10.2528/PIER07082403

Progress In Electromagnetics Research, PIER 77, 425–491, 2007

CENTRAL FORCE OPTIMIZATION: A NEW METAHEURISTIC WITH APPLICATIONS IN APPLIED ELECTROMAGNETICS

R. A. Formato

Registered Patent Attorney & Consulting Engineer
P.O. Box 1714, Harwich, MA 02645, USA

Abstract—Central Force Optimization (CFO) is a new deterministic multi-dimensional search metaheuristic based on the metaphor of gravitational kinematics. It models "probes" that "fly" through the decision space by analogy to masses moving under the influence of gravity. Equations are developed for the probes' positions and accelerations using the analogy of particle motion in a gravitational field. In the physical universe, objects traveling through three-dimensional space become trapped in close orbits around highly gravitating masses, which is analogous to locating the maximum value of an objective function. In the CFO metaphor, "mass" is a user-defined function of the value of the objective function to be maximized. CFO is readily implemented in a compact computer program, and sample pseudocode is presented. As tests of CFO's effectiveness, an equalizer is designed for the well-known Fano load, and a 32-element linear array is synthesized. CFO results are compared to several other optimization methods.

1. INTRODUCTION

Central Force Optimization (CFO) is a new *metaheuristic* for an optimization evolutionary algorithm (EA). CFO searches a multi-dimensional decision space for the extrema of an objective function to be maximized. To the author's knowledge, CFO is a new optimization metaphor that has not been described previously. It is based on an analogy to classical particle kinematics in a gravitational field. CFO is inherently deterministic, unlike other widely used metaheuristics (for example, Particle Swarm Optimization, "PSO" [1, 2] and Ant Colony Optimization, "ACO" [3, 4]).

In the same way that ACO was first introduced, CFO is introduced here as a metaheuristic, not as a fully developed algorithm resting on a firm theoretical foundation. Indeed, as pointed out in [3] "...ACO has been driven by experimental work, with the aim of showing that the *ideas* [emphasis added] underlying this technique can lead to successful algorithms. After this initial phase, researchers tried to deepen their understanding of the technique by building theoretical foundations." Even though the first "Ant System" algorithm was published in 1996, the first proof of convergence did not appear for four years, and at that was limited to a "rather peculiar ACO algorithm" [3].

By definition "A *metaheuristic* is a set of algorithmic *concepts* [emphasis added] that can be used to define heuristic methods applicable to a wide set of different problems. In other words, a metaheuristic is a general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications." [3] Typically a metaheuristic is proffered without any mathematical proof, and often it is inspired by a metaphor drawn from biology (ACO and PSO being prime examples). CFO is suggested precisely in this manner, but instead of biology, it is based on an analogy drawn from the motion of masses in a gravitational field.

This note describes the CFO concept in detail and illustrates its effectiveness with two examples: matching network design and linear array synthesis. These specific examples were chosen because they permit a direct comparison between CFO and several other widely used optimization methods. Initial work on CFO suggests that it should be a useful optimization tool for applications in applied electromagnetics and other fields. To be sure, there remain unresolved issues concerning how CFO algorithms should be implemented, in particular choosing run parameters, and admittedly there is no detailed theoretical foundation at this time. It is the author's hope that this paper will inspire further work on CFO that will address these issues.

This paper is organized as follows. Section 2 introduces the CFO metaphor using concepts drawn from gravitational kinematics. Section 3 provides a precise statement of the optimization problem being addressed. Section 4 explains the theory of Central Force Optimization and develops its two fundamental equations. Section 5 presents pseudocode for the specific CFO implementation used for the runs reported in this paper. Sections 6 and 7 apply CFO to two real-world problems in applied electromagnetics, viz., matching the canonical Fano load and synthesizing an optimized 32-element linear array. Section 8 tests CFO against a variety of 30- and 2-dimensional functions and one 4-dimensional function, all drawn from recognized "benchmark suites" (complex functions with analytically

known maxima). Section 9 describes an interesting CFO property, its ability to cluster probes in regions of maxima, which may be useful as a decision space "topology mapper." Section 10 discusses what appears to be CFOs tendency to become occasionally trapped in local maxima and a possible mitigation technique. Section 11 is the conclusion, followed by Appendix A which shows the benchmark functions.

2. THE CFO METAPHOR

Newton's universal law of gravitation describes the gravitational attraction of two masses m_1 and m_2. The magnitude of the force of attraction is [5]

$$F = \gamma \frac{m_1 m_2}{r^2}. \tag{1}$$

The distance between m_1 and m_2 is r. The coefficient γ is the "gravitational constant." Because the force acts along the line connecting the masses' centers, gravity is a *central force*. Hence, "Central Force Optimization" is used to describe the new metaheuristic.

Each mass is accelerated towards the other. The vector acceleration experienced by mass m_1 due to mass m_2 is given by

$$\vec{a}_1 = -\gamma \frac{m_2 \hat{r}}{r^2}, \tag{2}$$

where $\hat{r}$ is a unit vector. It points towards mass m_1 from mass m_2 along the line joining the two masses.

The position vector of a particle subject to *constant* acceleration during the interval t to $t + \Delta t$ is given by [6]

$$\vec{R}(t + \Delta t) = \vec{R}_0 + \vec{V}_0 \Delta t + \frac{1}{2} \vec{a} \Delta t^2. \tag{3}$$

The mass's position at time $t + \Delta t$ is $\vec{R}(t + \Delta t)$, where $\vec{R}_0$ and $\vec{V}_0$ are the position and velocity vectors at time t, respectively. In a standard three-dimensional Cartesian coordinate system, the position vector is $\vec{R} = x\hat{i} + y\hat{j} + z\hat{k}$, where $\hat{i}, \hat{j}, \hat{k}$ are the unit vectors along the x, y and z axes, respectively. Because CFO searches spaces of arbitrary dimensionality, these equations will be generalized to a decision space with N_d dimensions.

A simple example illustrates the physical basis of CFO's gravitational metaphor. Consider the following hypothetical problem: Determine the location of the solar system's largest planet with no prior knowledge of the solar system's topology. Because the largest planet

presumably produces the greatest gravitational field, one approach would be "fly" a group of probe satellites through the solar system while radioing back the location of each satellite at discrete time steps.

After a long enough time, most of the probes, whose trajectories are governed by equations (2) and (3), likely will cluster in orbits around the planet with the largest gravitational field. Equations (2) and (3) collectively are referred to as the "equations of motion." CFO generalizes the equations of motion in three-dimensional physical space to search a multidimensional decision space for the extrema of an objective function to be maximized. CFO's analog of the planetary mass is a user-defined function of the value of the objective function at each probed point.

3. PROBLEM STATEMENT

CFO is a metaheuristic for solving the following problem: A decision space is defined by $x_i^{\min} \leq x_i \leq x_i^{\max}$, $i = 1, \ldots, N_d$ where the x_i are decision variables. Determine the locations in that space of the global maxima of an objective function $f(x_1, x_2, \ldots, x_{N_d})$ whose value is referred to as the "fitness." $f(x_1, x_2, \ldots, x_{N_d})$'s topology in the decision space is unknown. It may be continuous or discontinuous, highly multimodal or "smooth," and it may be subject to a set of constraints Ω among the decision variables.

4. THE CFO ALGORITHM: THEORY

The 3-D decision space shown in Fig. 1 will be used to explain the CFO algorithm. CFO "flies" a set of "probes" through the space over a set of discrete "time" steps. This nomenclature has been chosen solely to reflect the analogy to gravitational kinematics. At every time step each probe's position is described by the three spatial coordinates computed from the equations of motion. At each point in a probe's trajectory through the decision space there is a corresponding value of the objective function, a "fitness" value.

The position vector $\vec{R}_j^p$ specifies the location of each probe at each time step. The indices p and j are the probe number and time step number, respectively. In a decision space with N_d dimensions, the position vector is $\vec{R}_j^p = \sum_{k=1}^{N_d} x_k^{p,j} \hat{e}_k$. The $x_k^{p,j}$ are probe p's coordinates at time step j, and $\hat{e}_k$ is the unit vector along the x_k axis.

As time progresses, the probes fly through the decision space along trajectories governed by the equations of motion under the influence

of the "gravitational" forces created by a user-defined function of the fitness at each of the other probes' locations. For example, probe p moves from position $\vec{R}^p_{j-1}$ at time step $j-1$ to position $\vec{R}^p_j$ at time step j, with "time" interval between steps $j-1$ and j being Δt.

The fitness at time step $j-1$ at probe p's location is given by $M^p_{j-1} = f(x^{p,j-1}_1, x^{p,j-1}_2, \ldots, x^{p,j-1}_{N_d})$. Each of the other probes also has associated with it a fitness $M^k_{j-1},\ k = 1, \ldots, p-1, p+1, \ldots, N_p$, where N_p is the *total* number of probes.

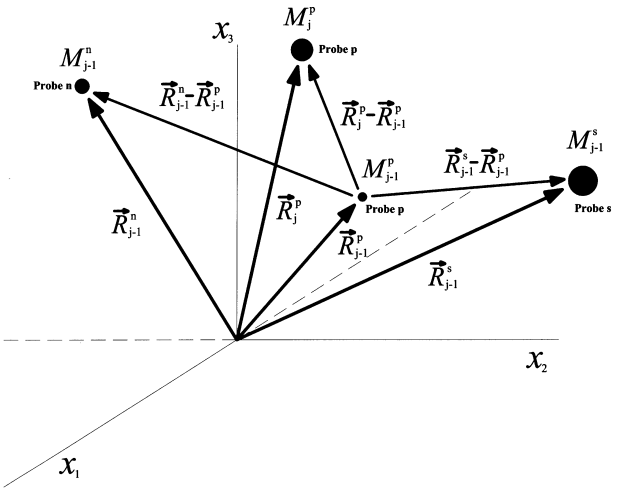

Figure 1. Typical 3-D CFO decision space.

In Fig. 1, the fitness at each probe's location is represented by the size of the blackened circle at the tip of the position vector, the metaphorical correspondence being to the size of the "planet" in the solar system analogy. Larger circles correspond to greater fitness values (bigger "planets" with greater gravitational attraction). Thus, the fitnesses in the figure, ranked from largest to smallest, are located at position vectors $\vec{R}^s_{j-1}, \vec{R}^p_j, \vec{R}^n_{j-1}$, and $\vec{R}^p_{j-1}$, respectively, the ranking being reflected in the relative size of the circles at the tip of each vector.

Probe p moves from location $\vec{R}^p_{j-1}$ to $\vec{R}^p_j$ along a trajectory that is determined by its initial position and by the *total* acceleration produced by the "masses" created by the fitnesses (or some function defined on them) at each of the other probes' locations. In the CFO implementation used in this note, the "acceleration" experienced by

probe p due to probe n is

$$\frac{G \cdot U\left(M_{j-1}^{n} - M_{j-1}^{p}\right) \cdot \left(M_{j-1}^{n} - M_{j-1}^{p}\right)^{\alpha} \cdot \left(\vec{R}_{j-1}^{n} - \vec{R}_{j-1}^{p}\right)}{\left|\vec{R}_{j-1}^{n} - \vec{R}_{j-1}^{p}\right|^{\beta}}$$

Similarly, probe s produces an acceleration of probe p given by

$$\frac{G \cdot U\left(M_{j-1}^{s} - M_{j-1}^{p}\right) \cdot \left(M_{j-1}^{s} - M_{j-1}^{p}\right)^{\alpha} \cdot \left(\vec{R}_{j-1}^{s} - \vec{R}_{j-1}^{p}\right)}{\left|\vec{R}_{j-1}^{s} - \vec{R}_{j-1}^{p}\right|^{\beta}}$$

G is CFO's "gravitational constant" ($G > 0$), and it corresponds to γ in eq. (1). Note that the minus sign in eq. (2) is taken into account in the order in which the differences in the acceleration expressions are taken. Terms in the numerator containing the objective function fitnesses, for example, $(M_{j-1}^{s} - M_{j-1}^{p})^{\alpha}$, correspond to the "mass" in eq. (2). An important departure from eq. (2) is the unit step function $U(\cdot)$, which is explained below. Following standard notation, the vertical bars denote vector magnitude, $|\vec{A}| = \left(\sum_{i=1}^{N_d} a_i^2\right)^{\frac{1}{2}}$, where a_i are the scalar components of vector $\vec{A}$.

The acceleration expression in "CFO space" is quite different from its physical space counterpart eq. (2). Specifically, there are no physical parameters corresponding to the exponents $\alpha > 0$ and $\beta > 0$, and there is no unit step $U(\cdot)$ in eq. (2). In physical space α and β would take on values of 1 and 3, respectively [note that the numerator does not contain a unit vector like eq. (2)].

In CFO space the algorithm designer is free to assign a completely different variation of gravitational acceleration with mass and distance than the one that occurs in the physical universe. This flexibility is included in the free parameters α and β. CFO test runs reveal that the algorithm's convergence is sensitive to the exponent values, and that some values of these exponents are better than others.

CFO's "gravity" and real gravity differ in two other very important ways. CFO "mass" is a user-defined function of the fitness values. In the implementation described in this note, "mass" is the difference of objective function fitnesses, $M_{j-1}^{s} - M_{j-1}^{p}$. The algorithm designer is free to choose other functions as well. As an example, one possibility might be some ratio of fitnesses or their differences. This notion is reminiscent of the "reduced mass" concept in gravitational kinematics, but it is not considered here.

The reason the difference of fitnesses is used, instead of the fitness values themselves, is to avoid excessive gravitational "pull" by other very close probes. Probes that are located nearby in the decision space are likely to have similar fitness values, which may lead to an excessive gravitational force on the subject probe. The fitness difference intuitively seems to be a better measure of how much gravitational influence there should be by the probe with a greater fitness on the probe with a smaller one.

The second difference is the unit step $U(z) = \left\{ \begin{array}{ll} 1, & z \geq 0 \\ 0, & \text{otherwise} \end{array} \right\}$. Because the CFO algorithm is based on a metaphor, CFO space can be a strange place in which physically unrealizable objects exist. Real mass must be positive, but not so mass in CFO space. Assuming the difference of fitnesses definition of mass described above, the mass can be positive or negative depending on which fitness is greater. The unit step function is included to avoid the possibility of "negative" mass. It forces CFO to create only positive masses that are consequently attractive in nature. If negative masses were allowed, the corresponding accelerations would be repulsive instead of attractive. The effect of a repulsive gravitational force is to fly probes away from large fitness values instead of toward them.

The expressions above represent only the accelerations experienced by probe p due to probes n and s. Taking into account the accelerations produced by each of the other probes on probe p, the *total* acceleration experienced by p as it "flies" from position $\vec{R}^p_{j-1}$ to $\vec{R}^p_j$ is given by summing over all other probes:

$$\vec{a}^p_{j-1} = G \sum_{\substack{k=1 \\ k \neq p}}^{N_p} U\left(M^k_{j-1} - M^p_{j-1}\right) \cdot \left(M^k_{j-1} - M^p_{j-1}\right)^{\alpha} \frac{\left(\vec{R}^k_{j-1} - \vec{R}^p_{j-1}\right)}{\left|\vec{R}^k_{j-1} - \vec{R}^p_{j-1}\right|^{\beta}} \tag{4}$$

The new position vector for probe p at time step j therefore becomes

$$\vec{R}^p_j = \vec{R}^p_{j-1} + \vec{V}^p_{j-1}\Delta t + \frac{1}{2}\vec{a}^p_{j-1}\Delta t^2, \quad j \geq 1. \tag{5}$$

$\vec{V}^p_{j-1}$ is probe p's "velocity" at the end of time step $j-1$: $\vec{V}^p_{j-1} = \frac{\vec{R}^p_{j-1} - \vec{R}^p_{j-2}}{\Delta t}$, $j \geq 2$. In these equations, both the velocity term and the time increment Δt have been retained primarily as a formalism that preserves the analogy to gravitational kinematics. Neither is required (but Δt obviously cannot be zero).

For simplicity, $\vec{V}^p_{j-1}$ and Δt were arbitrarily set equal to zero and unity, respectively, for the CFO runs reported here. A constant value of

Δt is best absorbed into the gravitational constant G. Varying Δt has the effect of changing the interval at which the probes "report" their positions. Whether or not doing so improves CFO's convergence has not been investigated, and consequently this remains an open question.

5. THE CFO ALGORITHM: IMPLEMENTATION

The entire CFO algorithm comprises equations (4) and (5). CFO is simple, and easily implemented in a compact computer program. The basic steps are: (1) compute initial probe positions, the corresponding fitnesses, and assign initial accelerations; (2) successively compute each probe's new position based on previously computed accelerations; (3) verify that each probe is located inside the decision space, making corrections as required; (4) update the fitness at each new probe position; (5) compute accelerations for the next time step based on the new positions; and (6) loop over all time steps.

As the CFO algorithm progresses, it is possible that some probes will "fly" to points outside the defined decision space. In order to avoid searching for maxima in unallowable regions, the probe should be returned to the decision space if this happens.

There are many possible approaches to returning errant probes. Among the simplest would be returning the probe to a specific point, say, its starting point or its last position. Some very tentative testing, however, suggests that this method does not work well. Forcing errant probes back to previously occupied positions appears to interfere too much with CFO's ability to control the probes' trajectories toward maxima.

Therefore a very simple deterministic repositioning scheme was used for the CFO runs described in this paper. Any probe that "flew" out of the decision space was returned to the midpoint between its starting position and the minimum or maximum value of the coordinate lying outside the allowable range.

Another possible relocation scheme is to randomly reposition errant probes. This also is a simple approach because it can utilize the compiler's built-in random number generator, which presumably returns essentially uncorrelated floating-point numbers. Adding some measure of randomness to CFO in this way, or possibly in how the initial probe distribution is generated, is entirely discretionary because the underlying CFO equations are deterministic. Thus, unlike ACO or PSO, CFO does not *require* randomness in any of its calculations.

All of the calculations discussed in this paper are entirely deterministic. CFO's deterministic nature is a major distinction setting it apart from other optimization metaheuristics. Most non-

analytical optimization algorithms are inherently stochastic. PSO and ACO are representative of this type of algorithm. Indeed, PSO and ACO require some degree of randomness in the calculations at every iteration in searching for solutions. In general, every ACO or PSO run starting with the same set of run parameters generates an entirely different solution. Some solutions are very good, but others may be quite poor. Characterizing how well a stochastic algorithm works thus requires a statistical analysis of its behavior, as discussed in [3, 12]. This requirement is completely eliminated in CFO because every CFO run returns exactly the same answer as long as the run parameters are unchanged.

5.1. CFO Pseudocode

The CFO algorithm used to optimize the Fano load equalizer and to synthesize the linear array comprises the following steps:

(1) Create Data Structures

(A) Two 3-dimensional arrays $R(p,i,j)$ and $A(p,i,j)$ are created for the probe position and acceleration vectors, respectively, where the indices $1 \le p \le N_p,\ 1 \le i \le N_d$, respectively, are the probe number and the coordinate (dimension) number in the decision space. The index $0 \le j \le N_t$ is the time step number, with $j = 0$ corresponding to the initial spatial distribution of probes and their accelerations. The velocity term in eq. (5) is set to zero, $\vec{V}_j^p = 0\ \forall\ p, j$, and consequently is not included in this CFO implementation.
(B) A 2-dimensional array $M(p,j)$ is created for the fitness values. Its elements are the fitness values at each probe's location at each time step, $M(p,j) = f(R(p,i,j))$, where the N_d-dimensional objective function to be maximized is $f(x_1, \ldots, x_i, \ldots, x_{N_d})$ with $x_i = R(p,i,j)$.

(2) Initialize Run

(A)(i) 3-D Fano Load Matching Network: Fill the position vector array with a uniform distribution of probes on each coordinate axis at time step 0 [see §6(i) for 2-D model initialization]:

$$\text{For } i = 1 \text{ to } N_d,\ n = 1 \text{ to } \frac{N_p}{N_d}:\ p = n + \frac{(i-1)N_p}{N_d},$$

$$R(p,i,0) = x_i^{\min} + \frac{(n-1)\left(x_i^{\max} - x_i^{\min}\right)}{\dfrac{N_p}{N_d} - 1}.$$

The number of probes per dimension, $\frac{N_p}{N_d}$, is specified by the user.

(A)(ii) 32-element Linear Array: Fill the position vector array with the following distribution of probes slightly off the decision space principal diagonal at time step 0:

For $p = 1$ to $N_p,\ i = 1$ to N_d :

$$R(p,i,0) = x_i^{\min} + \frac{(n-1)\left(x_i^{\max} - x_i^{\min}\right)\left[N_d(p-1)+i-1\right]}{N_p N_d - 1}.$$

(B) Set the initial acceleration to zero:

$$A(p,i,0) = 0, \quad 1 \le p \le N_p, \quad 1 \le i \le N_d$$

(C) Compute initial fitness:

$$M(p,0) = f(R(p,i,0)), \quad 1 \le p \le N_p, \quad 1 \le i \le N_d$$

(3) Loop on Time Step, j: Start with $j = 1$.
(A) Compute New Probe Positions

(a) For $p = 1$ to $N_p,\ i = 1$ to N_d :

$$R(p,i,j) = R(p,i,j-1) + \frac{1}{2}A(p,i,j-1)\Delta t^2,\ \Delta t^2 = 1$$

(b) Retrieve errant probes, if any:

$$\begin{aligned}
&\text{If} \quad R(p,i,j) < x_i^{\min} \text{ then} \\
&\qquad R(p,i,j) = x_i^{\min} + \frac{1}{2}\left(R(p,i,j-1) - x_i^{\min}\right) \\
&\text{If} \quad R(p,i,j) > x_i^{\max} \text{ then} \\
&\qquad R(p,i,j) = x_i^{\max} - \frac{1}{2}\left(x_i^{\max} - R(p,i,j-1)\right)
\end{aligned}$$

(B) Update Fitness Matrix for This Time Step

$$\text{For } p = 1 \text{ to } N_p: \ M(p,j) = f(R(p,i,j))$$

(C) Compute Accelerations for Next Time Step

For $p = 1$ to $N_p,\ i = 1$ to N_d :

$$\begin{aligned}
A(p,i,j) = G \sum_{\substack{k=1 \\ k \ne p}}^{N_p} & U(M(k,j) - M(p,j))(M(k,j) - M(p,j))^{\alpha} \\
& \times \frac{R(k,i,j) - R(p,i,j)}{\left|\vec{R}_j^k - \vec{R}_j^p\right|^{\beta}}
\end{aligned}$$

where

$$\left|\vec{R}_j^k - \vec{R}_j^p\right| = \sqrt{\sum_{m=1}^{N_d} \left(R(k,m,j) - R(p,m,j)\right)^2}$$

(D) Increment $j \Rightarrow j+1$, and repeat from Step (3)(A) until $j = N_t$ or some other stopping criterion has been met.

Note that for the 2-component CFO algorithm described below, Step 2(A)(i) was modified to compute an initial probe distribution on a 2-D grid, instead of the uniformly spaced set of probes on each of the coordinate axes.

This particular CFO implementation was written for the Power Basic/Windows 8.0 (2005) compiler [7]. Copies of the source code and executable files are available as described in Section 8.

6. FANO LOAD EQUALIZER

The first CFO example is a standard network problem: Optimally match the "Fano load" using the equalizer circuit shown in Fig. 2. The canonical Fano load comprises an inductance $L_{fano} = 2.3$ henries in series with the parallel combination of capacitor $C_{fano} = 1.2$ farads and resistor $R_{fano} = 1\,\Omega$ [8]. The equalizer comprises parallel capacitors C_1 and C_3 connected by series inductor L_2. Power is delivered to the load from a generator whose internal impedance is $R_g = 2.205\,\Omega$ (purely resistive).

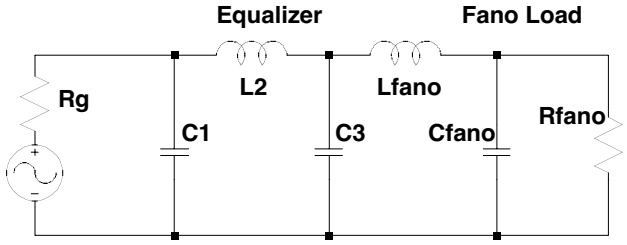

Figure 2. Fano load and equalizer topology.

The equalizer's component values are to be determined by an optimization algorithm. The objective of the optimization problem is to match the source to the load in the radian frequency range $0 \le \omega \le 1\,\text{rad/sec}$ by transferring the maximum possible power to the load at all frequencies. This requirement translates to a "minimax" criterion, viz., minimize $\delta = \max\{1 - T(\omega)\}$, where $T(\omega)$ is the "transducer power gain" (TPG, defined as the fraction of the maximum available power

delivered to the load) [9]. Because CFO is a maximization algorithm, not a minimization algorithm, the equivalent requirement for CFO is to maximize $T(\omega)$ in the interval $0 \leq \omega \leq 1\,\text{rad/sec}$.

Two different CFO optimizers were applied to this problem: the first a 2-D algorithm that optimized L_2 and C_3 with $C_1 = 0.386$ farad as determined in [9]; the second a 3-D algorithm that optimized all three equalizer components, C_1, L_2 and C_3. In the general CFO formulation, these cases correspond to $N_d = 2$ with $x_1 = L_2,\ x_2 = C_3$, and to $N_d = 3$ with $x_1 = C_1,\ x_2 = L_2,\ x_3 = C_3$, respectively.

The reasons for performing the 2-D optimization are that the objective function can be plotted as a 3-D surface, and the CFO probe locations can be plotted in the L_2-C_3 plane. The probe position plots show how the probes converge on the maximum, a visualization that is helpful in illustrating how CFO works.

The function to be maximized is the minimum value of $T(\omega)$ computed at 21 uniformly spaced frequencies in the interval $0 \leq \omega \leq 1\,\text{rad/sec}$. Following the formulation in [9], TPG may be expressed as $T(\omega) = 1 - |\Gamma_{in}|^2$, where $\Gamma_{in} = \frac{Z_{in} - R_g}{Z_{in} + R_g}$. Γ_{in} is the reflection coefficient, and Z_{in} is the input impedance seen by the generator at the equalizer input terminals.

$T(\omega)$ for the 2-D case is plotted in Fig. 3 for $0.1 \leq C_3 \leq 10$ and $0.1 \leq L_2 \leq 10$ (note that the size of the decision space was chosen arbitrarily). Figs. 3(a) and (b) provide perspective views intended to show the general topology of the maxima, not their precise locations. The TPG exhibits two peaks, one larger than the other, with the smaller peak occurring near the minimum values of L_2 and C_3. The plan view in Fig. 3(c) shows the actual locations of the maxima in the C_3-L_2 plane. The global maximum occurs in the vicinity of $L_2 \approx 3$ and $C_3 \approx 1$ as indicated by the brighter coloration in that region.

(i) 2-D Fano Optimizer

The 2-D optimization run was made with the following set of CFO parameters:

$$G = 15, \quad \alpha = 2, \quad \beta = 2, \quad a_{init} = 0$$

There is no particular reason why these specific values were chosen, except that they seem to work reasonably well for the purpose of illustrating CFO's effectiveness in optimizing the Fano load equalizer. Indeed, exactly how CFO's parameters should be chosen is an important open question.

The initial probe distribution was a uniform grid of 25 probes ($N_p = 25$) as shown in Fig. 4(a). The algorithm was run for 50 time steps ($N_t = 50$).

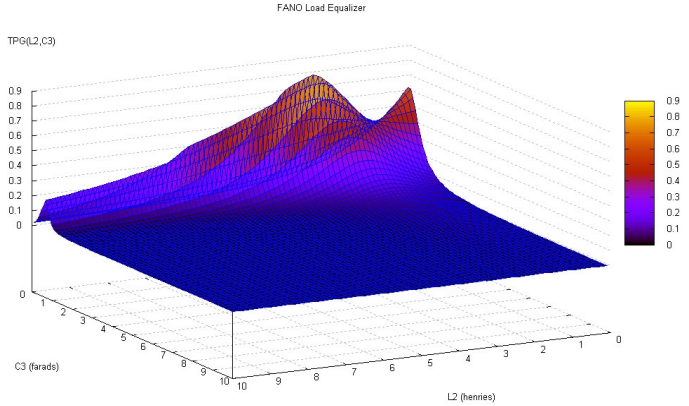

Figure 3a. $T(\omega)$ vs C_3 and L_2 perspective view.

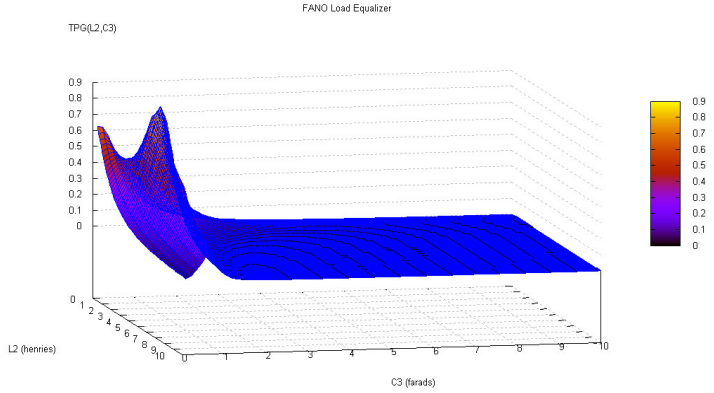

Figure 3b. $T(\omega)$ vs C_3 and L_2 perspective view.

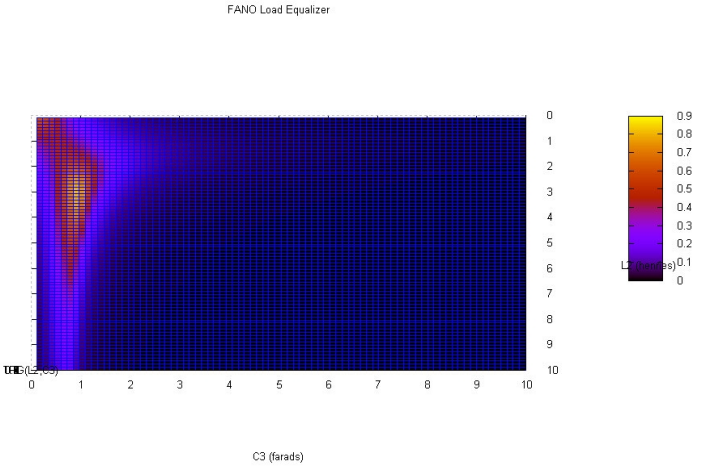

Figure 3c. $T(\omega)$ vs C_3 and L_2 plan view.

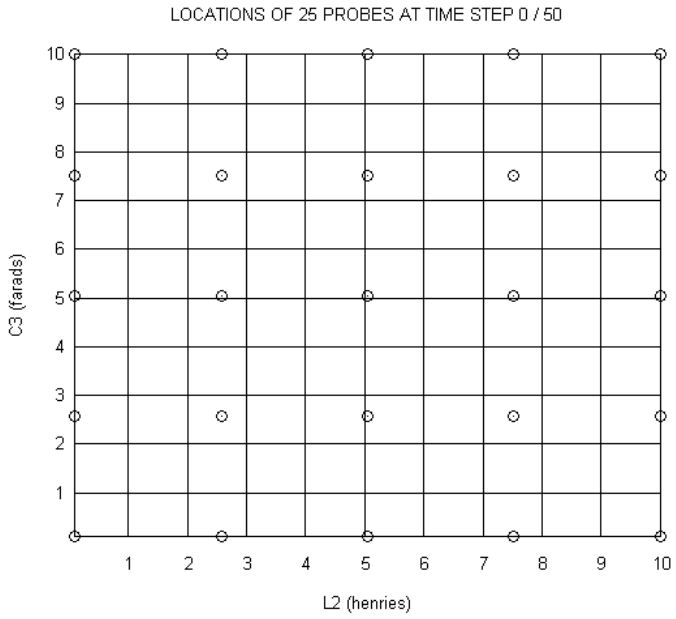

Figure 4a. 2-D initial probes (uniform grid).

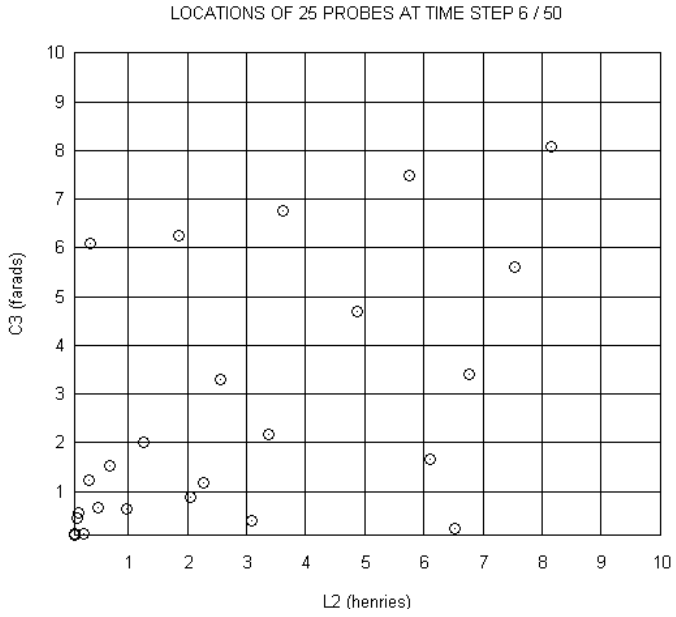

Figure 4b. 2-D probe positions at Step 6.

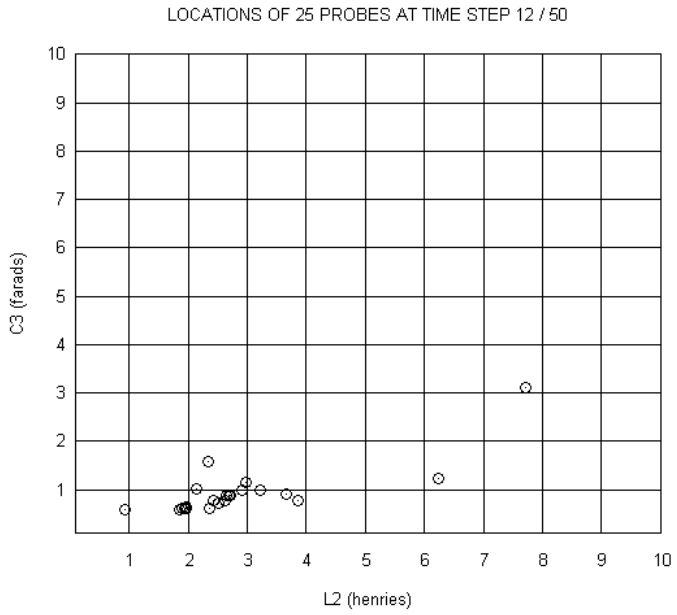

Figure 4c. 2-D probe positions at Step 12.

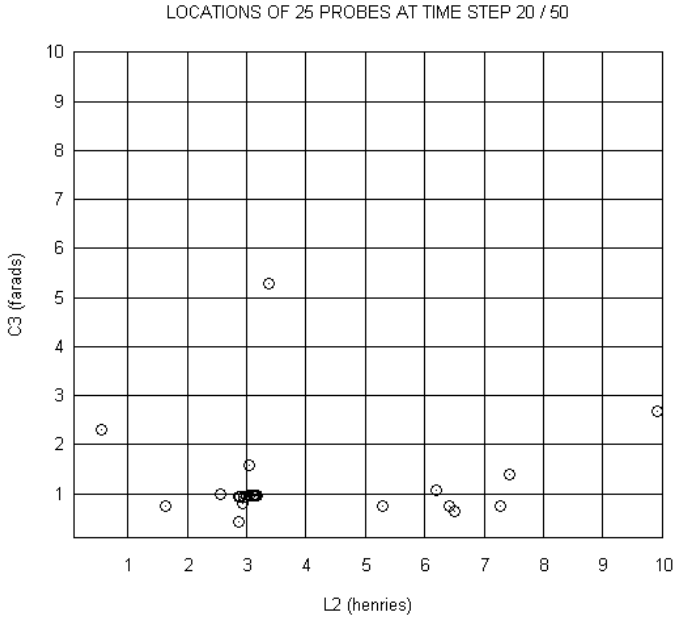

Figure 4d. 2-D probe positions at Step 20.

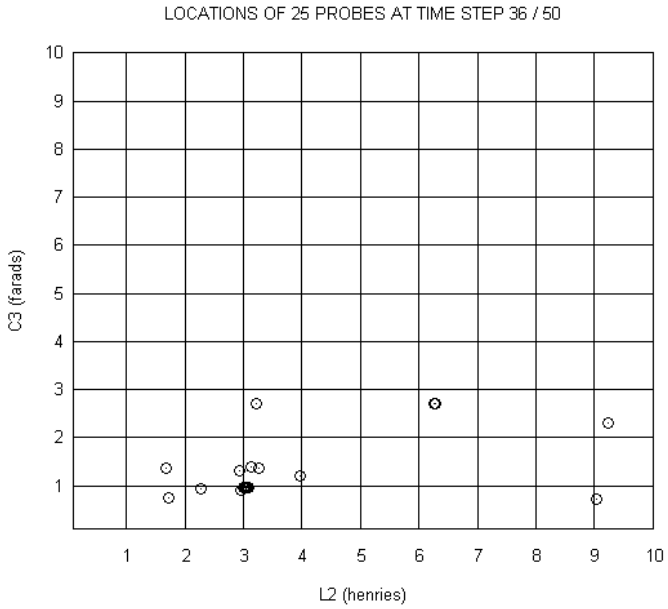

Figure 4e. 2-D probe positions at Step 36.

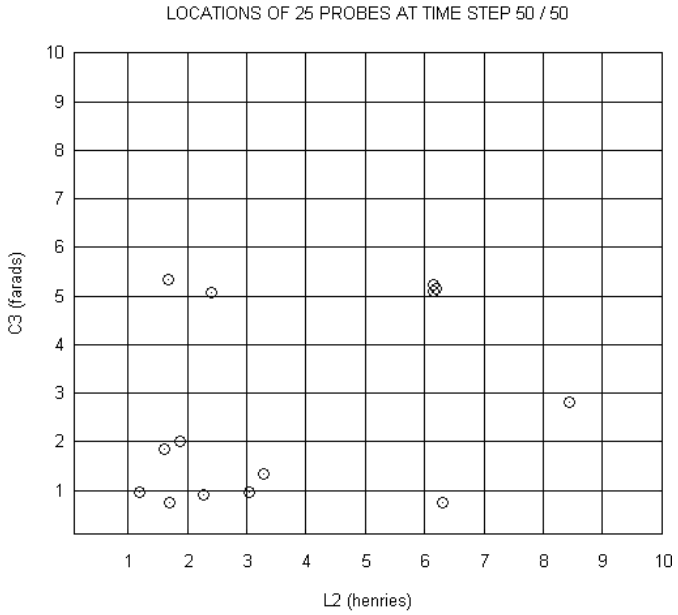

Figure 4f. 2-D probe positions at Step 50.

Figures 4(b) through 4(f) show the probes' positions in the decision space at various time steps. At step 6 the probes begin to cluster around the smaller peak near the minimum values of L_2 and C_3. At step 12 the probes are moving away from the smaller peak toward the global maximum. Convergence is evident at step 36. At the end of the run, many probes have converged on the maximum and cannot be separated visually. The maximum fitness at step 50 is 0.853 at $L_2 = 3.041$ henry and $C_3 = 0.961$ farad.

Fig. 5 shows how the fitness evolves as the run progresses. It rises very rapidly between time steps 6 and 10, and then more slowly through step 20. By that time, as is evident from Fig. 4(d), most probes have clustered near the global maximum, so that the fitness increases much more slowly after step 20.

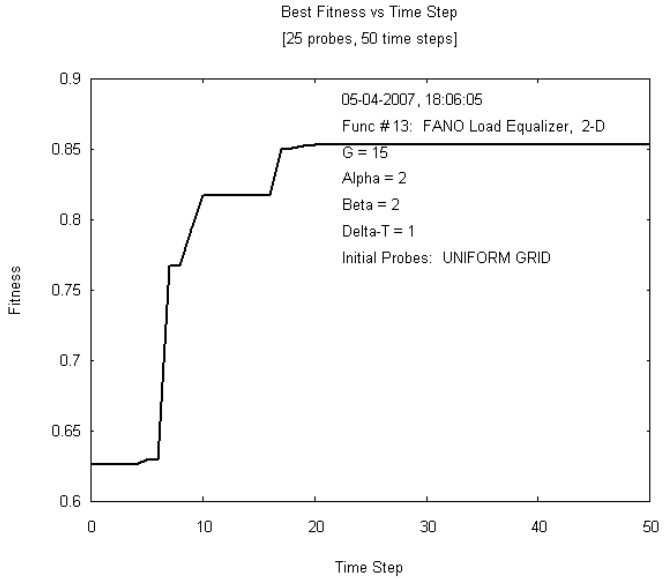

Figure 5. Fitness vs time step.

Fig. 6 plots the average distance between the probe with the best fitness value and all the other probes as a function of the time step. The average probe "distance" is normalized to the "size" of the decision space as measured by the length of its diagonal

$$Diag = \sqrt{\sum_{i=1}^{N_d} \left(x_i^{\max} - x_i^{\min}\right)^2}.$$

The curve shows a clear tendency for the probes to move toward

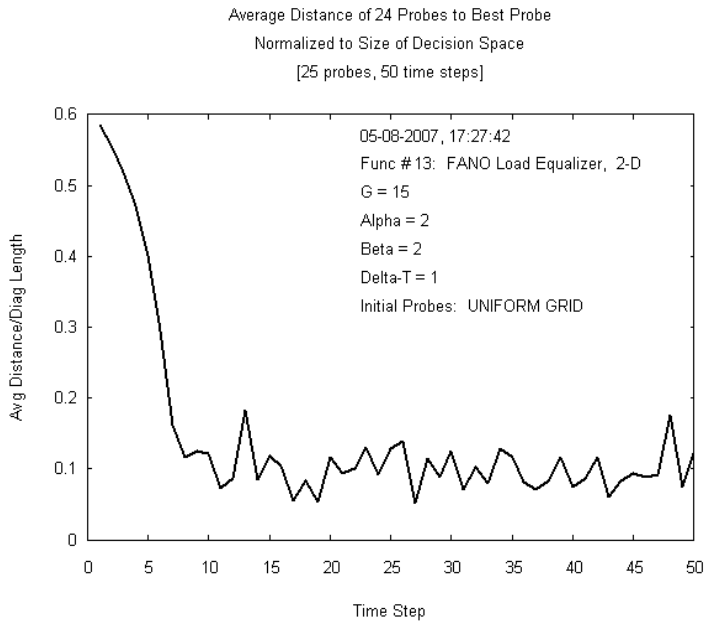

Figure 6. Normalized 2-D average probe distance.

the global maximum under the gravitational influence of the objective function's maxima.

(ii) 3-D Fano Optimizer

The 3-D optimizer used the same CFO parameters as the 2-D algorithm. It was run for 40 time steps ($N_t = 40$) using a total of 210 probes ($N_p = 210$). Unlike the 2-D case, the initial probe distribution was a uniformly spaced group of 70 probes along each of the three decision space coordinate axes as described above in section 5.1(2)(A)(i). Run time was approximately 3 minutes on a dual-boot MacBook Pro laptop (2 GHz Intel T2500 CPU with 1 GB RAM running Mac "Bootcamp" and Windows XP Pro/SP2). A total of 8,400 function calls evaluating $T(\omega)$ were made.

The optimized values computed by CFO for equalizer components C_1, L_2, C_3, respectively, are 0.460 farad, 2.988 henry, and 1.006 farad. The resulting maximum fitness value is 0.852.

Fig. 7 plots the best fitness as a function of time step. It increases very quickly through step 6, and thereafter much more slowly. However, even at time step 39 a very slight increase in fitness is seen to occur. Such slight increases might be expected as the number of

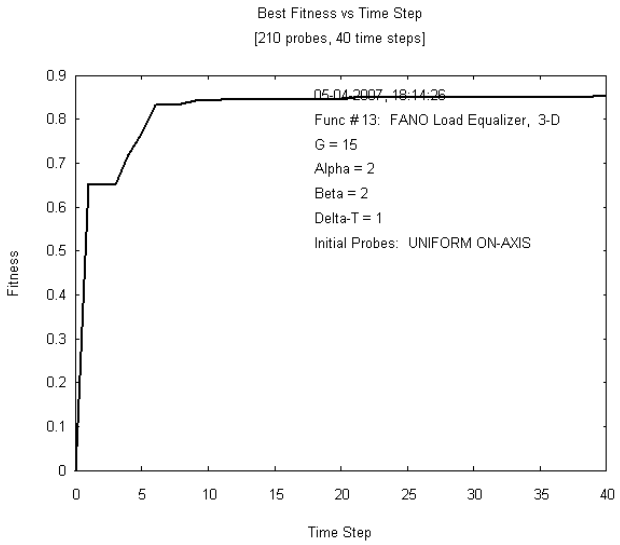

Figure 7. 3-D CFO fitness vs time step.

time steps increases, but there is a trade-off between the quality of the result and the length of a run. The balance to be struck is a matter of engineering judgment.

As in the 2-D case, probes cluster near the global maximum with increasing time step as seen in the plot in Fig. 8. By step 16 the average distance to the probe with the largest fitness has begun to stabilize near a value of 0.15.

The GA-Simplex method described in [9] was validated by comparing it to two other recognized optimization schemes: Carlin's RFT (Real Frequency Technique) [10], and RSE (Recursive Stochastic Optimization) [11]. The optimized equalizer component values using these three methods are summarized in Table 1 in [9]. In order to compare CFO, its data are reproduced in Table 1 below along with the CFO results.

It is apparent that all four methods converge on very similar solutions. The equalizer response for each set of component values is plotted in Fig. 9. These results clearly show that CFO solves the Fano load equalizer problem at least as effectively as other recognized optimization methods.

Not only does CFO compare favorably with the other techniques in terms of its result, it also compares favorably in terms of computational efficiency and rate of convergence. CFO required 8,400 function calls to reach convergence. The GA-Simplex method described in [9] employed

Table 1. Optimized equalizer component values.

-	RFT	RSE	GA-Simplex	CFO
Min T(ω)	0.849	0.855	0.852	0.852
C_1 (farads)	0.352	0.409	0.386	0.460
L_2 (henries)	2.909	3.054	2.976	2.988
C_3 (farads)	0.922	0.974	0.951	1.006

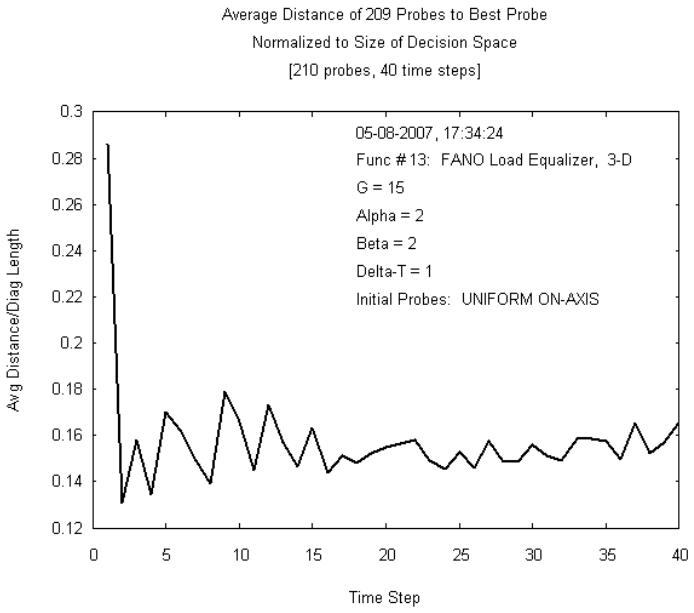

Figure 8. Normalized average 3-D probe distance.

a genetic algorithm with a population of 100 individuals evolved for 100 generations, followed by a Nelder-Mead Simplex algorithm that ran for 100 iterations using 170 function calls. Thus the GA-Simplex method required a total of 10,170 function evaluations. The RFT method [10] is a quasi-analytical/graphical method that involves segmenting the equalizer's resistance vs. frequency curve using piece-wise linear segments. Consequently, the method cannot be described in terms of computational efficiency measured by the number of required function calls. It is included here because it appears in [9] and permits a comparison of the final results. In contrast, the RSE method [11] can

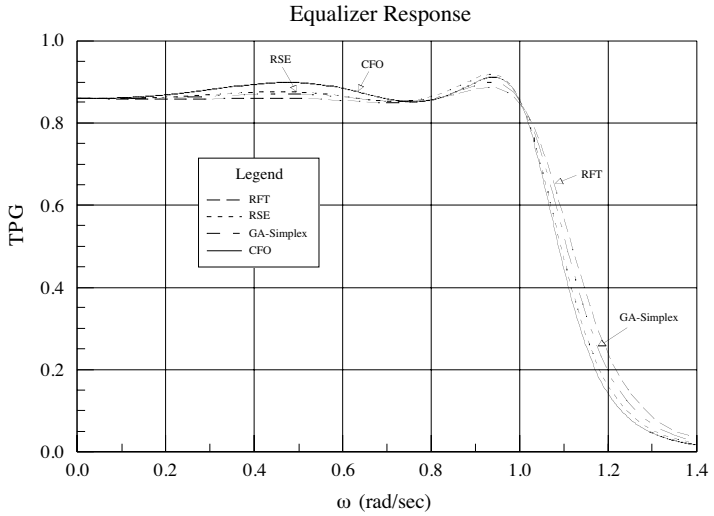

Figure 9. Optimized equalizer TPG.

be compared in a manner similar to GA-Simplex. RSE also comprised two calculation steps: 2,000 iterations using a stochastic Gauss-Newton optimization routine to locate an approximate solution which was then refined using a 20,000 iteration random search algorithm. The RSE method thus required 22,000 function evaluations. The literature does not report run times or rates of convergence similar to Fig. 7 for GA-Simplex or RSE, so that a direct comparison in that regard cannot be made. However, it seems reasonable that a good measure of computational efficiency is the total number of function evaluations that is required, and by that measure CFO is quite a bit better than the other methods. In terms of coding complexity, the flowchart in Fig. 2 in [9] and the pseudocode in Section E of [11] suggest that CFO probably is simpler to implement. CFO certainly is not more complex.

7. LINEAR ARRAY SYNTHESIS

The second CFO example is the synthesis of a 32-element linear array. This problem was solved using ACO [12]. The published results permit a direct comparison between CFO and ACO.

The reference linear array is shown schematically in Fig. 10. It comprises $2N_d$ elements equally spaced by a half wavelength ($\lambda/2$). [Note that N_d is the dimensionality of the CFO decision space as defined previously.] The array elements are positioned symmetrically about the origin along the X-axis. Each element is fed in-phase

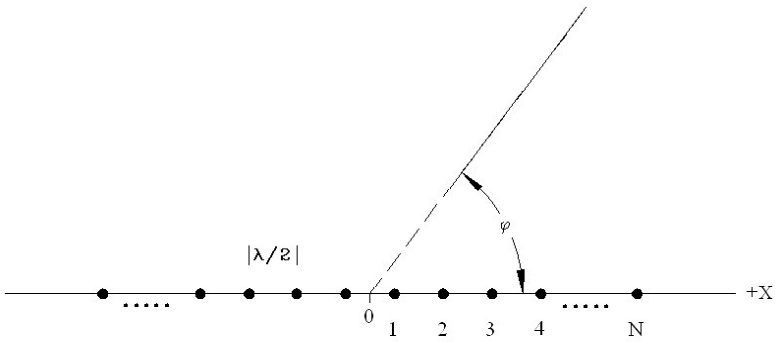

Figure 10. Initial linear array comprising $2N_d$ elements spaced $\lambda/2$ apart.

with equal amplitude excitation. In this case, the array factor simplifies to $F(\varphi, x_i) = 2\sum_{i=1}^{N_d} \cos(kx_i \cos\varphi)$, $k = \frac{2\pi}{\lambda}$, where x_i, $i = 1, \ldots, N_d$ are the (dimensional) element coordinates [12]. Normalizing x_i to $\lambda/2$, $F(\varphi, x_i') = 2\sum_{i=1}^{N_d} \cos(\pi x_i' \cos\varphi)$ where $x_i' = \frac{x_i}{0.5\lambda}$. The element coordinates in the uniformly spaced reference array are $x_i' = 0.5, 1.5, 2.5, \ldots, (N_d - 0.5)$. The array factor $F(\varphi, x_i')$ has a maximum value of $2N_d$. The array's normalized radiation pattern (directivity) in dB is given by $D'(\varphi, x_i') = 10\log_{10}\left(\frac{1}{2N_d} F(\phi, x_i')\right)^2$ [13].

The goal of the optimization procedure is to meet specific design goals for the array's pattern by changing only the positions of the array elements, x_i, not the excitation amplitude or phase. In this example the specific objective is to achieve a main lobe beamwidth $\leq 7.7°$, maximum sidelobe level $\leq -15\,\text{dB}$, and a deep null in the direction $\varphi_{null} = 81°$, and by symmetry also at $\varphi_{null} = 99°$, in a 32-element array. These values are taken from [12, §3.5].

In the context of CFO, the problem may be stated: Determine the array element coordinates x_i, where $x_i^{\min} \leq x_i \leq x_i^{\max}$, $i = 1, \ldots, N_d$, $N_d = 16$ so as to maximize a user-defined fitness function

$$f(x_i') = g\{BW[D'(\varphi, x_i')], \quad SLL[D'(\varphi, x_i')], \quad ND[D'(\phi_{null}, x_i')]\}$$

in which the functions $BW[D'(\varphi,x_i')], SLL[D'(\varphi,x_i')], ND[D'(\phi_{null},x_i')]$ respectively, return the main lobe beamwidth between first nulls, the maximum sidelobe level in dB, and the null depth in the specific direction φ_{null}. BW is positive definite, while SLL and ND are negative definite. This is a *constrained* optimization problem because

the x_i also must meet the requirement that no array elements occupy the same position, that is, $x_i \neq x_j$ for $i \neq j,\ i, j = 1, \ldots, N_d$.

The fitness function $g\{\varphi, x_i'\}$ may be any function the antenna designer wishes to use as a measure of the array's merit (how well the optimization objective is met). In [12, §3.5], for example, the "desirability function" (synonymous with "fitness") is defined as

$$f(\varphi, x_i') = \left|F(\varphi_{null}, x_i')\right|^{\frac{1}{\beta_3}} \left[\frac{1}{BW - \varphi_f}\right]^{\frac{1}{\beta_2}} |SLL|,$$

where $\varphi_f = 7°$, and $\beta_2 = 20$ and $\beta_3 = 6$ are empirically chosen "modulator parameters." [Note that variable symbols have been translated to be consistent with those in this paper.] It is emphasized in [12] that "...the definition of this function is one of the critical issues..." in developing the ACO algorithm. The other ACO run parameters in [12] also were determined empirically, which the authors point out "...is a common issue in most optimization algorithms."

CFO optimized the 32-element array using 1° pattern resolution in a decision space defined by $0.1 \leq x_i' \leq 32.5,\ i = 1, \ldots, 16$, with parameters $N_p = 48$ (3 probes per dimension), $N_t = 7,\ G = 2,\ \alpha = 2,\ \beta = 2,\ a_{init} = 0$. The initial probes were placed slightly off the decision space diagonal according to the prescription in section 5.1(2)(A)(ii) above [this deployment is consistent with the element position constraint]. CFO thus requires six independent parameters to define a run.

ACO also uses six user-specified parameters, viz., numbers of "ants" and iterations, probability function exponents (α, β), pheromone elimination period (γ), and pheromone elimination period coefficient (ρ) [12]. In addition to the "desirability" function, ACO also requires the definition of another "concentration pheromone function" [12]. Consequently CFO is less complex than ACO in setting up a run. Also, unlike ACO which is inherently discrete ([12], wherein $\Delta x_i = 0.1\lambda$), CFO is continuous, not requiring any artificial discretization of the decision space.

Optimization algorithms often are given a starting point in the decision space, usually a "best guess," or a previous run's best result, or, as in the linear array case, a "reference" design. Thus, following [12] in which the ACO search was begun with the uniform reference array, the CFO search also was commenced with the uniform array's coordinates inserted into probe #1 at time step #0, that is, $R(1, i, 0) = i - 0.5,\ i = 1, \ldots, N_d$.

The array designer is free to use any form of fitness function, and different forms will yield different results because the decision space

topology changes. For this example, the CFO fitness function was defined as

$$f(x_i') = c_1|SLL[D'(\phi, x_i')]| + c_2|ND[D'(\phi_{null}, x_i')]| - BW[D'(\phi, x_i')],$$

where the coefficients c_1 and c_2 were determined empirically as $c_1 = 1.5,\ c_2 = 0.2$. These values provided the desired balance between the three array parameters. This CFO fitness function seems to be much simpler than the ACO function, and, in the author's, opinion reflects an intuitively sensible way of combining the parameters to be optimized.

Table 2. CFO/ACO results for the optimized linear array.

-	N_{eval}	*BW*(deg)	*SLL(dB)*	*ND(dB)*
Goal	-	7.7	-15	-60
ACO	5,300*	7.35	-17.1	<-60*
CFO	336	6.00	-14.84	-62.8

* estimated from Figs. 16 and 14 in [12]

Table 3. Element coordinates (in $\lambda/2$) for the CFO-optimized 32-element linear array.

Coord	***Value***	***Coord***	***Value***
x_1'	1.2450	x_9'	10.0577
x_2'	1.3991	x_{10}'	11.3133
x_3'	2.5050	x_{11}'	12.5702
x_4'	3.7688	x_{12}'	13.8260
x_5'	5.0269	x_{13}'	15.0818
x_6'	6.2867	x_{14}'	16.3403
x_7'	7.5465	x_{15}'	17.6670
x_8'	8.8021	x_{16}'	18.9318

CFO optimization results appear in Table 2. This table also shows the target design values and the ACO results from [12]. Element coordinates for the CFO-optimized 32-element array are given in Table 3. CFO run time was less than 10 seconds. Figs. 11, 12, and 13 plot the evolution with time step of CFO's best fitness value, the average distance of other probes to the probe with the best fitness, and the array element coordinates.

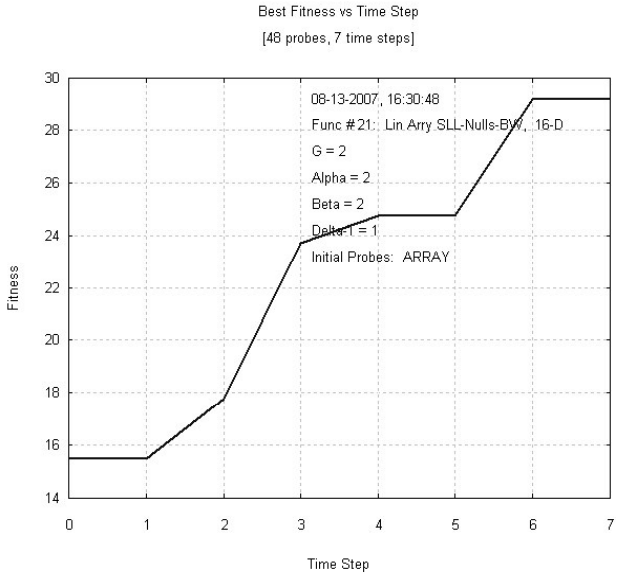

Figure 11. Evolution of linear array fitness.

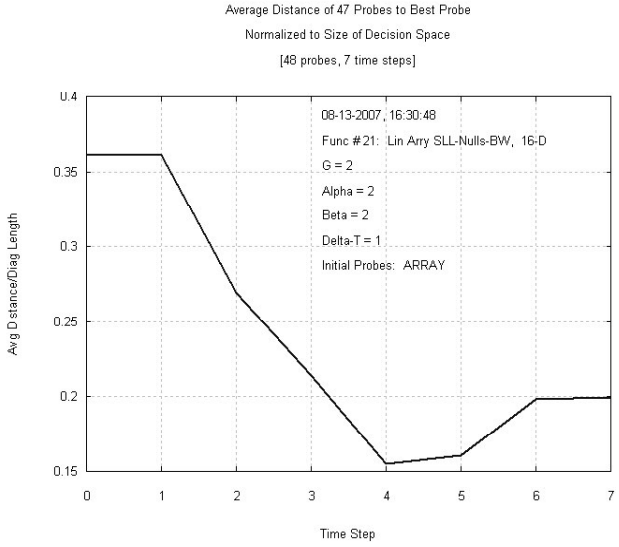

Figure 12. Average distance to best probe.

The data in Table 2 show how effective a properly set up CFO run can be. Using only 336 function evaluations (N_{eval}), compared to 5,300 for ACO, CFO produced an array with a narrower main beam and comparable null depth in less than 10 seconds. CFO missed the sidelobe level objective by a mere 0.16 dB, and this shortfall likely can be removed by "tweaking" the CFO run parameters or the fitness function coefficients c_1 and c_2.

Referring to the plotted CFO results, the array fitness (Fig. 11) increases very quickly, reaching a plateau in only 6 time steps. The average probe distance (Fig. 12) rapidly decreases through step 4, thereafter increasing moderately. CFO runs with 500 time steps (results not shown), all other parameters unchanged, confirm that the fitness increases only very slightly through step 80 and is flat thereafter. The average probe distance settles into a small amplitude oscillation around ≈ 0.315.

Fig. 13 shows how CFO evolves the element coordinates. The initial uniform array's coordinates increase very quickly from step 1 to 2, falling nearly as rapidly from step 2 to 3. Between steps 3 and 4 the coordinates again increase, followed by a plateau until step 5, and then a decrease to essentially their final values at step 6. It is interesting that in almost every case the degree to which a coordinate varies is proportional to its coordinate number, the exceptions being x_1, x_2 and x_{14}, x_{15}. However, it is not obvious what the significance of this behavior might be, if indeed there is any.

Fig. 14 plots the optimized array's normalized radiation pattern

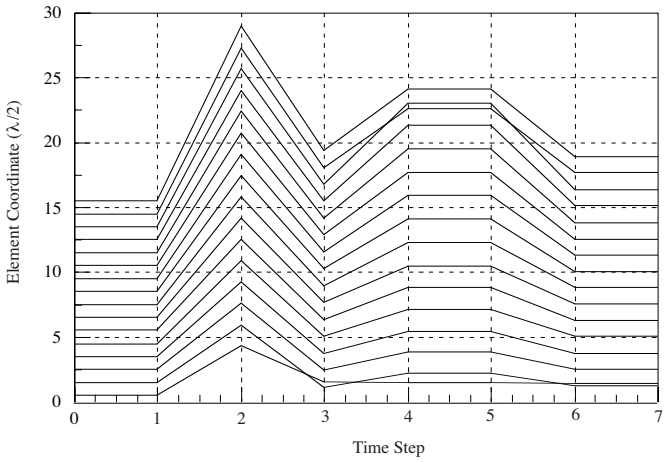

Figure 13. Evolution of array coordinates.

(heavy solid line), the (a) curve being the entire pattern, while the (b) curve provides more detail in the range $75^\circ \leq \varphi \leq 105^\circ$ [both (a) and (b) plots were computed with $\Delta\varphi = 0.25^\circ$]. On each graph the initial uniform array's pattern is plotted using a thin solid line in (a) and a dashed line in (b). It is apparent from Fig. 14(a) that the main beam is narrowed and the null at $\varphi_{null} = 81^\circ$ is created generally at the expense of increased sidelobe level away from the main beam. However, in close to the main beam the sidelobes actually decrease somewhat relative to the initial pattern.

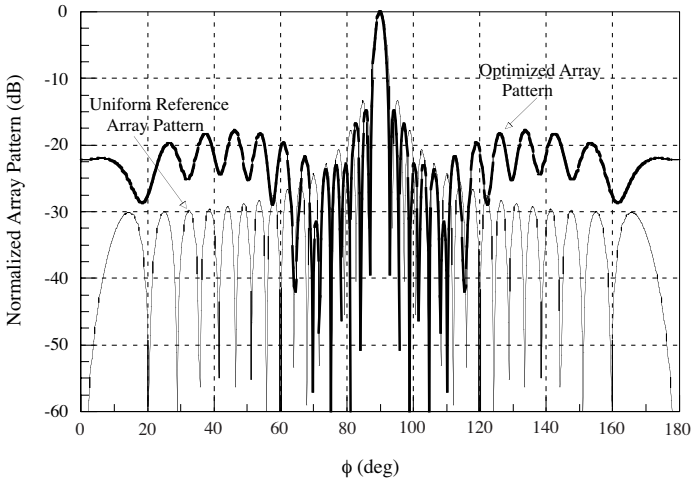

Figure 14a. CFO-optimized array pattern.

Fig. 14(b) provides a more detailed view of the pattern close to the main beam. The deep nulls at precisely $\varphi_{null} = 81^\circ, 99^\circ$ are clearly evident in the pattern. A measure of how effective the CFO optimization procedure has been is the fact that these null directions fall essentially on what are the second sidelobe maxima in the initial pattern. That sidelobe level was reduced by about 42 dB. It also is evident from the plot that the main beam has been narrowed, and that the maximum sidelobe level has been maintained around -15 dB as required. In fact, over the angular range of the plot, the optimized array's sidelobes are uniformly lower than those in the initial array's pattern.

These results clearly demonstrate that CFO works as well as ACO in optimizing a long linear array against three disparate performance measures: main lobe beamwidth, maximum sidelobe level, and nulling the pattern in a specific direction.

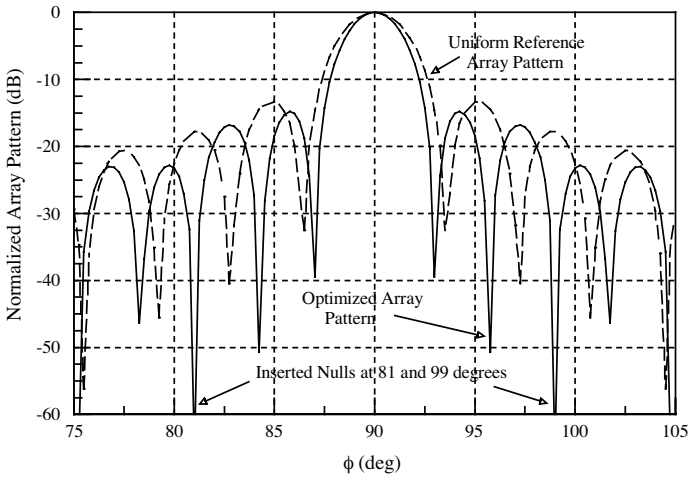

Figure 14b. Detailed array pattern.

Exactly how well a CFO run works does depend on the run parameters, and choosing a different set of parameters likely will yield different results. Fortunately, as this example illustrates, CFO frequently runs so quickly (in this example less than 10 seconds) that looping over sets of run parameters is easily done. Until a firm theoretical basis is established for defining run parameters, this brute force approach may well be suitable for a very wide range of optimization problems.

8. BENCHMARK TEST FUNCTIONS

In addition to testing against typical engineering applications as described in Sections 6 and 7, Central Force Optimization also has been tested against a variety of standard benchmark functions. The reason for testing against benchmarks is that their extrema are known, thus providing a numerically precise standard by which the effectiveness of an algorithm can be evaluated. This paper reports results for seven 30-dimensional ("30-D") functions, four 2-dimensional ("2-D") functions, and one four-dimensional function ("4-D"), all drawn from standard test suites. They were chosen to illustrate CFO's strengths and weaknesses, and appear explicitly in the Appendix. Except for two, all functions are drawn from the benchmark suite described in [14]. The modified Keane's Bump appears [15], and the Colville Function in [16]. CFO also has been tested against an additional twenty-two 2-D functions [30] not described here.

The following parameter values were used for all the CFO runs reported in this section: $G = 2,\ \alpha = 2,\ \beta = 2,\ a_{init} = 0$. As in the previous sections, there is no particular reason why these specific values were chosen, except that they seem to work reasonably well for the purpose of illustrating CFO's effectiveness. Indeed, as was emphasized earlier, exactly how these parameters should be chosen is an important open question. One interesting possibility is that the set of CFO run parameters itself might be determined by an optimization algorithm. This approach may be feasible for certain types of objective functions, viz. highly multimodal ones, because of CFO's rapid convergence.

Because CFO is inherently deterministic, it is not necessary to characterize its performance statistically by making multiple runs. Every CFO run with the same parameters returns the same result, so that the data tabulated below are derived from only one CFO run for each function.

Each of the test functions was searched using the algorithm described in Section 5. For the first eleven functions, the initial probe distribution comprised an even number of probes $\frac{N_p}{N_d}$ uniformly distributed along each of the coordinate axes, including the endpoints marking the limits of the decision space. Recall that N_p is the total number of probes used, not the number of probes per coordinate axis. For the last test function, Keane's Bump, a uniform 2-D grid of N_p probes was used for reasons explained below.

Since CFO searches for maxima, not minima, the negative of the functions in [14, 15] and [16] was computed. In addition, in order to avoid any bias resulting from the locations of maxima relative to the initial probe distribution, the maxima were offset if necessary. For example, the minimum of the original Griewank function in [14] is zero at the origin in the 30-D symmetric decision space $-600 \leq x_i \leq 600,\ i = 1, \ldots, 30$. The "modified" Griewank is the negative of the original with an offset maximum. Its maximum is zero at $[75.123]^{30}$, where the square bracket notation signifies $x_i = 75.123 \forall i$ in the 30-D space. Functions with offset maxima or with slightly different search regions are marked "Mod" in the tables below.

Results for the 30-D/4-D and 2-D test functions are summarized in Tables 4 and 5, respectively. Except for the discontinuous Step Function and Keane's Bump, all test functions are continuous on their domains of definition. Keane's Bump is constrained, whereas the others test functions are not. In the tables, N_p, N_t and N_{eval}, respectively, are the total number of probes, the number of time steps, and the total number of function evaluations, $N_{eval} = N_p \times N_t$. The number of function evaluations appears to be the best measure of how well CFO performs, and it is useful for comparing CFO to other

algorithms.

The next two columns in the tables, "CFO Fitness" and "Max Fitness," respectively, are the maximum fitness value of the objective function returned by CFO and the actual known maximum value. The last two columns, "CFO Coords" and "Actual Coords," respectively, are the coordinates of the CFO maximum fitness and the actual known location of the objective function's maximum. For all of the 30-D functions and the single 4-D function, the maximum's actual coordinates are the same for each dimension, while CFO returns slightly different values that vary by dimension. Consequently, the two values listed for the CFO coordinates are the minimum and maximum coordinate values returned by CFO. For the 2-D functions, the coordinates returned by CFO are tabulated.

8.1. 30-D/4-D Benchmark Functions

Turning to Table 4, without doubt the best example of CFO's 30-D performance is Schwefel's Problem 2.26. In only 1,920 evaluations CFO converged to a maximum of 12569.1 when the actual maximum is 12569.5. In comparison, the FEP ("Fast Evolutionary Programming") and CEP ("Classical Evolutionary Programming") algorithms in [14] returned minima of -12554.5 and -7917.1, respectively, using 900,000 function evaluations per run averaged over fifty runs.

Table 4. Summary of results for 30-D test functions $G = 2,\ \alpha = 2,\ \beta = 2,\ a_{init} = 0$ for all runs.

Function	N_p	N_t	N_{eval}	CFO Fitness	Max Fitness	CFO Coords	Actual Coords
Schwefel 2.26	240	8	1,920	12569.1	12569.5	(420.306ñ420.665)	$[420.9687]^{30*}$
Mod Griewank	780	6	4,680	-0.0459	0	(74.9000ñ75.2653)	$[75.123]^{30}$
Mod Ackleyís	780	5	3,900	-1.0066	0	(3.63045ñ4.24016)	$[4.321]^{30}$
Mod Rastrigin	600	8	4,800	-30.5308	0	(2.12839ñ2.13474)	$[1.123]^{30}$
Mod Step Function	600	4	2,400	-1	0	(73.6842-75)	$[75.123]^{30**}$
Mod Sphere	15,000	2	30,000	-0.0836	0	(75.0854-74.9168)	$[75.123]^{30}$
Mod Rosenbrock	60	250	15,000	-3.8	0	(26.0078-26.1289)	$[26.123]^{30}$
Mod Colville	56	15	840	-19.387	0	(7.74637,7.83799)	$[8.123]^{4}$

* notation: all coordinates in the 30-D or 4-D decision space have the same value shown in the square bracket.
** maximum occurs in a square region containing this point.

CFO's performance against the other benchmark functions is more of a mixed bag. In each case except the Sphere Model, convergence required less than 5,000 function evaluations. For the Sphere, 30,000 were required using a very large number of probes (15,000), which has

Table 5. Summary of results for 2-D test functions $G = 2,\ \alpha = 2,\ \beta = 2,\ a_{init} = 0$ for all runs.

Function	N_p	N_t	N_{eval}	CFO Fitness	Max Fitness	CFO Coords	Actual Coords
Mod Camel-Back	220	5	1,100	1.02956	0.0316285	(1.11294,0.28745)	(1.08983,0.2874)
	260	16	4,160	1.02772	1.0316285	(0.926972,1.69281)	(0.91017,1.7126)
Branin	400	18	7,200	-0.398689	-0.398	(3.12948,2.29433)	(3.142,2.275)
	490	15	7,350	-0.398294	-0.398	(9.42202,2.45343)	(9.425,2.425)
						not found	(-3.142,2.275)
Shekelís "Foxholes"	240	2	480	-1.2023	□1	(-31.9419,-32.768)	(-32,-32)
Mod Keaneís Bump	196	20	3,920	0.364915	unknown	(1.60267,0.46804)	unknown

a very substantial negative impact on runtime.

As to how well the maxima were located, CFO handled the Griewank and Sphere well, but results for Ackley's, Rastrigin's, Step, Rosenbrock and Colville functions were not nearly as good. The reason appears to be how thoroughly the initial probes sample the decision space's topology and some degree of trapping at a local maximum. In some cases the initial probe distribution that was used (uniform on the coordinate axes) provides adequate sampling of the objective function, whereas in others it appears not to do so. A different initial probe distribution presumably will provide different (hopefully better) results, but the question of the relationship of the initial probe distribution and CFO's performance in locating maxima is beyond the scope of this paper whose sole purpose is to introduce the CFO concept.

CFO's convergence rates are very high. As Table 4 shows, CFO converged in less than eight iterations in most cases, requiring only fifteen for the Colville Function. The Rosenbrock function required the greatest number of iterations at 250, with an attendant increase in the number of function evaluations. In the author's opinion, the best indicator of convergence rate is the total number of function evaluations, not the number of time steps alone. By that measure, CFO converges quite rapidly, requiring fewer than 5,000 function evaluations in most cases, and 30,000 in one case. Other algorithms (see [14], for example) typically require tens or hundreds of thousands of evaluations, and results vary from run to run because of the algorithm's stochastic nature.

One of CFO's weaknesses is that its computation time increases dramatically as the number of probes increases because of the

summation in eq. (4). All test runs reported here were made on a dual-boot 2 GHz Intel-based (T2500 CPU) MacBook Pro with 1 GB RAM running Windows XP/Pro SP2 and Mac "Bootcamp." The run times for the first five functions in Table 4 requiring from 1,920 to 4,800 function evaluations were between approximately 1.16 (Colville) and 280 (Griewank) seconds. In marked contrast, the runtime for the Sphere Model using 15,000 probes and 30,000 function evaluations was many hours, an unacceptably long time. An important question therefore is how to minimize the number of probes used by CFO. The answer may well lie in how the initial probes are deployed.

8.2. 2-D Benchmark Functions

Table 5 summarizes test function data for four 2-D objective functions. All of these runs were very quick, the shortest being about 3.9 seconds and the longest 14.2 seconds. One advantage of testing against 2-D objective functions is that the probe positions can be plotted, so that convergence is visually apparent. For example, Fig. 15(a) shows the initial distribution of 220 probes for the 6-Hump Camel-Back function. Fig. 15(b) shows the probe distribution at time step 5. Clustering of the probes around five of the six local maxima is evident at that

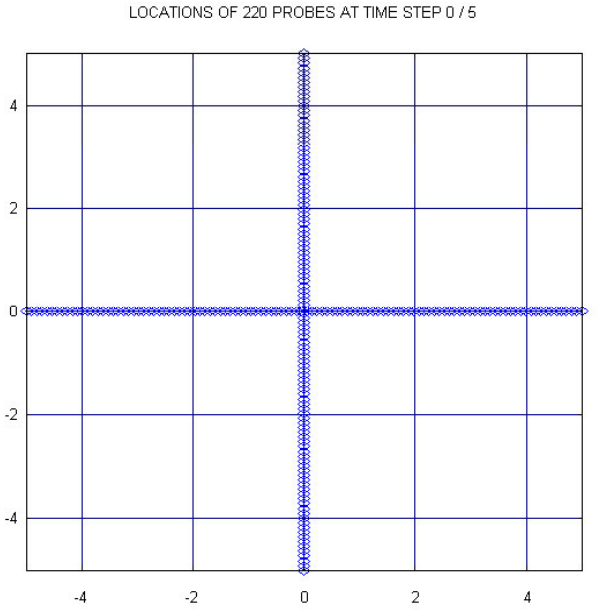

Figure 15a. Camel-back initial probes in x_1-x_2 plane.

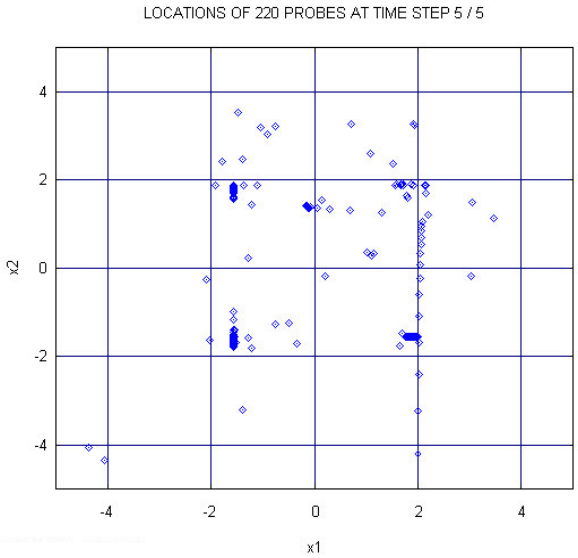

Figure 15b. Camel-back probes at time step 5 in x_1-x_2 plane.

time. CFO's tendency to cluster probes in this manner appears to be a unique and possibly useful attribute as discussed in the next section. The Camel-Back has two global maxima as shown in Table 5, and, interestingly, changing the number of initial probes toggles CFO between these maxima. With 220 probes, CFO converges on one of the maxima (1.02956), while 260 initial probes results in convergence on the other (1.02772). This effect suggests some measure of trapping near a local maximum, which clearly is an issue that should be addressed in future CFO research.

A similar effect is seen with the Branin function. It has global maxima at three points as shown, and CFO toggles between two of them depending on the number of initial probes. This observation lends further support to the speculation that CFO's initial probe distribution relative to the objective function's topology is an important factor in determining how CFO converges, and that some local trapping results from the specific probe distribution. In this case, the likely reason that the Branin's third maximum is not found is its asymmetrical decision space, $-5 \leq x_1 \leq 10,\ 0 \leq x_2 \leq 15$, which places more probes in the first quadrant, thereby favoring solutions in that region.

CFO's maximum fitnesses for both the Camel-Back and Branin functions are close to the actual values, while its maximum for Shekel's "Foxholes" function is not quite as good. Because the negative of the

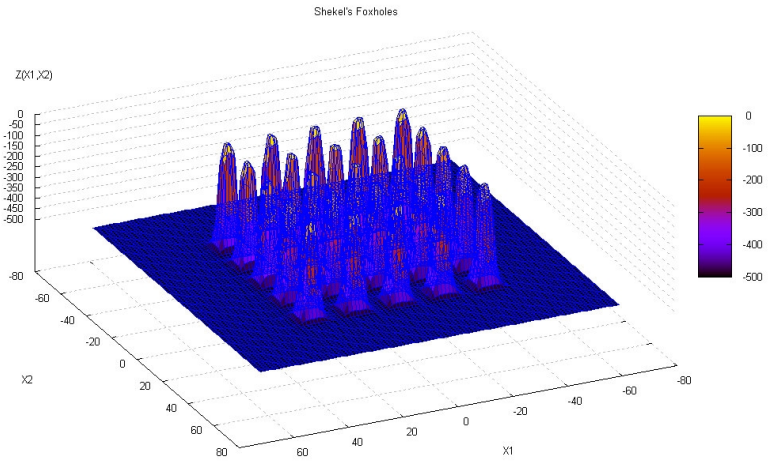

Figure 16a. 2-D Shekel's "Foxholes" function.

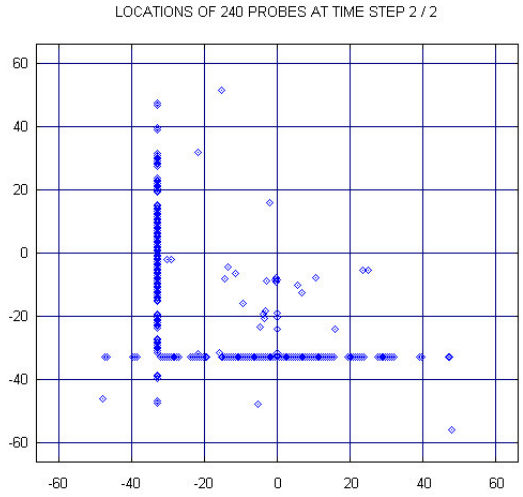

Figure 16b. Probe locations at time step 2 for Shekel's "Foxholes".

original Shekel's Foxholes test function is taken, the function used here perhaps is better described as "inverted foxholes," hence the quotation marks in its name. A perspective view of this function appears in Fig. 16(a). The final probe distribution at time step 2 is shown in Fig. 16(b). The primarily linear distribution of probes is intuitively consistent with the grid-like arrangement of the function's peaks.

The last 2-D test function is Keane's Bump, a constrained objective function. The locations of its maxima are not known, but they can be approximated by examining the perspective and plan views in Fig. 17. The global and nearest local maxima are contained in two "ridge line" regions near $x_1 = \mp 1.6$, $x_2 = \mp 0.47$. These regions are nearly, but not precisely, symmetrical.

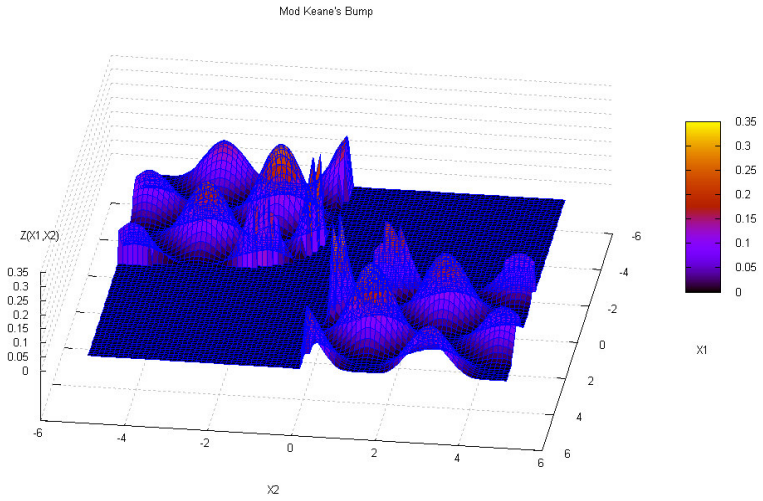

Figure 17a. Perspective view of Keane's bump.

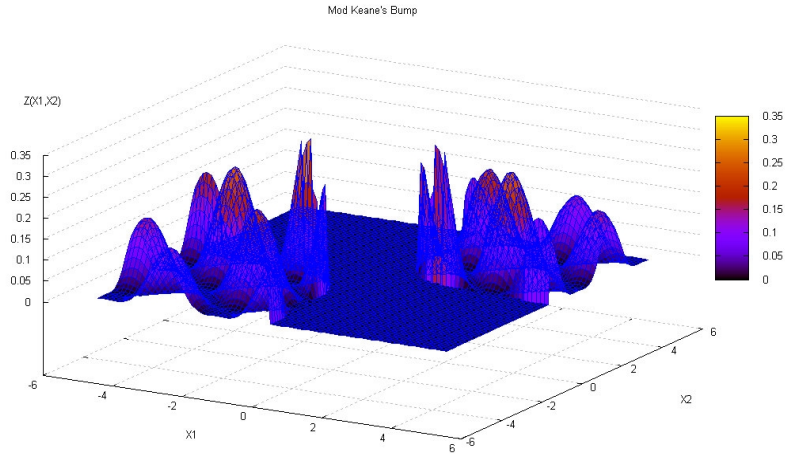

Figure 17b. Perspective view of Keane's bump.

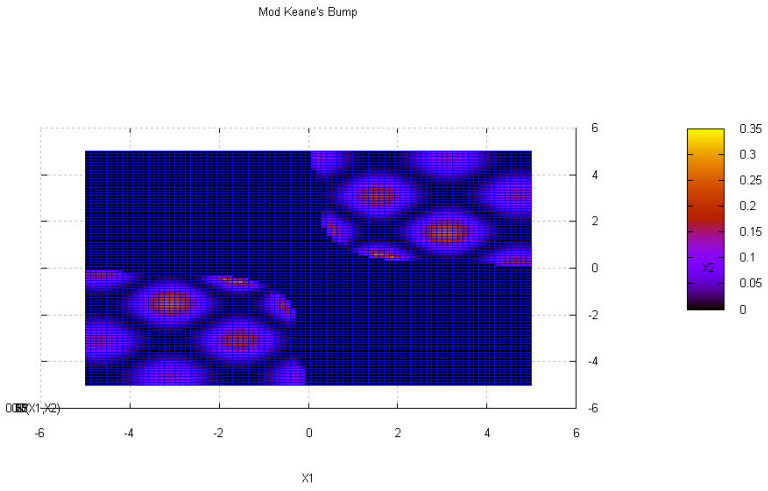

Figure 17c. Plan view of Keane's bump.

Unlike the other test functions, the initial probe distribution for Keane's Bump was a uniform 2-dimensional *grid* of probes instead of a uniform distribution of probes along each coordinate axis. The reason for this departure is that an examination of Fig. 17(c) reveals that the function value is zero along each coordinate axis, so that initial probes deployed there can provide no information about the function. This is yet another example of why the initial CFO probe distribution must somehow be related to the objective function's topology in its decision space. Fig. 18 shows the initial probe distribution that was used, a square grid of 14 probes on each edge for a total of 196 probes.

Because Keane's Bump is highly multimodal, it is useful in illustrating the effect of varying CFO's "gravity." Runs were made with four different values of the gravitational constant G: $+7, +2, +0.5$ and -2. Results appear in Table 6. CFO's convergence is clearly influenced by the value of G, and it is perhaps counter-intuitive that the greatest gravitational constant does not provide the best convergence (fitness of 0.357... for $G = 7$ vs. 0.364... for $G = 2$). Choosing G clearly is important for obtaining good results, but exactly how it should be done remains elusive.

Another interesting attribute of the CFO algorithm is how it performs with a negative gravitational constant. When $G < 0$ CFO's "gravity" is repulsive, so that instead of attracting probes towards good solutions, the negative gravity pushes them away. The results for $G = -2$ clearly show this effect. The maximum fitness does not

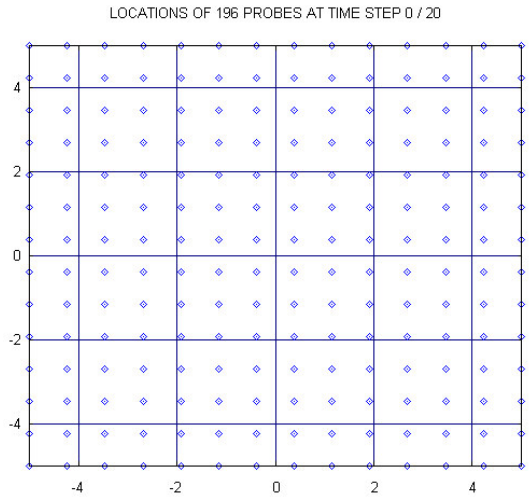

Figure 18. Initial probe distribution for Keane's bump.

Table 6. Results for Keane's bump with varying gravity $N_p = 196,\ \alpha = 2,\ \beta = 2,\ a_{init} = 0$ for all runs.

G	N_t	CFO Fitness	CFO Coords
2	20	0.364915	(1.60267,0.46804)
7	50	0.357819	(1.61608,0.472266)
0.5	50	0.362238	(1.6124,0.468207)
ñ2	50	0.141965	(3.46154,1.15385)

even approximate the correct value, nor are the coordinates correct. Figs. 19(a) and (b), show the final probe locations for $G = 2$ and $G = -2$, respectively. While Fig. 19(a) clearly shows clustering of the probes near the ridge lines containing the maxima, Fig. 19(b) shows the probes clustering near the decision space boundaries. There are no probes near the maxima for $G = -2$ because they have been pushed to the very edges of the decision space.

Fig. 20 provides further insight into the effect of varying G. It plots the normalized average distance between the probe with the best fitness value and all the other probes as a function of the time step. As before, the average probe distance is normalized to the size of the

decision space as measured by the length of its diagonal. For $G = 2$, Fig. 20(a), the distance decreases nearly monotonically from a value near 0.37 to just over 0.17 at time step 20. This is a result of the

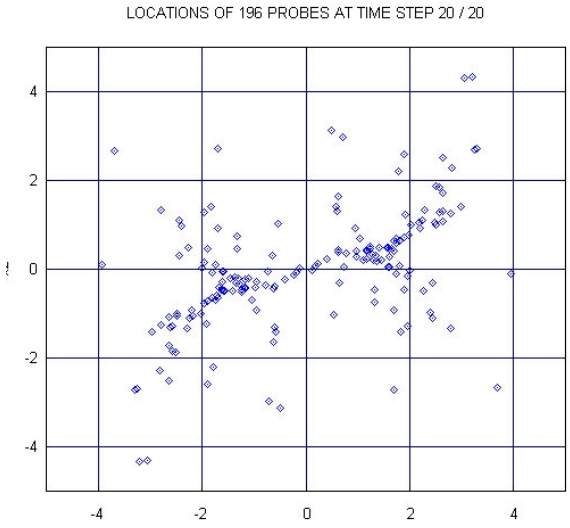

Figure 19a. Final probe locations for $G = 2$.

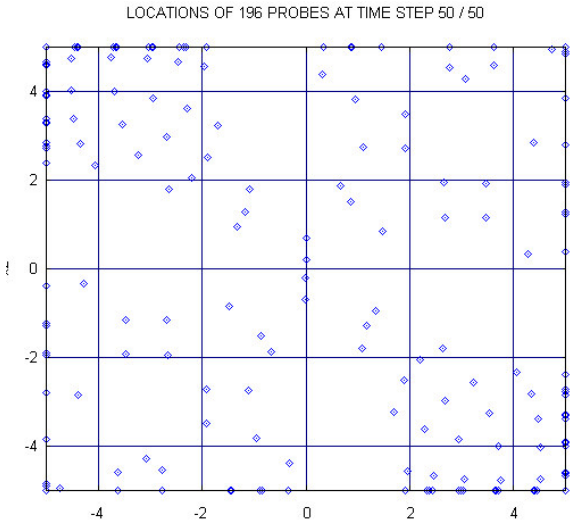

Figure 19b. Final probe locations for $G = -2$.

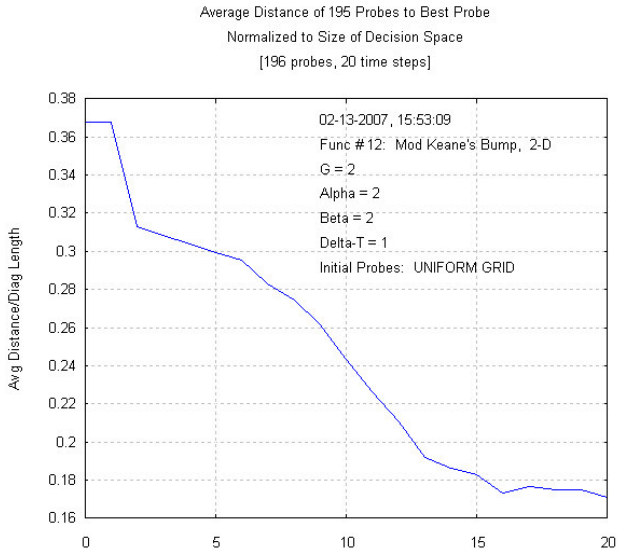

Figure 20a. Average probe distance vs. time step for $G = 2$.

Figure 20b. Average probe distance vs. time step for $G = -2$.

clustering of probes near the maxima. As time progresses, more and more probes get closer and closer together. Just the opposite happens when gravity is negative. In Fig. 20(b), $G = -2$, the probes are seen to fly away from each other. The average distance to the best fitness probe increases almost linearly with time step from the starting value near 0.37 to a value slightly under 0.44 at step 50.

This example using Keane's Bump function shows how important the appropriate initial probe distribution and the appropriate value of the gravitational constant are to designing an effective CFO algorithm. How to choose these parameters is an important unresolved question.

9. CFO AS A TOPOLOGY MAPPER?

Central Force Optimization exhibits what appears to be a unique and potentially useful characteristic: the algorithm clusters its probes near maxima, both local and global, without necessarily flying all the probes to a single "best" point (as, for example, PSO and ACO generally do). Thus CFO may be useful as a tool for "mapping" the topology of a decision space by locating its approximate maxima. Doing so may improve an optimizer's convergence by permitting a reduction in the size of the decision space that is searched. Instead of searching a very wide range in the decision variables, CFO may allow for multiple searches in much smaller regions, ones that CFO has identified as containing maxima.

CFO appears to converge very quickly for highly multimodal functions, so that its use as a "mapping preprocessor" may make sense. In addition, many engineering applications are not necessarily best served by locating the actual global maximum. For fitness functions comprising many maxima of similar amplitude, real world design and fabrication issues may well make a sub-optimal solution actually the "best" solution. By clustering probes around maxima, CFO may point an optimizer toward solutions that are not globally optimal but nevertheless merit consideration. This section describes CFO's clustering behavior with three 2-D example functions.

9.1. 2-D Sine Function

The 2-D sine curve is defined as

$$f(x_1, x_2) = \sin\left(7.5\sqrt{(x_1 - 2.5)^2 + (x_2 - 2.5)^2}\right),\ 0 \le x_1,\ x_2 \le 5.$$

This function has an infinite number of indistinguishable maxima with a value of unity as shown in Fig. 21. CFO tends to distribute its probes

Table 7. 2-D sine function CFO run parameters.

Run	N_p	N_t	N_{eval}	***Init. Dist.***
CFO-1	36	25	900	Uniform Grid
CFO-2	16	55	880	Random

over all the maxima points, whereas most, if not all, other algorithms will converge on a single point.

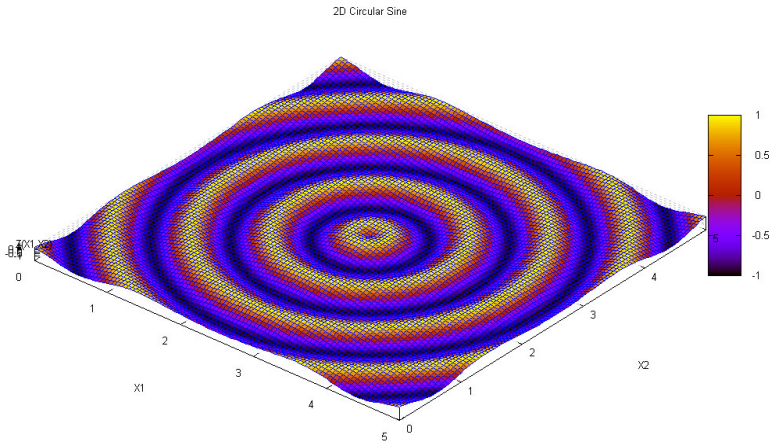

Figure 21. 2-D circular sine function.

Two CFO runs were made with the parameters shown in Table 7. The runs utilized approximately the same total number of function evaluations, $N_{eval} \sim 900$. The *Init. Dist.* column shows the type of initial probe distributions that were used because the initial probe distribution can have a significant effect on an algorithm's convergence. For example, if an initial CFO probe (or PSO particle, or ACO "ant") is placed right on one of the function maxima, accidentally or otherwise, then the best probe fitness returned at the very beginning of the run is the maximum possible value, in this case, unity. The initial distribution thus biases the run toward or away from the actual maxima. Because CFO is inherently deterministic, in this case both deterministic and random uniform initial probe distributions were used. This is a departure from all the previous runs reported in this paper. Plots of the initial distributions are shown in Fig. 22. The uniform grid of initial probes does not bias the algorithm by placing probes on or near maxima.

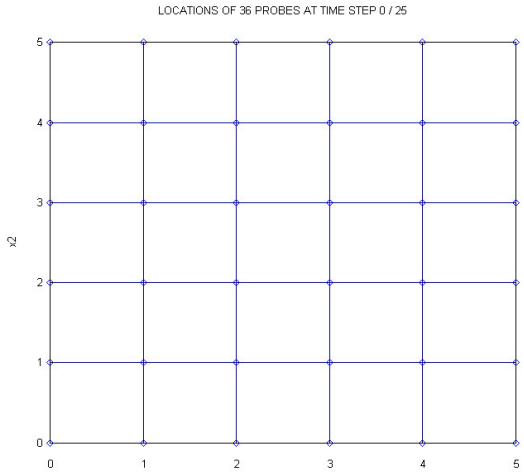

Figure 22a. Initial uniform grid of CFO probes x_1-x_2 plane.

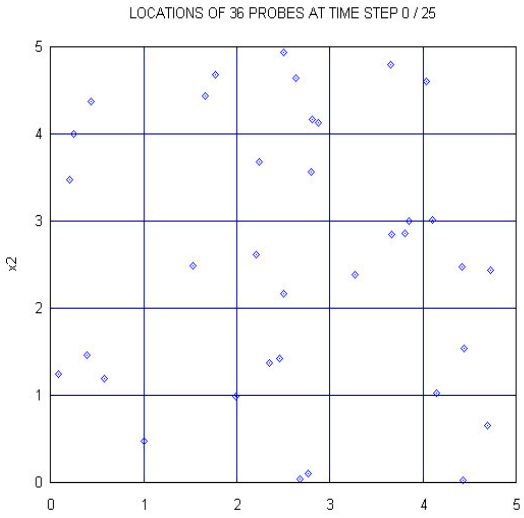

Figure 22b. Initial random CFO probes x_1-x_2 plane.

Probe locations at time step 25 are shown in Fig. 23(a). It is clearly evident that the CFO probes cluster symmetrically along concentric circles at the maxima. The reason that probes in the outer circle are grouped around lines radiating from center to corners is that there is more mass in those directions. If decision space were circular

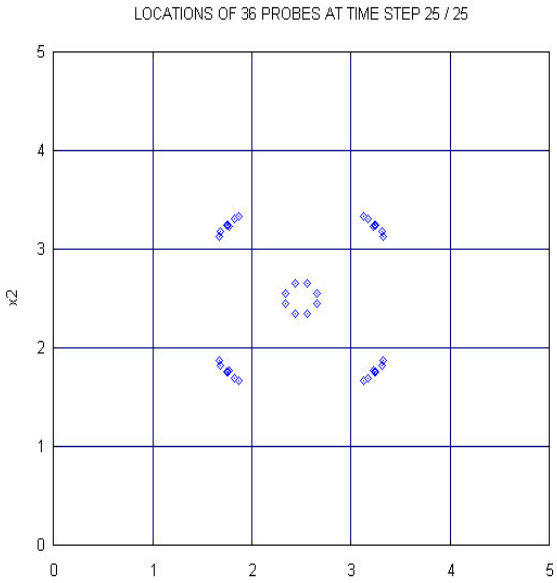

Figure 23a. CFO probe locations x_1-x_2 plane at Step 25.

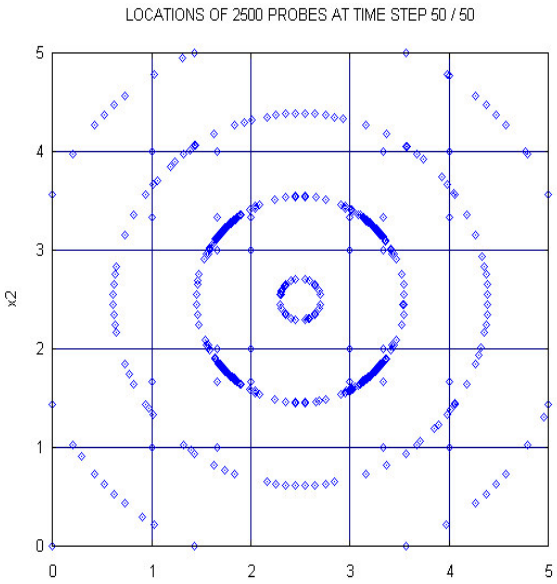

Figure 23b. 2500 CFO probe x_1-x_2 plane at Step 50.

instead of square, the probes presumably would show a more or less uniform distribution along the maxima circles. To further illustrate this point, Fig. 23(b) shows the positions of 2,500 probes after 50 time steps using the same run parameters. The circular structure of the sine curve's maxima is clearly evident, as is the higher concentration of probes in the directions of the decision space's diagonals resulting from more "mass" in the direction of the corners.

This characteristic of CFO, its ability to cluster probes around many maxima, appears to be unique among optimization methodologies, and may be useful in "mapping" the topology of an unknown decision space. The results of a preliminary CFO run might, for example, be used to seed another optimization run, whether it is CFO, PSO, ACO, or any other optimizer. Starting an optimization run in the vicinity of known maxima can greatly improve convergence efficiency, and, as this example shows, CFO appears to do quite well in locating multiple maxima.

The maximum fitnesses returned by the first CFO run with $N_{eval} = N_p \times N_t = 900$ are in the range 0.954328–0.999964 (Table 8). These data show a high degree of symmetry, which reflects the 2D sine's circular symmetry. The best fitness as a function of time step is shown in Fig. 24(a). It increases very quickly between steps 6 and 7, and very slowly thereafter. The average distance from all probes to the best probe normalized to the size of the decision space (length of the principal diagonal) is plotted in Fig. 24(b) as a function of time step. The flattening of the curve after step 23 suggests that a stable probe

Table 8. 2D sine function maxima computed by CFO using uniform initial probes.

Maximum Fitness	x_1	x_2
0.999964	3.24128	3.24128
0.999869	3.33134	1.86676
0.999869	1.86676	3.33134
0.999869	1.86676	1.66866
0.996793	1.75197	1.75197
0.995861	1.76810	3.23190
0.995861	1.76810	1.76810
0.995540	1.82214	1.68534
0.995540	1.82214	3.31466
0.954328	2.55764	2.34115
0.954328	2.34115	2.55764

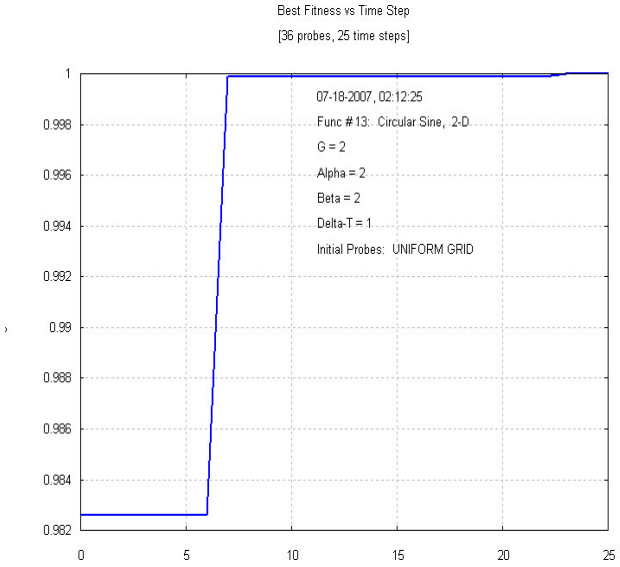

Figure 24a. CFO run 1 best fitness vs. time step.

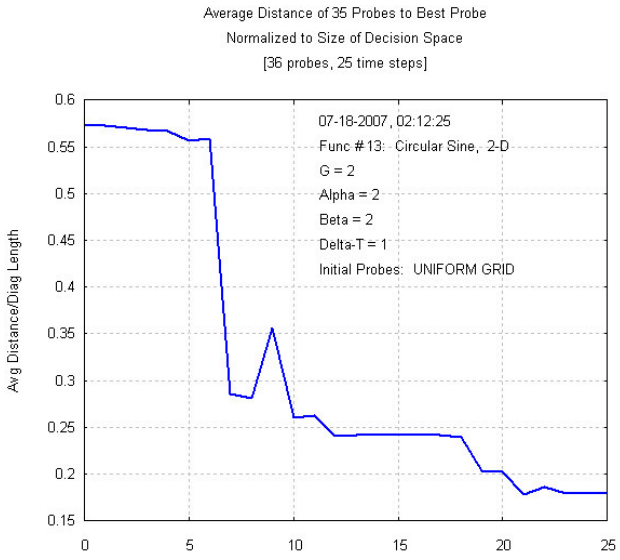

Figure 24b. CFO run 1 average distance to best probe vs. time step.

Table 9. Summary of CFO results for the 2D sine function.

Run	*Max Fitness*	x_1	x_2
CFO-1	0.999964	3.24128	3.24128
CFO-2	0.999784	0.30135	4.10126

configuration has been obtained, and such flattening may be a useful measure of convergence. Table 9 summarizes the best fitness results for the two CFO runs. In both cases CFO has come very close to the actual maximum of unity.

9.2. Modulated R^2 Function

Another illustration of CFO's ability to cluster probes is provided by the modulated R-squared function defined as

$$f(x_1, x_2) = -r^2 - \sin(10\pi r),\ r = \sqrt{x_1^2 + x_2^2},\ -5 \le x_1,\ x_2 \le 5.$$

Its global maximum value is ≈ 0.977545 at $r \approx 0.1497$, these values being determined numerically from the data used to create the radial plot in Fig. 25. A perspective view of this function appears in Fig. 26.

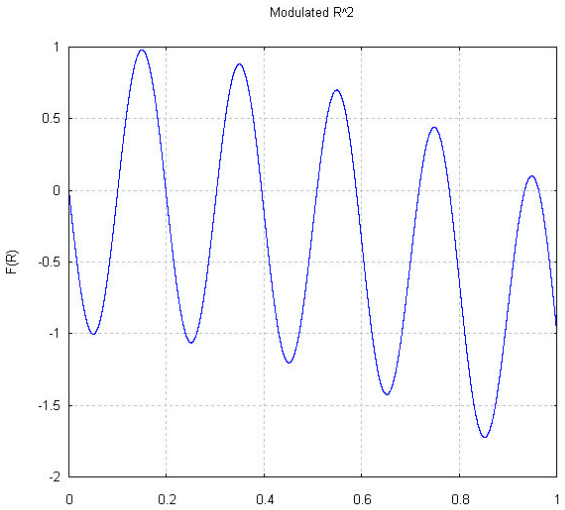

Figure 25. Modulated R-squared function vs. $r = \sqrt{x_1^2 + x_2^2}$.

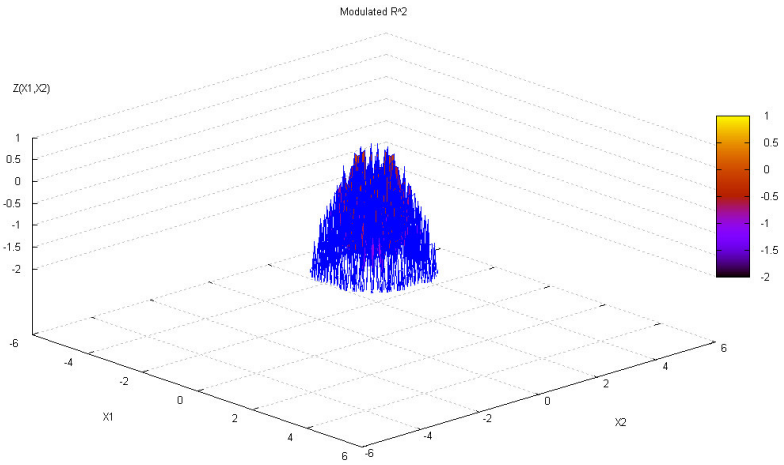

Figure 26. 2-D modulated R-squared function.

The CFO run was made using a uniform grid of initial probes with $N_p = 225,\ N_t = 2048,\ G = 5,\ \alpha = 4,\ \beta = 2,\ \Delta t = 1$. These values are somewhat different than those used previously, and, as in all previous cases, were determined empirically. The best fitness returned by CFO was 0.977494 at $x_1 = -0.128425,\ x_2 = 0.076285$. The radial distance to this point is 0.149373, which agrees very well with the numerically determined radius.

CFO's clustering effect is evident in Figs. 27(a) and (b) showing the probe distributions at time steps 384 and 2048, respectively. While convergence was essentially obtained much earlier in the run, the large number of time steps was chosen because it shows how tightly CFO clusters probes on the very small circle of global maxima centered on the origin.

9.3. Three Cylinders

The previous test functions were continuous, which raises the question of how well CFO deals with discontinuous objective functions. This test demonstrates that CFO successfully locates and distinguishes multiple clustered maxima even when the objective function is highly discontinuous.

The three cylinders function is defined on $0 \le x_1,\ x_2 \le 5$ by the following equations:

$$r_1 = \sqrt{(x_1 - 3)^2 + (x_2 - 2)^2}, \quad r_2 = \sqrt{(x_1 - 4)^2 + (x_2 - 4)^2},$$

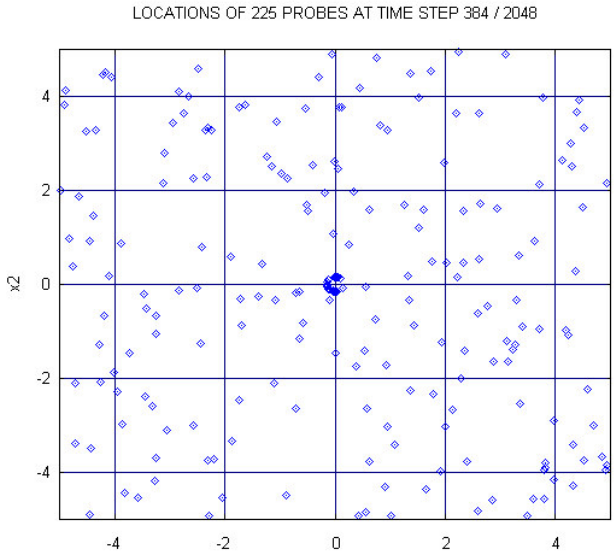

Figure 27a. Modulated R-squared probes in x_1-x_2 plane at Step 384.

Figure 27b. Modulated R-squared probes in x_1-x_2 plane at Step 2048.

$$r_3 = \sqrt{(x_1-1)^2+(x_2-3)^2}$$

$$f(x_1,x_2) = \begin{cases} 1, & r_1 \le 0.75 \\ 1.05, & r_2 \le 0.375 \\ 1.05, & r_3 \le 0.25 \\ 0, & \text{otherwise} \end{cases}$$

This function is plotted in Fig. 28 (note that the cylinder walls do not appear to be perfectly perpendicular to the x_1, x_2-plane because of the granularity with which the plot is calculated). The cylinders are centered on the points (3,2), (4,4) and (1,3). The fattest cylinder with radius 0.75 has a height of 1.0, while the other two smaller diameter cylinders at (4,4) and (1,3), radii 0.375 and 0.25, respectively, have heights of 1.05. Thus, the global maxima are an infinite number of points on the ends of the smaller diameter cylinders whose values are only 5% greater than the local maxima centered on (3,2) and occupying a much larger area.

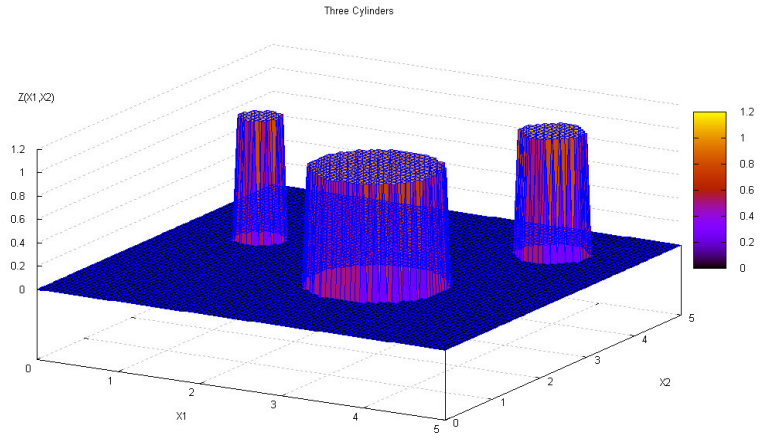

Figure 28. Three cylinders.

The CFO run was made with $N_p = 225$, $N_t = 8192$, $G = 5$, $\alpha = 4$, $\beta = 2$, $\Delta t = 1$ and a uniform initial probe distribution. Probe locations in the x_1-x_2 plane appear in Fig. 29. At time step 16, CFO clearly has located the three cylinders. In fact, it appears that the optimizer will cluster probes on all three maxima circles, global and local. As the CFO run progresses, however, probes are drawn away from the fat cylinder to the two smaller diameter cylinders because of their greater gravitational attraction. At step 4096, for example, the probes have significantly dispersed away from the fat cylinder. By the end of the run all but fifteen probes have clustered

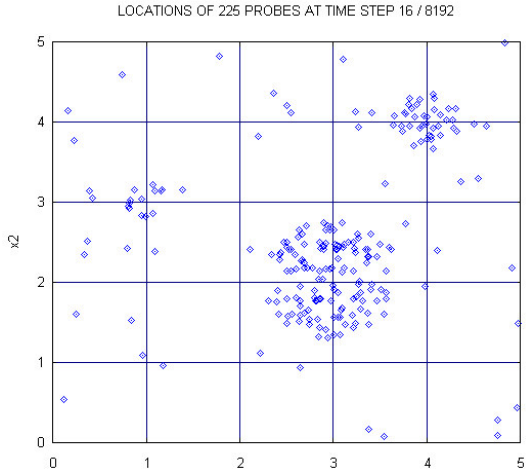

Figure 29a. Three cylinders probes in x_1-x_2 plane at Step 16.

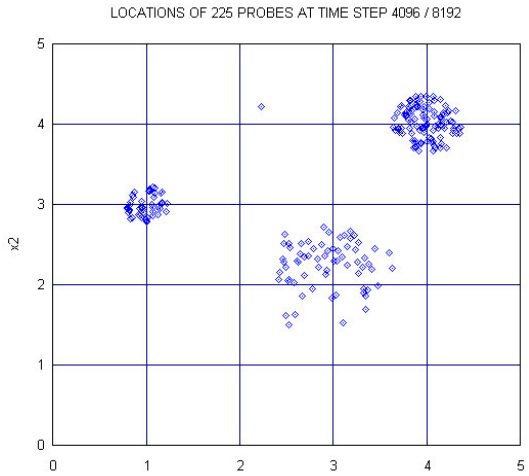

Figure 29b. Three cylinders probes in x_1-x_2 plane at Step 4096.

at the actual global maxima. CFO successfully located the maxima of this highly discontinuous objective function with nearby local maxima of comparable amplitude, clustering probes throughout the regions containing the maxima. This is a further example of CFO's ability to cluster probes around dispersed maxima.

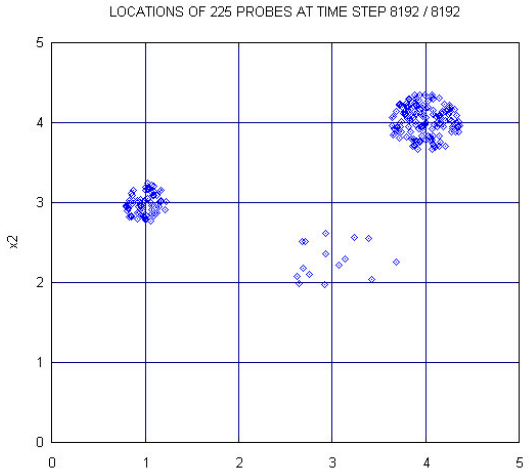

Figure 29c. Three cylinders probes in x_1-x_2 plane at Step 8192.

10. AVOIDING TRAPPING AT LOCAL MAXIMA

The benchmark runs described in Section 8 suggest that CFO can become trapped at a local maximum. This section discusses a possible solution to this problem. The Step function is revisited in two dimensions in order to illustrate an *adaptive* approach that may avoid trapping.

Two CFO runs were made with $G = 2$, $\alpha = 2, \beta = 2$ and a uniform on-axis distribution of initial probes. Twenty probes were uniformly distributed on the x_1 and x_2 axes as shown in Fig. 30. The global maximum of zero was offset from the origin to the point (75.123, 75.123) in order to avoid any bias resulting from the initial probe distribution as discussed earlier (see Benchmark 5 in the Appendix). Table 10 shows the CFO run parameters. The first run was made with $N_t = 250$ and the second with $N_t = 4$.

For the longer CFO run, Fig. 31(a) shows that all but two of the probes have converged at step 250. The reason for making the shorter

Table 10. 2D step function run parameters.

Run	N_p	N_t	N_{eval}	***Init. Dist.***
CFO-1	40	250	10,000	On-Axes
CFO-2	40	4	160	On-Axes

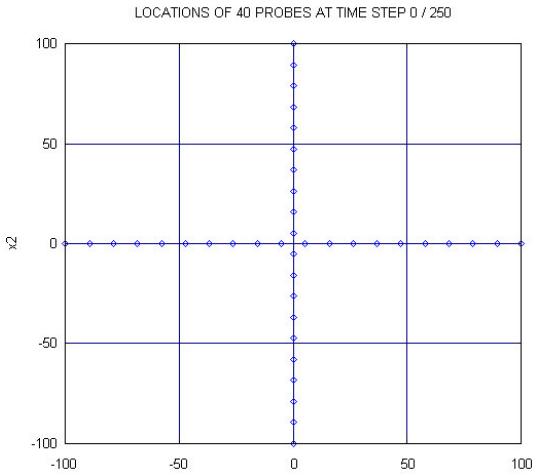

Figure 30. 2-D step function initial probe distribution in x_1-x_2 plane.

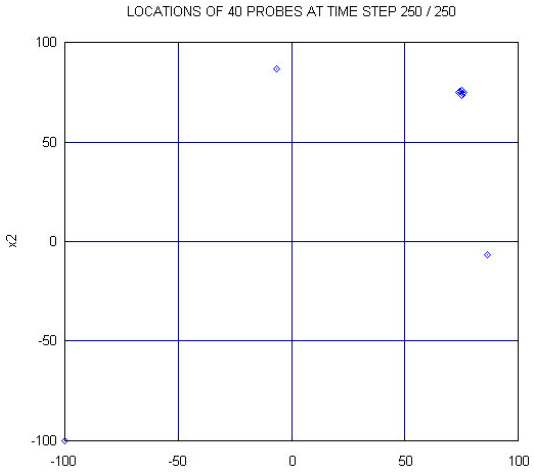

Figure 31a. CFO probes in x_1-x_2 plane at Step 250.

CFO run is to illustrate that many probes have clustered near the maximum as early as step 4 as shown in Fig. 31(b). In fact, CFO returns the same maximum value at step 4 as it does at step 250, the only difference being slightly different x_1 and x_2 coordinates. The best fitnesses at the end of the longer CFO run are tabulated in Table 11, and results for the two runs appear in Table 8. For the long run, instead

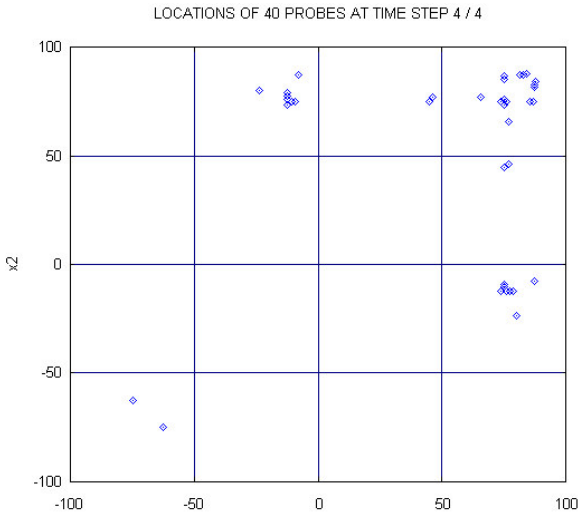

Figure 31b. 2-D step probes in x_1-x_2 plane at Step 4.

of converging on the global maximum of zero at (75.123,75.123), CFO converged on a maximum of -1 at (74.315,75.4982). This is apparently the result of trapping at a local maximum.

Table 11. 2D step function maxima computed by CFO at Step 250.

Maximum Fitness	x_1	x_2
-1	74.315	75.4982
-6868	-6.66667	86.6667
-61250	-100	-100

Table 12. Summary of CFO/PSO results for the 2D step function.

Run	***Max Fitness***	x_1	x_2	N_{eval}	N_p	N_t
CFO-1	-1	75	76.3158	160	40	4
CFO-2	-1	74.315	74.4982	10,000	40	250

Figs. 32(a) and (b), respectively, are CFO's best fitness and average distance curves. The very rapid increase in fitness to -1 at step 4 is evident, and the best value does not change after that point because the algorithm apparently has been trapped. The probe

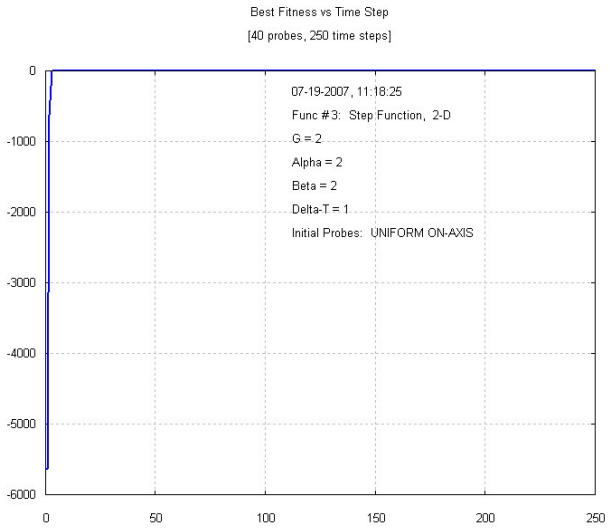

Figure 32a. 2-D step function best fitness vs. time step.

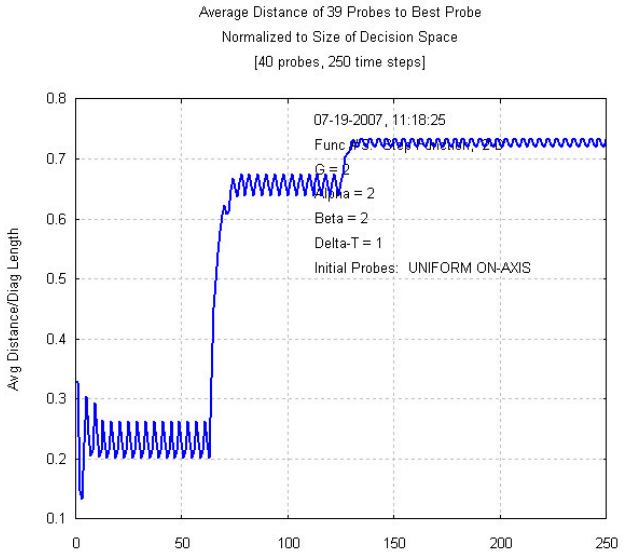

Figure 32b. 2-D step function average distance to best probe vs. time step.

distance plot shows a very interesting behavior. The minimum value occurs at step 3, the step before the fitness climbs to −1. Thereafter the average distance oscillates about values that plateau and then increase in a step-like fashion. A very long CFO run ($N_t = 30,000$, results not shown) reveals that the curve remains flat and continues to oscillate around the value of 0.728 as seen in Fig. 32(b).

CFO appears to converge very rapidly to the *vicinity* of a function's global maximum. In every case in section 8, for example, the maximum CFO fitness began to plateau no later than step 20 (at step 15 the Rosenbrock run had converged on $x_i = 25.3125$ for 28 of the 30 coordinates). This is a very rapid convergence on an approximation of the global maximum, and it seems to be a characteristic behavior of CFO when the initial run parameters are properly chosen.

If indeed this is a persistent CFO characteristic, as it seems to be in view of the quite different test functions, then an *adaptive* CFO implementation that shrinks the decision space around an approximate maximum located after only a few time steps may avoid local trapping and converge very quickly. As an example, if the 2-D Step function decision space is truncated from $-100 \leq x_1,\ x_2 \leq 100$ to $70 \leq x_1,\ x_2 \leq 95$ (based on the convergence seen by step 4) and CFO is re-run with 40 = p N probes distributed uniformly along the lines $x_1, x_2 = 82.5$ (dividing the region into four quadrants), then CFO locates the global maximum of zero at (75.338,75.5149) in only 3 steps. Assuming 20 steps to start, this specific adaptation (all other CFO parameters unchanged) avoids local trapping and locates the 2-D Step functions global maximum in 23 steps with a total of 920 functions evaluations.

This example shows that a properly implemented adaptive CFO algorithm should be able to avoid becoming trapped in local maxima. Another possible approach might be to deterministically or randomly redistribute some fraction of the probes when saturation becomes evident in the evolution of either the best fitness or the average distance curves.

11. CONCLUSION

This paper introduces Central Force Optimization (CFO) as a new optimization metaheuristic. Preliminary analysis suggests that CFO is an effective *deterministic* search algorithm for solving multidimensional optimization problems. CFO's effectiveness has been illustrated by designing a 3-element equalizer for the canonical Fano load, and by synthesizing a 32-element linear array with three specific design criteria. In both cases CFO produced results that were as good or better than those produced by several other optimization algorithms

applied to the same problems.

While initial CFO testing has been very encouraging, the fact remains that Central Force Optimization at this point is a *metaheuristic* as defined in the introduction. Many improvements in designing CFO algorithms undoubtedly are possible, and to be sure, unresolved questions remain, among them: (a) how to choose the initial probe distribution; (b) how to define "mass" in CFO-space; (c) how to choose the CFO parameters G, α, β, N_p and N_t; and (d) how to define termination criteria. Questions relating to the choice of run parameters, however, are not at all unique to CFO. As the linear array example shows, the same questions apply to setting up ACO runs, and in the linear array case the answers were entirely empirical.

Hopefully these issues will be addressed by future research spurred on by this description of the CFO concept. CFO's status at this time is much like that of ACO when it was first introduced in 1996. With the attention of researchers far better qualified than the author, CFO perhaps will achieve a similar measure of success as an optimization algorithm. The use of evolutionary algorithms for antenna design and for solving other problems in applied electromagnetics has grown dramatically through the years. It is an active research area, recent representative examples being found in [17–29]. Perhaps CFO could be applied to some of those problems in order to compare its performance to the wide variety of EA's that have successfully been used.

CFO demonstration programs (executables and source code, including the Fano load and array synthesis programs) and other materials using many different test functions [30] are available upon request. Interested persons should email the author at rf2@ieee.org to request copies.

ACKNOWLEDGMENT

The author wishes to thank the PIER Editorial Board and the Reviewers for their efforts and many helpful suggestions. This paper has been improved considerably as a result.

APPENDIX A. TEST FUNCTIONS

This appendix contains the definitions of the benchmark functions discussed in Section 8. Each function $f(x) = f(x_1, x_2, \ldots, x_{N_d})$ is defined on an N_d-dimensional decision space, where $N_d = 2$ or 30 as indicated. In several cases the maxima are offset from the locations described in [14] and [15] in order to avoid possible bias resulting from

proximity of the initial probes to the maxima. The functions appear in the order listed in Tables 4, 5 and 6 above.

Benchmark 1: Schwefel Problem 2.26

The generalized Schwefel Problem 2.26 is a 30-D function defined as

$$f(x) = \sum_{i=1}^{30} x_i \sin\sqrt{|x_i|}, \quad -500 \le x_i \le 500$$

Its maximum value is 12569.5 occurring at $x_i = 420.9687$, $i = 1, \ldots, 30$. As an indicator of its complexity, the 2-D Schwefel is plotted below in Fig. A1. This function is continuous and extremely multimodal.

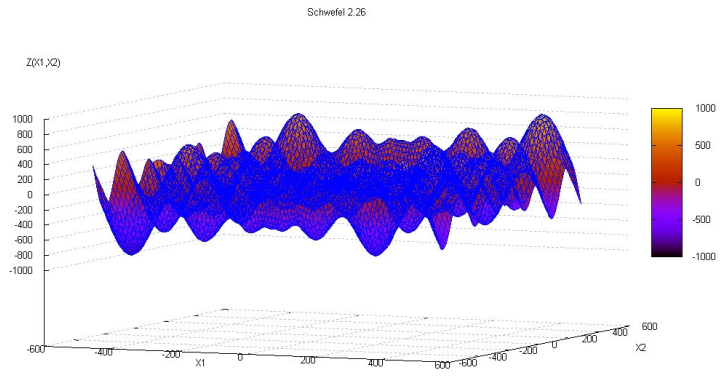

Figure A1. 2-D schwefel 2.26.

Benchmark 2: Modified Griewank

The modified Griewank function is a 30-D function defined as

$$f(x) = -\frac{1}{4000}\sum_{i=1}^{30}(x_i - x_o)^2 + \prod_{i=1}^{30}\cos\left\{\frac{(x_i - x_o)}{\sqrt{i}}\right\} - 1, \ -600 \le x_i \le 600.$$

Its maximum value of zero is offset from the origin and occurs at $f(x_o, x_o, \ldots, x_o)$ where $x_o = 75.123$. The Griewank is continuous and very highly multimodal, as shown in the 2-D plot in Fig. A2.

Benchmark 3: Modified Ackley's Function

The modified version of the 30-D Ackley function has its maximum

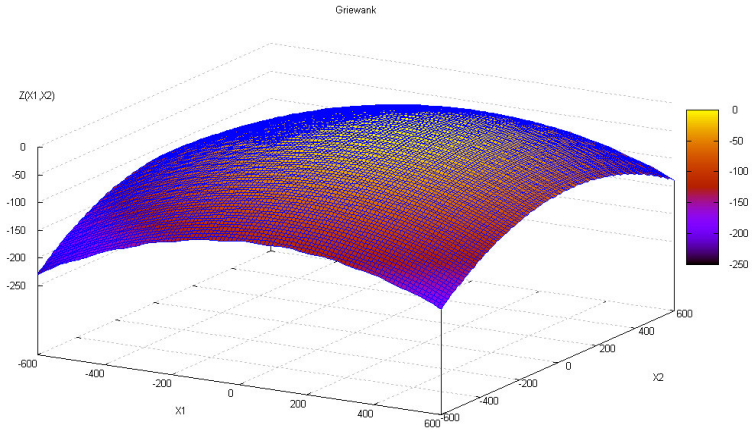

Figure A2a. 2-D Griewank over entire domain.

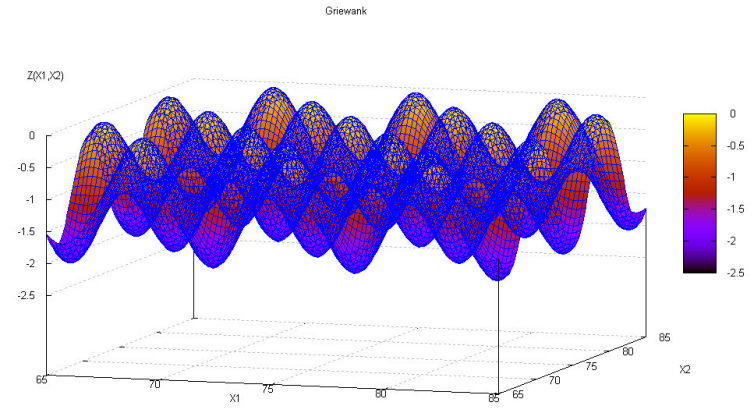

Figure A2b. 2-D Griewank showing detail in vicinity of maximum.

offset from the origin and is defined by

$$f(x) \;=\; 20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30}(x_i - x_o)^2}\right) + \exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos\{2\pi(x_i - x_o)\}\right) - 20 - e,\;\; -32 \le x_i \le 32$$

The maximum value is 0 at $(x_o, \ldots, x_o)$, where $x_o = 4.321$. A plot of this function in two- dimensions appears in Fig. A3.

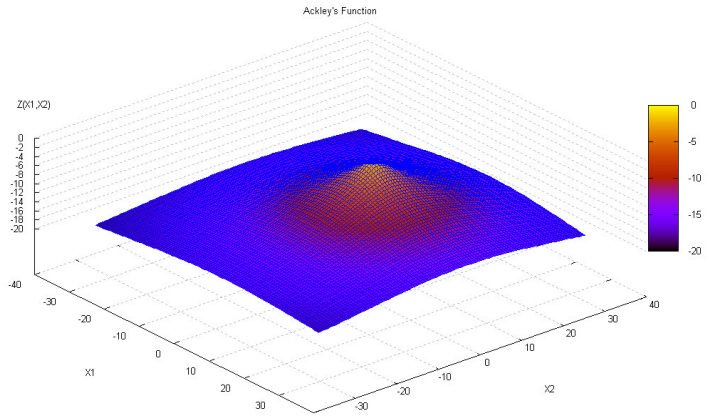

Figure A3. 2-D modified Ackley function.

Benchmark 4: Modified Rastrigin

The modified Rastrigin function is defined as

$$f(x) = -\sum_{i=1}^{30} \left[y_i^2 - 10\cos(2\pi y_i) + 10 \right], \; y_i = x_i - x_o, \; -5.12 \leq x_i \leq 5.12$$

The maximum value of zero occurs at $f(x_o, x_o, \ldots, x_o)$ where $x_o = 1.123$. It possesses many local maxima that are close in value. The 2-D version of this function appears in Fig. A4.

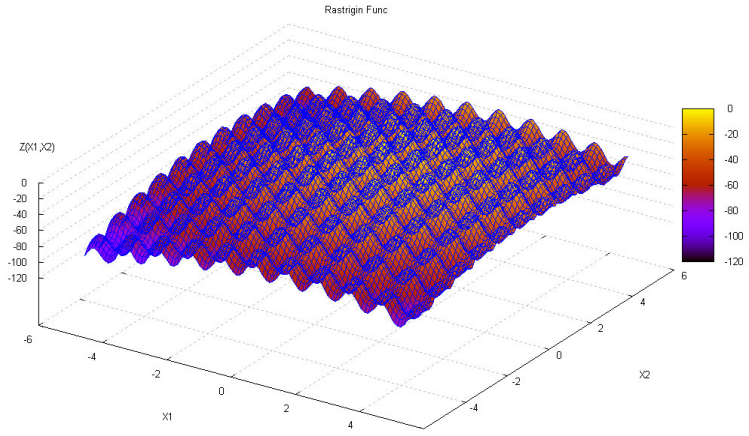

Figure A4. 2-D modified rastrigin function.

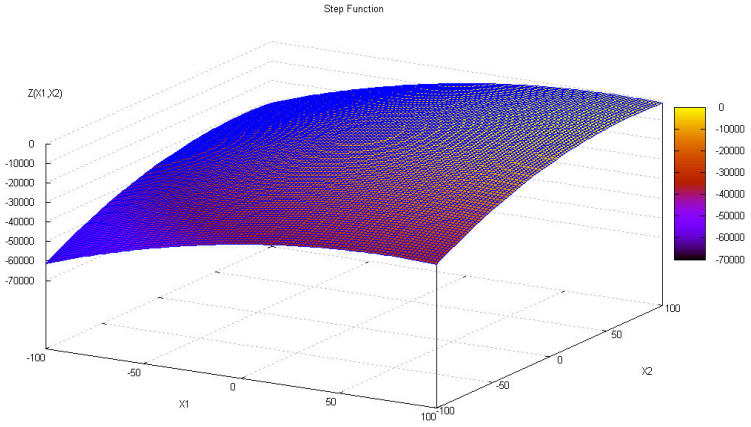

Figure A5a. 2-D step function on test domain.

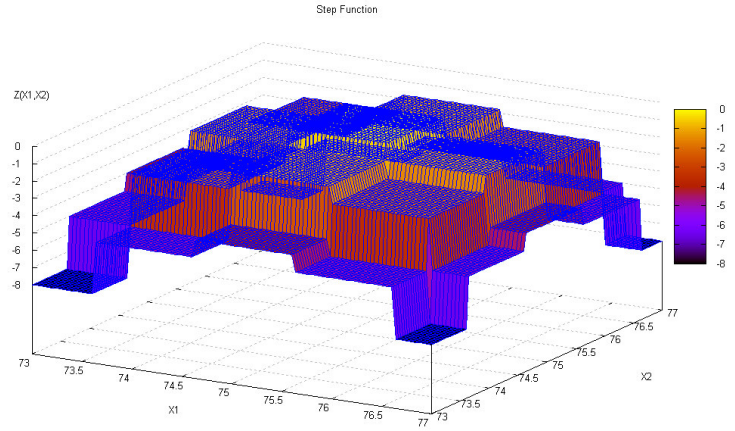

Figure A5b. Expanded view of 2-D step function in vicinity of maximum.

Benchmark 5: Modified Step

The modified step is defined as

$$f(x) = -\sum_{i=1}^{30} \left(\lfloor x_i - x_o + 0.5 \rfloor\right)^2, \ -100 \leq x_i \leq 100$$

where the offset shifts the maximum from the origin to $(x_o, \ldots, x_o)$, $x_o = 75.123$. Fig. A5a plots the 2-D version of this function over its domain of definition for this test case. The step's

value varies over a very wide range, from a maximum of zero at (75.123,75.123) to a minimum below $-60,000$. Fig. A5b shows an expanded view of the step's structure in the vicinity of the maximum.

Benchmark 6: Modified Sphere Model

The modified sphere model is defined as

$$f(x) = -\sum_{i=1}^{30} (x_i - x_o)^2, \quad -100 \le x_i \le 100$$

where the maximum value of zero occurs at $f(x_o, \ldots, x_o),\ x_o = 75.123$. The two-dimensional version of the Sphere Model appears in Fig. A6.

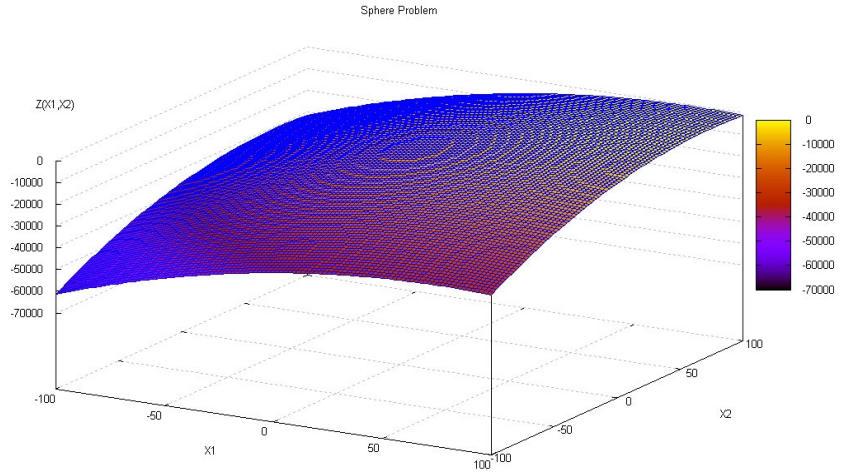

Figure A6. Modified sphere model.

Benchmark 7: Rosenbrock's Function

$$f(x) = -\sum_{i=1}^{29} \left[100 \left\{ (x_{i+1} - x_o) - (x_i - x_o)^2 \right\}^2 + \{(x_i - x_o) - 1\}^2 \right]$$

where $x_o = 25.123$. This function has a maximum value of zero at $x_i = 26.123,\ i = 1, \ldots, 30$. The 2-D version is plotted in Fig. A7.

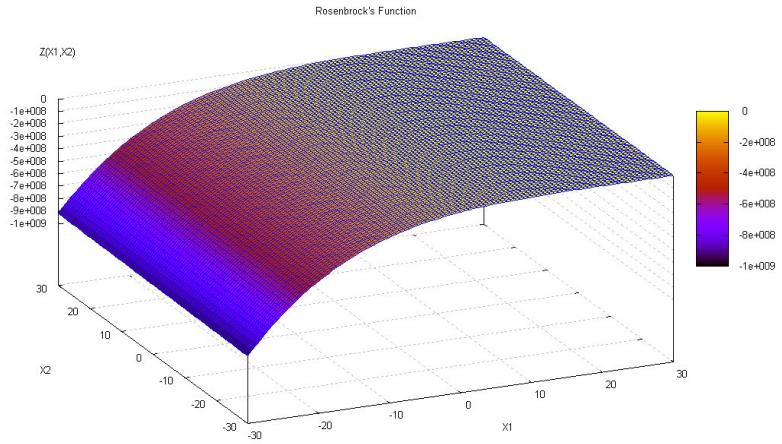

Figure A7. 2-D Rosenbrock's function.

Benchmark 8. Colville Function

The modified Colville function is defined as

$$\begin{aligned} f(x_1,x_2,x_3,x_4) &= -100\cdot\left(x_2'-x_1'^2\right)^2-(1-x_1')^2 \\ &= -90\cdot\left(x_4'-x_3'^2\right)^2-(1-x_3')^2 \\ &= -10.1\cdot\left((x_2'-1)^2-(x_4'-1)^2\right) \\ &= -19.8\cdot(x_2'-1)\cdot(x_4'-1) \end{aligned}$$

where $x_i' = x_i - 7.123,\ i = 1,\ldots,4,\ -10 \le x_i \le 10.$

It has a maximum value of zero at $x_i = 8.123,\ i = 1,\ldots,4.$ By definition, the Colville function is four-dimensional and consequently cannot be plotted.

Benchmark 9: Six-Hump Camel-Back

The 2-D Six-Hump Camel-Back function is defined as

$$\begin{aligned} f(x) &= -4(x_1-1)^2+2.1(x_1-1)^4-\frac{1}{3}(x_1-1)^6 \\ &\quad -(x_1-1)(x_2-1)+4(x_2-1)^2-4(x_2-1)^4, \quad -5\le x_i\le 5 \end{aligned}$$

The maximum value is 1.0316285 at (1.08983,0.2874) and (0.91017,1.7126). The Camel-Back appears in Fig. A8.

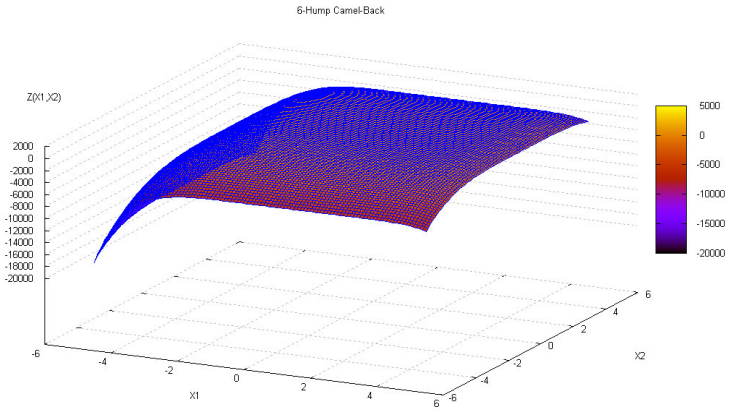

Figure A8. 2-D six-hump Camel-back function.

Benchmark 10: Branin Function

The Branin function is defined as

$$f(x) = -\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 - 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 - 10, \qquad -5 \le x_i \le 15$$

The standard search interval of $-5 \le x_1 \le 10,\ 0 \le x_2 \le 15$ has been expanded so that each coordinate has the same minimum and maximum values. The Branin's maximum value is -0.398 occurring at the three points:

$(-3.142, 12.275)$
$(3.142, 2.275)$
$(9.425, 2.425)$

The 2-D Branin function is plotted in Fig. A9.

Benchmark 11: Shekel's Inverted Foxholes

Shekel's Inverted Foxholes is a 2-dimensional test function defined by

$$f(x) = -\left[\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right]^{-1}, \quad -65.536 \le x_i \le 65.536$$

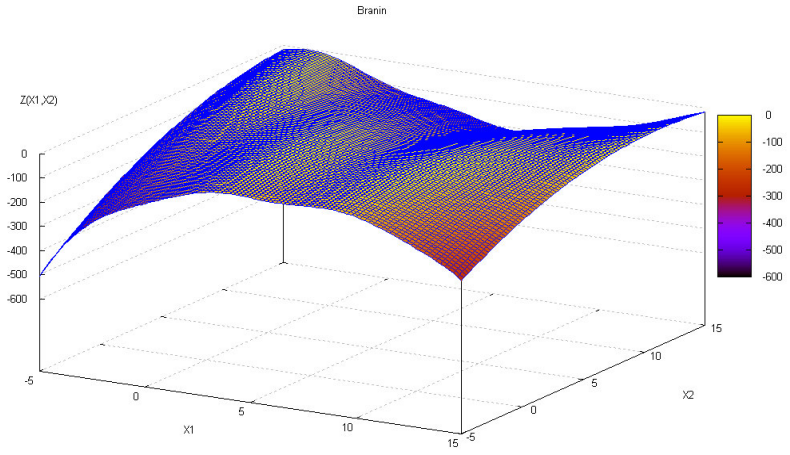

Figure A9. 2-D Branin function.

where

$$(a_{ij}) = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \cdots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \cdots & 32 & 32 & 32 \end{pmatrix}$$

Its maximum value is ≈ -1 occurring at $(-32, -32)$. The 2D Shekel's "Foxholes" appears in Fig. 16 above.

Benchmark 12. Modified Keane's Bump

The modified Keane's Bump is a constrained objective function defined as

$$f(x_1, x_2) = \left\{ \begin{array}{l} 0 \text{ for } x_1 + x_2 \geq 15 \text{ or } x_1 x_2 \leq 0.75; \text{ otherwise} \\ \dfrac{\cos^4(x_1) + \cos^4(x_2) - 2\cos^2(x_1)\cos^2(x_2)}{\sqrt{x_1^2 + 2x_2^2}} \end{array} \right\}$$

$$-5 \leq x_1,\ x_2 \leq 5$$

The precise location(s) and value(s) of the maxima are unknown. Various views of Keane's Bump appear in Fig. 17.

REFERENCES

1. Kennedy, J. and R. Eberhart, "Particle swarm optimization," *Proc. IEEE Conf. On Neural Networks*, Vol. 4, 1942–1948, Nov./Dec. 1995.

2. Special Issue on Particle Swarm Optimization, *IEEE Trans. Evol. Comp.*, Vol. 8, No. 3, Jun. 2004.
3. Dorigo, M., M. Birattari, and T. Stützle, "Ant colony optimization," *IEEE Comp. Intell. Mag.*, Vol. 1, No. 4, 28–39, Nov. 2006.
4. Special Section on Ant Colony Optimization, *IEEE Trans. Evol. Comp.*, Vol. 6, No. 4, Aug. 2002.
5. Marion, J., *Classical Dynamics of Particles and Systems*, 2nd edition, §2.7, Harcourt Brace Jovanovich, New York, NY, 1970.
6. Brand, L., *Differential and Difference Equations*, §56, John Wiley & Sons, Inc., New York, NY, 1966.
7. Power Basic, Inc., http://www.powerbasic.com
8. Fano, F., "Theoretical limitations on the broadband matching of arbitrary impedances," *J. Franklin Inst.*, Vol. 249, 57–83, Jan. 1950; 139–154, Feb. 1950.
9. Rodriguez, J., I. García-Tuñón, J. Taboada, and F. Basteiro, "Broadband HF antenna matching network design using a real-coded genetic algorithm," *IEEE Trans. Ant. Propag.*, Vol. 55, No. 3, 611–618, Mar. 2007.
10. Carlin, H., "A new approach to gain-bandwidth problems," *IEEE Trans. Circuits Syst.*, Vol. CAS-24, No. 4, 170–175, Apr. 1977.
11. Dedieu, H., C. Dehollain, J. Neirynck, and G. Rhodes, "A new method for solving broadband matching problems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, Vol. 41, No. 9, 561–570, Sep. 1994.
12. Rajo-Iglesias, E. and O. Quevedo-Teruel, "Linear array synthesis using an ant-colony-optimization-based algorithm," *IEEE Ant. Prop. Mag.*, Vol. 49, No. 2, 70–79, Apr. 2007.
13. Balanis, C., *Antenna Theory: Analysis and Design*, §6.4, Harper & Row, Publishers, New York, NY, 1982.
14. Yao, X., Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evolutionary Computation*, Vol. 3, No. 2, 82–102, Jul. 1999.
15. Emmerich, M. T. M., K. C. Giannakoglou, and B. Naujoks, "Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Trans. Evolutionary Computation*, Vol. 10, No. 4, 421–439 [see Appendix I(E)], Aug. 2006.
16. Doo-Hyun and O. Se-Young, "A new mutation rule for evolutionary programming motivated from backpropagation learning," *IEEE Trans. Evolutionary Computation*, Vol. 4, No. 2,

188–190, Jul. 2000.

17. Chen, T., Y. Dong, Y. Jiao, and F. S. Zhang, "Synthesis of circular antenna array using crossed particle swarm optimization algorithm," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 13, 1785–1795, 2006.
18. Lee, K.-C. and J.-Y. Jhang, "Application of particle swarm algorithm to the optimization of unequally spaced antenna arrays," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 14, 2001–2012, 2006.
19. Lu, Y.-Q. and J.-Y. Li, "Optimization of broadband top-loaded antenna using micro-genetic algorithm," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 6, 793–801, 2006.
20. Ayestarán, R. Laviada, and F. Las-Heras, "Synthesis of passive-dipole arrays with a genetic-neural hybrid method," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 15, 2123–2135, 2006.
21. Mitilineos, S., S. Thomopoulos, and C. Capsalis, "Genetic design of dual-band, switched-beam dipole arrays, with elements failure correction, retaining constant excitation coefficients," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 14, 1925–1942, 2006.
22. Tian, Y. and J. Qian, "Ultraconveniently finding multiple solutions of complex transcendental equations based on genetic algorithm," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 4, 475–488, 2006.
23. Capozzoli, A. and G. D'Elia, "Global optimization and antennas synthesis and diagnosis, Part one: concepts, tools, strategies and performances," *Progress In Electromagnetics Research*, PIER 56, 195–232, 2006.
24. Capozzoli, A. and G. D'Elia, "Global optimization and antennas synthesis and diagnosis, Part two: applications to advanced reflector antennas synthesis and diagnosis techniques," *Progress In Electromagnetics Research*, PIER 56, 233–261, 2006
25. Chen, X., X., T. Grzegorczyk, and J. A. Kong, "Optimization approach to the retrieval of the constitutive parameters of a slab of general bianisotropic medium," *Progress In Electromagnetics Research*, PIER 60, 1–18, 2006.
26. Donelli, M., S. Caorsi, F. De Natale, M. Pastorino, and A. Massa, "Linear antenna synthesis with a hybrid genetic algorithm," *Progress In Electromagnetics Research*, PIER 49, 1–22, 2004.
27. Inman, M., J. Earwood, A. Elsherbeni, and C. Smith, "Bayesian optimization techniques for antenna design," *Progress In Electromagnetics Research*, PIER 49, 71–86, 2004.

28. Sijher, T. and A. Kishk, "Antenna modeling by infinitesimal dipoles using genetic algorithms," *Progress In Electromagnetics Research*, PIER 52, 225–254, 2005.
29. Misra, I. S., R. Chakrabarty, and B. Mangaraj, "Design, analysis and optimization of V-dipole and its three-element Yagi-Uda array," *Progress In Electromagnetics Research*, PIER 66, 137–156, 2006.
30. Formato, R., Reg. nos. TX 6-459-271, TX 6-461-552, TX 6-468-062, TX 6-464-965, TX 6-522-082, TX 6-540-042, 16 Nov. 2006, et seq., Copyright Office, U.S. Library of Congress, Washington, DC (others pending).