A Comparison of Antenna Placement Algorithms

Abhinav Jauhri

April 22, 2015

Motivation

- Antenna placement study is often ignored
- Placing new antennas requires a long, manual effort to complete an antenna placement study, if at all
- Parasitic effects due to fixed or mobile platform
- With multiple antennas systems offer interference, and thereby reduce each antenna's efficiency

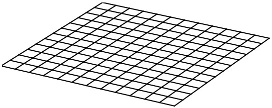
Outline of this talk

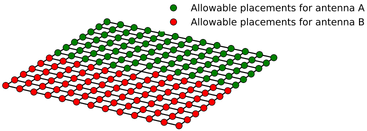
- Part 1: Introduction to the antenna placement problem
- Part 2: Description of stochastic algorithms, their properties and operators
- Part 3: Evaluation of test cases

Part 1: Introduction to the antenna placement problem

Antenna Placement Problem

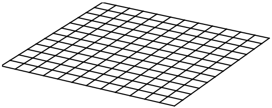
Given, platform + allowable placements of antennas

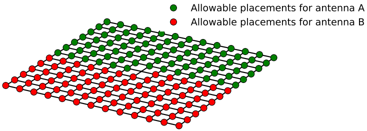




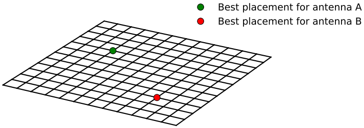
Antenna Placement Problem

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 n antennas $A = A_1, A_2, \ldots, A_n$ such that $n > 1$
- for each A_i, L_i denote the set of allowable placements $\in \mathbb{R}^3$ such that $\mid L_i \mid = m_i$ and $\forall i, m_i > 1$

$$L_i = \{(x_1, y_1, z_1) \ldots (x_{m_i}, y_{m_i}, z_{m_i})\}$$

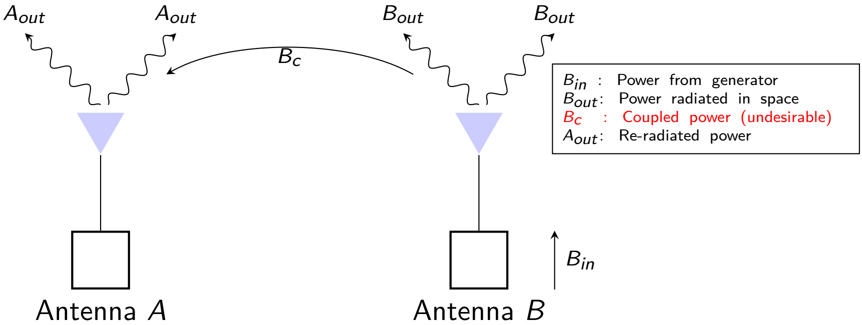
Problem: Find a set of n optimal antenna placements on P to maximize gain and minimize coupling.

Size of search space $= \mathbf{m^n}$, if $m_i = m, \forall i \in [1, n]$

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 power.



Minimize Mutual Coupling (MC)

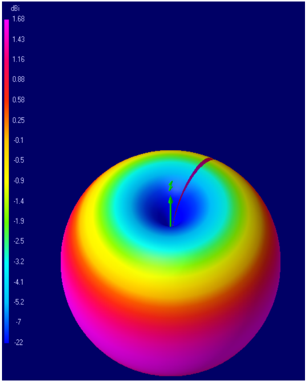
$$F_{MC} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} CP(A_i, A_j), \qquad (1)$$

where

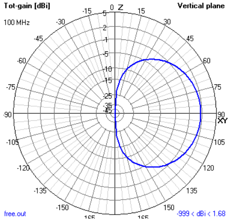
- $CP(\cdot, \cdot) \in \mathbb{R}$ is the coupling between two antennas, and computed using a simulator
- There will be $\binom{n}{2}$ coupling terms

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

Free Space Gain Pattern / Radiation Pattern



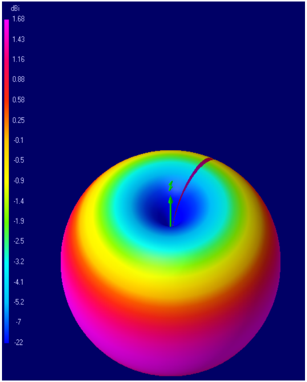
Free-space pattern without platform or other antennas



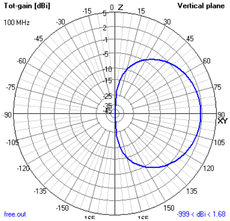
2D view of the free-space gain pattern

This is ideal pattern since there is no interference

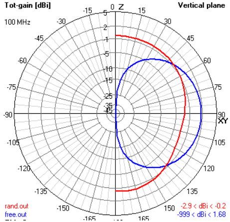
Gain Pattern



Free-space pattern without platform or other antennas



2D view of the free-space gain pattern



In-situ gain pattern for random antenna placement different from free-space gain pattern

Minimize Difference in Gain Pattern (GP)

$$F_{GP} = \sum_{i=1}^{n} \sum_{\theta=0}^{\frac{180^\circ}{S}} \sum_{\phi=0}^{\frac{360^\circ}{S}} \left(FSG_i(S\theta, S\phi) - ISG_i(S\theta, S\phi)\right)^2, \quad (2)$$

where

- S is the step size
- θ, ϕ spherical coordinates in degrees
- $FSG(\cdot,\cdot) \in \mathbb{R}$ 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 configuration such that **fitness** F is minimal:

$$F = \alpha F_{MC} + \beta F_{GP}, \tag{3}$$

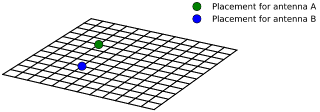
where α, β are adjustable weights for each of the objectives

Part 2: Stochastic Algorithms

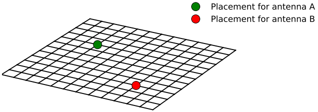
Individual(s)

An **individual** is a member of a set of feasible solutions.

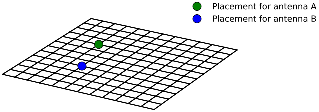
- An algorithm operates on an individual:



- Some algorithms operate on a population of individuals:

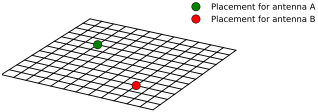


. . .

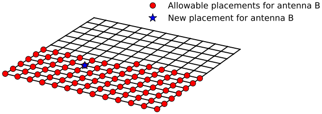


Mutation Operator

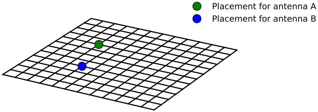
1. Given an individual, select an antenna uniformly at random, say antenna B:



2. Select uniformly at random from other allowable placements of antenna *B*:

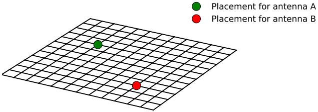


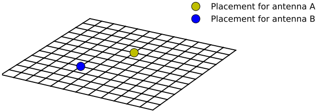
3. Change position for antenna B in individual, whereas antenna A's position remains same:



Crossover Operator

1. Select two individuals from population:

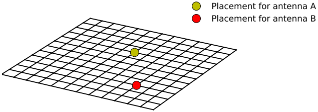


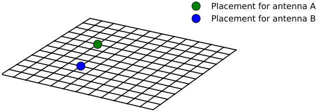


2. Select a crossover point, and swap placements prior to the point:

$$\begin{array}{ll|l} \text{ind}_1: & A & B \\ & \downarrow \; \uparrow & \\ \text{ind}_2: & A & B \end{array}$$

3. Two new offsprings created:





Stochastic Algorithms

We will consider algorithms which are based on randomization principle:

- Operate on a population of individuals:

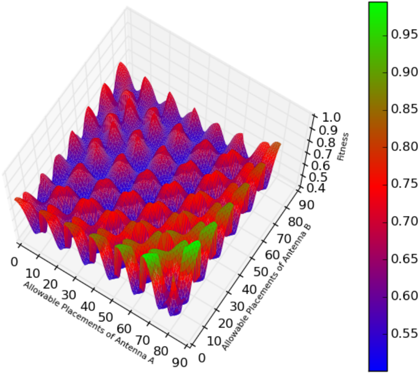
  1. **Genetic Algorithm**
  2. **Evolutionary Strategy**

- Operate on a single individual:

  3. **Simulated Annealing**
  4. **Hill Climbing**

Question: Why use stochastic algorithms?

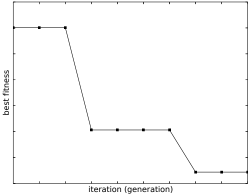
Fitness Plot



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

Genetic Algorithm

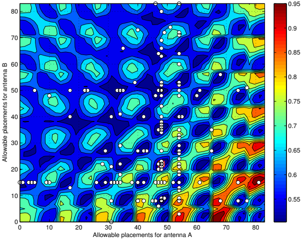
Generate initial populaiton P_0;
Compute fitness of each individual;
$i \leftarrow 1$;
while $i < gen_{max}$ **do**
    $P_i \leftarrow \emptyset$;
    Elitism: Copy some percentage of fittest inidividuals to P_i ;
    for *(population_size - elites)* /2 **do**
        Select a pair of individuals ;
        Perform *crossover* with some probability;
        Add new or original pair as it is to P_i;
    Apply *mutation* to a fraction of individuals in P_i;
    Update $i \leftarrow i + 1$;



Example: Progress of GA applied fitness minimization problem. Each point shows the fitness of the best individual over iterations.

Genetic Algorithm

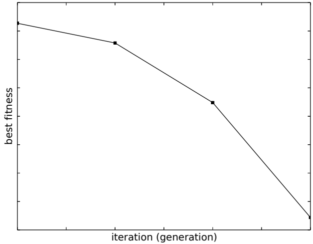
Generate initial populaiton P_0;
Compute fitness of each individual;
$i \leftarrow 1$;
while $i < gen_{max}$ **do**
    $P_i \leftarrow \emptyset$;
    Elitism: Copy some percentage of fittest inidividuals to P_i ;
    for *(population_size - elites)* /2 **do**
        Select a pair of individuals ;
        Perform *crossover* with some probability;
        Add new or original pair as it is to P_i;
    Apply *mutation* to a fraction of individuals in P_i;
    Update $i \leftarrow i+1$;



Fitness of population shown with ∘ for last iteration over the contour plot. Population has less diversity.

Evolutionary Strategy

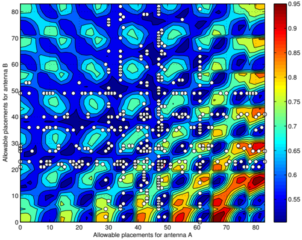
Generate initial populaiton P_0;
Compute fitness of each individual;
$i \leftarrow 1$;
while $i < gen_{max}$ **do**
    $P_i \leftarrow \emptyset$;
    Apply *mutation* operator multiple times to each individual in P_{i-1} to create offsprings ;
    Compute fitness for all offsprings ;
    Copy a fraction of P_{i-1} individuals ordered by fitness into P_i ;
    Update $i \leftarrow i+1$



Example: Progress of ES applied to fitness minimization problem

Evolutionary Strategy

Generate initial populaiton P_0;
Compute fitness of each individual;
$i \leftarrow 1$;
while $i < gen_{max}$ **do**
    $P_i \leftarrow \emptyset$;
    Apply *mutation* operator multiple times to each individual in P_{i-1} to create offsprings ;
    Compute fitness for all offsprings ;
    Copy a fraction of P_{i-1} individuals ordered by fitness into P_i ;
    Update $i \leftarrow i+1$

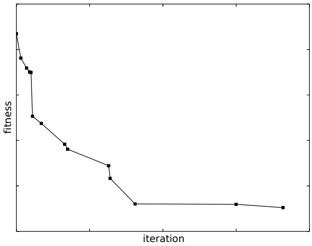


Fitness of a population shown with ∘ for last iteration. Greater diversity in comparison to GA[1]

[1] Spears, William M., and Kenneth A. DeJong. "Dining with GAs: operator lunch theorems."

Hill Climbing

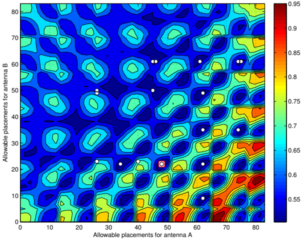
```
Generate a random inidividual ind_curr ;
Compute fitness of ind_curr ;
i ← 1 ;
while i < i_max do
    Create another individual ind_new by mutation of ind_curr ;
    if fitness(ind_new) < fitness(ind_curr) then
        ind_curr ← ind_new
    i ← i + 1
```



Example: Progress of HC applied to fitness minimization problem

Hill Climbing

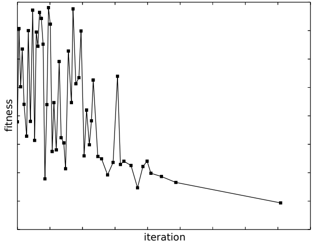
1. Generate a random inidividual ind_{curr} ;
2. Compute fitness of ind_{curr} ;
3. $i \leftarrow 1$;
4. **while** $i < i_{max}$ **do**
    - Create another individual ind_{new} by *mutation* of ind_{curr} ;
    - **if** $fitness(ind_{new}) < fitness(ind_{curr})$ **then**
        - $ind_{curr} \leftarrow ind_{new}$
    - $i \leftarrow i + 1$



Fitness of ind_{curr} individuals over an entire run shown with $\circ$. Search is restricted due to greedy approach to accept only fitter (low fitness) individuals

Simulated Annealing

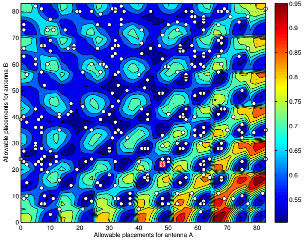
Generate a random inidividual ind_{curr} ;
Compute fitness of ind_{curr} ;
$i \leftarrow 1$;
while $i < i_{max}$ **do**
    Create another individual ind_{new} by *mutation* of ind_{curr} ;
    if $fitness(ind_{new}) > fitness(ind_{curr})$ **then**
        if $rand() < e^{-\delta f / T}$ **then**
            $ind_{curr} \leftarrow ind_{new}$
    else
        $ind_{curr} \leftarrow ind_{new}$
    $T \leftarrow T \cdot f_{cooling}$;
    $i \leftarrow i + 1$;



Example: Progress of SA applied to fitness minimization problem. As iterations increase, worse individuals with lower delta fitness (δf) are accepted.

Simulated Annealing

Generate a random inidividual ind_{curr} ;
Compute fitness of ind_{curr} ;
$i \leftarrow 1$;
while $i < i_{max}$ **do**
  Create another individual ind_{new} by *mutation* of ind_{curr} ;
  if $fitness(ind_{new}) > fitness(ind_{curr})$ **then**
    if $rand() < e^{-\delta f / T}$ **then**
      $ind_{curr} \leftarrow ind_{new}$
  else
    $ind_{curr} \leftarrow ind_{new}$
  $T \leftarrow T \cdot f_{cooling}$;
  $i \leftarrow i + 1$;



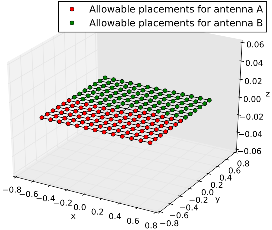
Fitness of ind_{curr} individuals over an entire run shown with $\circ$. Search is distributed across the terrain

Part 3: Evaluation of test cases

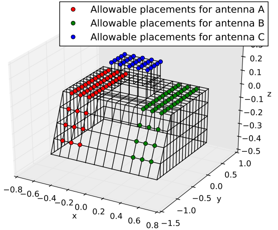
Experimental Setup

1. We use a popular NEC2 simulator to get fitness parameters
2. Evaluated the entire search space using an exhaustive algorithm to find the optimal antenna locations which is not ordinarily possible
3. Termination criteria was set to be at most 50% evaluations of the search spcae
4. 1000 independent runs of each test case against each algorithm with $\alpha = \beta = 1/2$

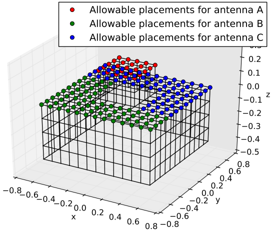
Experiments Test Cases



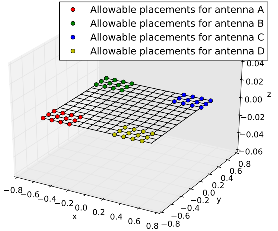
Test Case #1: search space size of 7056 (84*x*84) allowable placements



Test Case #2: search space size of 50625 (45*x*45*x*25) allowable placements



Test Case #3: search space size of 126025 (71*x*71*x*25) allowable placements

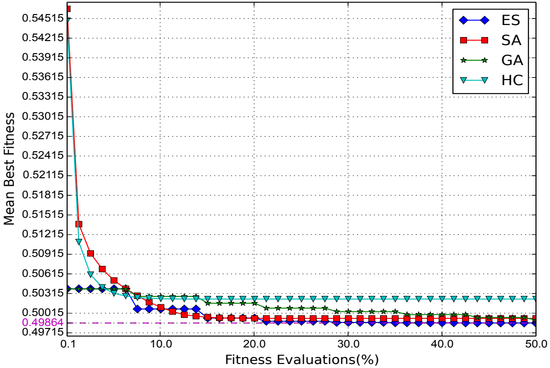


Test Case #4: search space size of 20736 (12*x*12*x*12*x*12) allowable placements

Results - Test Case 1

Sample size = 1000

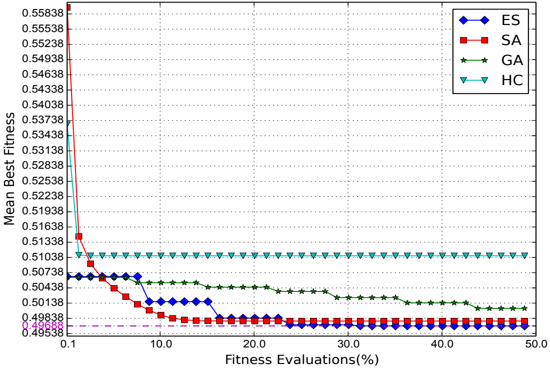
Algorithm	%Evaluations vs. Exhaustive		Best fitness	
	Mean	Std. Dev.	Mean	Std. Dev.
ES	11.88	10.48	0.49865	0.00009
SA	8.28	4.47	0.49935	0.00163
GA	17.21	15.69	0.49949	0.00182
HC	2.50	2.20	0.50230	0.00501



Results - Test Case 2

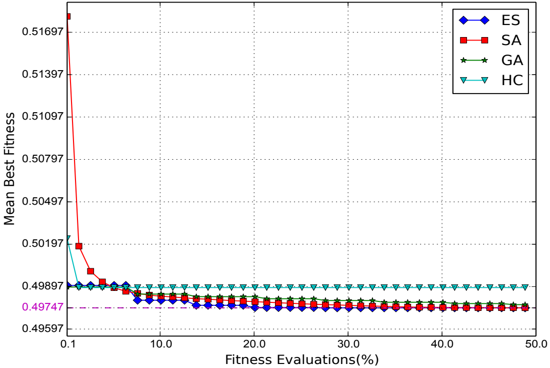
Sample size = 1000

Algorithm	%Evaluations vs. Exhaustive		Best fitness	
	Mean	Std. Dev.	Mean	Std. Dev.
ES	16.08	7.72	0.49688	0.00000
SA	7.96	3.33	0.49784	0.00233
GA	25.98	15.51	0.50034	0.00341
HC	0.40	0.31	0.51071	0.01305



Results - Test Case 3

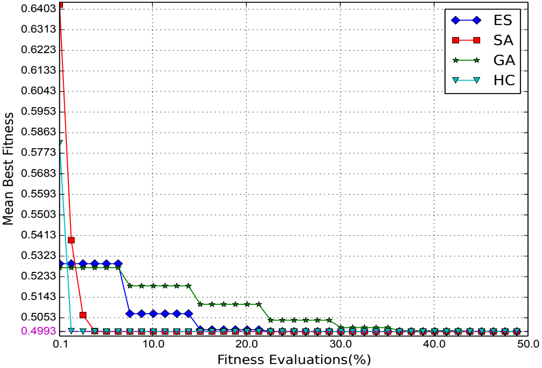
Sample size = 1000				
Algorithm	%Evaluations vs. Exhaustive		Best fitness	
	Mean	Std. Dev.	Mean	Std. Dev.
ES	11.04	6.72	0.49747	0.00000
SA	19.61	11.16	0.49747	0.00003
GA	23.05	16.25	0.49770	0.00038
HC	0.21	0.17	0.49890	0.00182



Results - Test Case 4

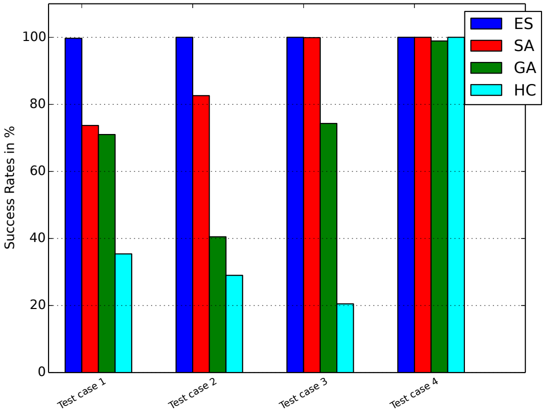
Sample size = 1000

Algorithm	%Evaluations vs. Exhaustive		Best fitness	
	Mean	Std. Dev.	Mean	Std. Dev.
ES	12.48	5.61	0.49926	0.00000
SA	2.76	0.83	0.49926	0.00000
GA	22.42	9.94	0.49934	0.00072
HC	0.44	0.26	0.49926	0.00000



Results - Success Rates

Success rate reports percentage of runs in which the algorithm is able to find the optimum with 50% evaluations as termination criteria



Conclusions

- First study to investigate optimizing multiple antenna placement on a single platform using multiple stochastic algorithms
- Formulated an automated procedure for the antenna placement problem which aims to improve the working of multiple antennas on a platform

Conclusions

- Results show Simulated Annealing was less successful but faster to converge compared to Evolutionary Strategy
- Evolutionary Strategy was slower to converge but success rate $\approx$ 100% with a mean of at most 16% evaluations of search space
- Algorithms reduce search time to at most 1/4 in comparison to an exhaustive algorithm
- Future work - Consider other techniques like *Differential Evolution*, *Particle Swarm Optimization* and *ALPS*