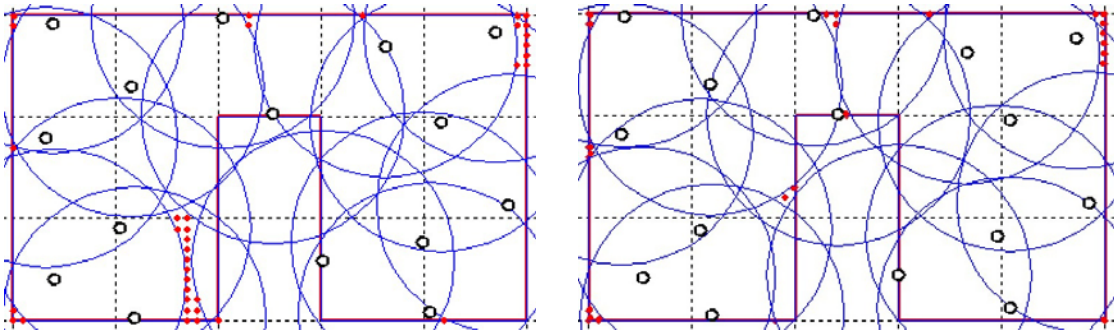


Figure 10. Performance curves for DLS and GA for Horseshoe with *n* = 15

Figure 10 shows the aggressive nature of DLS. The search rapidly descends to solutions with objective function values that are significantly better than those that the GA finds in the initial stages of the search (e.g., within the first 10 minutes of processing time). Even after the 30-minute limit established in our original experiment, DLS is capable of improving the best solution at a faster rate than the GA. The final solutions found after the stopping criteria are satisfied also favor DLS in terms of the corresponding objective function values. Figure 11 shows the designs found at the end of the search depicted in Figure 10. The red dots in Figure 11 indicate the 0.46 m^2 and 0.23 m^2 of area where the system is not available in the GA and DLS designs, respectively. It is interesting to note that, given enough time, the procedures arrive at somewhat similar designs.



a) GA Design

b) DLS Design

Figure 11. Designs for the Horseshoe case with 15 beacons. The red dots indicate the unavailable area

We performed one final experiment for which we constructed an area of 69.23 m^2 with walls and obstacles (test case 4). We used the same parameter values as indicated above and set $n_{\min} = 40$. The characteristics of the solutions found during the search are shown in Table 5.

No. of Beacons (n)	$f(\Omega)$ [\$/m^2]	Inaccuracy f_1 (DOP)	Unavailability $f_2 \cdot Av$[m^2]	Cost $k_3 \cdot f_3$ [\$/m^2]
61	194.00	1.77	0	176.22
60	191.31	1.79	0	173.33
59	188.63	1.81	0	170.44
58	185.90	1.83	0	167.55
57	183.25	1.85	0	164.66
56	180.58	1.88	0	161.77
55	177.98	1.90	0	158.89
54	175.36	1.93	0	156.00
53	172.72	1.96	0	153.11
52	170.16	1.99	0	150.22
51	167.60	2.02	0	147.33
50	165.18	2.07	0	144.44
49	162.95	2.13	0.01	141.55
48	160.33	2.15	0.01	138.66
47	158.66	2.19	0.13	135.77
46	157.03	2.29	0.16	132.89
45	155.43	2.34	0.28	130.00
44	153.93	2.46	0.30	127.11
43	152.74	2.60	0.34	124.22
42	152.14	2.70	0.52	121.33
41	156.83	2.65	1.64	118.44
40	159.01	2.90	2.00	115.55

Table 5. Results of the experiment with test case 4

The results in Table 5 indicate that the best solution found (according to the weights used for the objective function value) corresponds to the design with 42 beacons. The resulting total cost is $121.33 and the solution has reasonable values of inaccuracy and unavailability. The actual design is shown in Figure 12. The red dots in the lower corners of the design indicate the 0.52 m^2 of area where positioning is not available.

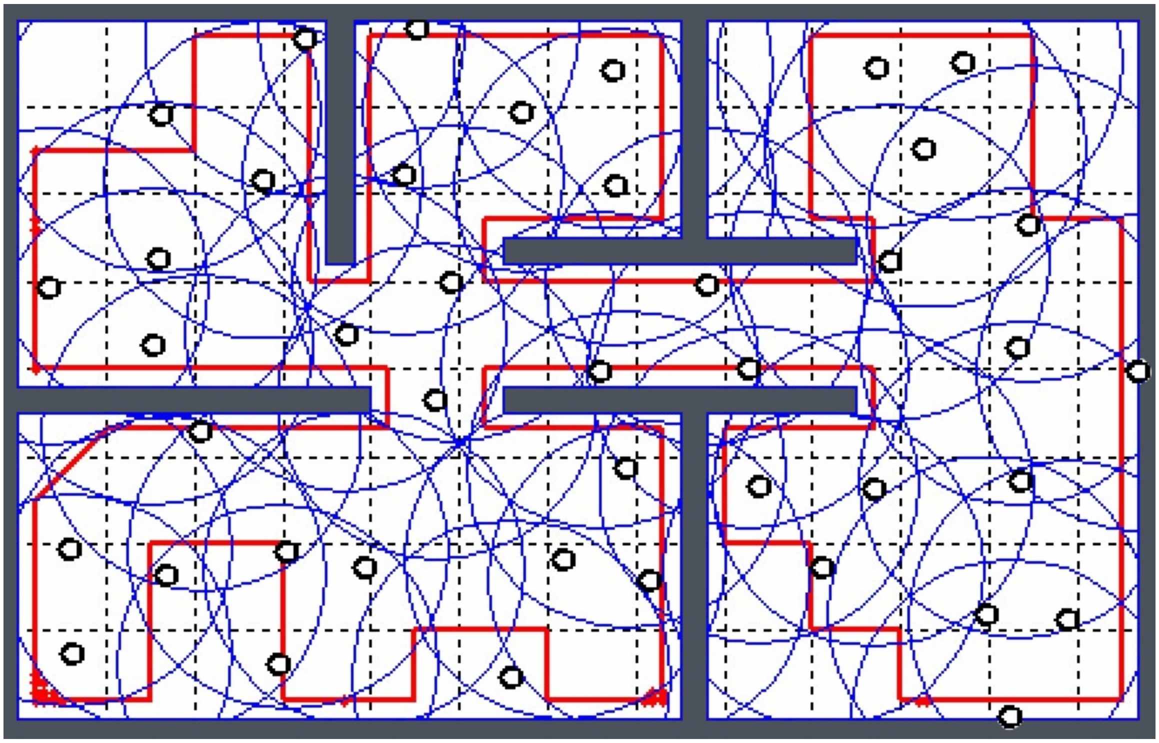


Figure 12. Best design with 42 beacons for test case 4

The coverage calculations in test case 4 take into consideration that the navigation and beacon areas are different. In particular, the navigation area is delimited by the red lines and the beacon area is the larger surface with the white background. The wide dark bars are walls that are neither part of the navigation area nor the beacon area. The process to find the design in Figure 12 required approximately 28 hours.

5 Conclusions

We have described the development of an optimization procedure for the problem of finding the optimal location of beacons in an indoor positioning facility. The procedure consists of two alternating phases that result in an aggressive search trajectory toward high-quality designs. These designs avoid singularities and provide coverage with high accuracy. These characteristics are important for the deployment of sensor networks with minimum infrastructure costs. Our tests show that the proposed procedure is more effective than the only other one that currently exists for this engineering design problem. We have created a set of 4 test cases that show the merit of our proposal. We believe that there is room for improvement regarding the effectiveness and efficiency of solution procedures for this problem but, at the same time, we have set a benchmark that we hope will motivate future research.

6 References

Glover, F. and M. Laguna (1997) *Tabu Search*, Kluwer Academic Publisher, Boston.

Hazas, M. and A. Ward (2003) “A High Performance Privacy-Oriented Location System,” In Proceedings of PerCom 2003: *First IEEE International Conference on Pervasive Computing and Communications*, Dallas-Fort Worth, USA, pp 216–223.

Hightower, J. and Borriello, G. (2001) “Location systems for ubiquitous computing” *IEEE Computer*, vol. 34 (8) pp. 57-66.

Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

Jin, Y., Olhofer, M. and Sendhoff, B. (2001) “*Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?*” In Proceedings of the Genetic and Evolutionary Computation Conference GECCO, pp. 1042-1049.

Kang, Y. K., H. C. Park, W. Hwang and K. S. Han (1996) “Optimum Placement of Piezoelectric Sensor/Actuator for Vibration Control of Laminated Beams,” *AIAA Journal*, vol. 34, no. 9.

Kang, Y. K., H. C. Park and B. Agrawal (1998) “Optimization of Piezoelectric Sensor/Actuator Placement for Vibration Control of Laminated Plates,” *AIAA Journal*, vol. 36, no. 9.

Kelly, J. P., M. Laguna and F. Glover (1994) “A Study on Diversification Strategies for the Quadratic Assignment Problem,” *Computers and Operations Research*, vol. 21, no. 8, pp. 885-893.

Manthey, W., N. Kroemer and V. Magori (1992) “Ultrasonic Transducers and Transducer Arrays for Applications in Air,” *Measurement Science and Technol*gy, vol. 3, no. 3, pp. 249-261.

Minkoff, J. (1992) *Signals, Noise and Active Sensors*, John Wiley & Sons.

Oh, D. Y. and H. C. No (1995) “Determination of the Minimal Number of Optimal Sensor Location in a Nuclear System with Fixed Incore Detectors,” *Nuclear Engineering and Design*, vol. 152, pp. 197-212.

Prieto, J. C., Jiménez, A. R., Guevara, J. I., Ealo, J. L, Seco, F. A., Roa J. O. and Ramos, F. X. (2007) “Subcentimeter-accuracy localization through broadband acoustic transducers” *in proceedings of the IEEE International Symposium on Intelligent Signal Processing* (WISP'2007), pp. 1-6.

Roa, J., Jiménez, A.R., Seco, F., Prieto, C., Ealo, J. and Ramos, F. (2005) “Primeros Resultados en la Optimización de la Ubicación de Balizas para Localización utilizando

Algoritmos Genéticos” in *proceedings of the XXVI Jornadas de Automática*, ISBN: 84-689-0730-8. pp. 75-86.

Sinriech, D. and S. Shoval (2000) “Landmark Configuration for Absolute Positioning of Autonomous Vehicles,” *IIE Transactions*, vol. 32, pp. 613-624.

Steuer, R. E. (1986) “*Multiple Criteria Optimization: Theory, Computation, and Application*” New York: Wiley & Sons, Inc.

Yarlagadda, R., I. Ali, N. Al-Dhahir and J. Hershey (2000) “GPS GDOP metric” *IEE Proc.-Radar, Sonar Navig.*, vol 147, no. 5, pp. 259-264.

YI, S-Y (2003) “Global Ultrasonic System for Self-Localization of Mobile Robot” *IEICE Transaction on Comunications*, vol. E86-B, no. 7, pp. 2171-2177.