Advanced Computation:
Computational Electromagnetics

Optimization

1

Outline

- Introduction
- The Merit Function
  - The "Rectangle" Algorithm
- Direct Methods
  - Complete search
  - Others
- Stochastic Optimization
  - Genetic algorithms
  - Simulated annealing
  - Particle swarm optimization

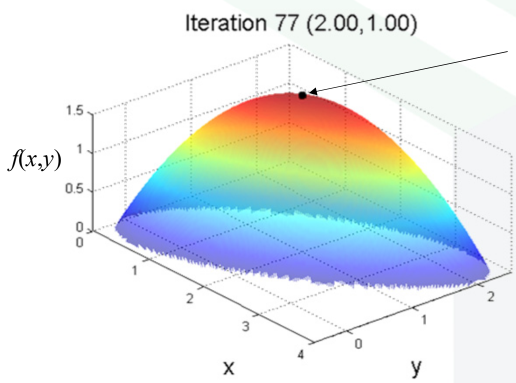
EMPossible Slide 2

2

Introduction

What is Optimization?

Optimization is simply finding the minimum or a maximum of a function. In this regard, it is similar to root finding.

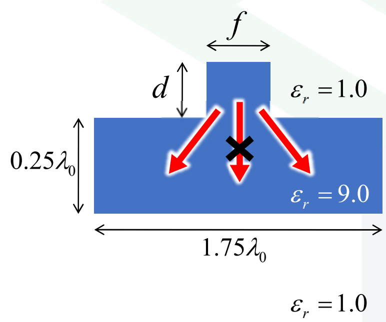


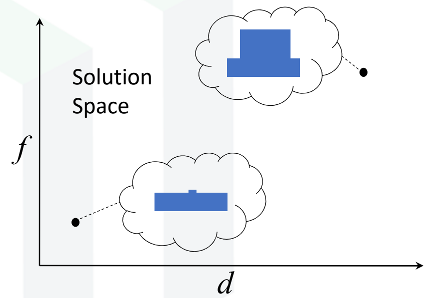
The function is usually chosen to be a measure of how "good" something is.

Merit function
Fitness function
Quality function

A Simple Example

Suppose we need to choose f and d so as to prevent diffraction into the zero-order transmitted mode for a normally incident wave. How do we do this?





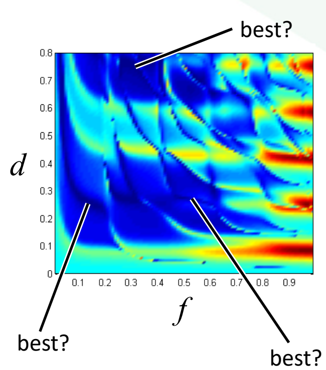
This is a simple problem where it is feasible to simulate every point in the entire solution space and pick the global minimum. Many times, it will take centuries to simulate the entire solution space. This is not feasible and another approach is needed.

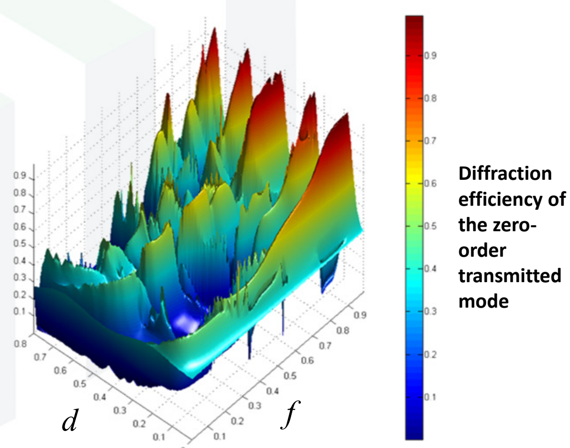
EMPossible Slide 5

5

Global Best Vs. Local Best

The solution space is often sprinkled with many possible solutions (local extrema). It is the primary goal of optimization to find the absolute best solution (global extrema). Without having some apriori knowledge of the solution, however, it is usually impossible to determine if the solution is a global best solution.





EMPossible Slide 6

6

Common Optimization Algorithms

- Direct Methods
  - Complete Search — Only guaranteed method for finding the global extrema.
  - Gradient Methods — Converges very quickly to a local extrema. No global search.
- Stochastic Optimization
  - Particle Swarm Optimization
  - Genetic Algorithms
  - Simulated Annealing — Random search to converge to a local best solution.

(Particle Swarm Optimization, Genetic Algorithms:) Searches globally. Usually finds a good solution. No guarantee it is a global best solution.

EMPossible

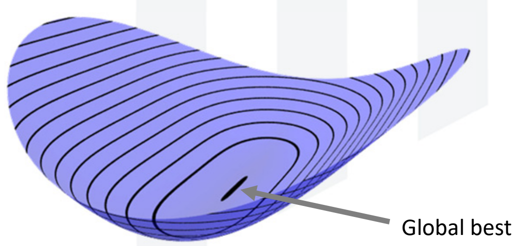
Slide 7

7

Can We Find the Global Best Solution?

Other than performing a complete search of the solution space, there is no known method to determine if an "optimized" solution is the global best.

Further, there is no known method to determine if a problem even has a global best. At present, only continuous convex problems are guaranteed to have a global best solution.



EMPossible

Slide 8

8

Notes on Optimization

- Stochastic methods are used most effectively when very little is know about the solution space. That is, when the engineer has no idea what the best design will look like.
- Unless something is known about the solution space, it is not possible to certify that the global best solution has been found. As engineers, we are usually satisfied with "good enough" solutions.
- Only an exhaustive complete search can guarantee a global best solution has been found.
- Direct methods converge very fast, but can only find local best solutions.
- It's all about the merit function, not the algorithm.

EMPossible

Slide 9

9

The Merit Function

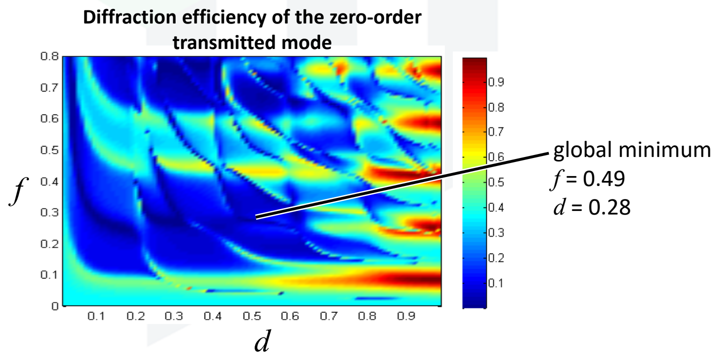
Slide 10

10

The Merit Function

We need a single number that tells us how "good" a particular solution is. This is called the merit function. The optimization can attempt to minimize or maximize this merit function. In this case, our merit function is the diffraction efficiency in the zero-order transmitted mode and we want to minimize it.

$$M\left(f,d\right) = T_0 \equiv \text{transmittance of zero-order mode}$$

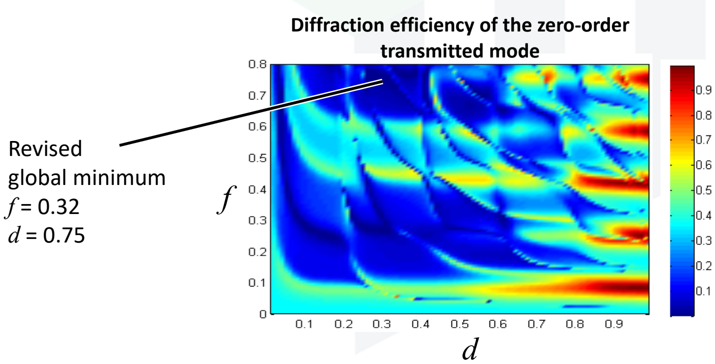


EMPossible

Rethink the Merit Function!!!

In this case, the global best is a very narrow region. This probably isn't feasible when fabrication tolerances are considered. It also assumes the model is perfectly accurate. It is often good practice to include some type of "bandwidth" in your merit function to pick a minimum that is also broad and robust.

$$M\left(f,d\right) = T_0 \cdot W_0 \qquad\qquad W_0 \equiv \text{width of extrema}$$



EMPossible

Multiple Considerations

The merit function can be challenging to formulate when multiple things must be considered.

There is no cookbook way of doing this. It is up to the ingenuity of the engineer to arrive at this.

A common approach is to form a product where teach term is a separate consideration.

$$M = \frac{A_1 \cdot A_2 \cdot A_3 \cdots A_M}{B_1 \cdot B_2 \cdot B_3 \cdots B_N}$$

Parameters we wish to maximize (minimize)

Parameters we wish to minimize (maximize)

EMPossible 13

13

Incorporating Relative Importance

If the considerations are not of equal importance, we need a way to enhance or suppress their impact on the merit function.

We usually cannot just scale them by a constant.

$$M = aA_1 \cdot bA_2 \cdot cA_3 = abc\left(A_1 \cdot A_2 \cdot A_3\right)$$

There are some other effective ways of doing this.

$$M = A_1^{\alpha} \cdot A_2^{\beta} \cdot A_3^{\gamma}$$ exponents

$$M = A_1 \cdot \log\left(A_2\right) \cdot A_3$$ logarithms

$$M = A_1 \cdot \left(1 + A_2\right) \cdot A_3$$ adding constants

$$M = A_1^{\alpha} \cdot \left(1 + A_2\right)^{\beta} \cdot \log\left(1 + A_3\right)$$ hybrids

EMPossible 14

14

Example (1 of 2)

Suppose we wish to optimize the design of an antenna.

We are probably most concerned about its bandwidth B and efficiency E. Based on this, we could define a merit function as

$$M = B \cdot E$$

Are these equally important?

The *Shannon capacity theorem* sets a limit on the bit rate C of data given the bandwidth B of the channel and the signal-to-noise ratio SNR.

$$C = B \log_2 \left(1 + \mathrm{SNR}\right)$$

This shows that bandwidth and efficiency are not equally important.

Example (2 of 2)

We can come up with a better merit function that is inspired by the *Shannon capacity theorem.*

$$M = B \log_2 \left(1 + E\right)$$

Maybe size L is also a consideration. We want data rate as high as possible using an antenna as small as possible. A new merit function that considers size could be

$$M = \frac{B}{L} \log_2 \left(1 + E\right)$$

This merit function, however, would approach infinity as L approaches zero, leading to a false solution.

This problem can be fixed by adding a constant to L.

$$M = \frac{B}{1 + L} \log_2 \left(1 + E\right)$$

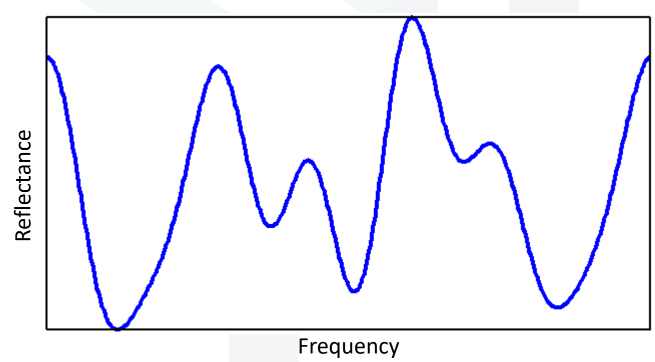
The “Rectangle” Algorithm

Slide 17

17

Goal of the Algorithm

Suppose we wish to design a broadband reflector. We construct an initial design and simulate the reflectance. Using a single quantity, what is the performance of this preliminary design?



EMPossible

18

18

Common Considerations for the Merit Function

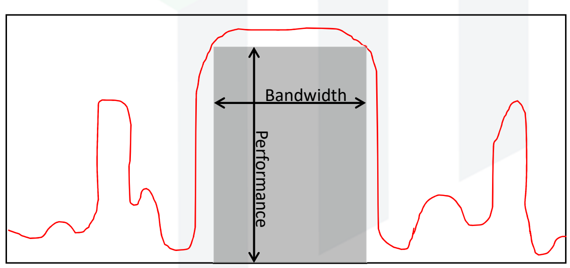
- Bandwidth
- Reflectance across the band
- Ease of fabrication
- Simplest design
- Are you also interested in the transmittance?
- Are you also interested in polarization?
- More…

19

A Very Common Merit Function

In electromagnetics, we are most often interested in maximizing performance (i.e. reflectance here) over some bandwidth.

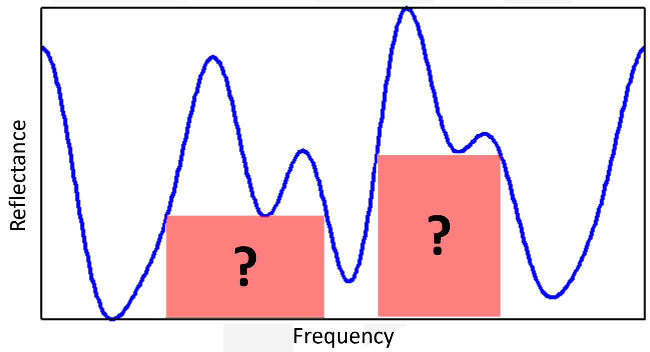
$$\text{MF} = (\text{Bandwidth}) \times (\text{Performance})$$



20

The Rectangle Algorithm

We can determine the merit function by finding the biggest rectangle that fits under a curve. But…which rectangle is biggest?



EMPossible 21

21

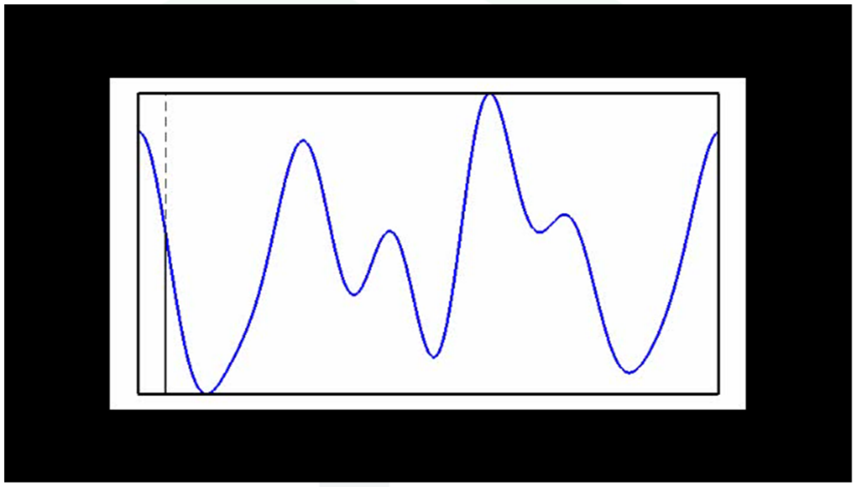
Steps in the Rectangle Algorithm

- Step 0 – Decide on a merit function.
- Step 1 – Calculate the spectra of a device
- Step 2 – Loop over all points in spectra.
  - i. Seek left.
  - ii. Seek right.
  - iii. Calculate merit function.
- Step 3 – Overall merit function is the area of the largest rectangle.

EMPossible 22

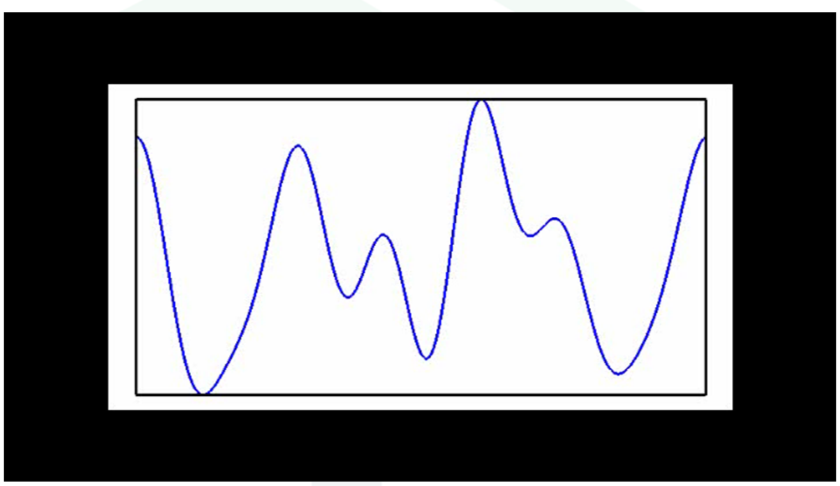
22

Animation of Rectangle Construction



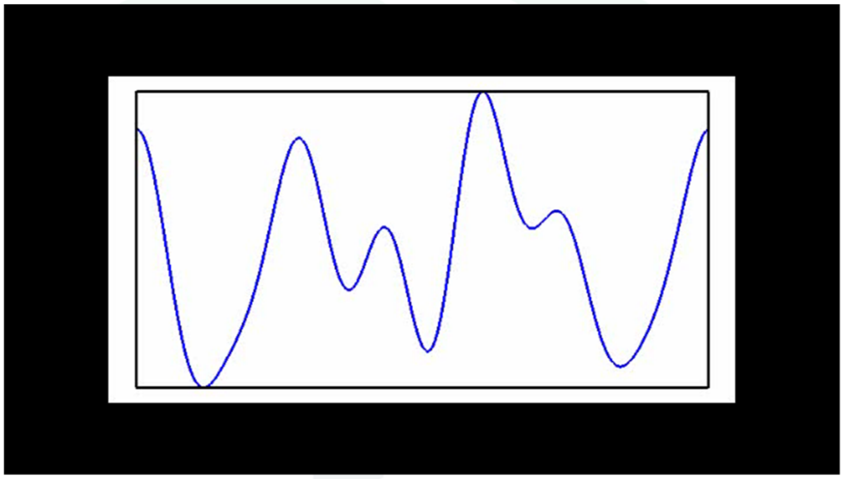
Animation of the Rectangle Algorithm (1 of 3)

$$\text{MF} = w \cdot h$$



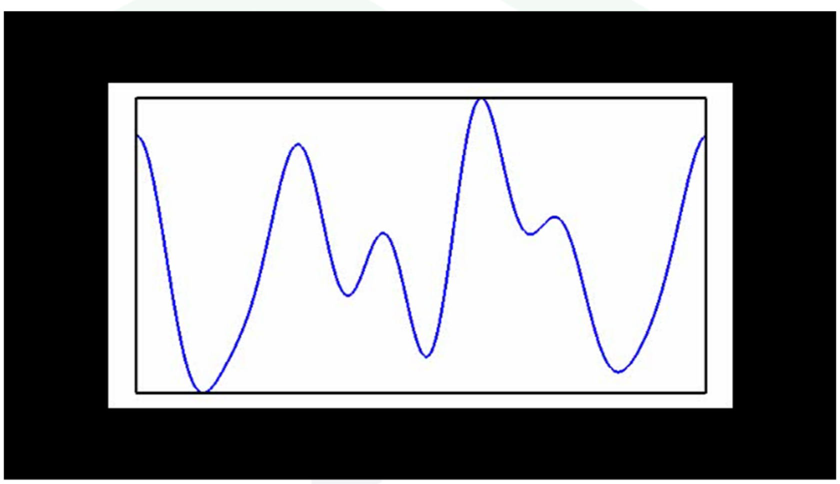
Animation of the Rectangle Algorithm (2 of 3)

$$\text{MF} = \log(1+w)\cdot h^3$$



Animation of the Rectangle Algorithm (3 of 3)

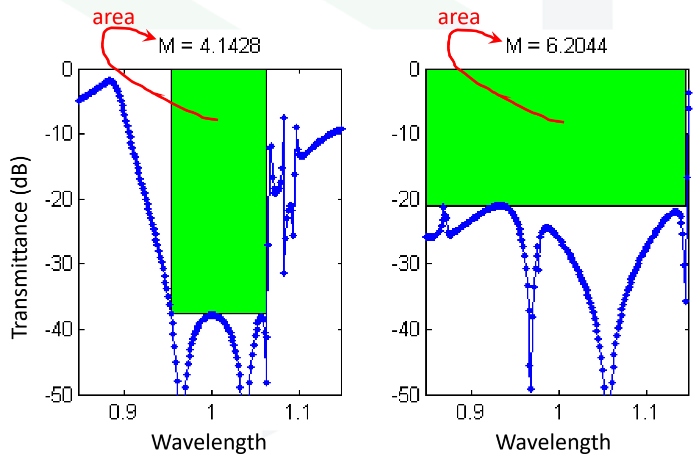
$$\text{MF} = w^3 \cdot \log(1+h)$$



An Example Merit Function

Suppose you want to minimize transmission through a device where your application requires it to be broadband. You need both minimum transmission and wide bandwidth.

$$M \sim |T| \cdot \mathrm{BW}$$



EMPossible

Slide 27

27

Complete Search

Slide 28

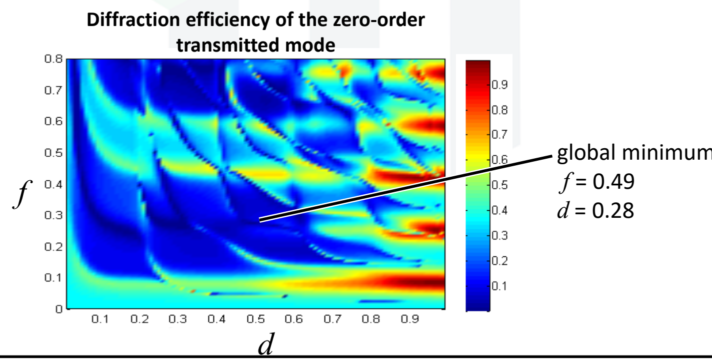
28

Complete Search

The data below represents a complete search where every point in the solution space has been simulated.

This is the only know way to guarantee that the global best solution has been found.

This is very computationally intensive and time consuming. So much so, that typical applications would require centuries of computation time to complete.



Other Direct Methods

- Single Variable Optimization
  - Parabolic interpolation
  - Newton's method
  - Golden-section search
  - more…
- Multi Variable Optimization
  - Powell's method
  - Steepest ascent (or descent) method
  - Newton's method for multiple variables
  - more…

For more on these methods, see Computational Methods in EE
https://empossible.net/academics/emp4301_5301/

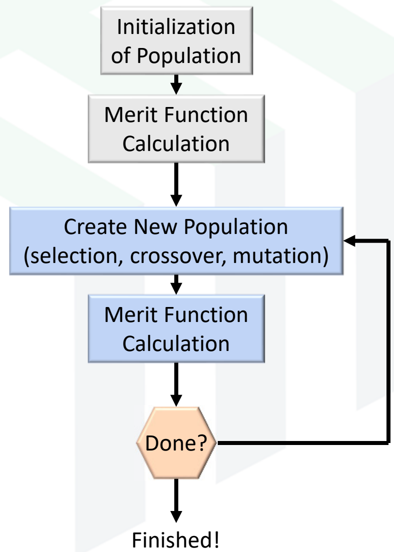
Genetic Algorithms

What are Genetic Algorithms?

Genetic algorithms code designs into “chromosomes.” A population of candidate designs are simulated. Chromosomes are interchanged between the best few designs to produce the next generation of candidate designs. This process continues until convergence.

$$\text{Parent \#1} = [10010011010100110101]$$
$$\text{Parent \#2} = [01011010101100101010]$$
$$\downarrow$$
$$\text{Child \#1} = [10011010010100111010]$$
$$\text{Child \#2} = [10010011011100110101]$$
$$\text{Child \#3} = [01010010101100101010]$$

Block Diagram of Basic GA



EMPossible

Slide 33

33

Notes on Genetic Algorithms

- Genetic algorithms are a special case evolutionary algorithms inspired by natural evolution.
- GAs are the most popular stochastic optimization algorithm.

EMPossible

Slide 34

34

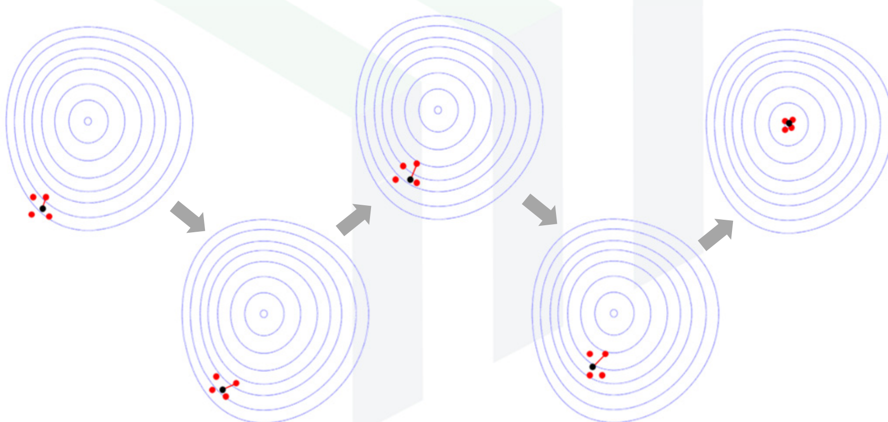
Simulated Annealing

Slide 35

35

What is Simulated Annealing?

Simulated annealing is a method that applies a series of random changes to a candidate design in hopes that a better solution is found. As the method progresses, the magnitude of the random changes is continually decreased.

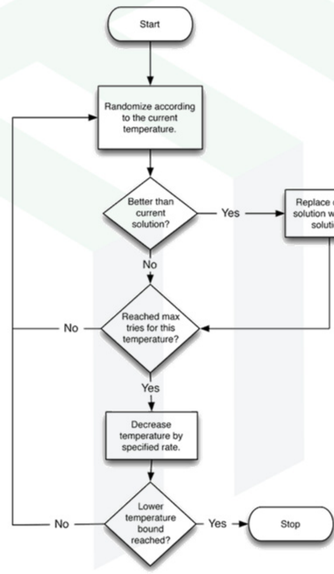


EMPossible

Slide 36

36

Block Diagram of SA



EMPossible

Slide 37

37

Notes on Simulated Annealing

- Name is derived from annealing commonly employed in metallurgy. A series of heating and cooling cycles reduces defects in crystals.
- Usually used to “fine tune” a design.
- Can still be used for global optimization.

EMPossible

Slide 38

38

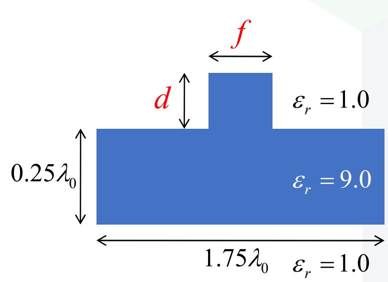
Particle Swarm Optimization

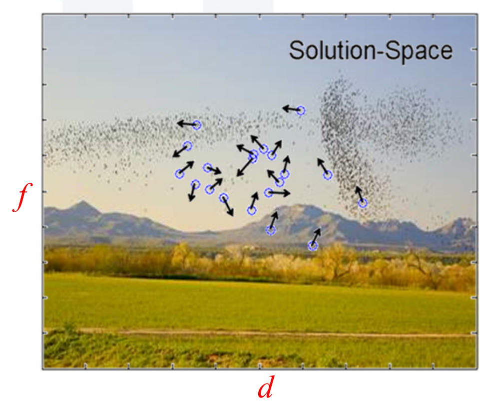
Slide 39

39

What is PSO?

Particle swarm optimization (PSO) allows us to intelligently search the solution-space for a global best solution. In contrast to an exhaustive search, PSO significantly reduces the time to optimize a device. It was originally developed by biologists to be able to mathematically describe the swarming of birds.



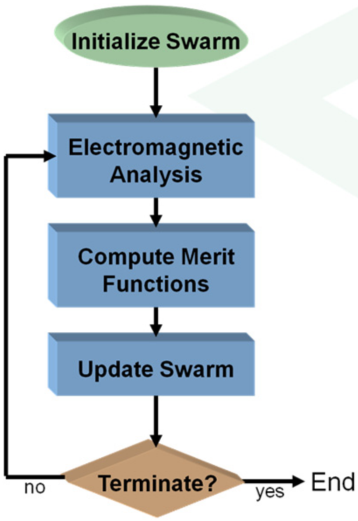


EMPossible

Slide 40

40

How Does it Work?



$$v_i^{k+1} = wv_i^k + cr_1^k\left(p_i^k - x_i^k\right) + sr_2^k\left(p_g^k - x_i^k\right)$$

$$x_i^{k+1} = x_i^k + \chi v_i^{k+1}$$

EMPossible

Slide 41

41

Setting Up the Problem

- Identify Parameters of Design
  - You want to describe as many geometries as possible with as few parameters as possible
    - The parameters describe the design.
- Particle Position
  - Each point in solution space corresponds to a specific design that can be simulated.
  - Use scalability to reduce the number of parameters
- Merit Function (or Fitness Function)
  - This may be the most critical consideration.
  - You must quantify how “good” a design is with a single number.
  - This is particularly difficult when multiple factors must be optimized at the same time.

EMPossible

Slide 42

42

Scaling the Particle Parameters

The particle parameters (i.e. position) are stored in a column vector.

$$\mathbf{x}_k = \begin{bmatrix} x_1^k \\ x_2^k \\ \vdots \\ x_N^k \end{bmatrix}$$

$k \equiv$ PSO iteration number
$N \equiv$ Number of partical parameters

It is best to keep these parameters scaled to fall in the same range…say 0 to 1.

$$0 < x_i^k < 1$$

This makes it easier to move the particles uniformly across the solution space. That is r, c, and s will affect the parameters the same.

EMPossible Slide 43

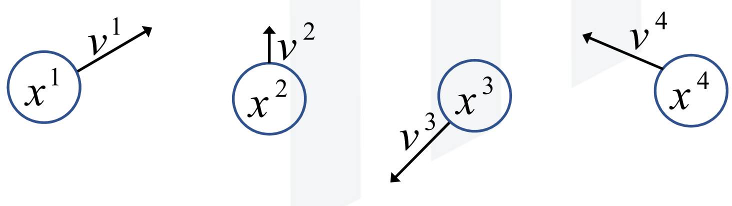
43

PSO Update Algorithm

The basic PSO algorithm is composed of two mathematical steps.

1. Intelligently update the velocity of the particle.
2. Update the position of the particles based on their velocity.

We control the behavior of the particles by how we update their velocity at each iteration.



EMPossible Slide 44

44

PSO Velocity Update

The velocity of each particle is update according to

$$v_{i+1}^{k} = \underbrace{w v_i^k}_{\text{inertia term}} + \underbrace{c r_1^k \left(p_i^k - x_i^k \right)}_{\text{cognitive term}} + \underbrace{s r_2^k \left(p_g^k - x_i^k \right)}_{\text{social term}}$$

The **inertia term** controls how quickly a particle will change direction.

The **cognitive term** controls the tendency of a particle to move toward the best solution observed by that particle.

The **social term** controls the tendency of a particle to move toward the best solution observed by any of the particles.

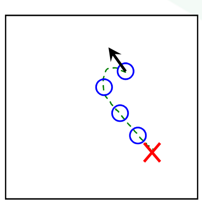
$x_i^k \equiv$ position of i^{th} particle at k^{th} iteration
$v_i^k \equiv$ velocity of i^{th} particle at k^{th} iteration
$p_i \equiv$ best solution observed by i^{th} particle
$p_g \equiv$ best solution observed by any particle
$w \equiv$ inertia coefficient
$c \equiv$ cognitive coefficient
$s \equiv$ social coefficient
$r \equiv$ randomness factor

EMPossible

PSO Inertia Term

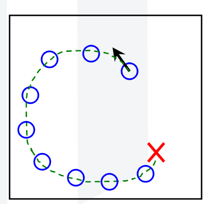
The **inertia term** in the velocity update equation controls how tightly the particles can turn a curve. The higher the inertia constant, the longer it takes the particle to turn, thus scanning a larger portion of the solution space.

$$w v_i^k \equiv \text{ inertia term}$$



Small inertia coefficient w

- Converges faster
- Searches less space



Large inertia coefficient w

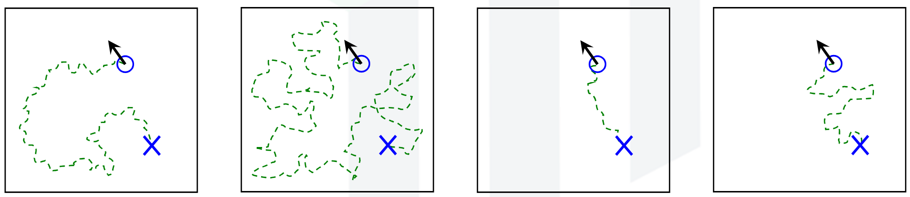
Note: Each particle can have a unique inertia coefficient!

EMPossible

PSO Cognitive Term

The cognitive term controls how “smart” the particle is and how willing it is to go pursue it’s own local best solution. Increasing the cognitive coefficient will make the particle want to back toward the best solution it found.

$$cr_1^k\left(p_i^k - x_i^k\right) \equiv \text{cognitive term}$$



Small c and small r | Small c and large r | Large c and small r | Large c and large r

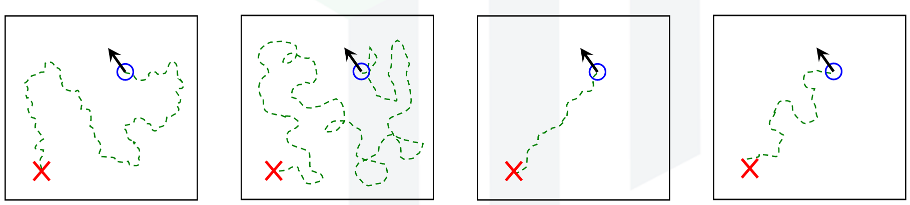
Note: Each particle can have a unique cognitive coefficient!

EMPossible

PSO Social Term

The social term controls the particle’s tendency to move toward the global solution found among all the particles. By increasing the social coefficient, the particle will be more inclined to move toward the global best solution.

$$sr_2^k\left(p_g^k - x_i^k\right) \equiv \text{social term}$$



Small s and small r | Small s and large r | Large s and small r | Large s and large r

Note: Each particle can have a unique social coefficient!

EMPossible

PSO Final Formulae

Putting it all together we arrive at the final form of the formulae. Each particle can be programed individually with its own inertia, cognitive and social factors. You can also combine particles into "Tribes" or "Neighborhoods" to scan certain portions of the solution space.

$$v_i^{k+1} = wv_i^k + cr_1^k\left(p_i^k - x_i^k\right) + sr_2^k\left(p_g^k - x_i^k\right)$$
$$x_i^{k+1} = x_i^k + \chi v_i^{k+1}$$

Note: The χ term in the position update equation is called the constriction term. It is used to tighten up the group up toward the end of optimization. It can be set to zero or neglected.

EMPossible

Slide 49

49

Concluding Remarks

Slide 50

50

Final Remarks

- Almost never a guarantee a global best solution has been found.
- Dr. Rumpf tends to apply the methods this way…
  - Continuous design variables → Particle swarm optimization
  - Discrete design variables → Genetic algorithm
  - Fine tuning a solution → Simulated annealing, gradient descent, Newton's method