Antenna Array Synthesis by Non-uniform Element Positions Using Cross-Entropy Method

Li Bian[1, 2] ,Qun Wu[1,3],Guohui Yang[1]

[1] Department of Microwave Engineering, Harbin Institute of Technology, Harbin, China
[2] College of Electronic and Information Engineering, Heilongjiang Institute of Science and Technology, Harbin, China
[3]State Key Laboratory of Millimeter Wave, Southeast University, Nanjing, China
branran@163.com,qwu@hit.edu.cn

***Abstract*-Non-uniform element positions make antenna array synthesis become highly non-linear optimization problems. A new approach for the synthesis of non-uniform positions linear arrays is presented. The method is based on Cross-Entropy theory. Linear antenna array with 32 isotropic elements, uniformly weighted, symmetrically placed is synthesized by optimizing the element positions using Cross-Entropy Method. Three score functions are selected to synthesis the array. Sidelobe power and beamwidth of best array designs for each score function are compared. And absolute element positions from origin of each score function are presented. It is benefit work utilizing the Cross Entropy method for solving electromagnetic optimization problems.**

I. INTRODUCTION

By altering the spacings between elements, one can achieve greater control over the shape of the radiation pattern sidelobes especially towards array endfire since a design that is well-behaved near the end of the aperture can prevent the appearance of grating lobes. The introduction of non-uniform spacings makes the optimization problem highly non-linear since the element spacings are defined in the argument of the exponential for the array factor. A new approach for the synthesis of non-uniform positions linear arrays is presented. The method is based on Cross-Entropy theory. Previously, the synthesis of array spacings has been solved using GA, Simulated Annealing and Particle Swarm Optimization [1-5].

II. CROSS ENTROPY METHOD

Cross-Entropy (CE) is general stochastic optimization technology based on a fundamental principle of information theory called cross entropy (or Kullback-Leibler).In 1997, CE was first introduced by Reuven Y.Rubinstein [6] as an adaptive importance sampling for estimating probabilities of rare events, and was widely used in complex optimization thereafter.

Generally, the CE method involves an iterative procedure which each iteration can be broken into two steps:

1) Generate a random data sample (trajectories, vectors, etc.) according to a certain mechanism.

2) Update the parameters of the random mechanism based on the data to produce a "better" sample for the next iteration.

The fundamental characteristic of the CE is that it operates based on a parameterized probability density distribution during the optimization procedure, as opposed to similar stochastic techniques such as the Genetic Algorithm (GA) or Simulated Annealing (SA), which operate directly based on the samples in the candidate population.

Let $\boldsymbol{X}=(X_1\ldots X_n)$ be a random vector taking values in some space χ. Let $\{f(\cdot;v)\}$ be a family of probability density functions (pdfs) on χ, with respect to some base measure υ.Here υ is a real-valued parameter (vector). For any measurable function H, Thus

$$E[H(\boldsymbol{X})]=\int_{\chi} H(\boldsymbol{x})f(\boldsymbol{x},\boldsymbol{v})\upsilon(d\boldsymbol{x}) \tag{1}$$

Where, $E[\cdot]$ is mean operation, and υ is either accounting measure or the Lebesgue measure. Assume $\upsilon d(x)=dx$ for simplicity.

Let S be some real-valued function on χ. It is interested in the probability that $S(X)$ is greater than or equal to given real number γ, under $f(\cdot;\mu)$.This probability l can be expressed as

$$l=P_{\boldsymbol{\mu}}(S(\boldsymbol{X})\geq\gamma)=E_{\boldsymbol{\mu}}[I_{\{S(\boldsymbol{X})\geq\gamma\}}] \tag{2}$$

Where, $I_{\{\cdot\}}$ is shown as a set.A straightforward way to estimate l is to use Crude Monte-Carlo simulation: Draw a random sample from $f(\cdot;\mu)$; then an unbiased estimator of l is

$$\hat{l}_{M-C}=\frac{1}{N}\sum_{i=1}^{N} I_{\{S(X_i)\geq\gamma\}} \tag{3}$$

Where, N is sample number.

If l is less than10^{-5},$\{S(X)\geqslant\gamma\}$ is called rare event. In that case a large numbers simulation effort is required in order to estimate l with small Relative Error (RE) or a narrow confidence interval. The Importance Sampling (IS) technology [7] is presented by Rubinstein to solve the questions above.

Let g is another probability density function. When $g(x)=0$, $I_{\{S(\boldsymbol{x})\geq\gamma\}}f(\boldsymbol{x};\boldsymbol{\mu})=0$, g can be used to represent l, Eq.(2) can be simplified below.

$$l=\int I_{\{S(\boldsymbol{x})\geq\gamma\}}\frac{f(\boldsymbol{x};\boldsymbol{\mu})}{g(\boldsymbol{x})}g(\boldsymbol{x})d\boldsymbol{x}=E_g[I_{\{S(\boldsymbol{X})\geq\gamma\}}\frac{f(\boldsymbol{X};\boldsymbol{\mu})}{g(\boldsymbol{X})}] \tag{4}$$

Where, $E_g[\cdot]$ expresses mean operation about g. $g(x)$ is defined as important sampling density. Let $W(\boldsymbol{x};\boldsymbol{\mu})=\frac{f(\boldsymbol{x};\boldsymbol{\mu})}{g(\boldsymbol{x})}$,called Likelihood Ratio(LR)，then Eq.(4) can be simplified below.

$$l=\int I_{\{S(\boldsymbol{x})\geq\gamma\}}\frac{f(\boldsymbol{x};\boldsymbol{\mu})}{g(\boldsymbol{x})}g(\boldsymbol{x})d\boldsymbol{x}=E_g[I_{\{S(\boldsymbol{X})\geq\gamma\}}W(\boldsymbol{X};\boldsymbol{\mu})] \tag{5}$$

From the Eq. (4), the unbiased estimator if l turns to be

$$\hat{l} = \frac{1}{N}\sum_{i=1}^{N} I_{\{S(\boldsymbol{X}_i)\geq\gamma\}} \frac{f(\boldsymbol{X}_i;\boldsymbol{\mu})}{g(\boldsymbol{X}_i)} = \frac{1}{N}\sum_{i=1}^{N} I_{\{S(\boldsymbol{X}_i)\geq\gamma\}} W(\boldsymbol{X};\boldsymbol{\mu}) \quad (6)$$

Where, $\hat{l}$ is called as the importance sampling estimates or likelihood ratio estimates.

It is well known that the best way to estimate l is to use the change of measure with density

$$g^*(\boldsymbol{x}) = \frac{I_{\{S(\boldsymbol{x})\geq\lambda\}} f(\boldsymbol{x},\boldsymbol{\mu})}{l} \quad (7)$$

Put Eq. (7) into Eq. (6), then

$$I_{\{S(\boldsymbol{X}_i)\geq\lambda\}} \frac{f(\boldsymbol{X}_i;\boldsymbol{\mu})}{g^*(\boldsymbol{X}_i)} = l \quad (8)$$

Where, Eq. (8) is established for all i.

Eq. (7) shows $g^*(\boldsymbol{x})$ depends the unknown parameter l, so it is difficult to be defined clearly. Moreover, it is often convenient to choose a g in the family of densities $\{ f(\cdot;\boldsymbol{v}) \}$. Parameter vector $\boldsymbol{v}$ is called reference parameter. The idea now is to choose $\boldsymbol{v}$ to make the distance between the densities g^* and $f(\cdot;\boldsymbol{v})$ minimal. A particular convenient measure of distance between two densities g and h is the Kullback-Leibler distance, which is also termed the cross-entropy between g and h. The Kullback-Leibler distance is defined as:

$$\begin{aligned} D(g,h) &= E_g \ln \frac{g(\boldsymbol{X})}{h(\boldsymbol{X})} \\ &= \int g(\boldsymbol{x}) \ln g(\boldsymbol{x}) d\boldsymbol{x} - \int g(\boldsymbol{x}) \ln h(\boldsymbol{x}) d\boldsymbol{x} \end{aligned} \quad (9)$$

Make g^* and $f(\cdot;\boldsymbol{v})$ take the place of g and h in Eq. (9)

$$D(g,f) = \int g^*(\boldsymbol{x}) \ln g^*(\boldsymbol{x}) d\boldsymbol{x} - \int g^*(\boldsymbol{x}) \ln f(\boldsymbol{x};\boldsymbol{v}) d\boldsymbol{x} \quad (10)$$

If Eq. (10) has the minimum, $\int g^*(\boldsymbol{x}) \ln f(\boldsymbol{x};\boldsymbol{v}) d\boldsymbol{x}$ has a maximum, which is equivalent to solving the maximization problem.

$$\max_{\boldsymbol{v}} D(\boldsymbol{v}) = \max_{\boldsymbol{v}} \int g^*(\boldsymbol{x}) \ln f(\boldsymbol{x};\boldsymbol{v}) d\boldsymbol{x} \quad (11)$$

Put Eq. (7) into Eq. (11) :

$$\max_{\boldsymbol{v}} D(\boldsymbol{v}) = \max_{\boldsymbol{v}} \int \frac{I_{\{S(\boldsymbol{x})\geq\lambda\}} f(\boldsymbol{x},\boldsymbol{\mu})}{l} \ln f(\boldsymbol{x};\boldsymbol{v}) d\boldsymbol{x} \quad (12)$$

Eq. (12) is equivalent to the maximum program at last.

$$\max_{\boldsymbol{v}} D(\boldsymbol{v}) = \max_{\boldsymbol{v}} E_u [I_{\{S(\boldsymbol{X})\geq\gamma\}} \ln f(\boldsymbol{X};\boldsymbol{v})] \quad (13)$$

Set another $f(\cdot;\omega)$. Using again importance sampling for, Eq. (13) can be rewritten as

$$\max_{\boldsymbol{v}} D(\boldsymbol{v}) = \max_{\boldsymbol{v}} E_{\boldsymbol{\omega}} [I_{\{S(\boldsymbol{X})\geq\gamma\}} W(\boldsymbol{X};\boldsymbol{\mu},\boldsymbol{\omega}) \ln f(\boldsymbol{X};\boldsymbol{v})] \quad (14)$$

For any reference parameter $\boldsymbol{\omega}$, where

$$W(\boldsymbol{x};\boldsymbol{u},\boldsymbol{\omega}) = \frac{f(\boldsymbol{x};\boldsymbol{\mu})}{f(\boldsymbol{x};\boldsymbol{\omega})}$$

is the likelihood ratio, at $\boldsymbol{x}$, between $f(\cdot;\boldsymbol{\mu})$ and $f(\cdot;\boldsymbol{\omega})$. The optimal solution of(14) can be written as

$$\boldsymbol{v}^* = \arg\max_{\boldsymbol{v}} E_{\boldsymbol{\omega}} [I_{\{S(\boldsymbol{X})\geq\gamma\}} W(\boldsymbol{X};\boldsymbol{\mu},\boldsymbol{\omega}) \ln f(\boldsymbol{X};\boldsymbol{v})] \quad (15)$$

In this condition,

$$\max_{\boldsymbol{v}} \hat{D}(\boldsymbol{v}) = \max_{\boldsymbol{v}} \frac{1}{N}\sum_{i=1}^{N} I_{\{S(\boldsymbol{X}_i)\geq\gamma\}} W(\boldsymbol{X}_i;\boldsymbol{\mu},\boldsymbol{\omega}) \ln f(\boldsymbol{X}_i;\boldsymbol{v}) \quad (16)$$

Where $X_1 \ldots X_N$ is a random sample from $f(\cdot;\boldsymbol{\omega})$, N refers to the sample frequency.

From Eq. (16), the Maximum value is about the variable $\boldsymbol{v}$, and the solution of (16) may be readily obtained by solving the following system of equations:

$$\frac{1}{N}\sum_{i=1}^{N} I_{\{S(\boldsymbol{X}_i)\geq\gamma\}} W(\boldsymbol{X}_i;\boldsymbol{\mu},\boldsymbol{\omega}) \nabla \ln f(\boldsymbol{X}_i;\boldsymbol{v}) = \boldsymbol{0} \quad (17)$$

Then, the importance sampling estimator or Likelihood ratio estimator of Eq. (6) is

$$\hat{l} = \frac{1}{N}\sum_{i=1}^{N} I_{\{S(\boldsymbol{X}_i)\geq\gamma\}} W(\boldsymbol{X};\boldsymbol{\mu},\boldsymbol{\omega}) \quad (18)$$

Eq. (15) shows that the probability of target event $\{S(\boldsymbol{X}) \geq \gamma\}$ can not be too small, and it requires $l \geq 10^{-5}$ as usual. It is hard for Eq. (5) to calculate when $l < 10^{-5}$, $I_{\{S(\boldsymbol{X}_i)\geq\gamma\}}$ is empty set because of medium-sized samples. This condition also limits the calculation of the left side of Eq. (17).

Multi-level algorithm [8] can overcome the difficulty above. The main idea is to construct the sequence of reference parameters $\{\boldsymbol{v}_t, t > 0\}$ and level sequence $\{\gamma_t, t > 1\}$, then to make $\boldsymbol{v}_t$ and γ_t update iteration at the same time.

Sum up, program flow graph and iteration steps to CE method for rare event are presented, as shown in Figure.1.

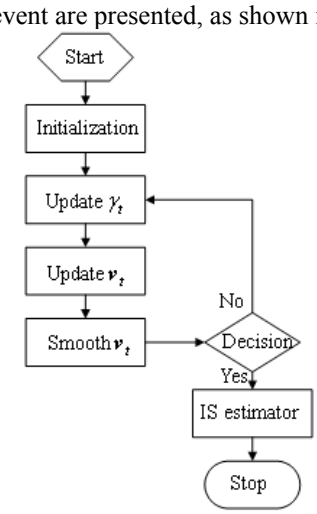


Figure 1. Program flow graph and iteration steps

Sum up, program flow graph and iteration steps to CE method for rare event are presented [9], as shown in Figure.1.

Specific steps are:

1) Initialize parameters:

Set initial $\hat{\boldsymbol{v}}_0$, sample select coefficient ρ, set population size K, smoothing constant α and set iteration counter t=1.

2) Adaptive update γ_t :

Given $\hat{\boldsymbol{v}}_{t-1}$, let γ_t be the $(1-\rho)$-quintile of $S(\boldsymbol{X})$ satisfying $P_{\hat{\boldsymbol{v}}_{t-1}}(S(\boldsymbol{X}) \geq \gamma_t) \geq \rho$ and $P_{\hat{\boldsymbol{v}}_{t-1}}(S(\boldsymbol{X}) \leq \gamma_t) \geq 1-\rho$,Where X~ $f(\cdot;\hat{\boldsymbol{v}}_{t-1})$ and $f(\cdot;\hat{\boldsymbol{v}}_{t-1})$ choose normal distribution.

According to the random sample $X_1,\ldots,X_N$ which are produced by $f(\cdot;\hat{\boldsymbol{v}}_{t-1})$,calculate $S(\boldsymbol{X}_i)$ for i (=1,…N) and arrange $S(\boldsymbol{X}_i)$ from small to large: $S_{(1)} \leq \ldots \leq S_{(N)}$,then the updated value of γ_t is $\hat{\gamma}_t = S_{(\lceil (1-\rho)N \rceil)}$,where $\lceil \cdot \rceil$ means rounding.

3) Adaptive update $\boldsymbol{v}_t$

Given $\hat{\boldsymbol{v}}_{t-1}$ and $\hat{\boldsymbol{v}}_{t-1}$, determine $\boldsymbol{v}_t$ by solving the CE programs by setting the updating value of $\boldsymbol{\omega} = \hat{\boldsymbol{v}}_{t-1}$

$$\hat{\boldsymbol{v}}_t = \arg\max_{\boldsymbol{v}} \frac{1}{N}\sum_{i=1}^{N} I_{\{S(\boldsymbol{X}_i)\geq\hat{\gamma}_t\}} W(\boldsymbol{X}_i;\boldsymbol{\mu},\hat{\boldsymbol{v}}_{t-1}) \ln f(\boldsymbol{X}_i;\boldsymbol{v}) \quad (19)$$

4) Optimization of $\boldsymbol{v}_t$

To accelerate the speed of convergence rate of CE method, the smoothing coefficient α is used to optimize $\boldsymbol{v}_t$,where $\boldsymbol{v}_t = \alpha\boldsymbol{v}_t + (1-\alpha)\boldsymbol{v}_{t-1}$ ($0<\alpha<1$).

5） When $t = t+1$， repeat step 2） to 4） until it meets the standard of the loop stops.

III. PROBLEM SETUP

The thinning process of uniform linear arrays typically produces sidelobes with a relatively constant height. By altering the spacings between elements, one can achieve greater control over the shape of the radiation pattern sidelobes especially towards array endfire since a design that is well-behaved near the end of the aperture can prevent the appearance of grating lobes. The introduction of non-uniform spacings makes the optimization problem highly non-linear since the element spacings are defined in the argument of the exponential for the array factor.

Linear antenna array with 32 isotropic elements, uniformly weighted, symmetrically placed is synthesized by optimizing the element positions using Cross-Entropy Method. The three score functions are defined as follows:

1. $Score = \max(20\log_{10}|FF(u)|)$, $\lambda/Nd \leq u \leq 1$

2. $Score = 10\log_{10}\left(\frac{2}{\Delta u}\int_u |FF(u)|^2 du\right)$, $\lambda/Nd \leq u \leq 1$

3. $Score \frac{1}{N_s}\sum_{i=1}^{N_s}(20\log_{10}|FF(u_i)| - Mask_{dB}(u_i))^2$

Where, $Mask_{dB} = -8.6957u - 21.3042(dB)$ .A plot of the *Mask* score function is shown Figure 2.

Score function #1 is the score used previously in the section on array thinning. This score function tends to produce sidelobe patterns with a constant height. A summation is included in the definition when minimizing sidelobe power over different intervals of u space. Score function #2 integrates the sidelobe power over interval [u_{li}, u_{ui}] of u-space.

This score function tends to maintain the sidelobe envelope of the array along with the locations of sidelobe maxima and nulls. During minimization these parameters may be spread slightly, but for the most part remain unchanged. Score function #3 is a square error score between the sidelobe powers of the far-field pattern for a given interval of u space and a Mask function representing a desired pattern shape. Only the samples of $|FF(u)|^2$ that violate the Mask function contribute to the total score. For this reason, this score function is often called the 'don't exceed' square error criterion. The total number of samples which violate Mask is Ns and the score is normalized by the total number of samples used for sampling $FF(u)$. This score function is quite good for shaping sidelobe patterns since it does not penalize null positions during the optimization process because most nulls are typically below sidelobe peaks and 'don't exceed' the mask. An optimal solution of score function #3 is defined as any solution which does not violate the mask, resulting in a score of zero.

Mask tries to enforce a peak sidelobe power of –22 dB at λ/Nd, the first null position of the original uniform linear array. The sidelobe envelope then tapers linearly towards a final value of –30 dB at u = 1. Inside the mainlobe region Mask is set to zero so that any point defining the mainlobe of the optimized pattern does not count towards the score. The score is then computed by counting the samples of the far-field pattern (expressed in dB) which are greater than Mask. The linear tapering allows an increase in the sidelobes between u = [0.3, 1] to allow for a reduction in the first sidelobe.

In order to perform a successful optimization procedure, Mask must not be too greedy. Certain compromises must be made in the pattern to achieve desired objectives.

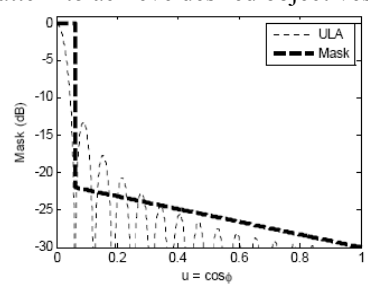


Figure 2. Mask function optimization plot

TABLE I CE DESIGN PARAMETERS

Symbol	Quantity	Value
α	Smoothing Parameter	0.7
ρ	Sample Selection Parameter	0.1
K	Population Size	100
x_n	Limits on Element Position	[0,(N/2-0.5)d]
$\mu^{(0)}$	Initial Mean of Element Positions	(n-0.5d)/λ
$\sigma^{2(0)}$	Initial Variance of Element Positions	100

IV. SIMULATION RESULTS

The CE design parameters are given in Table I. The continuous form of CE was used to perform the optimization. The initial values for the mean and variance of the Gaussian distributions were set to (n–0.5) d/λ, for n = 1, 2,…, N/2 and one-hundred respectively for each element in the array. The equation (n–0.5)d/λ for the mean of element n initially biases

the solution to be about the original position of element n in the uniform array. An initial variance of one-hundred allowed the initial solutions to explore the feasible region of the element positions for the original array defined by [0, (N/2–0.5)·d] for x > 0 and also was large enough to prevent the smoothing coefficient of 0.7 from freezing the algorithm in a suboptimal solution. So, there exists a tradeoff between speed of convergence and guaranteeing convergence to an arbitrarily small neighborhood of the optimal solution. The maximum value of the element positions was set to (N/2–0.5)·d = 7.75λ, which is the aperture length of the original uniform array for x > 0. The smoothing coefficient of 0.7 was chosen as a tradeoff with ρK for convergence and accuracy based on the results of the previous section. The choice of both the population size, K and the sample selection parameter, ρ are critical to the success of the algorithm. The separation between the total number of population samples considered and the number of best estimates used to update the parameters of the Gaussian distributions must be large enough to ensure there is an increase in the probability that the best samples from the current iteration appear in subsequent iterations. The optimization procedure was stopped when the maximum variance of all distributions was less than 2.2204x10-16, which equals the floating point precision of the computer as defined by MATLAB's eps command.

Sidelobe power and beamwidth of best array designs for each score function are compared. And absolute element positions from origin of each score function are presented in the Table II and Table III.

TABLE II BEST ARRAY DESIGNS FOR EACH SCORE

Score		Maximum SLL(dB)		FNBW(deg)	
NO	Best Score	$u=\lambda/Nd$	u>FNBW/2	Optimized	Original
#1	-19.69	-19.69	-19.69	8.056	7.16
#2	-26.92	-14.67	20.98	8.867	7.16
#3	0.366	-12.975	-22	9.33	7.16

TABLE III ABSOLUTE ELEMENT POSITIONS

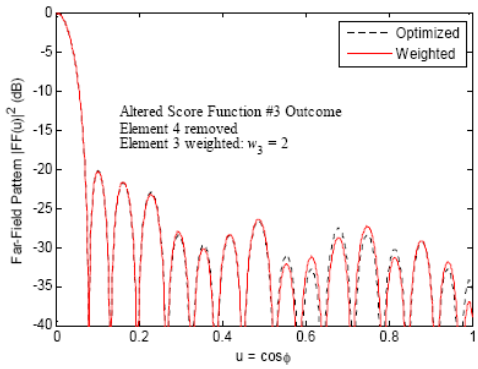
Element No.	±1	±2	±3	±4	±5	±6	±7	±8	±9	±10	±11	±12	±13	±14	±15	±16
x_n/λ	0.23	0.64	0.99	1.45	1.87	2.30	2.71	3.15	3.70	4.04	4.67	5.13	6.00	6.65	7.26	7.75
	0.21	0.59	0.96	1.42	1.67	2.24	2.44	3.03	3.29	3.85	4.23	4.79	5.31	6.04	6.87	7.66
	0.34	0.47	1.01	1.27	1.62	2.13	2.27	2.81	3.11	3.65	4.11	4.65	5.13	5.81	6.59	7.34

It is apparent from the results in Table Ⅱ, that score function #1 yields the best sidelobe power with respect to the original uniform array ($u=\lambda/Nd$). However, both score function #2 and #3 outperform score function #1 in the sidelobe region of the optimized array. Despite terminating the optimization procedure after the maximum variance of all elements exceeded eps, all three scores basically converged to their final values within 150 iterations.

One interesting note that is not touched upon in array geometry optimization is the fact that unless a constraint is placed on the spacing between absolute element positions then it is possible for elements to occupy the same location. This result can be mechanically impractical not to mention electrically.

These two elements basically occupy the same location. This result is mathematically transparent to the array factor and this result would occur whether one is optimizing absolute element positions or element spacing. What this result actually represents in reality is a single element with twice the amplitude. For example, element 4 can be removed and an amplitude weight w_3 = 2 can be applied to element 3. The resulting array pattern is shown in Figure 3 and is nearly identical to the original.

It is benefit work utilizing the Cross Entropy method for solving electromagnetic optimization problems.



ACKNOWLEDGEMENTS

This work is support by National Natural Science Foundation of China (Grant No. 60971064), Youth Science Fund of Heilongjiang Province of China (Grant No.QC2010023), and a grant from Youth Outstanding Ability Program in Heilongjiang Institute of Science and Technology, Project Program of State Key Laboratory of Millimeter Wave (Grant No. K201006), Science and Technology on Electromagnetic Environment Laboratory, and the Special funds projects for technological and innovative talent of Harbin City (2010RFXXG010).

REFERENCES

[1] A.Uenveren and A.Acan, "Multi-Objective Optimization with Cross Entropy Method: Stochastic Learning with Clustered Pareto Fronts", CEC 2007,IEEE Congress on.Sept.2007,pp.3065-3071

[2] Ali Abdulhadi Noaman, "Concentric Circular Array Antenna Null Steering Synthesis by Using Modified Hybrid Ant Colony System Algorithm", IJACT, Vol.2, No.2, pp.144~157, 2010.

[3] Y.zhang,et.al, "Cross Entropy Optimization of Multiple-Input Multiple-Output Capacity by Transmit Antenna Selection",IET Microwaves, Antenna & Propagation,Vol.1,No.6,pp.1131~1136,2007.

[4] R.Y.Rubinsten,and D.P.Kroese, "The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning", New York:Springer,2004.

[5] Z.Botev and D.P.Kroese, "Global Likelihood Optimization via the Cross-Entropy Method with an Application to Mixture Models", IEEE Proceedings of the 2004 Winter Simulation Conference, Vol.1, 2004.

[6] Jianfeng Ye,etc, "Phase control based on immune algorithm generates zero pattern", The Modern Radar, vol.29,No.12,pp.65~68,2007.

[7] Usman Qayyum, "Efficiency Enhancement of Neural Network with Phase Only Correlation", JDCTA, Vol. 2, No. 2, pp.42 ~ 47, 2008.

[8] R.Y.Rubinstein, "Optimization of Computer Simulation Models with Rare Events", European Journal of Operational Research, vol.99, pp.89-112, 1997.

[9] Ho, S.L., Shiyou Yang, "The Cross-Entropy Method and Its Application to Inverse Problems" IEEE Trans. Magn., Vo: 46 , no. 8,pp 3401–3404,2010