Improving Decision Space Exploration of Central Force Optimization Using Negative Gravity

Richard A. Formato
Consulting Engineer & Registered Patent Attorney

1. Introduction

Central Force Optimization (CFO) [1,2,3] is a global search and optimization (GSO) metaheuristic based on gravitational kinematics, the motions of real masses whose trajectories are controlled by real gravity. CFO searches a decision space (DS) for the *maxima* of an objective function whose value is its 'fitness' and whose topology on DS is unknown or unknowable

The CFO metaphor flies 'probes' that sample DS and converge on local extrema. It is inherently *deterministic*, that is, every run with the same setup yields precisely the same results. CFO's exploitation of discovered extrema (quickly converging on a local maximum) is quite good [4], but its exploration (sampling other regions of DS) can be inhibited by that very attribute because probes that have coalesced are no longer available to search elsewhere. This dichotomy plagues most GSO's. To quote " "*Exploration* and *exploitation* are the two cornerstones of problem solving by search." For more than a decade, Eiben and Schippers' advocacy for balancing between these two *antagonistic cornerstones* still greatly influences the research directions of evolutionary algorithms (EAs)..." [5, emphasis added] and additionally discussed in [6-8]. Like all GSO algorithms CFO is subject to the inescapable tension between exploration (adequately sampling DS) and exploitation (quickly converging on global maxima). Using *negative gravity* may mitigate this effect thereby improving CFO's exploration. Injecting a small amount hopefully discovers new maxima that otherwise would be missed. They may include global maxima or simply other extrema with similar fitness values. This note looks into this possibility with some examples.

CFO has been used both for *design* and for *optimization* (D/O) to solve 'real-world' problems, often in an engineering setting. *Design* refers to meeting a specified minimum performance, whereas *Optimization* refers to determining the greatest objective function value(s) and location(s) in DS (the 'best' fitnesses and coordinates). Importantly, running any GSO algorithm against 'benchmark' functions is altogether different from D/O on typical real world problems. Benchmark functions are known analytically *a priori*, as are their maxima value(s) and their location(s) in DS. Solving benchmarks is quite different from the usual 'real world' case of starting without any objective function at all. Therefore a key element of solving the problem is *formulating* a suitable objective function. Accomplishing this can be daunting, especially in complex cases, and using a deterministic algorithm like CFO can make a big difference because there is no inherent randomness in that approach. This note looks into this issue as well..

Most GSO's are stochastic in nature, so that even with the same setup successive runs produce different results. In order to evaluate the effect of a change in the objective function, even a slight change, say, using a different coefficient in some term, requires multiple runs of a stochastic GSO, often tens or hundreds in order to build sufficient statistics to reliably measure the effect of that new coefficient. In stark contrast, CFO requires only two runs, one with the original objective function and a second with the modified version because *all* changes in the runs' outputs are a consequence *only* of modifying the objective function, not random changes because the GSO is inherently stochastic.

CFO has been effectively used with excellent results to solve problems in a wide range of disparate disciplines, among them, as examples: training neural networks [9], patch antenna synthesis [10], power system state space pruning [11], antenna design [12-19], arid region water distribution [20], UAV flight

path planning [21], bandpass filter design [22], humanoid robot gait [23] ensembles of neural networks [24], satellite image fusion [25], medical image fusion [26], iris recognition [27], and filter design [28].

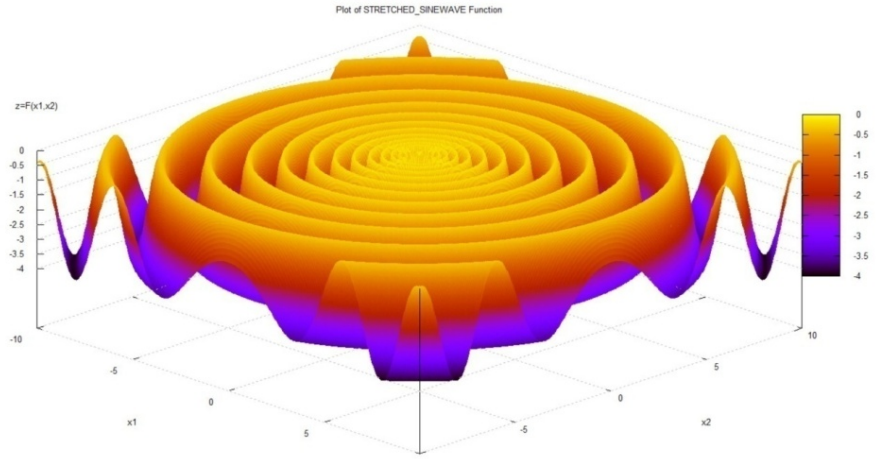
2. Effect of Negative Gravity

With respect to the sign of CFO's gravity, + or –, the question is, Does making it negative benefit CFO's performance, and if so, why, or does it impede it? Positive gravity causes CFO's probes always to move toward greater fitnesses, never away, and consequently to some degree positive gravity inevitably impedes CFO's exploration. In fact, CFO often converges very quickly [4], a favorable attribute, but not if rapid exploitation is accomplished at the expense of under-sampling DS, which may well be the case. Adding some negative gravity that has probes flying away from each other may address this issue by causing probes that otherwise would coalesce to explore more widely by flying into regions that have been under-sampled or perhaps not sampled at all. The test case reported here shows that a small amount of negative gravity indeed does benefit CFO's performance, ostensibly because it enhances CFO's exploration while retaining the algorithm's ability to exploit already located maxima. Before getting to that, however, a couple of two-dimensional (2D) functions will be used to illustrate the effect of negative gravity. Negative gravity was injected using π–fractions (Appendix A1). π–fractions #0-1,000,001 are downloadable at https://app.box.com/s/qdd8rzrhgaozne0ag1nes9jkm0bj6ark. These data are provided for any interested user and may be distributed without limitation.

2.1 Stretched Sine Wave: The first function illustrating negative gravity is the stretched Sine Wave [29] defined as

$$f(\vec{x}) = \sum_{i=1}^{N_d-1} (x_{i+1}^2 + x_i^2)^{0.25} \bullet \{sin^2[50(x_{i+1}^2 + x_i^2)^{0.1}] + 0.1\}$$

The second function is the Gaussian grid (below). While the Sine Wave is a recognized benchmark, the grid is not. The stretched Sine Wave is plotted below..



Stretched Sine Wave

For both functions a dense π–fraction initial probe distribution (IPD) was used to provide a large number of sampling points in order to visualize how CFO's probe distributions evolved with positive and negative gravity by plotting the probe positions as the run progressed. The gravitational constants (Appendix A2) were G=±2.

Fig. 1 shows the Sine Wave probe evolution at steps 0 (IPD), 2, 4 and 20 (clockwise) with positive gravity. This benchmark's maximum is zero at the origin of the 10x10 decision space, and the probes' convergence on the maximum is visually evident. CFO's positive gravity causes the probes to come together around the maximum. Even though the Sine Wave is circularly symmetric DS is not, and the effect of this asymmetry is evident in the probe distributions that cluster along the DS diagonals. Because of the dense IPD, the CFO run was intentionally short (20 steps) and returned a best fitness of -0.005543 at the point (-0.00288,0.00106).

Fig. 2 shows the Sine Wave probe evolution with negative gravity in the same format as Fig.1. The dramatic effect of G < 0 is quite apparent – CFO's probes are all forced away from the maximum and instead cluster along the DS diagonals in the far corners, again a result of the Sine Wave/DS asymmetry. The negative gravity flew the probes as far apart as possible. At the end of the run, a best fitness of -0.102165 was returned at the point (0.04672,+1.02846)

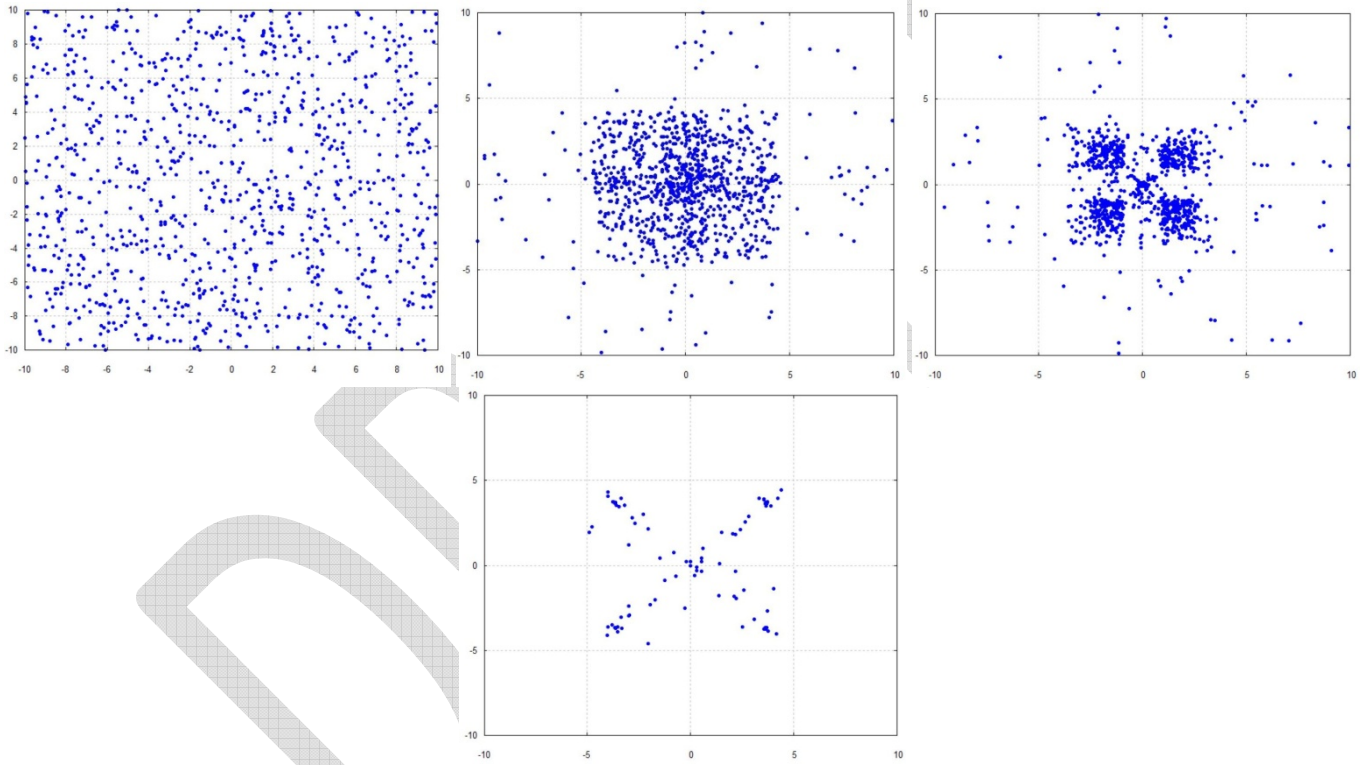


Fig. 1. Probes Locations, Stretched Sine Wave, Positive Gravity, G = +2

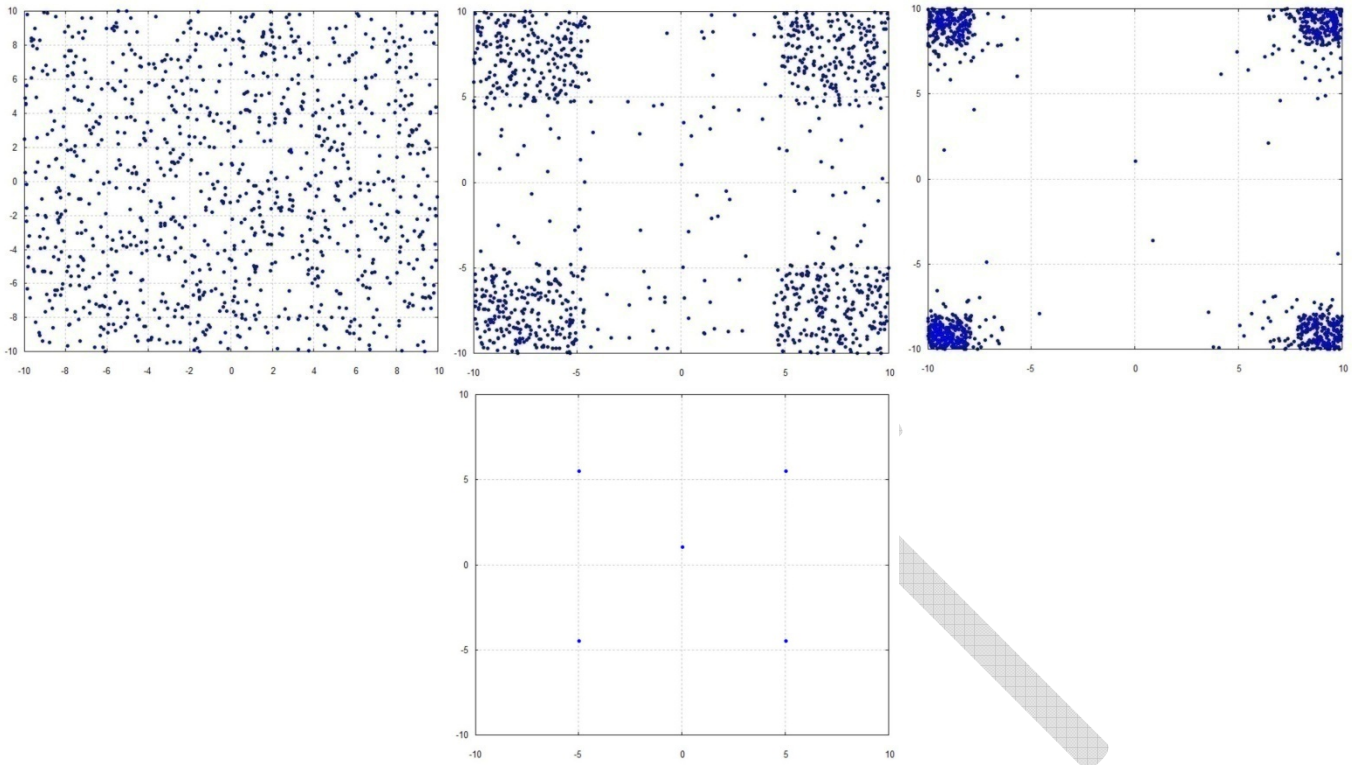


Fig. 2. Probes, Stretched Sine Wave, Negative Gravity, G = –2

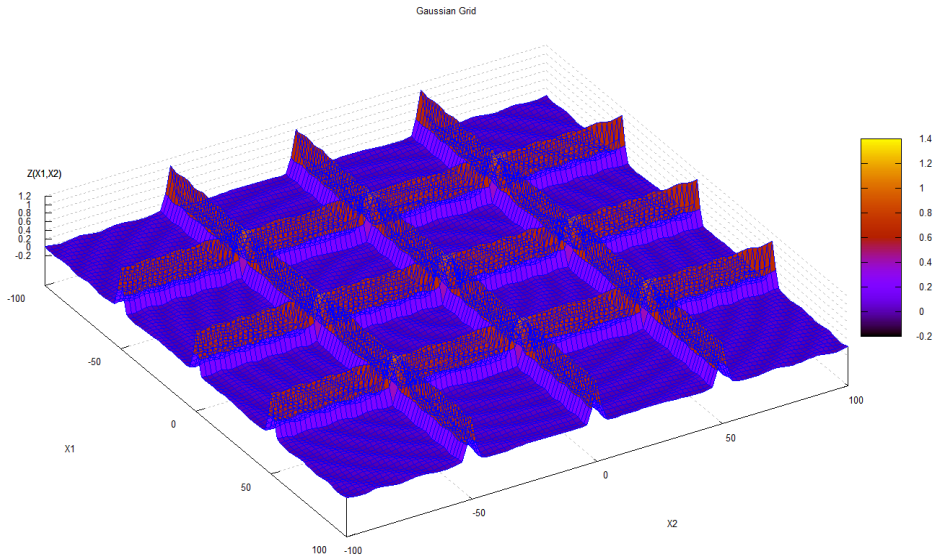
2.2 Gaussian Grid: The gaussian grid provides another compelling example of the validity of CFO's gravitational metaphor and the effect of negative gravity. An attractive force of gravity requires positive mass and a positive gravitational constant (Appendix A2). Because the CFO implementation employed in this note forces the mass to be positive, the effect of negative gravity is readily demonstrated by setting the gravitational constant G to a negative value, in this case $G=-2$, and as with the Sine Wave the result is quite dramatic.

The two-dimensional gaussian grid function is defined as

$$f(x_1,x_2)=MIN\left\{1,\sum_{i=1}^{3}\left(e^{-\frac{[x_1+50(i-2)]^2}{\sigma^2}}+e^{-\frac{[x_2+50(i-2)]^2}{\sigma^2}}\right)\right\}+\frac{1}{40}\sin\left(\frac{2\pi\sqrt{x_1^2+x_2^2}}{10}\right)$$

$$-100\le x_1,x_2\le 100,\ \sigma=2\,.$$

The grid is not a recognized benchmark, but it is useful to demonstrate CFO's behavior in distributing probes with and without negative gravity. The grid is plotted below. Its global maxima of 1.025 lie on the grid lines $x_1=0,\pm 50;\ x_2=0,\pm 50$.



Gaussian Grid

CFO was run again with a dense IPD, in this case a uniform grid of probes, in order to visualize the probes' evolution as the run progresses. Fig 3 shows the probe distribution at steps 24 and 50 with positive gravity, $G=+2$. At step 24 the grid's structure is quite evident, and by step 50 it is fully resolved. Interestingly, the grid lines passing through the origin do not contain probes for $|x_{1,2}|>50$. The likely reason is that probes on these segments were attracted to the large probe concentration near the origin. This thinning effect also seems to be occurring at step 50 on the segments $-50\le x_1\le 50$, $x_2=0$ and $-50\le x_2\le 50$, $x_1=0$ where the probe density near the segment center is lower than at the ends.

When the gravity is negative, $G=-2$, the probe distribution at step 50 (end of run) is shown in Fig. 4, in stark contrast to the positive gravity case, with $G=+2$. Instead of clustering along the grid lines, the probes are symmetrically forced away from maxima to the very edges of the decision space. With negative gravity CFO's probes fly away from each other instead of clustering near the grid's maxima. As an interesting corollary, the grid's landscape contains a continuum of local maxima, but in decision spaces like this, that is, DS's containing an uncountable or a very large number of maxima, most GSO's converge on only one. In many real-world problems, however, that maximum may not actually be the "best". In real-world D/O, especially engineering design, identical or nearly identical fitnesses do not necessarily correspond to fungible designs. As a practical matter, of many solutions with the same or nearly the same fitness, usually one is better than the others for reasons that may not be quantifiable in an objective function. For example, one design may be less expensive, or easier to fabricate, or to maintain, or to distribute, and so on with regard to any number of ancillary considerations that are not or cannot be reflected in an objective function. In cases where there are many indistinguishable or nearly indistinguishable maxima, a deterministic GSO like Central Force Optimization, with positive gravity and a small amount of negative gravity may be useful in exploring DS as a pre-processor to aid in locating many maxima, not just one, thus broadening the range of solutions.

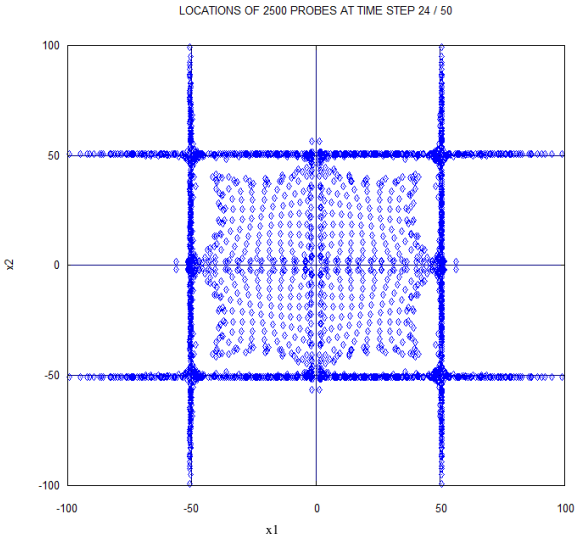


Fig. 3(a) Step 24 with $G = +2$

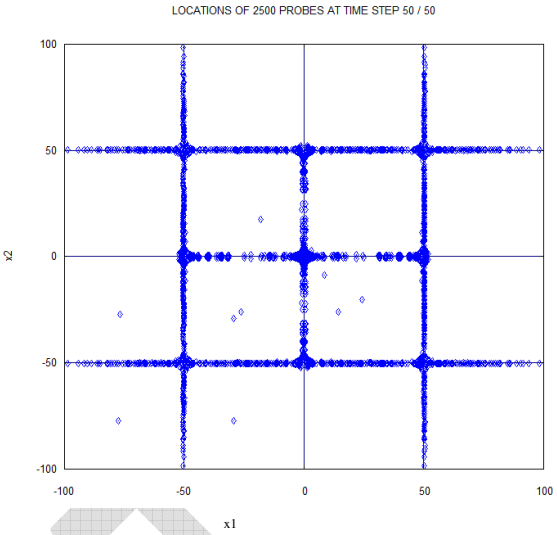


Fig. 3(b) Step 50 with $G = +2$

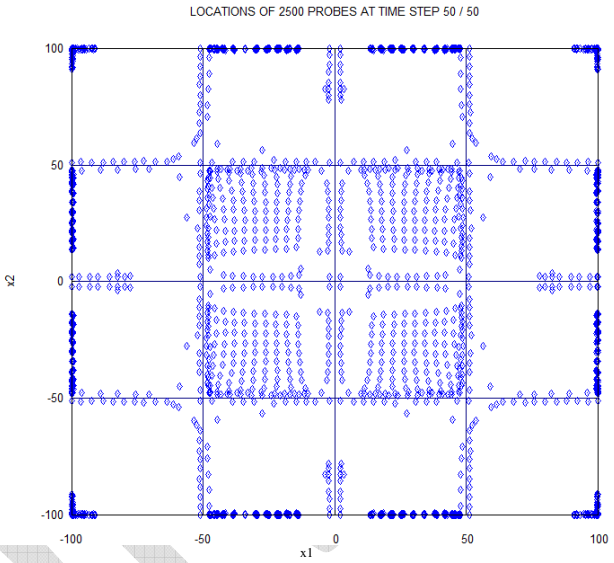


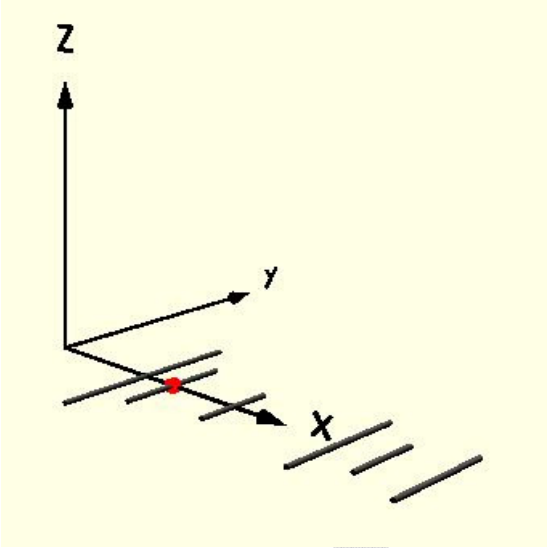
Fig. 4. Step 50 with $G = -2$

3. Example of Negative Gravity Benefit: A 'Real-World' Problem

The beneficial effects of injecting a small amount of negative gravity in CFO were discussed in considerable detail in some previously published papers [30, 31]. Highlights from that work are presented here to supplement the discussion so far. While the example used here is drawn from Yagi antenna design, the basic idea of adding some negative gravity to CFO applies to any CFO problem. The reader is not expected or required to have any background in antennas or electromagnetics and should feel free to peruse this section quickly if desired.

The structure under consideration is the six-element Yagi-Uda array shown below. Arrays of this type are used across the entire radio spectrum and range in size from sub-centimeter to tens of meters. The antenna comprises six parallel dipole elements mounted on a common axis ('boom'). One element is connected to the radio transmitter/receiver ('driven') while the others are not ('parasitic'). The antenna's performance over a range of frequencies ('bandwidth', BW) is measured primarily by how much energy it

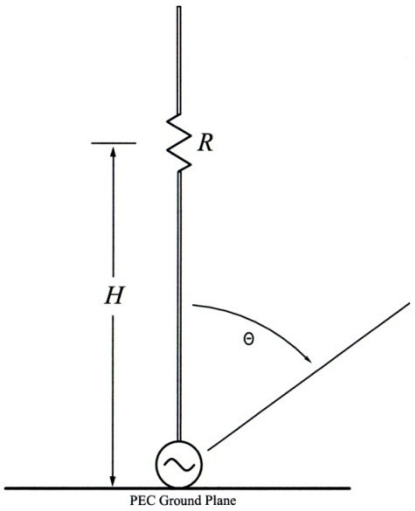
radiates in specific directions ('gain') and how well it accepts power from a radio transmitter ('standing wave ratio', SWR).



The problem here is to maximize an objective (fitness) function that combines these parameters: antenna gain, SWR and BW. The specific mathematical form of the fitness function is not known *a priori* and must be determined by the designer. There is a limitless number of ways these parameters can be mathematically combined, and not all of them are suitable as measures of how well a particular Yagi design performs. This problem is eighteen-dimensional (diameter/length of each element (12 variables); five spacings along the boom; and 'input impedance' that determines SWR). The very first question is, What mathematical combination of these eighteen variables does the best job of measuring how good a particular design is? Not any easy question!

3.1 Formulating a Fitness Function: As pointed out previously, formulating a suitable fitness function can be quite a difficult, but using a deterministic GSO like CFO makes the job easier. To further illustrate this point, before getting back to the Yagi problem, another antenna example will be used to emphasize how important it is to define a suitable objective function, especially in very complex cases like the Yagi array, and how difficult it can be even in the simplest cases.

The problem here is maximizing the bandwidth of the simplest possible antenna structure, a base-fed monopole (a straight wire) on an infinite perfectly conducting ground plane as shown below. The monopole's bandwidth will be increased by adding a fixed resistor R at a height H. The objective is to determine the best values of R and H. Unlike the Yagi problem, this problem is only two-dimensional so that its fitness function can be visualized. Of course, as is usually the case, the problem statement doesn't come with the required objective function, so the designer must be formulate it from scratch. What is a suitable form? There are many parameters that can be included, among them the monopole input impedance, R_{in}, X_{in}, its radiation efficiency, ε, (which is reduced by adding R), the characteristic impedance, Z_0, and the maximum gain, $G_{\max}$, all across the range of desired frequencies. The algorithm designer is free to define any objective function that works well to measure the monopole's fitness



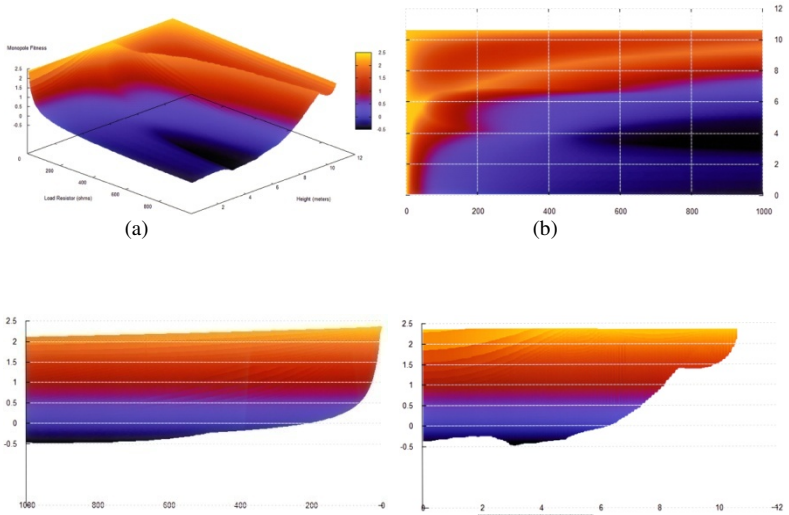
In order to illustrate the difficulty in formulating a suitable objective function, three forms will be considered, each one ostensibly suitable for the problem as defined. The candidates are:

$$f_1(R,H) = \frac{Min(\varepsilon) + Min(G_{\max})}{SWR}$$

$$f_2(R,H,Z_0) = \frac{Min(\varepsilon)}{\left|Z_0 - Max(R_{in})\right| \cdot \left|Max(X_{in})\right|}$$

$$f_3(R,H,Z_0) = \frac{Min(\varepsilon)}{\left|Z_0 - Max(R_{in})\right| \cdot \Delta SWR \cdot [Max(Xin) - Min(Xin)]}$$

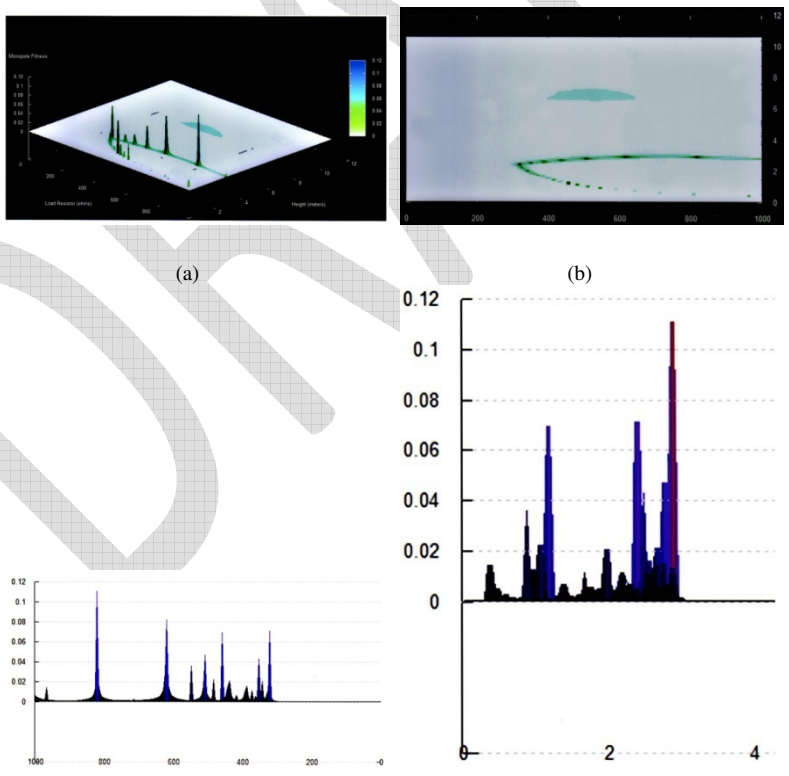
Their 2D landscapes are plotted below. A visual inspection reveals that two of the three are poor choices, one of them extremely poor. Functions 1 and 3 are smoothly varying whereas f_2 is pathologically spikey, even though at first blush its functional form seems to be good for maximizing BW. f_1 has a narrow maximum very close to the DS boundary, which my make it difficult to locate. As pointed out f_2 simply is pathological, and its spikey nature would make it difficult for most GSO's to locate maxima. Of the three candidates, f_3 clearly is the best. Its maximum is well within DS, and it is well defined.



(a) (b)

(c) (d)

Fig. 5. Landscape for Objective Function $f_1(R,H,50)$



(a) (b)

(c) (d)

Fig. 6. Landscape for Objective Function $f_2(R,H,50)$

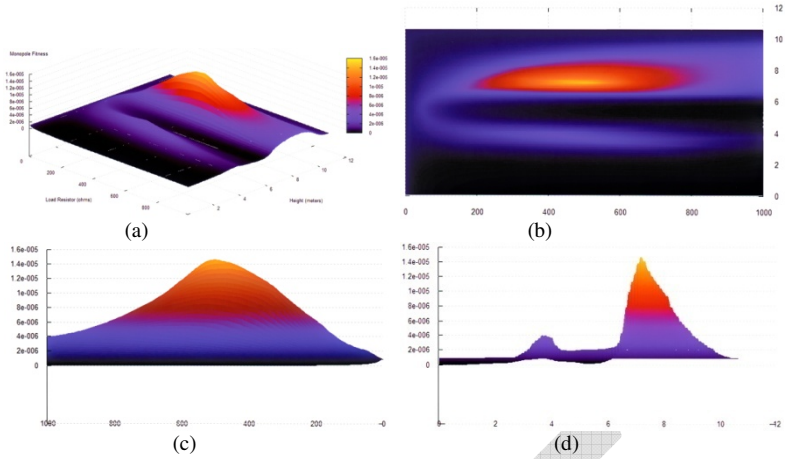


Fig. 7. Landscape for Objective Function $f_3(R, H, 50)$

How could the algorithm designer formulate a suitable objective function if, for example, another component is added, say, an inductor, whose value and location are to be determined the way R and H must be found? Now the problem is 4D instead of 2D. The landscape for candidate objective functions cannot be plotted. How, then, can candidate functions be evaluated? Using a deterministic GSO like CFO makes all the difference because then changes in the results from one run to the next are entirely a result of changes to the fitness function, and nothing else, like randomness in the GSO itself.

3.2 Yagi Array Results: Getting back to the Yagi problem, the objective was to maximize its BW and gain over a specified frequency range. After trying several fitness functions, the following simple form was settled on:

$$YAGI\ Fitness = c1*Gain(L) + c3*Gain(M) + c5*Gain(U) - c2*SWR(L) - c4*SWR(M) - c6*SWR(U)$$

where L/M/U are the lower/mid/upper frequencies at which the Yagi's power gain and SWR are computed. The weighting coefficients are : c1 = c2 = c5 = c6 = 1; c3 = c4 = 3, which intentionally favors midband performance, slightly. Their values were determined empirically by making successive CFO runs with different values and evaluating the results of each one, something that could not be done quickly with a stochastic GSO.

Negative gravity was injected into CFO as follows. At each step negative gravity was pseudo randomly assigned using π-fractions (data files available, see Appendix A1). At each step in the CFO run, the value of the π-fraction was tested against the target level of negative gravity and the sign of gravity adjusted accordingly (positive or negative). CFO pseudocode and the actual vs. target amount of negative gravity are shown below (details in [30,31]).

Alg: CFO - Six-Element Yagi Design

Initialization (Step 0)*

(i) Initialize π-fraction index (← 2)

(ii) Compute Initial Probe Distribution (π-frac prv's)

(iii) Compute Initial Fitness Matrix

(iv) Initial Probe Accelerations (← 0)

For Step#: 1 - #Steps

(i) Pseudo randomly assign positive or negative gravity (π-frac prv's)

(ii) Compute Probe Position Vectors

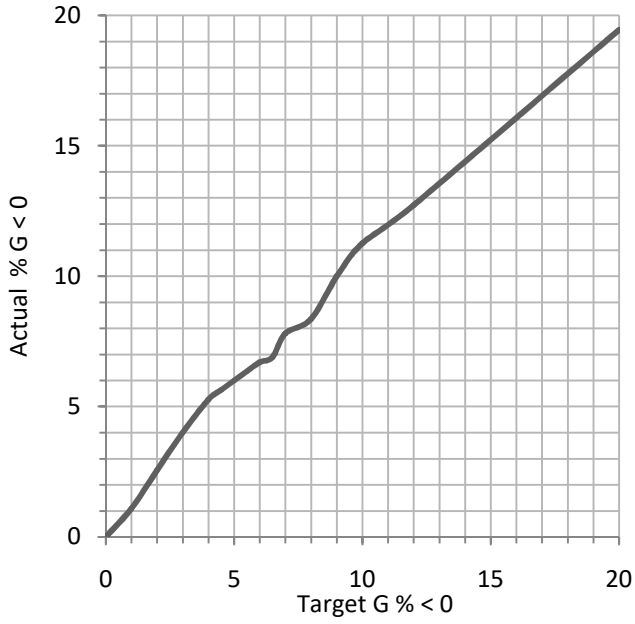
(iii) Retrieve Errant Probes

(iv) Compute Fitness Matrix for Current Probe Distribution

(v) Compute Accelerations from Current Probe Distribution/Fitnesses

(vi) Adjust Repositioning Factor F_{rep}

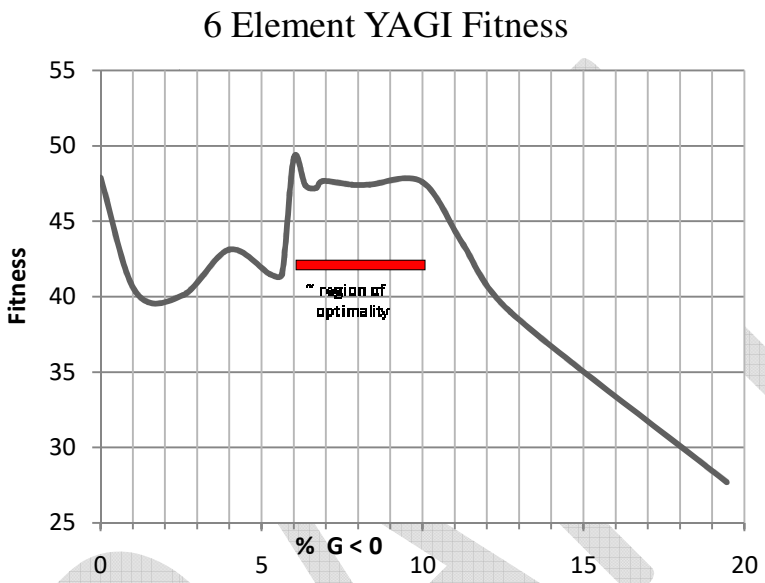
Next

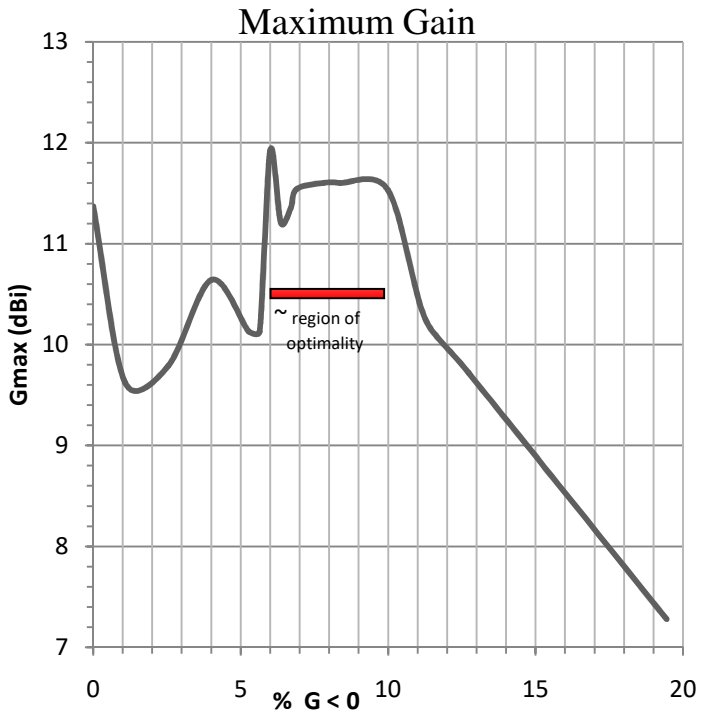


CFO Negative Gravity using π Fractions

The Yagi fitness and its maximum gain as a function of the amount of negative gravity are shown below. With zero negative gravity, the fitness and max gain, respectively, are about 48 and 12 dB, both very respectable values. Adding a small amount of negative gravity at first reduces both of these performance measures at first, but when the negative gravity level reaches about six percent both the fitness and gain increase dramatically, and they remain above their initial values with as much as about 10%

negative gravity, after which they gradually decrease. These two plots alone demonstrate convincingly that adding some small amount of negative gravity can help quite a bit. The negative gravity allows CFO's probes explore more widely and to discover even better array designs. The improved fitness and gain correspond to solutions that were missed by CFO because it failed to fly probes into their regions of DS. In fact, combining $G < 0$ in CFO with a couple of other extensions (Dynamic Threshold Optimization and Elitism) has resulted in more than 19% improvement in the fitness of a 6-element Yagi [30,31]. There is no question that some small amount of negative gravity is beneficial..





4. Conclusion It is evident that injecting a small amount of negative gravity into Central Force Optimization can produce substantially better results because the algorithm's exploration of DS is improved. The examples discussed here are compelling, and there is every reason to expect that injecting a small amount of negative gravity will enhance any CFO run against any GSO problem. Therefore, it is recommended that the negative gravity approach be used in all CFO implementations.

APPENDIX A1 – π FRACTIONS

GSO Sampling: Uniformity in randomly generated sample points is an important consideration in GSO. Sample points should be generated using a truly uniformly distributed random variable (rv) calculated from a probability distribution, but most if not all pseudorandom sequence generators fall short because their points in fact are not uniformly distributed. One alternative approach is using Low Discrepancy Sequences (LDS) which are becoming more popular. For example, De Rainville *et al*. [32,33] provide a summary of the uniformity problem and develop an evolutionary optimization approach to generating LDS. Pant *et al*. [34] describe an improved Particle Swarm Optimization (PSO) algorithm utilizing van der Corput and Sobol LDS. Other representative, not exhaustive, examples include LDS applied to liquid crystal display dot patterns [35], power system stabilizers [36], and financial analysis [37,38].

This appendix describes an alternative approach to generating uniformly distributed sample points using π Fractions that are computed from hexadecimal digit extraction from the mathematical constant π. Pi Fractions [39] are uniformly distributed and provide a basis for creating reproducible, deterministic sample point distributions that can be used in any GSO algorithm regardless of its fundamental nature, stochastic, deterministic or hybrid. The importance of determinism in electromagnetics problems is discussed in [47].

Another reason for considering π Fractions is what some practitioners, including the author, consider a willy-nilly proliferation of stochastic metaheuristics of questionable merit [42]. Examples range from "Anarchic societies" to "Zombies" [43]. Are these algorithms any good? How can they be efficiently compared head-to-head or to other well-established algorithms? Making them deterministic would be a good first step, and π Fractions can do that. For interested readers, π fractions#0-1,000,001 are downloadable at:
https://app.box.com/s/qdd8rzrhgaozne0ag1nes9jkm0bj6ark

BBP Algorithm: The Bailey-Borwein-Plouffe (BBP) algorithm quite remarkably extracts hexadecimal digits from the numerical constant π beginning at *any* digit without having to compute *any* of the preceding digits. BBP is based on the identity

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k}\left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6}\right)$$

whose derivation and use in BBP are described in detail in [39]. As an example, the hex digits of π starting at digit 1,000,000 are 26C65E52CB459350050E4BB1 and the corresponding π Fraction is 0.151464362347971272412488292131. For all practical purposes the first 215,829 π Fractions are uniformly distributed on [0,1) with a mean value of 0.499283729688375. The Cumulative Distribution Function (CDF) for these data is plotted in Fig. A1-1 (1,000 bins) in which $\Pr\{\pi_i \le X\}$ is the probability that π Fraction π_i is less than or equal to $0 \le X \le 1$. It is reasonable to speculate that all sequential π_i are uniformly distributed as well. Testing on various subsets of the 215,829 data set reveals a uniform distribution regardless of how many contiguous fractions are included in any fairly large sample or where the sequence is begun. It also seems reasonable to believe that any sufficiently large set of arbitrarily selected π_i also will be uniformly distributed in [0,1). These characteristics have not been investigated for the other constants discussed in [39], so it is not known whether or not they exhibit similar behavior.

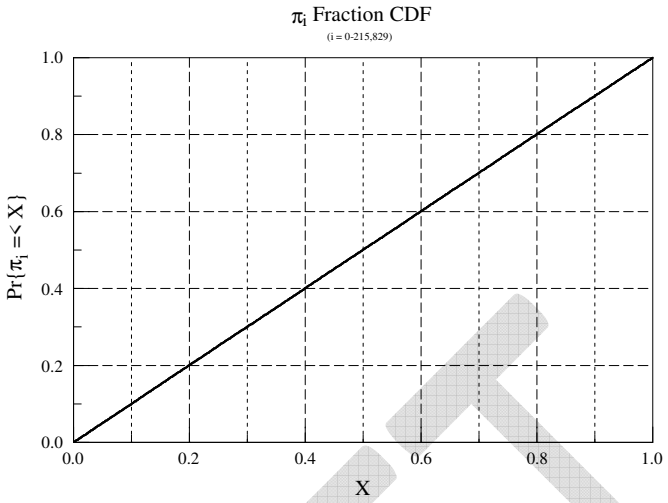


Fig. A1-1. π Fraction CDF

Dimensional Correlations: Nonuniformity in LDS sequences often is evident in bidimenional plots in high dimensionality spaces. Figs. 2 and 3 in [32] are good examples. They show, respectively, almost perfect linear correlations in monotonically increasing van der Corput sequences and correlations between dimensions 7 and 8 in a Halton sequence. Other striking visual examples appear in [42] and [43]. Testing of van der Corput and Halton sequences [44,45] reveals many undesirable correlations. A typical 30-dimensional Halton example for coordinates 27 and 28 appears Fig. A1-2.

Sample points based on π Fractions also can exhibit strong linear correlations, but apparently only under very limited circumstances. For example, Fig. A1-3 plots (x_{27},x_{28}) for 1,000 points in 30 dimensions using the π Fractions in their order of occurrence (index increment = 1 starting with the first π Fraction). The linear correlation is obvious, but it disappears completely when instead dimensions 27 and 29 are compared as seen in Fig. A1-4. Many test runs suggest that the π Fractions exhibit correlation *only* in successive dimensions and *only* when accessed in their order of occurrence, regardless of where the sequence starts. But when a different index greater than 1 is used, for example, a value of 2, there is no obvious correlation as shown in the (x_{27},x_{28}) plot in Fig. A1-5. These data suggest it is reasonable to believe that indeed the π Fractions provide uniformly distributed uncorrelated sample points as long as successive fractions are *not* used to compute the sample point coordinates.

An Example – πGASR Algorithm: The utility of π Fractions was investigated by generating uniformly distributed sample points and pseudorandom numbers in the genetic algorithm πGASR which is based on Li *et al.*'s novel GA [40]. A standard GA is improved in [40] by (i) allowing competition between child chromosomes in a new crossover operator resulting in better interpolation and extrapolation of decision space sample points and (ii)

introducing an iteration-dependent mutation operator. Li *et al.*'s algorithm is referred to here as "Genetic Algorithm with Sibling Rivalry" (GASR) because of the new crossover operator (see [40] for details and note that the "SR" descriptor is introduced here). Its implementation using π Fractions is algorithm πGASR. π Fractions are used to create the initial chromosome distribution and in testing for crossover, mutation, and elitism. Details of the scheme selecting the π_i are determined by the algorithm designer and in this case appear in the source code listing (see below). The manner in which the πGASR's fractions are selected avoids the bidimensional correlation issue discussed above. Note that πGASR, like CFO, maximizes the objective function instead of minimizing it.

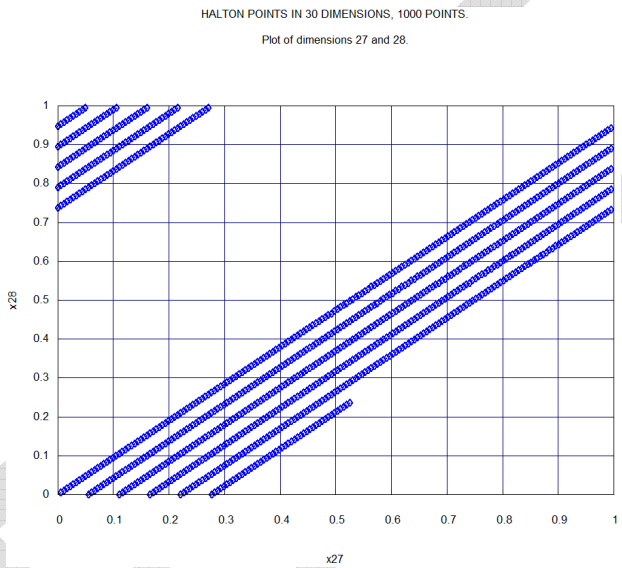


Fig. A1-2. Coordinates (x_{27},x_{28}) 30D Halton sequence (1,000 points).

I. Benchmark Results

πGASR was tested against the six-function benchmark suite shown in Table A1-1. Results are reported in Tables A1-2 and A1-3. In Table 1 DS is the decision space, x^* the location of the objective function's known maximum, and $f(x^*)$ its value. This suite was used in [41] to test the new algorithm vibrational-PSO (v-PSO).

πGASR was implemented with the following parameters: crossover probability = 0.8; mutation probability = 0.02; crossover weight factor w=0.5 [40]; mutation shape factor $\beta = 2$ [40]. The numbers of generations and chromosomes were 101 and 2500 in Table A1-2 and 100 and 2000 in Table A1-3, respectively. In both cases the best chromosome from the previous generation was randomly inserted into the next one ("elitism"). Runs in Table A1-2 were terminated early if the change in fitness between the current generation and the 20th previous generation was $\leq 10^{-6}$. In order to provide a head-to-head statistical comparison with v-PSO, the πGASR runs in Table A1-3 ran to completion using all 20,000,000 function evaluations.

Table 2 compares v-PSO and single-run πGASR results for the 10, 20 and 30-dimensional benchmarks. N_d is the DS dimensionality and N_{eval} the total number of πGASR function

evaluations. The v-PSO data are average values for 100 runs using 200,000 function evaluations per run (20,000,000 evaluations of each test function). Because v-PSO performs minimization the signs of its results have been changed for comparison to πGASR. In all cases in Table A1-2, a *single* πGASR run was made because every πGASR run with specific π Fraction distributions yields the same result every time since the fractions are pseudorandom and therefore known with absolute precision.

In terms of function evaluations πGASR's worst case figure of 656,308 is nearly 97% less than v-PSO's 20,000,000. In terms of solution quality, πGASR performed very well on f_2, f_3 and f_4 ; well on f_1 and f_5 ; and exceptionally well on f_6. For f_6 with N_d=30 πGASR required 294,763 evaluations (98.5% fewer than v-PSO) and returned a best fitness of -9.400238×10^{-3} compared to v-PSO's average value of -2,139.5±103.3. These results strongly suggest that pseudorandom π fractions can be very useful in implementing what amount to *deterministic* "stochastic" algorithms that avoid the need to make multiple runs to generate statistical data.

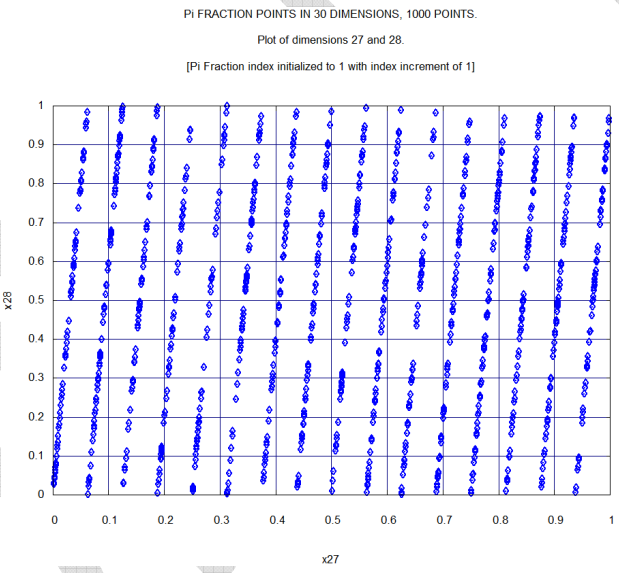


Fig. A1-3. Coordinates (x_{27},x_{28}) 30D π Fractions, index increment = 1.

Nevertheless, in order to directly compare v-PSO and πGASR, Table A1-3 shows statistical data using the same number of function evaluations. πGASR's average best fitness and its standard deviation are tabulated along with the best fitness returned over all runs. πGASR performed worse on f_1; essentially the same on f_2 and f_3; better on f_4; worse on f_5; and much better on f_6. However, even in cases where v-PSO outperformed πGASR the differences were not dramatic, and for function f_6 πGASR outperformed v-PSO by a very wide margin. Comparing the πGASR data in Tables A1-2 and A1-3 shows that the longer runs with far more function evaluations do yield uniformly better results, as expected.

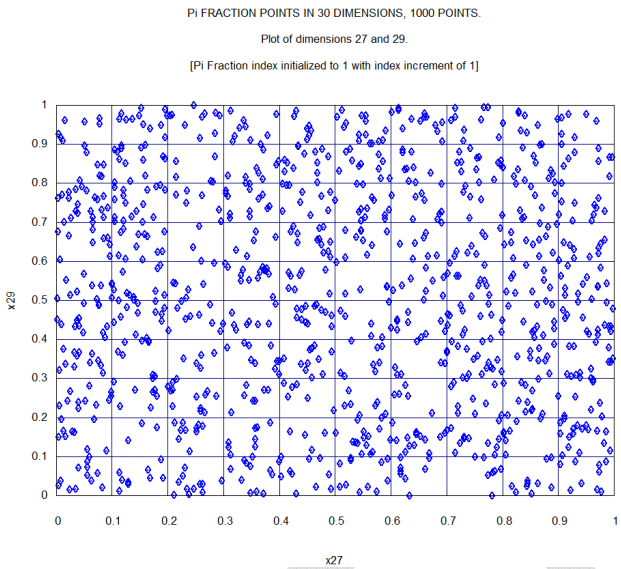


Fig. A1-4. Coordinates (x_{27},x_{29}) 30D π Fractions, index increment = 1.

Summary: π Fractions have been shown to be an effective approach to creating uniformly distributed decision space sample points for global search and optimization. Fractions associated with constants other than π also may be similarly useful, but they have not been investigated. Algorithm πGASR was used an example, and its performance tested against the v-PSO six benchmark suite with generally very good results and in one case much better results. π fraction pseudorandom sequences should be useful for improving the performance of any "stochastic" algorithm in several ways: (i) the resulting sequences are entirely deterministic so that all runs with the same setup produce exactly the same results thus rendering a stochastic algorithm effectively deterministic without compromising its ability to explore the decision space; (ii) making successive runs with different sequences likely will result in improved performance with far fewer function evaluations; and (iii) decision space adaptation is easily accomplished because the sequences are deterministic (for example, shrinking the decision space around a group of maxima). The π fraction data file used here and source code listings are available online at www.GitHub.com [46] (key word PiFractions).

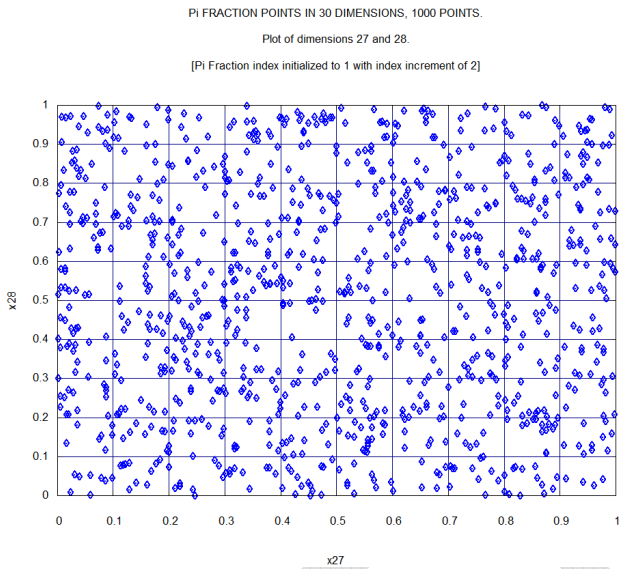


Fig. A1-5. Coordinates (x_{27},x_{28}) 30D π Fractions, index increment = 2.

Table A1-1. v-PSO Benchmark Functions.

Fnc #	*Function*	*f(x)*	DS	*x**	*f(x*)*
f_1	Ackley	$20\exp\left(-0.2\sqrt{\frac{1}{N_d}\sum_{i=1}^{N_d}x_i^2}\right)+$ $\exp\left(\frac{1}{N_d}\sum_{i=1}^{N_d}\cos(2\pi x_i)\right)-20-e$	$[-30,30]^{N_d}$	$[0]^{N_d}$	0
f_2	Cosine Mixture	$-\sum_{i=1}^{N_d}x_i^2+0.1\sum_{i=1}^{N_d}\cos(5\pi x_i)$	$[-1,1]^{N_d}$	$[0]^{N_d}$	$0.1N_d$
f_3	Exponential	$\exp(-0.5\sum_{i=1}^{N_d}x_i^2)$	$[-1,1]^{N_d}$	$[0]^{N_d}$	1
f_4	Griewank	$-\frac{1}{4000}\sum_{i=1}^{N_d}(x_i-100)^2+$ $\prod_{i=1}^{N_d}\cos\left(\frac{x_i-100}{\sqrt{i}}\right)-1$	$[-600,600]^{N_d}$	$[0]^{N_d}$	0
f_5	Rastrigin	$-\sum_{i=1}^{N_d}[x_i^2-10\cos(2\pi x_i)+10]$	$[-5.12,5.12]^{N_d}$	$[0]^{N_d}$	0
f_6	Schwefel	$-418.9829\ N_d+\sum_{i=1}^{N_d}[x_i\sin(\sqrt{\lvert x_i\rvert})$	$[-500,500]^{N_d}$	$[420.9687]^{N_d}$	0

Table A1-2. Single Run πGASR Data for v-PSO Benchmark Suite.

fnc#	N_d	Best Fitness		πGASR *Neval*
		v-PSO*	πGASR (single run)	
f_1	10	–1.84e-15±2.9e-16	–5.762878e-4	656,308
	20	–2.84e-15±1.5e-16	–1.161337e-2	328,243
	30	–4.93e-15±3.4e-16	–6.988124e-3	457,978
f_2	10	1±0	0.9999997	457,978
	20	2±0	1.9999993	457,978
	30	3±0	2.9999981	394,558
f_3	10	1±0	0.9999999	361,090
	20	1±3e-18	0.9999999	294,763
	30	1±1e-17	0.9999999	328,243
f_4	10	–0.020±0.006	–0.004429	492,372
	20	–0.0026±0.002	–0.015874	361,090
	30	–8.8568e-4±0.001	–0.002139	457,978
f_5	10	0±0	–1.057361e-4	425,640
	20	0±0	–1.203252e-3	394,558
	30	–5.6843e-16±1e-15	–9.932735e-5	492,372
f_6	10	–620.8131±50.4	–7.753379e-4	457,978
	20	–1.3384e+3±68.5	–7.666976e-4	394,558
	30	–2.1395e+3±103.3	–9.400238e-3	294,763

* average best fitness over 20,000,000 evaluations; data reproduced from Table IV in [41] with sign changed because πGASR maximizes *f(x)* while v-PSO minimizes.

Table A1-3. πGASR Statistical Data for v-PSO Benchmark Suite.

fnc#	N_d	Average Best Fitness (20×10^6 evals)		πGASR Overall Best Fitness**
		v-PSO*	πGASR Avg / Std Dev	
f_1	10	–1.84e-15±2.9e-16	–3.25606e-3 / 2.90e-3	–7.22411e-5
	20	–2.84e-15±1.5e-16	–3.71691e-3 / 4.04e-3	–1.58685e-4
	30	–4.93e-15±3.4e-16	–3.54867e-3 / 3.03e-3	–1.22941e-4
f_2	10	1±0	0.9999999 / 2.33e-7	0.9999999
	20	2±0	1.9999997 / 3.83e-7	1.9999999
	30	3±0	2.9999995 / 8.41e-7	2.9999999
f_3	10	1±0	0.9999999 / 9.40e-9	0.9999999
	20	1±3e-18	0.9999999 / 1.93e-8	0.9999999
	30	1±1e-17	0.9999999 / 3.55e-8	0.9999999
f_4	10	–0.020±0.006	–5.02147e-4 / 7.44e-4	–6.39425e-10
	20	–0.0026±0.002	–7.29040e-4 / 1.25e-3	–3.29534e-09
	30	–8.8568e-4±0.001	–8.45741e-4 / 1.354e-3	–6.00263e-07
f_5	10	0±0	–2.53755e-4 / 4.08e-4	–2.73693e-7
	20	0±0	–6.25339e-4 / 1.20e-3	–1.40063e-6
	30	–5.6843e-16±1e-15	–6.13252e-4 / 9.69e-4	–1.38614e-7
f_6	10	–620.8131±50.4	–1.84158e-4 / 8.37e-5	–1.27276e-4
	20	–1.3384e+3±68.5	–4.36412e-4 / 3.40e-4	–2.55634e-4
	30	–2.1395e+3±103.3	–6.58102e-4 / 4.18e-4	–3.82090e-4

* average v-PSO best fitness over 20,000,000 evaluations; data reproduced from Table IV in [41] with sign changed because πGASR maximizes *f(x)* while v-PSO minimizes.
** best πGASR fitness over 20,000,000 objective function evaluations

APPENDIX A2 – CENTRAL FORCE OPTIMIZATION

A2-1 The CFO Metaphor

Central Force Optimization (CFO) analogizes gravitational kinematics, that is, the motions of real bodies in the real Universe under the influence of real gravity. The governing physical law is Newton's Universal Law of Gravitation. Newton's Law formulates the magnitude of the gravitational force of attraction between the two masses m_1 and m_2 as (see [1] for specific references)

$$F = \gamma \frac{m_1 \, m_2}{r^2} \qquad \text{(a1)}$$

where r is the distance between them, and γ is the "gravitational constant." The force pf gravity always is attractive, never repulsive, and mass in the real Universe always is positive, never negative. The force of gravity is a *central force* because it acts only along the line connecting the mass centers, hence the name 'Central Force Optimization'. Mass m_1 experiences a vector acceleration due to mass m_2 given by

$$\vec{a}_1 = -\gamma \frac{m_2 \, \hat{r}}{r^2} \qquad \text{(a2)}$$

where $\hat{r}$ is a unit vector that points toward m_1 along the line joining the masses' centers.

A2-2 Problem Statement

The CFO metaheuristic addresses the following problem: In a decision space (DS) defined by $x_i^{\min} \le x_i \le x_i^{\max}, \quad i = 1,...,N_d$ where the x_i are decision variables, locate the global maxima of an objective ('fitness') function $f(x_1, x_2, ..., x_{N_d})$ possibly subject to a set of constraints $\boldsymbol{\Omega}$ among the decision variables. The value of $f(x_1, x_2, ..., x_{N_d})$ is called the "fitness." CFO explores DS by flying metaphorical "probes" whose trajectories are governed by equations of motion drawn from the gravitational kinematics analogy.

A2-3 Constant Acceleration

The vector location of a mass under constant acceleration is given by the position vector

$$\vec{R}(t + \Delta t) = \vec{R}_0 + \vec{V}_0 \, \Delta t + \frac{1}{2} \vec{a} \, \Delta t^2 \qquad \text{(a3)}$$

where $\vec{R}(t + \Delta t)$ is the position at time $t + \Delta t$. $\vec{R}_0$ and $\vec{V}_0$, respectively, are the position and velocity vectors at time t, and the acceleration $\vec{a}$ is constant during the interval Δt. In standard three dimensional Cartesian coordinates $\vec{R} = x\hat{i} + y\hat{j} + z\hat{k}$, where $\hat{i}$, $\hat{j}$, $\hat{k}$ are the unit vectors along the x, y, z axes, respectively. The CFO metaphor analogizes equations (a1)-(a3) by generalizing them to a decision space of N_d dimensions.

A2-4 Probe Trajectory

CFO's probes in a typical three-dimensional DS are shown schematically in Fig. A2-1. The location of each probe at each time step is specified by its position vector $\vec{R}_j^p$, in which p and j are the probe number and time step index, respectively. In an N_d-dimensional DS the position vector is $\vec{R}_j^p = \sum_{k=1}^{N_d} x_k^{p,j} \hat{e}_k$, where the $x_k^{p,j}$ are probe p's coordinates at time step j, and following standard notation $\hat{e}_k$ is the unit vector along the x_k axis.

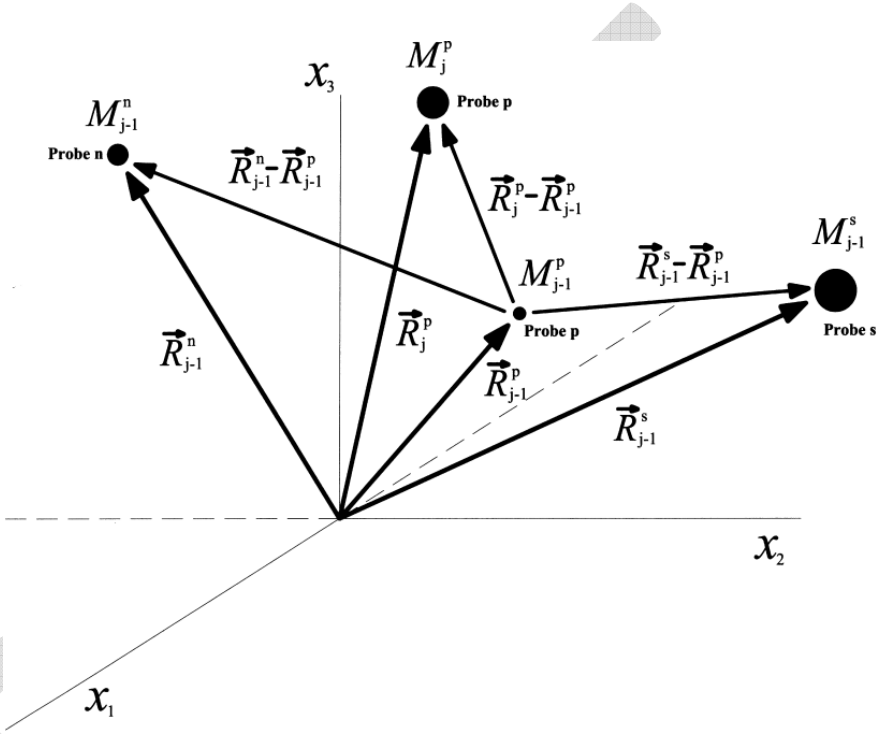


Fig. A2-1 Typical 3-D CFO Decision Space

Consider a typical probe, p. It moves from position $\vec{R}_{j-1}^p$ at time step $j-1$ to position $\vec{R}_j^p$ at time step j under the influence of the metaphorical "gravitational" forces that act on it. Those forces are created by the fitness at each of the other probes' locations at time step $j-1$. The "time" interval between steps $j-1$ and j is Δt.

At time step $j-1$ at probe p's location the fitness is $M_{j-1}^p = f(x_1^{p,j-1}, x_2^{p,j-1}, ..., x_{N_d}^{p,j-1})$. Each of the other probes also has associated with it a fitness of $M_{j-1}^k, k = 1, ..., p-1, p+1, ..., N_p$, N_p being the total number of probes. In this illustration, the value of the fitness is represented by the size of the blackened circle at the tip of the position vector. In keeping with the gravity metaphor, the blackened circles may be thought of as

"planets," say, in our Solar System. Larger circles correspond to greater fitness values, that is, bigger planets with correspondingly greater gravitational attraction. In Fig. A2-1 the fitnesses ordered from greatest to least occur at $\vec{R}^{s}_{j-1}$, $\vec{R}^{p}_{j}$, $\vec{R}^{n}_{j-1}$, and $\vec{R}^{p}_{j-1}$, respectively, as shown by the relative size of the circles.

Probe p's trajectory in moving from location $\vec{R}^{p}_{j-1}$ to $\vec{R}^{p}_{j}$ is determined by its initial position and by the *total acceleration* produced by the "masses" that are created by the fitnesses (or some function defined on them) at each of the other probes' locations. In the CFO implementation used in this paper the "acceleration," analogous to eq. (a2), experienced by probe p due to the single probe n is given by

$$\frac{G \cdot U\left(M^{n}_{j-1} - M^{p}_{j-1}\right) \cdot \left(M^{n}_{j-1} - M^{p}_{j-1}\right)^{\alpha} \cdot \left(\vec{R}^{n}_{j-1} - \vec{R}^{p}_{j-1}\right)}{\left|\vec{R}^{n}_{j-1} - \vec{R}^{p}_{j-1}\right|^{\beta}} \qquad \text{(a4)}$$

where G is CFO's "gravitational constant" corresponding to γ in eq. (a1). Note that in the real Universe $G > 0$, always. In CFO space, however, G can be positive (attractive force of gravity) or negative (repulsive force of gravity). Returning to the forces acting on probe p, in a similar fashion to probe n's effect, the acceleration of probe p due to a different probe s is given by

$$\frac{G \cdot U\left(M^{s}_{j-1} - M^{p}_{j-1}\right) \cdot \left(M^{s}_{j-1} - M^{p}_{j-1}\right)^{\alpha} \cdot \left(\vec{R}^{s}_{j-1} - \vec{R}^{p}_{j-1}\right)}{\left|\vec{R}^{s}_{j-1} - \vec{R}^{p}_{j-1}\right|^{\beta}} \qquad \text{(a5)}$$

Note that the minus sign in eq. (a2) has been included in the order in which the differences are taken in these acceleration expressions. "Mass" in eq. (a2) corresponds to the terms in the numerator involving the fitnesses. Importantly, ***it does not correspond to the fitness itself***. In these equations $U(\cdot)$ is the unit step function $U(z) = \begin{cases} 1, & z \geq 0 \\ 0, & otherwise \end{cases}$. And following standard notation the vertical bars denote vector magnitude, $\left|\vec{X}\right| = \left(\sum_{i=1}^{N_d} x_i^2\right)^{\frac{1}{2}}$, where x_i are the scalar components of $\vec{X}$.

There are no parameters in eq. (a2) corresponding to the "CFO exponents" $\alpha > 0$ and $\beta > 0$, nor to the unit step $U(\cdot)$. In real physical space α and β would take on values of 1 and 3, respectively. Note, too, that the numerators in eqs. (a4) and (a5) do not contain a unit vector like eq. (a2). The exponents are included to give the algorithm designer a measure of flexibility by assigning, if desired, a different variation of gravitational acceleration with mass and with distance.

A2-5 Mass in CFO Space

Two other important differences between real gravity and CFO's version are: (i) the definition of "mass," which above is the *difference of fitnesses*, for example, $M^{s}_{j-1} - M^{p}_{j-1}$, not the fitness value itself; and (ii) inclusion of the unit step $U(z) = \begin{cases} 1, & z \geq 0 \\ 0, & otherwise \end{cases}$. The difference of fitnesses is used to avoid excessive gravitational "pull" by other close by probes that presumably will have fitnesses with similar values. The unit step is included to avoid the possibility of "negative" mass. In the physical Universe, mass is positive, always, but in CFO-space the mass could be positive or negative depending on which fitness is greater. The unit step forces CFO to allow only positive masses, that is, attractive masses. If negative fitness differences were allowed, then some accelerations would be repulsive instead of attractive, thus forcing probes away from large fitnesses instead of toward them. The algorithm designer is free to consider other definitions of mass as well. One possibility, for example, might be a ratio of fitnesses similar to the "reduced mass" concept in gravitational kinematics.

A2-.6 Total Acceleration and Position Vector for a Single Probe

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 the sum of the gravitational effects over all other probes, that is,

$$\vec{a}^{p}_{j-1} = G \sum_{\substack{k=1 \\ k \neq p}}^{N_p} U\left(M^{k}_{j-1} - M^{p}_{j-1}\right) \cdot \left(M^{k}_{j-1} - M^{p}_{j-1}\right)^{\alpha} \times \frac{\left(\vec{R}^{k}_{j-1} - \vec{R}^{p}_{j-1}\right)}{\left|\vec{R}^{k}_{j-1} - \vec{R}^{p}_{j-1}\right|^{\beta}} \qquad \text{(a6)}$$

Probe p's new position vector at time step j is therefore given by

$$\vec{R}^{p}_{j} = \vec{R}^{p}_{j-1} + \vec{V}^{p}_{j-1} + \frac{1}{2}\vec{a}^{p}_{j-1}\Delta t^{2}, \quad j \geq 1 \qquad \text{(a7)}$$

where (see discussion)

$$\vec{V}^{p}_{j-1} = 0$$

(a7) is the analog of (a3) where $\vec{V}^{p}_{j-1}$ is the probe's "velocity" at the end of time step $j-1$. In eq. (a7) the coefficient ½, the velocity term, and the time increment Δt have been retained primarily as a formalism to highlight the analogy to gravitational kinematics, but they are not required, and in fact $\vec{V}^{p}_{j-1}$ should be set to zero. For the CFO implementation used here, as a matter of convenience Δt is arbitrarily set to 1. Of course, if desired, any constant value of Δt as well as the factor ½ can be absorbed into the gravitational constant G. The velocity term $\vec{V}^{p}_{j-1}$ in (a7) has been retained purely as a formality and *should be set to zero* as it has been here and shown above. Some tentative numerical experiments showed that including the $\vec{V}^{p}_{j-1}$ term may actually impede probe convergence. The reason for this

seemingly contradictory behavior is that while CFO's probe trajectories are piecewise linear, in general they are curvilinear. In curvilinear motion the acceleration and velocity vectors are not necessarily parallel. For example, in the limiting case of circular motion, the velocity vector is tangent to the circle while the acceleration vector is radially inward along the circle's radius, that is, in the case of circular motion case the acceleration and velocity vectors are actually orthogonal. In the original CFO paper $\vec{V}_{j-1}^{\,p}$ serendipitously had been set to zero as a matter of convenience so that the acceleration-velocity directionality issue was avoided entirely.

A2-7 Errant Probes

An important concern is how to handle an "errant" probe, that is, one that has flown outside DS. It is possible that the total acceleration experienced by a probe will fly it into regions of unfeasible solutions beyond the DS boundary. There are many ways to deal with this contingency, and a simple one was implemented in the basic version of CFO used here, namely, the use of a "repositioning factor," $0 \le F_{rep} \le 1$. This factor is used to reposition an errant probe according to the formulas

$$If\ x_i^{p,j} < x_i^{\min} \therefore\ x_i^{p,j} = x_i^{\min} + F_{rep} \cdot (x_i^{p,j-1} - x_i^{\min}) \qquad \text{(a8)}$$

$$If\ x_i^{p,j} > x_i^{\max} \therefore\ x_i^{p,j} = x_i^{\max} - F_{rep} \cdot (x_i^{\max} - x_i^{p,j-1}) \qquad \text{(a9)}$$

F_{rep} is assigned an initial value and incremented at each step by a fixed amount ΔF_{rep}, and if it exceeds unity is reset to the initial value. This simple approach guarantees that all probes remain inside DS. Note that while this procedure is pseudo random in nature, numerical experiments have shown that it is not as effective as pseudo randomly injecting a small amount of negative gravity.

A2-8 D_{avg} Convergence Metric

Perhaps the best measure of CFO's convergence is the "Average Distance" metric computed as $D_{avg} = \dfrac{1}{L_{diag} \cdot (N_p - 1)} \sum_{p=1}^{N_p} \sqrt{\sum_{i=1}^{N_d} (x_i^{p,j} - x_i^{p^*,j})^2}$, where $p*$ is the number of the probe with the best fitness, and the superscripts *p* and *j* denote, respectively, the probe and step numbers as above. $L_{diag} = \sqrt{\sum_{i=1}^{N_d} (x_i^{\max} - x_i^{\min})^2}$ is the length of the decision space principal diagonal. If every one of CFO's probes have coalesced onto a single point, then $D_{avg} = 0$. How closely this metric approaches zero is a good indicator of how CFO's probe distribution has evolved around a maxima. D_{avg} also is useful in identifying potential local trapping because oscillation in D_{avg} appears to signal trapping at a local maxima.

A2-9 Initial Probe Distribution

Every CFO run begins with a *user-specified* Initial Probe Distribution (IPD) defined by two parameters: **(1)** N_p, the total number of probes used; and **(2)** where the probes are placed inside DS. Many CFO implementations have employed a *pseudorandom variable* ('prv') IPD comprising an orthogonal array of N_p/N_d probes per axis pseudorandomly deployed on "probe lines" that are parallel to the coordinate axes and intersecting at a point along DS's principal diagonal. *Pseudorandomness* is defined as an arbitrary numerical sequence that is precisely known by specification or calculation. CFO's fundamentally deterministic nature is not altered by injecting pseudorandomness because at every step CFO's calculations are repeatable with absolute precision (see [3] for a discussion of why pseudorandomness is important in CFO).

Fig. A2-2 provides a two-dimensional (2D) example of this particular type of IPD, in this case nine probes shown on each probe line, two overlapping (but any number may be used). The probe lines are parallel to the x_1 and x_2 axes intersecting at a point on DS's principal diagonal marked by position vector $\vec{D} = \vec{X}_{\min} + \gamma(\vec{X}_{\max} - \vec{X}_{\min})$, where $\vec{X}_{\min} = \sum_{i=1}^{N_d} x_i^{\min} \hat{e}_i$ and $\vec{X}_{\max} = \sum_{i=1}^{N_d} x_i^{\max} \hat{e}_i$ are the diagonal's endpoint vectors. The parameter $0 \le \gamma \le 1$ [not to be confused with the gravitational constant in equation (a2)] determines where the probe lines intersect along the diagonal. Fig. A2-3 shows a typical 2D IPD for different values of γ, and Fig. A2-4 shows a 3D example. Of course, this procedure is generalized to the N_d-dimensional decision space to create N_d probe lines parallel to the N_d coordinate axes.

While Fig. A2-2 shows equal numbers of probes on each probe line, a different number of probes per axis can be used instead. For example, if equal probe spacing were desired in a decision space with unequal boundaries, or if overlapping probes were to be excluded in a symmetrical space, then unequal numbers could be used. Unequal numbers also might be appropriate if there is any *a priori* knowledge of DS's landscape, however it may have been obtained. For example, denser sampling in one region (more probes) may be appropriate if there appear to be more maxima there. While the variable N_p/N_d IPD of Fig. A2-2 was used for the results reported here, any number of other altogether different variable IPD's could be used instead. The key idea is that the IPD must be pseudorandom in the sense of uncorrelated with the decision space landscape in order to provide better sampling of the landscape. Typical CFO pseudocode implementing the variable probe-line IPD approach appears in Fig. A2-5.

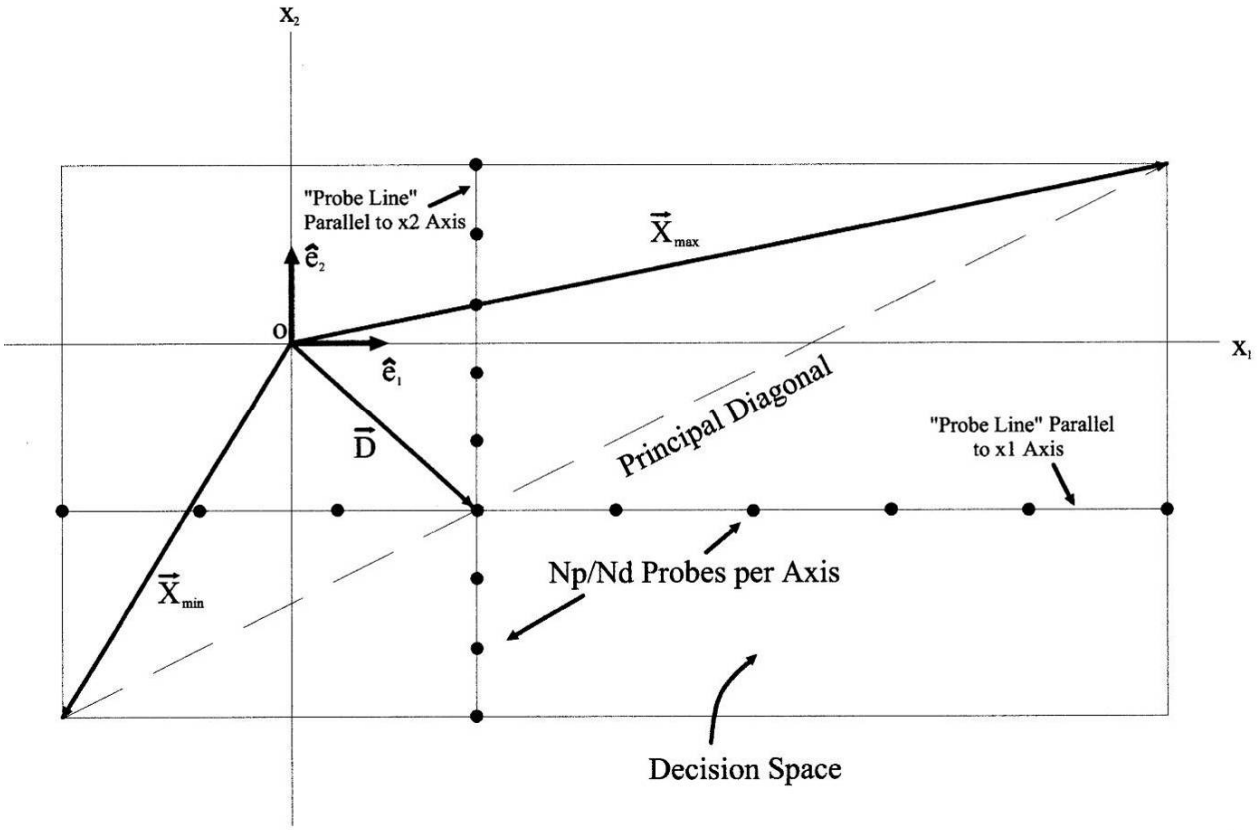


Fig. A2-2. Typical Variable 2D Initial Probe Distribution.

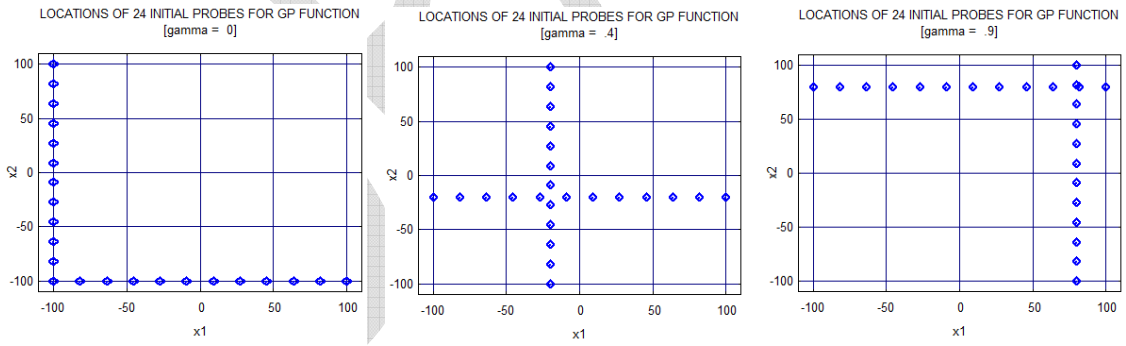


Fig. A2-3. Typical 2D IPD's for Different Values of γ (0./0.4/0.9).

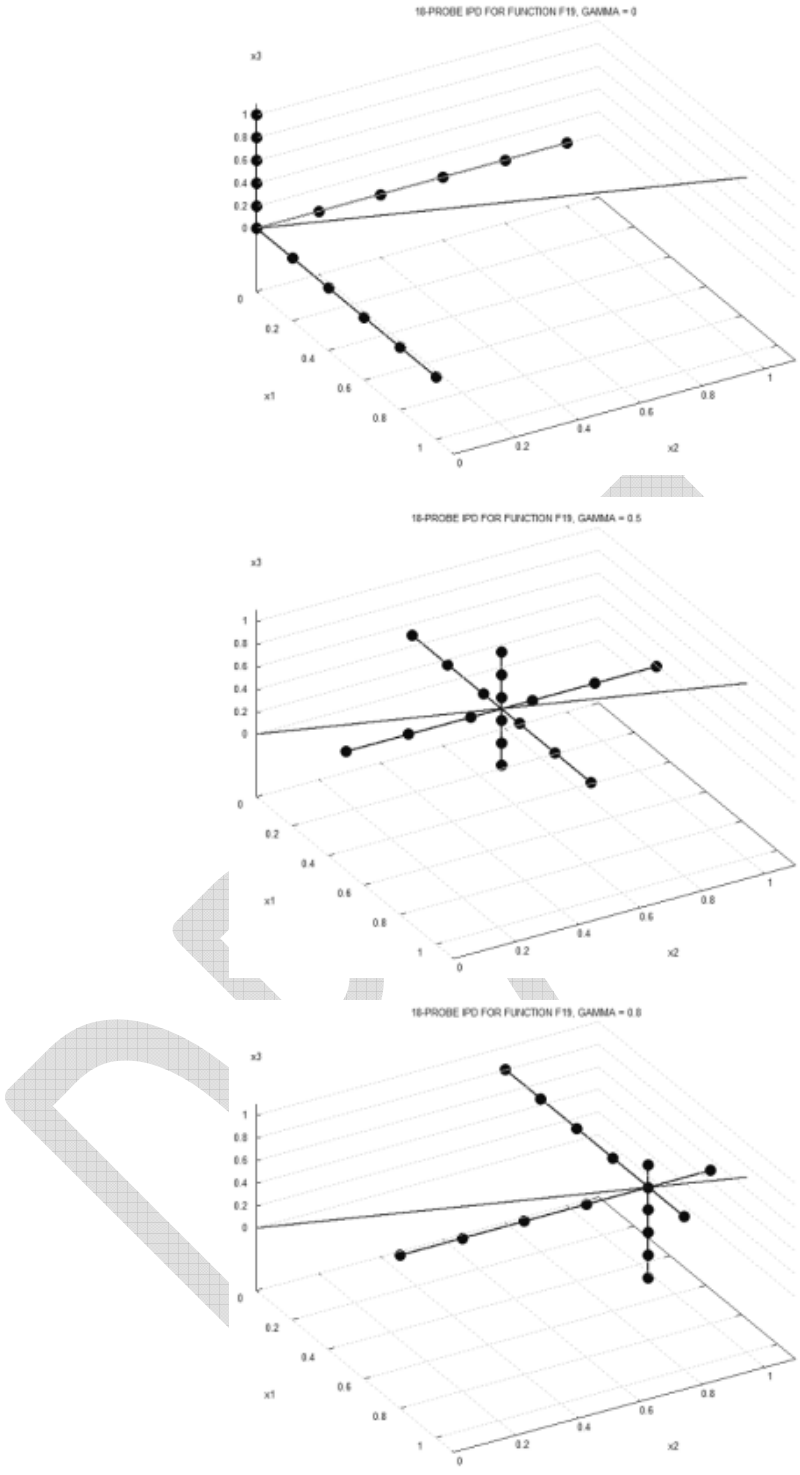
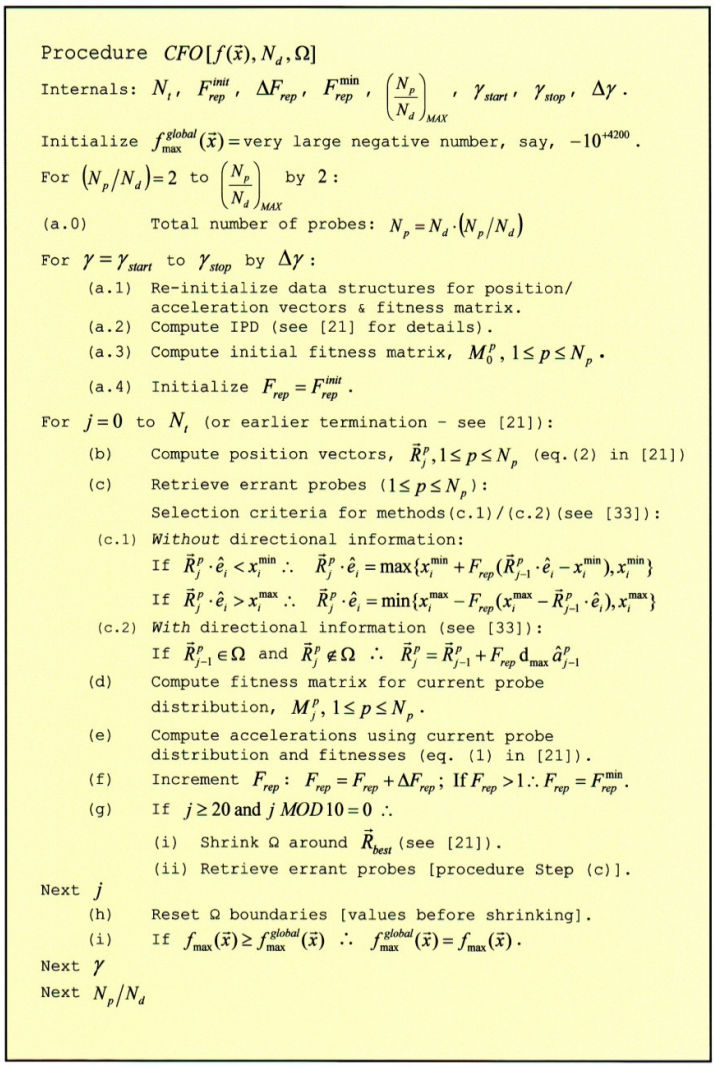


Fig. A2-4. Typical 3D IPD 6-probes/probe line, γ=(0.0/0.5/0.8, top to bottom)



(important note – above, Ω is the Decision Space)

Fig. A2-5. Typical CFO Pseudocode using Probe Line IPD

REFERENCES

[1] Formato, R. A., "Central Force Optimization: A New Metaheuristic with Applications in Applied Electromagnetics," *Prog. Electromagnetics Research*, **PIER 77**, pp. 425-491 (2007), online at https://www.jpier.org/
doi:10.2528/PIER07082403.

[2] Formato, R. A., "Central Force Optimization: A New Computational Framework for Multidimensional Search and Optimization," in **Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)**, *Studies in Computational Intelligence 129* (N. Krasnogor, G. Nicosia, M. Pavone, and D. Pelta, Eds.), vol. 129, pp. 221-238, Springer-Verlag, Heidelberg (2008).

[3] Formato, R. A., "Central Force Optimisation: a New Gradient-Like Metaheuristic for Multidimensional Search and Optimisation," *Int. J. Bio-Inspired Computation*, vol. 1, no. 4, pp. 217-238 (2009).
doi: 10.1504/IJBIC.2009.024721

[4] Ding D, Luo X, Chen J, Wang X, Du P, Guo Y. A Convergence Proof and Parameter Analysis of Central Force Optimization Algorithm. *Journal of Convergence Information Technology* (JCIT). 2011;6(10):16-23.
https://www.researchgate.net/publication/236985390

[5] Crepinsek, M., Liu, S-H., Mernik, M., "Exploration and exploitation in evolutionary algorithms: A survey," ACM Comp. Surveys, Art. No. 35, July 2013. [https://doi.org/10.1145/2480741.2480752]

[6] Luke, S., "Essentials of Metaheuristics, Second Edition", Online Ver. 2.2, Oct. 2015.
https://cs.gmu.edu/~sean/book/metaheuristics/

[7] Korosec, P. and Eftimov, T., "Insights into Exploration and Exploitation Power of Optimization Algorithm Using DSCTool," Mathematics, vol. 8, p. 1474, 2020. https://doi.org/10.3390/math8091474

[8] Cuevas, E., Echavarria, A., Ramirez-Ortegon, M., "An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation," Appl. Intell., vol. 40, pp.256–272, 2014.
https://doi.org/10.1007/s10489-013-0458-0

[9] Green, R., Wang, L., Alam, M. (2012). Training neural networks using Central Force Optimization and Particle Swarm Optimization: Insights and comparisons. *Expert Systems with Applications*. *39*(1), 555-563 (2012). doi:10.1016/j.eswa.2011.07.046

[10] Qubati, G., Dib, N. (2010). Microstrip patch antenna optimization using modified central force optimization. *Progress In Electromagnetics Research B*, 21, 281-298.

[11] Green, R.C., Wang, L., Alam, M. (2012). "Intelligent state space pruning with local search for power system reliability evaluation", presented at 3rd IEEE PES International Conference and Exhibition on *Innovative Smart Grid Technologies (ISGT Europe), 14-17 Oct. 2012* , Berlin, pp.1-8
doi: 10.1109/ISGTEurope.2012.6465775

[12] Mahmoud, KR (2011) Central force optimization: Nelder-Mead hybrid algorithm for rectangular microstrip antenna design. *Electromagnetics*, 31(8):578-592.
doi: 10.1080/02726343.2011.621110.

[13] Asi M, Dib NI (2010) Design of multilayer microwave broadband absorbers using central force optimization. *Progress In Electromagnetics Research B*, 26:101-113, 2010. doi:10.2528/PIERB10090103

[14] Montaser, A.M., Mahmoud, K.R., Elmikati, H.A. (2012). Integration of an optimized E-shaped patch antenna into a laptop structure for bluetooth and notched-UWB standards using optimization techniques. *ACES Journal* (Applied Computational Electromagnetics Society). 27(10), p. 784 (2012).

[15] Montaser, A.M., Mahmoud, K.R., Abdel-Rahman, A.B., Elmikati, H.A. (2013) "Design bluetooth and notched-UWB e-shape antenna using optimization techniques," *Progress In Electromagnetics Research B*, 47:279-295, 2013. doi:10.2528/PIERB12101407

[16] Montaser, A.M., Mahmoud, K.R., Elmikati, H.A. (2012). "Tri-band slotted bow-tie antenna design for RFID reader using hybrid CFO-NM algorithm", presented at 29th *National Radio Science Conference (NRSC), 10-12 April 2012, Cairo*, pp.119-126.
doi: 10.1109/NRSC.2012.6208515.

[17] Gonzalez, J.E., Amaya, I., Correa, R., (2013). Design of an optimal multi-layered electromagnetic absorber through central force optimization. Paper presented at PIERS, Stockholm, Sweden, 12-15 Aug. 2013, *Proceedings* p. 1082.

[18] A.khaleel, A. A. J. (2009). Linear phase finite-impulse response filter design using central force optimization algorithm. M.Sc. Elec. Eng. Thesis, online at http Ahmed, E., Mahmoud, K.R., Hamad, S., Fayed, Z.T. (2013) CFO parallel implementation on GPU for adaptive beam-forming applications. *International Journal of Computer Applications*. 70(12):10-16, May 2013. doi: 10.5120/12013-8001

[19] Montaser, A.M., Mahmoud, K.R., Abdel-Rahman, A.B., Elmikati, H.A. (2013). Design of a compact tri-band antenna for RFID handheld applications using optimization techniques. Paper presented at 30th National Radio Science Conference (NRSC), Cairo, Egypt, 16-18 April, 2013. doi: 10.1109/NRSC.2013.6587905.

[20] Haghighi A., and Ramos, A. (2012). HM Detection of leakage freshwater and friction factor calibration in drinking networks using central force optimization. *Water Resources Management,* Vol. 26, No. 8, pp.2347-2363 (2012).

[21] Chen, Y., Yu, J., Mei, Y., Wang, Y., Su, X. (2015). Modified central force optimization (MCFO) algorithm for 3D UAV path planning. *Neurocomputing*. Online 26 July 2015,http://www.sciencedirect.com/science/article/pii/S0925231215010267.
doi: 10.1016/j.neucom.2015.07.044

[22] Abdel-Rahman, A.B., Montaser, A.M., Elmikati, H.A., (2012). Design of a novel bandpass filter with microstrip resonator loaded capacitors using CFO-HC algorithm.

presented at *Middle East Conference on Antennas and Propagation, Cairo, Egypt, 29-31 Dec. 2012.*
doi:10.1109/MECAP.2012.6618188

[23] T.T. Huan and H.P.Y. Anh, "Optimal stable gait for nonlinear uncertain humanoid robot using central force optimization algorithm *Engineering Computations*, 18 January 2019 ISSN: 0264-4401

[24] Chao, M., Xin, S. Z., Min, L.S.,"Neural network ensembles based on copula methods and Distributed Multiobjective Central Force Optimization algorithm", Engineering Applications of Artificial Intelligence, Vo. 32, June 2014, pp.203-212
https://doi.org/10.1016/j.engappai.2014.02.009

[25] Talal, T.M., Attiya, G., Metwalli, M.R. *et al.* Satellite image fusion based on modified central force optimization. *Multimed Tools Appl*, Vol. 79, pp.21129–21154 (2020). https://doi.org/10.1007/s11042-019-08471-7.

[26] H.M. El-Hoseny, W.A. El-Rahman,E.M. El-Rabaie,F.E.A. El-Samie, and O.S. Faragallah, "An efficient DT-CWT medical image fusion system based on modified central force optimization and histogram matching", *Infrared Physics & Technology*, Vol. 94, Nov. 2018, pp.223-231
https://doi.org/10.1016/j.infrared.2018.09.00

[27] Shaikh, N.F., Doye, D.D. (2014). A novel iris recognition system based on central force optimization. *International Journal of Tomography and Simulation*. Vol. 27, No. 3

[28] A.khaleel, A. A. J. (2009). Linear phase finite-impulse response filter design using central force optimization algorithm. M.Sc. Elec. Eng. Thesis, online at https://theses.ju.edu.jo/Original_Abstract/JUF0706056/JUF0706056.pdf

[29] Momin, J. and Yang, X-S., "A literature survey of benchmark functions for global optimisation problems", *Int. J. Mathematical Modelling and Numerical Optimization,* Vol. 3, No. 2, 2033

[30] Formato, R.A., "Central Force Optimization with Gravity < 0,Elitism, and Dynamic Threshold Optimization: An Antenna Application,6-Element Yagi-Uda Arrays", *Wireless Engineering and Technology*,Vol 12, pp.53-82 (2021)
https://doi.org/10.4236/wet.2021.124004

[31] Formato, R.A., "Six-Element Yagi Array Designs Using Central Force Optimization with Pseudo Random Negative Gravity," *Wireless Engineering and Technology*, Vol. 12, pp.23-51 (2021)
https://doi.org/10.4236/wet.2021.123003

[32] De Rainville, F.-M., Gagné, C., Teytaud, O., and Laurendeau, D., "Evolutionary Optimization of Low-Discrepancy Sequences," *ACM Trans. Modeling and Computer Simulation* (TOMACS), **22**, 2, Article No. 9, March 2012.

[33[De Rainville, F.-M., Gagné, C., Teytaud, O., and Laurendeau, D., "Optimizing Low-Discrepancy Sequences with an Evolutionary Algorithm," *Genetic and Evol. Comp. Conf. 2009* (GECCO 2009), July 8-12, Montreal, Canada

[34] Pant, M., Thangaraj, R., Grosan, C., and Abraham, A., "Improved Particle Swarm Optimization with Low-Discrepancy Sequences," *IEEE Cong. Evol. Comp. 2008* (CEC 2008), pp. 3011-3018.
doi: 10.1109/CEC.2008.4631204.

[35] Imamichi, T., Numata, H., Mizuta, H. and Ide, T., "Nonlinear Optimization to Generate Non-overlapping Random Dot Patterns," *Proc. 2011 Winter Simulation Conf.*, S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds. pp. 2419-2430 (978-1-4577-2109-0/11/$26.00 ©2011 IEEE).

[36] Alabduljabbar, A. A., Milanovi, J. V., Al-Eid, E. M., "Low Discrepancy Sequences Based Optimization Algorithm for Tuning PSSs," *Proc. 10th Intl. Conf. Prob. Meth. Applied Power Systems 2008* (PMAPS '08), 25-29 May 2008, Rincon, pp.1-9.

[37] Krykova, I., "Evaluating of Path-Dependent Securities with Low Discrepancy Methods," M.S. Thesis (Financial Mathematics), Worcester Polytechnic Institute, Dec. 2003.

[38] Jørgensen, K., "Himalaya Options: an analysis of pricing Himalaya options in a Monte Carlo and Quasi-Random Monte Carlo framework," Master's Thesis, Dept. Business Studies, School of Business, Aarhus Univ. (Denmark), Aug. 2008.

[39] Bailey, D., Borwein, P., and Plouffe, S., "On the Rapid Computation of Various Polylogarithmic Constants," *Math. Comp.*, **66**, 218, April 1997:903-913 [S 0025-5718(97)00856-9].

[40] Li, W.T., Shi, X.W., Hei, Y. Q., Liu, S. F., and Zhu, J., "A Hybrid Optimization Algorithm and Its Application for Conformal Array Pattern Synthesis," IEEE Trans. Ant. Prop., **58**, 10, Oct. 2010:3401-3406.

[41] Pehlivanoglu, Y. V., "A New Particle Swarm Optimization Method Enhanced With a Periodic Mutation Strategy and Neural Networks," IEEE Trans. Evol. Comp., **17**, 3, June 2013:436-452.

[42] Sorensen, K., "Metaheuristics—the metaphor exposed," Intl. Trans. in Op. Res. **22**:3–18 (2015).
doi: 10.1111/itor.12001

[43] Aranha, C., and Campelo, F., "Evolutionary Computation Bestiary," online at https://github.com/fcampelo/EC-Bestiary;http://conclave.cs.tsukuba.ac.jp/research/bestiary/

[44] van der Corput, J. G., "Verteilungsfunktionen." *Proc. Ned. Akad. v. Wet.*, **38**:813–821, 1935.

[45] Halton, J. H. "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals." *Numerische Mathematik,* **2:**84-90, 1960.

[46] https://github.com (search term: PiFractions).

[47] Formato, R. A., "*Determinism in Electromagnetic Design & Optimization - Part I: Central Force Optimization*". [Online at www.e-fermat.org.]