

A Comparison of Antenna Placement Algorithms

Abhinav Jauhri

April 5, 2015

Motivation

- ▶ Systems with multiple antennas offer interference, and thereby reduce each antenna's efficiency
- ▶ Parasitic effects due to the platform
- ▶ Frequency bands change over time requiring new antennas, and therefore need to find new placements

Outline of this talk

- ▶ Part 1: Introduction to the antenna placement problem
- ▶ Part 2: Description of stochastic algorithms, their properties and operators
- ▶ Part 3: Evaluation of test cases

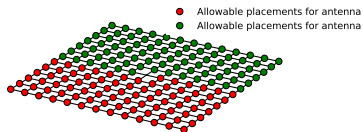
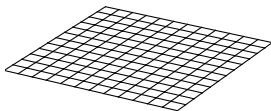
Part 1: Introduction to the antenna placement problem

Antenna Placement Problem

Given, platform

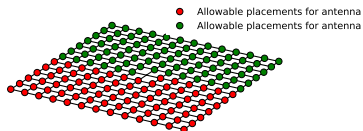
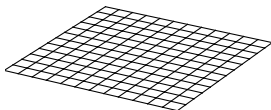
+

allowable placements of antennas

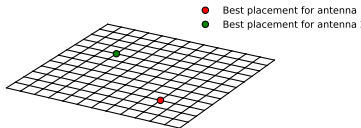


Antenna Placement Problem

Given, platform + allowable placements of antennas



Problem: Find best antenna placements to maximize gain and minimize coupling



Antenna Placement Problem

Given:

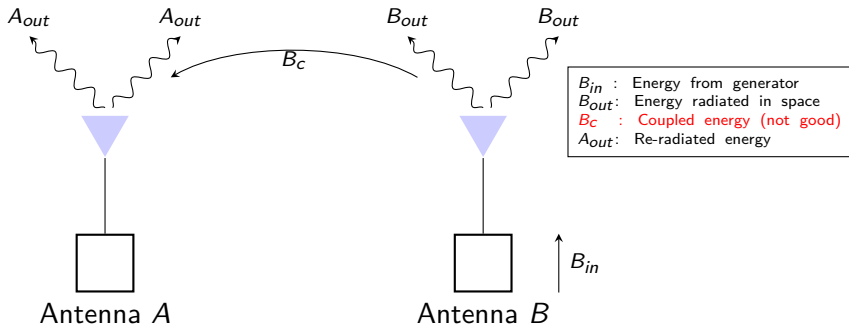
- ▶ platform P with its surface gridded such that end points represent possible antenna placements
- ▶ set of m antennas $A = A_1, A_2, \dots, A_m$ such that $m > 1$
- ▶ for each A_i , L_i denote the set of allowable placements $\in \mathbb{R}^3$ such that $|L_i| = n_i$ and $\forall i, n_i > 1$;
$$L_i = \{(x_1, y_1, z_1) \dots (x_{n_i}, y_{n_i}, z_{n_i})\}$$

Problem: Find a set of n optimal antenna placements on P to maximize gain and minimize coupling

Question: How is a good antenna placement quantified in the context of platform and other antennas?

Mutual Coupling

When two antennas are in proximity, and one is transmitting, the second will receive some of the transmitted energy.



Minimize Mutual Coupling

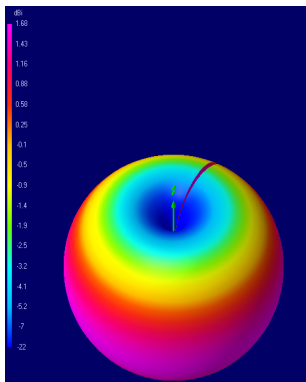
$$F_{MC} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n CP(A_i, A_j), \quad (1)$$

where

- ▶ $CP(\cdot, \cdot) \in \mathbb{R}$ is the coupling between two antennas, and computed using a simulator

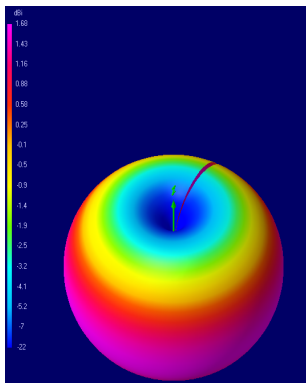
Example: If $n = 3$, then $F_{MC} = CP(A_1, A_2) + CP(A_1, A_3) + CP(A_2, A_3)$

Radiation Pattern

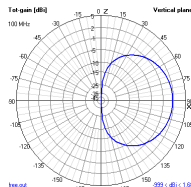


Free-space pattern without platform or other antennas

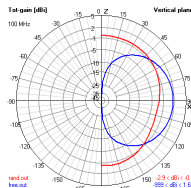
Radiation Pattern



Free-space pattern without platform or other antennas



2D view of the free space gain pattern



In-situ pattern (in red) for random antenna placements

Minimize Difference in Radiation Pattern

$$F_{RP} = \sum_{i=1}^n \sum_{\theta=0}^{\pi/S} \sum_{\phi=0}^{2\pi/S} (FSG_i(S\theta, S\phi) - ISG_i(S\theta, S\phi))^2, \quad (2)$$

where

- ▶ S is the step size
- ▶ θ, ϕ spherical coordinates
- ▶ $FSG(\cdot, \cdot) \in \mathbb{R}$ is the free-space gain pattern computed by the simulator
- ▶ $ISG(\cdot, \cdot) \in \mathbb{R}$ is the in-situ gain pattern computed by the simulator

Fitness Evaluation

Find a placement such that F is minimal:

$$F = \alpha F_{MC} + \beta F_{RP}, \quad (3)$$

where α, β are adjustable weights for each of the objectives

Part 2: Stochastic Algorithms

Stochastic Algorithms

We will consider algorithms which rely on *randomization* principle.

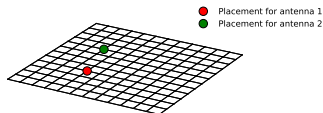
- ▶ Genetic Algorithm
- ▶ Evolutionary Strategy
- ▶ Simulated Annealing
- ▶ Hill Climbing

Each algorithm maintains a candidate solution or pool of candidate solutions called population

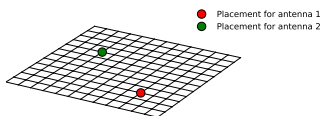
Stochastic Algorithms: Operand

Candidate solution or an **individual** is a member of a set of possible solutions.

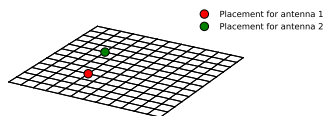
- ▶ Simulated Annealing and Hill Climbing maintain single individual



- ▶ Genetic Algorithm and Evolutionary Strategy maintain a population of individuals

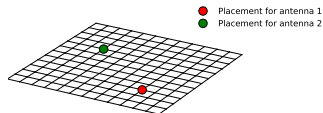


...

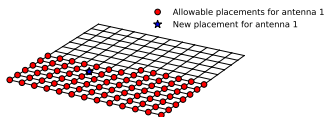


Stochastic Algorithms: Mutation Operator

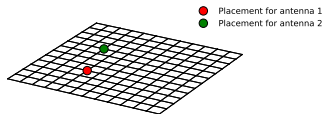
1. Given an individual, select an antenna uniformly at random, let's say antenna 1:



2. For antenna 1, select any other allowable placement:

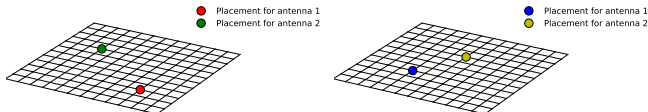


3. Change position for antenna 1 in individual:

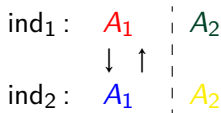


Stochastic Algorithms: Crossover Operator

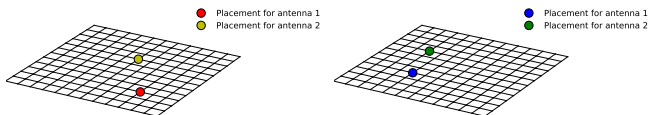
1. Select two individuals from population:



2. Select a crossover point, and swap placements prior to the point:

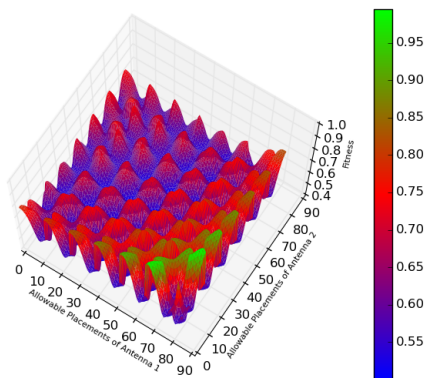


3. Two new offsprings created:



Question: Why use stochastic algorithms?

Multi-Modal Search Space



Search space for one of the test cases evaluated. There are multiple local minimas which makes convergence difficult. z-axis is the combined fitness F (refer Eq. 3)

Genetic Algorithm

```

1  $P \leftarrow$  generate  $p$  random individuals. Compute
   $fitness(ind_i), i \in [1, p];$ 
2  $i = 0$  ;
3 while  $i < gen_{max}$  do
4   Elitism: Select  $n_e$  fittest individuals to add to  $P'$  ;
5   for  $(p - n_e)/2$  times do
6     /* 'select' returns a pair of individuals */
7      $M \leftarrow select(P, 2)$  ;
8     if  $rand(0, 1) < p_c$  then
9        $O \leftarrow crossover(M)$  ;
10      Add  $O$  to  $P'$  ;
11    else
12      Add  $M$  to  $P'$  ;
13  Uniformly select  $p_m \cdot (p - n_e)$  individuals from  $P$ ,
14  and apply mutation operator to each ;
15  Update  $P \leftarrow P'$  ;
16  Compute  $fitness(ind_i), i \in [1, p];$ 
17  Update  $i \leftarrow i + 1$  ;

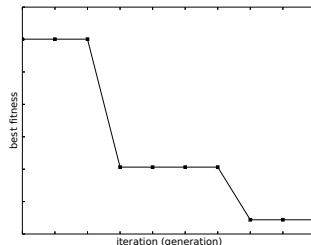
```

Genetic Algorithm

```

1  $P \leftarrow$  generate  $p$  random individuals. Compute
   $fitness(ind_i), i \in [1, p]$ ;
2  $i = 0$  ;
3 while  $i < gen_{max}$  do
4   Elitism: Select  $n_e$  fittest individuals to add to  $P'$  ;
5   for  $(p - n_e)/2$  times do
6     /* 'select' returns a pair of individuals */
7      $M \leftarrow select(P, 2)$  ;
8     if  $rand(0, 1) < p_c$  then
9        $O \leftarrow crossover(M)$  ;
10      Add  $O$  to  $P'$  ;
11    else
12      Add  $M$  to  $P'$  ;
13  Uniformly select  $p_m \cdot (p - n_e)$  individuals from  $P$ ,
14  and apply mutation operator to each ;
15  Update  $P \leftarrow P'$  ;
16  Compute  $fitness(ind_i), i \in [1, p]$ ;
17  Update  $i \leftarrow i + 1$  ;

```



Plateaus suggesting
stagnation of search

Evolutionary Strategy

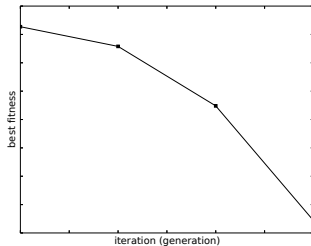
```
1  $\mathbf{P} \leftarrow$  generate  $\mu$  random individuals ;
2  $i = 0$  ;
3 while  $i < gen_{max}$  do
4   Create  $\lambda/\mu$  offsprings from each  $\mu$  individuals by
   applying mutation operator;
5   Add all offsprings to  $\mathbf{P}$  ;
6   Compute  $fitness(ind_i), i \in [1, \lambda + \mu]$  ;
7   Keep  $\mu$  best individuals in  $\mathbf{P}$ , and discard remaining
    $\lambda - \mu$  individuals ;
8   Update  $i \leftarrow i + 1$ 
```

Evolutionary Strategy

```

1  $\mathbf{P} \leftarrow$  generate  $\mu$  random individuals ;
2  $i = 0$  ;
3 while  $i < gen_{max}$  do
4     Add all offsprings to  $\mathbf{P}$  ;
5     Create  $\lambda/\mu$  offsprings from each  $\mu$  individuals by
applying mutation operator ;
6     Compute  $fitness(ind_i), i \in [1, \lambda + \mu]$  ;
7     Keep  $\mu$  best individuals in  $\mathbf{P}$ , and discard remaining
 $\lambda - \mu$  individuals ;
8     Update  $i \leftarrow i + 1$ 

```



Less likely to stagnate
search space exploration

Simulated Annealing

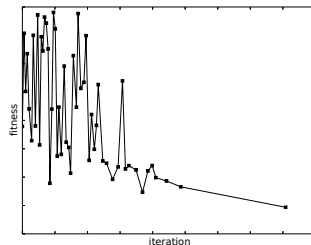
```
1 c ← generate a random individual ;
2 i = 0 ;
3 while i < imax do
4     n ← mutate(c) ;
5     if fitness(c) < fitness(n) then
6         if rand(0,1) < e-δf/T then
7             /* replace current individual by a higher
              fitness (less fitter) individual          */
              c ← n
8         else
9             c ← n ;
10    T ← T · fcooling ;
11    i ← i + 1 ;
```

Simulated Annealing

```

1  c ← generate a random individual ;
2  i = 0 ;
3  while i < imax do
4      n ← mutate(c) ;
5      if fitness(c) < fitness(n) then
6          if rand(0,1) < e-δf/T then
7              /* replace current individual by a higher
11             fitness (less fitter) individual */
12             c ← n
13          else
14              c ← n ;
15          T ← T · fcooling ;
16          i ← i + 1 ;

```



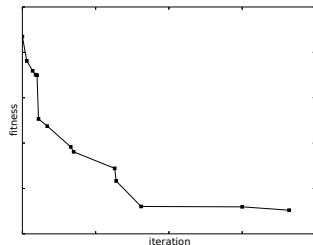
Fluctuation in fitness
gradually reduces due to
cooling

Hill Climbing

```
1 Initialize  $c \leftarrow$  generate a random individual ;
2 Compute  $fitness(c)$  ;
3  $i = 0$  ;
4 while  $i < i_{max}$  do
5      $n \leftarrow mutate(c)$  ;
6     if  $fitness(n) < fitness(c)$  then
7          $c \leftarrow n$ 
8      $i \leftarrow i + 1$ 
```

Hill Climbing

```
1 Initialize  $c \leftarrow$  generate a random individual ;
2 Compute  $fitness(c)$  ;
3  $i = 0$  ;
4 while  $i < i_{max}$  do
5    $n \leftarrow mutate(c)$  ;
6   if  $fitness(n) < fitness(c)$  then
7      $c \leftarrow n$ 
8    $i \leftarrow i + 1$ 
```



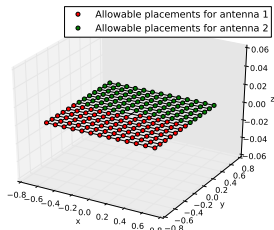
Greedy approach to
accept only fitter (low
fitness) individuals

Part 3: Evaluation of test cases

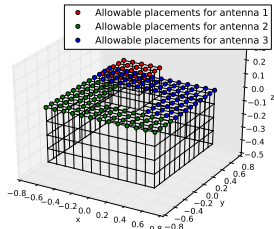
Experimental Setup

1. All test cases describe platforms which are replicas of real-world use cases like mobile devices, tanks, and cars
2. We use a popular *NEC2* simulator ¹ to get fitness parameters
3. Evaluate the entire search space using an exhaustive algorithm to find the optimal antenna locations

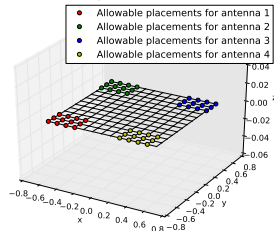
Experiments: Test Cases



Test Case #1 with 7056 (84×84) allowable placements



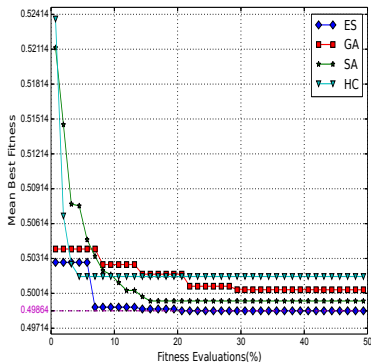
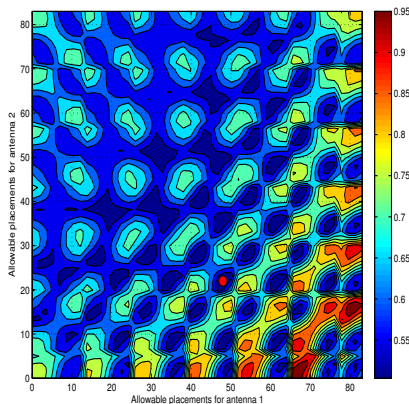
Test Case #2 with 50625 ($45 \times 45 \times 45$) allowable placements



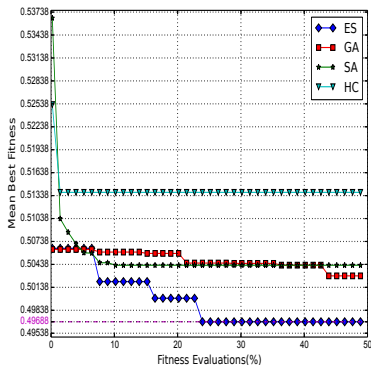
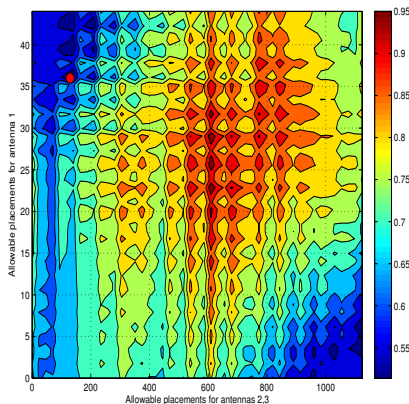
Test Case #3 with 126025 ($71 \times 71 \times 25$) allowable placements

Test Case #4 with 20736 ($12 \times 12 \times 12 \times 12$) allowable placements

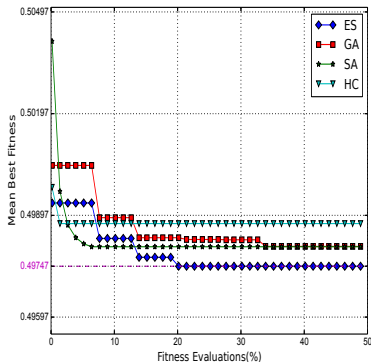
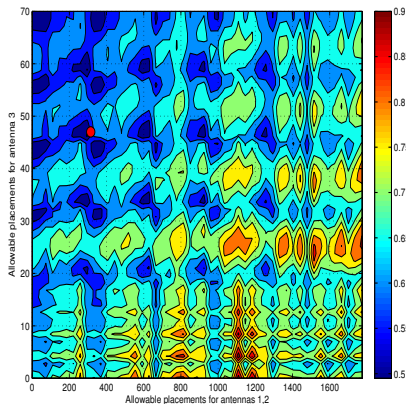
Results - Test Case 1



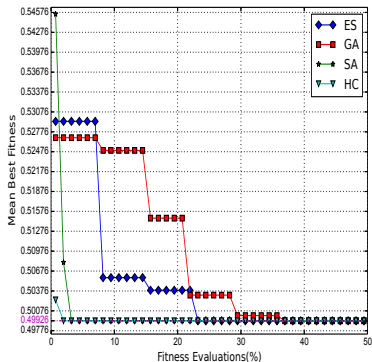
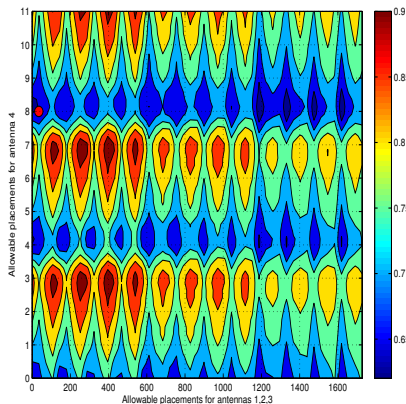
Results - Test Case 2



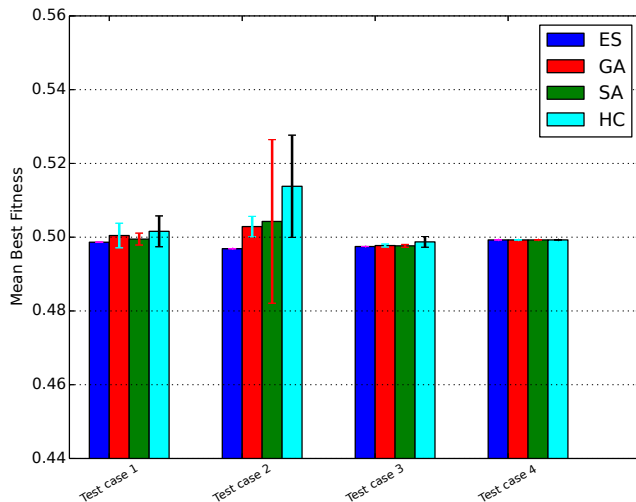
Results - Test Case 3



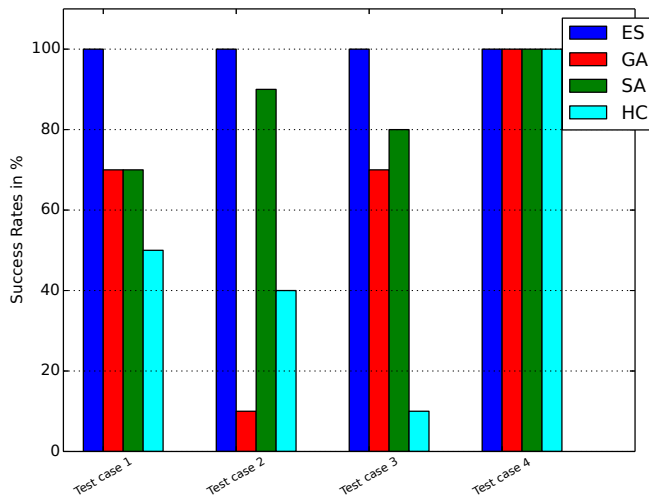
Results - Test Case 4



Results - Mean Best Fitness With Std. Dev.



Results - Success Rates



Conclusion

- ▶ Formulation of the antenna placement problem
- ▶ Generic problem formulation to accommodate multiple antennas and platforms
- ▶ Optimal placements found using Evolutionary Strategy with at most 25% evaluations of search space