

Central Force Optimization Applied to the PBM Suite of Antenna Benchmarks

Richard A. Formato 

Registered Patent Attorney & Consulting Engineer, Harwich, MA, USA

Email: rf2@ieee.org

How to cite this paper: Formato, R.A. (2026) Central Force Optimization Applied to the PBM Suite of Antenna Benchmarks. *Journal of Applied Mathematics and Physics*, 14, 735-769. <https://doi.org/10.4236/jamp.2026.142039>

Received: December 12, 2025

Accepted: February 11, 2026

Published: February 14, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Real-world optimization problems often require an external “modeling engine” that computes fitnesses or data that are then input to an objective function. These programs often have much longer runtimes than evaluating fitnesses solely with built-in compiler routines. The PBM suite of “real world” antenna benchmarks is an example. Using a stochastic optimizer on real-world problems can be quite challenging because every run returns a different “best” fitness. This issue is addressed by making many runs, often hundreds, possibly thousands, in order to generate meaningful statistics, but doing so may be prohibitive if external modeling is required. And even then the statistical nature of the results may obscure true global extrema. In addition, most of the time real-world problems do not come with a well-defined, clearly appropriate objective function. It falls to the practitioner to define one, which in itself can be a daunting task made even more difficult by using a stochastic optimizer because then every candidate fitness function must be evaluated statistically. This paper examines these and related issues by applying CFO to the PBM suite.

Keywords

Optimization, Global Search and Optimization, π Fractions, Central Force Optimization, CFO, PBM, GASR, Genetic Algorithm, PBM Antenna Benchmarks, Antenna, Deterministic Algorithm, Stochastic Algorithm, Pseudorandomness, Metaheuristic, Evolutionary Algorithm, Multidimensional Search

1. Introduction

The PBM suite is a set of typical “real-world” engineering problems developed by Pantoja, Bretones and Martin. These problems do not have known solutions. They

are designed to test the effectiveness of antenna optimization algorithms using the Numerical Electromagnetics Code as an external modeling engine. A major concern whenever external modeling is required with a stochastic optimizer is having to make multiple runs to evaluate how effective the optimizer is, but not so with a *deterministic* optimizer.

Central Force Optimization (CFO) [1]-[4] is a deterministic gradient-like multi-dimensional search and optimization metaheuristic based on the metaphor of gravitational kinematics. It comprises two simple equations, and because the law of gravity is deterministic, so too is CFO, unlike many nature-inspired algorithms that are fundamentally stochastic.

In this paper, CFO is applied to the PBM suite and its results compared to published performance data for other optimization algorithms. CFO acquits itself quite well. CFO's gradient-like nature is discussed, and it is speculated that a "generalized hyperspace derivative" might be defined for optimization problems as a new mathematical construct based on the Unit Step function. What appears to be a sufficient but not necessary condition for local trapping of CFO, oscillation in the probe average distance curve, is discussed in the context of the theory of gravitational "resonant returns" that gives rise to strikingly similar oscillatory curves. It is suggested that this theory may be applicable to CFO as an aid to understanding trapping and to developing effective mitigation techniques, possibly based on a concept of "energy" in CFO space. It also is suggested that CFO may be re-formulated as a "total energy" model by analogizing conservation of energy for orbiting masses in physical space.

This paper serves several purposes. The primary purpose is to report CFO's performance against the PBM suite. Results from algorithms π CFO and π GASR are compared to the published PBM data and to CFO implemented without π fractions. π fractions are used to inject pseudorandomness into an algorithm and are discussed in detail. The comparison data are quite good. They demonstrate the general utility of π fractions in global search and optimization, in particular in rendering deterministic an otherwise probabilistic algorithm.

The secondary purposes of this paper are i) to describe the CFO metaheuristic and to present its basic equations; ii) to suggest the possibility of a new mathematical construct, a "hyperspace directional derivative" based on the Unit step function; iii) to discuss the strikingly similar appearance of velocity curves for NEOs (Near Earth Objects in gravitationally trapped close encounters) with CFO's average probe distance curves; iv) to suggest the possibility of using probe total "energy" as a measure of local trapping; and v) to suggest the possibility of re-formulating CFO using a conservation of energy model.

CFO was introduced as a novel metaheuristic [1] that was applied to two problems in applied electromagnetics (EM): i) designing an equalizer for the canonical *Fano Load*; and ii) synthesizing a linear dipole array. CFO's results were compared to data from several other algorithms, and it also was tested against many benchmark functions with analytically known maxima. The results suggested that CFO

merited further development as a promising and robust optimization methodology, which indeed has been the case. CFO has been used to synthesize linear and circular antenna arrays, and it performed very well compared to the Quadrature Programming Method (QPM), to a Particle Swarm Optimization (PSO) algorithm, and to a Genetic Algorithm (GA) as reported in [5]. Even though CFO is perhaps not as well developed as many of the widely used evolutionary algorithms (EAs), it nevertheless has shown very robust performance and considerable promise for future development as discussed below.

An important consideration in implementing CFO algorithms is the inclusion of some measure of pseudorandomness because doing so evidently improves decision space (DS) exploration. To that end, π fractions are applied to a version called π CFO and its performance is compared to the GA called π GASR using the PBM suite [6] as the test platform.

The PBM paper [6] describes its purpose as follows: “This paper presents a new test suite formed not by artificial analytical functions but by a set of numerical electromagnetic problems aimed at facilitating the comparison of the performance of different EAs when they are applied in computational electromagnetics.”

The PBM suite is specifically designed to objectively evaluate the performance of EAs used to solve complex “real world” EM problems. Representative, not exhaustive, examples of such algorithms might include: PSO-based array synthesis [7]; swarm intelligence optimization of layered media [8]; GA antenna modeling [9]; array design using GAs, and memetic and tabu search algorithms [10]; crack detection using a finite-difference frequency domain/PSO methodology [11]; Vee-dipole optimization using a bacteria foraging algorithm [12]; and GA-optimized MRI coil design [13]. These are the kinds of problems PBM is meant to address by objectively evaluating how well the associated optimizers perform. CFO is a candidate algorithm.

EAs solve these problems by “evolving” solutions that are unavailable analytically or numerically. The plethora of nature-inspired EAs makes comparison very difficult without a standardized set of benchmarks for testing, which is why the PBM suite was developed. PBM is based on the Numerical Electromagnetics Code (NEC) [14] and addresses this need for standardized testing of EA’s that are applied to optimized antenna design. It therefore is also a useful tool for evaluating how well CFO performs.

An EA’s performance is measured in two ways: (i) by its *effectiveness* (accurately locating global maxima, values and locations) and (ii) by its *efficiency* (locating global maxima with minimum computational effort). An algorithm that fails to accurately locate global maxima obviously fails to accomplish its intended purpose and consequently is ineffective. EAs that are, on a relative basis, computationally intensive are less desirable than ones requiring fewer calculations. This paper uses these performance measures to compare CFO to the four EAs discussed in [6]. The results are, for the most part, very encouraging. CFO performs

very well compared to the PBM EAs, and in many cases much better. But CFO does suffer limitations, primarily a consequence of its being deterministic, which renders it prone to local trapping.

Because this paper touches on several topics, it is organized as follows.

1. Introduction
 2. PBM Benchmarks: Overview & NEC4
 - 2.1. The PBM Problems
 - 2.2. NEC4 Validation
 3. CFO Performance Summary
 - 3.1. Effectiveness
 - 3.2. Efficiency
 4. Pseudorandom CFO Using Pi Fractions
 - 4.1. π fractions
 - 4.2. Utility of π fractions in CFO
 5. Algorithms Compared
 6. Comparison Results
 7. The CFO Metaphor
 - 7.1. CFO, NEOs and Energy
 - 7.2. CFO Equations
 - 7.3. New “Hyperspace Directional Derivative”
 - 7.4. CFO Parameters
 8. Why a Deterministic Optimizer?
 9. Conclusion
- Appendix A: PBM Benchmarks—Geometries & Landscapes
Appendix B: CFO Pseudocode

2. PBM Benchmarks: Overview and NEC4

2.1. The PBM Problems

The PBM suite comprises five antenna problems designed to test an EA’s effectiveness and efficiency. They do not have known analytical solutions although, of course, the two-dimensional (2D) topologies can be visualized. **Table 1** lists the suite’s properties. In each case, the objective is to maximize the antenna’s directivity (“fitness”). Four of the problems are 2D, while the fifth is $(N_{el} - 1)D$ [N_{el} is the number of dipole elements in a collinear array]. The 2D problem topologies (“landscapes”) are plotted in the Appendix to illustrate the nature and complexity of the decision space. Problems #1 and #4 are unimodal with a single global maximum. The first problem is “lumpy” with strong local maxima, whereas the fourth is “smooth”. Problem #2 is “noisy” in a complex landscape with large amplitude nearby local maxima. The third problem’s topology is extremely multimodal with four global maxima. Problem #5 is a unimodal high-dimensionality problem that optimizes the element spacing of an N_{el} -element collinear dipole array.

Table 1. Properties of the PBM benchmark problems.

| PBM Benchmark # | Problem Characteristics (in each case objective is to maximize directivity) |
|--------------------|---|
| 1 | Variable length center-fed dipole. 2D, unimodal, single global maximum, strong local maxima. |
| 2 | Uniform 10-element array of center-fed $\frac{\lambda}{2}$ -dipoles. 2D, added Gaussian noise, single global maximum, multiple strong local maxima. |
| 3 | 8-element circular array of center-fed $\frac{\lambda}{2}$ -dipoles. 2D, highly multimodal, four global maxima. |
| 4 | Vee Dipole. 2D, unimodal, single global maximum, “smooth” landscape. |
| 5 | Collinear N_{el} -element array of center-fed $\frac{\lambda}{2}$ -dipoles. $(N_{el} - 1)D$, unimodal, single global maximum. |

2.2. NEC4 Validation

Because the PBM problems do not have analytic solutions, in [6] they were solved numerically using the Numerical Electromagnetics Code (“NEC”). Anecdotally, NEC is considered by many engineers to be the “gold standard” for wire structure EM modeling, but it must be used correctly. The first step in evaluating CFO’s performance therefore is validation of the published PBM results. The five PBM benchmark antennas were modeled using the Numerical Electromagnetics Code Version 4.1 Double Precision (NEC4.1D) [14], which may be (probably is) a different version of NEC than the one used in [6]. The segmentation and wire radii used in [6] were replicated. More detailed discussion of the PBM problems and NEC4 input and output files are available online at the following website:

<https://app.box.com/s/j6km9sy4udstujy5fx9e40p6ci70qs85>. Each PBM antenna was modeled using the coordinates for the maxima reported in [6]. In several cases, they were estimated from graphical data and, as a consequence, are necessarily approximate. **Table 2(a)** and **Table 2(b)** summarize the results. In the “Domain” column, x_1 and x_2 , respectively, refer to the abscissa and ordinate; λ is the wavelength; N_d is the problem’s dimensionality; and D_{\max}^{PBM} and D_{\max}^{NEC4} , respectively, are the maximum directivities from the PBM paper and as computed by NEC4.1D for this paper.

While the general conclusion is that NEC4 essentially recovers the PBM results, there are some noteworthy differences. For problems #1 and #2, NEC4’s computed directivities are slightly less than PBM’s. For problems #3 and #4, they are lower by wider margin. The best agreement is on problem #5 where the NEC4 and PBM data show very good agreement. What accounts for these discrepancies is not clear. There are several possible explanations, ranging from possibly different versions of NEC to compiler differences in creating the executables to slight differences in the antenna models, for example, source modeling (NEC4 excitation was modeled following the guidelines for the “EX0” card [14] (Part I, p. 48 *et seq.*). These differences between the PBM results and NEC4 notwithstanding, the vali-

dition test shows that NEC4 can be used effectively to assess CFO’s performance against the PBM benchmarks, and that CFO should recover maxima with similar amplitudes at about the same points in DS.

Table 2. (a). PBM and NEC4 Results for Benchmarks #1-4; (b) PBM and NEC4 Results for Benchmark #5.

| (a) | | | | | | |
|-----------|--|--------------------------|---------|------------------|-------------------|--|
| Problem # | PBM | | | NEC4 | | |
| | Domain | x_1 | x_2 | D_{\max}^{PBM} | D_{\max}^{NEC4} | |
| 1 | $0.5\lambda \leq L \leq 3\lambda$ $0 \leq \theta \leq \frac{\pi}{2}$ | 2.58λ | 0.63 | 3.32 | 3.2 | |
| 2 | $0.5\lambda \leq d \leq 3\lambda$ $0 \leq \theta \leq \pi$ | $\sim 5.85\lambda^{(1)}$ | $\pi/2$ | ~ 18.3 | 18.11 | |
| 3 | $0 \leq \beta \leq 4$ $0 \leq \theta \leq \pi$ | 0.5, 1.5, 2.5, 3.5 | $\pi/2$ | ~ 7.05 | 6.15 | |
| 4 | $0.5\lambda \leq L_{arm} \leq 1.5\lambda$ $\frac{\pi}{18} \leq \alpha \leq \frac{\pi}{2}$ | 1.5λ | 0.834 | ~ 5.8 | 4.8 | |

| (b) | | | | | | |
|-----------|-----------|-------|--|---------------|--------------------|-------------------|
| Problem # | # Dipoles | N_d | Domain | d_i | D_{\max}^{PBM} | D_{\max}^{NEC4} |
| 5 | 6 | 5 | $0.5\lambda \leq d_i \leq 15\lambda$ $1 \leq i \leq 5$ | 0.99λ | $\sim 11.25^{(2)}$ | 11.22 |
| 5 | 10 | 9 | $0.5\lambda \leq d_i \leq 15\lambda$ $1 \leq i \leq 9$ | 0.99λ | ~ 19 | 19.10 |
| 5 | 16 | 15 | $0.5\lambda \leq d_i \leq 15\lambda$ $1 \leq i \leq 15$ | 0.99λ | ~ 31 | 30.97 |
| 5 | 24 | 23 | $0.5\lambda \leq d_i \leq 15\lambda$ $1 \leq i \leq 23$ | 0.99λ | ~ 47 | 46.88 |

Note: ⁽¹⁾ Values marked with ~ are estimated from Figure 6, Figure 9 or Figure 11 in [6]; ⁽²⁾ Values marked with ~ are estimated from Figure 13 in [6].

3. CFO Performance Summary

3.1. Effectiveness

This section summarizes CFO’s performance (Appendix A contains detailed discussions of each benchmark). CFO’s *effectiveness* is measured by how accurately it locates the PBM maxima (coordinates and maxima values, that is, best fitnesses). **Table 3** summarizes the coordinate results and **Table 4** the fitnesses (“Δ” denotes difference in values). Inspection of these tables shows that the agreement generally is quite good.

Turning to **Table 3**, for 2D problems #1 through #4, respectively, the coordinates agree to within (1.12%, 1.9%), (1.26%, 0.89%), (3.95%, 0.16%), and (0.03%, 14.75%) [relative to PBM’s coordinates]. The only significant disagreements are in the abscissas for problem #3 and the ordinates for problem #4. In the case of problem #3, the location of the global maximum at $\beta = 0.5$, $\theta = \frac{\pi}{2}$ is known analytically. No computed value was reported in [6], so that the true degree of agreement is not known. For problem #4, the discrepancy in the inner angle α may be a result of modeling differences, or possibly using different versions of NEC. Of course, from a practical (read “engineering”) point of view, agreement to within about 15% probably is acceptable. NEC4 guidelines suggest that an Average Gain Test (AGT) value within about 15% indicates a good EM model. All models discussed here meet that test. For problem #5, the dipole separation for maximum directivity agreed to within 1% for all array sizes, which is quite good. If the 24-element array, for which $d_i = 1\lambda$, is excluded, then the agreement is even better, to within 0.7%.

Table 3. Comparison of PBM and CFO maxima coordinates.

| PBM Problem # | PBM Coordinates | | CFO Coordinates | | Δ | |
|------------------|---------------------------------|---------|---------------------------------|---------|------------------------------|-------------------------|
| | x_1 | x_2 | x_1 | x_2 | $x_1^{PBM} - x_1^{CFO}$ | $x_2^{PBM} - x_2^{CFO}$ |
| 1 | 2.58λ | 0.63 | 2.55088λ | 0.61805 | 0.02912 λ | 0.01195 |
| 2a [no noise] | $\sim 5.85\lambda$ | $\pi/2$ | 5.92359λ | 1.55685 | -0.07359 λ | 0.01395 |
| 2b [noise] | nr ⁽¹⁾ | nr | 6.93601λ | 1.54721 | --- | --- |
| 3 | 0.5 | $\pi/2$ | 0.48024 | 1.57327 | 0.01976 | -0.00247 |
| 4 | 1.5λ | 0.834 | 1.49520λ | 0.71098 | 0.00048 λ | 0.12302 |
| --- | $d_i, i = 1, \dots, N_{el} - 1$ | | $d_i, i = 1, \dots, N_{el} - 1$ | | $MAX(d_i^{PBM} - d_i^{CFO})$ | |
| 5 | 0.99 λ | | 0.98310 $\lambda - 1\lambda$ | | 0.01 λ | |

Notes: ⁽¹⁾ Not reported in [6].

Table 4 compares CFO’s and PBM’s best fitnesses as measured by the difference of directivities, $D_{max}^{PBM} - D_{max}^{CFO}$. The differences are 3.43%, 0.36%, 27.74%, and 1.47%, respectively, for problems #1 through #4 [relative to PBM values]. The agreement is quite good, except for problem #3, which would be troublesome but for the data in **Table 2(a)**. NEC4 returned a directivity 6.15 at the points $(\beta, \theta) = \left(i - 0.5, \frac{\pi}{2}\right)$, $i = 1, 2, 3, 4$, instead of 7.05 as reported in [6]. If, in fact, 6.15 is the correct value, then the difference decreases to a more modest 5.47%. At a minimum, there appears to be a question concerning the accuracy of the result reported in [6], and CFO’s overall performance suggests that it accurately located the first global maximum for problem #3 after all. On problem #5 the directivity values are in excellent agreement across all array sizes. The differences range from

only 0.08% ($N_{el} = 16$) to 0.52% ($N_{el} = 10$). In this case CFO recovered the PBM's suite best fitnesses with very high accuracy.

Table 4. Comparison of PBM and CFO best fitnesses.

| PBM Problem # | CFO Results | | | D_{\max}^{PBM} | Δ $D_{\max}^{PBM} - D_{\max}^{CFO}$ |
|--------------------|---------------------------------|---------|------------------|--------------------|---|
| | x_1 | x_2 | D_{\max}^{CFO} | | |
| 1 | 2.55088λ | 0.61805 | 3.20627 | 3.32 | 0.11373 |
| 2a (without noise) | 5.92359λ | 1.55685 | 18.3654 | $18.3^{(1)}$ | -0.0654 |
| 2b (with noise) | 6.93601λ | 1.54721 | 18.6880 | nr ⁽²⁾ | nr |
| 3 | 0.48024 | 1.57327 | 6.48634 | $7.05^{(1)}$ | 0.56366 |
| 4 | 1.49520λ | 0.71098 | 5.71479 | $5.8^{(1)}$ | 0.08521 |
| --- | $d_i, i = 1, \dots, N_{el} - 1$ | | --- | --- | --- |
| 5 (6 el) | 0.99105λ | | 11.2202 | $\sim 11.25^{(3)}$ | 0.0298 |
| 5 (7 el) | 0.98310λ | | 13.1826 | nr | --- |
| 5 (10 el) | 0.99421λ | | 19.0985 | $\sim 19^{(2)}$ | -0.0985 |
| 5 (13 el) | 0.99629λ | | 25.0611 | nr | --- |
| 5 (16 el) | 0.98958λ | | 30.9742 | $\sim 31^{(2)}$ | 0.0258 |
| 5 (24 el) | 1.00000λ | | 46.8813 | $\sim 47^{(2)}$ | 0.1187 |

Notes: ⁽¹⁾ Values marked with are estimated from the figures in [6]. ⁽²⁾ nr – not reported in [6]. ⁽³⁾ Values marked with ~ are estimated from Figure 13 in [6].

These data show that CFO essentially accurately recovered the global maximum for every one of the five PBM benchmark problems, with the caveat that it located only one of the four maxima for problem #3. This performance is better than any of the algorithms reported in [6]. None of those algorithms achieved 100% effectiveness. CFO therefore is very effective against the PBM antenna benchmarks.

3.2. Efficiency

An EA's *efficiency* is measured by how computationally intensive it is. As the detailed results below show, CFO generally converges quickly, at least to the vicinity of a global maximum if trapping has not occurred. After its usually rapid rise, the best fitness often continues to increase, but much slower pace with only slight additional gains.

Here, efficiency is measured by the total number of calculations (objective function evaluations) required for the best fitness to saturate. The PBM paper examines four algorithms: i) *GA-FPC*, ii) *μGA*, iii) *GA-RC*, and iv) *PSO* (see [6] for details). The first three are GA variants, while the fourth is a particle swarm. Because these algorithms are stochastic, each one was run twenty times to develop

statistics. The “mean hit time” in Table II in [6] is the average generation number at which the global maximum was located. Thus, the *average* number of function evaluations is the product of mean hit time and the population size. **Table 5** shows this measure for the PBM algorithms, along with the total number of objective function evaluations for saturation of CFO’s best fitness [$N_{eval} = (j^* + 1)N_p$, where j^* is the CFO step at which the fitness saturated, and N_p is the number of CFO “probes” (see §7.2)]. In **Table 5**, the *best* values are shown in bold red font, the *second best* in bold blue.

On problem #1 CFO performed much better than all the other algorithms, requiring only 60 calculations to locate the global maximum compared to 1530 for the next best algorithm, *PSO*. On problem #2, without and with noise, CFO did not perform as well as *GA-FPC* and μ *GA* by a factor of about 2 to 3. CFO was on par with *GA-RC*, and it did substantially better than *PSO*. For problem #3, CFO out-performed all the algorithms except *PSO*, whose performance was about 17% better. Two different CFO initial probe distributions were used on problem #4, with the result that CFO’s performance was comparable to *GA-FPC*’s, and quite a bit better than μ *GA*’s and *GA-RC*’s by at least a factor of 2. Compared to *PSO*, however, CFO did not do as well by a factor of about 3. Problems #5a and 5b in [6] (Table II) are 7 and 13 element collinear arrays, respectively [6] (p. 1119). For problem #5a CFO was nearly 15 times more efficient than the next most efficient algorithm, *PSO*. On problem #5b, CFO was more than 12 times more efficient, again compared to *PSO*.

Table 5. CFO efficiency (# function evaluations).

| Problem # | CFO N_{eval} | Results from PBM Paper [6], Table II (mean hit time x population) | | | |
|-----------------|------------------------------------|--|-----------------|-------------------|-------------|
| | | <i>GA-FPC</i> | μ <i>GA</i> | <i>GA-RC</i> | <i>PSO</i> |
| 1 | 60 | 3140 | 5065 | 8920 | 1530 |
| 2a [no noise] | 1320 | 360 | 450 | 1400 | 2280 |
| 2b [with noise] | 768 | | | | |
| 3 | 1050 | 1940 | 1685 | 5040 | 900 |
| 4 | 1488 ⁽¹⁾ 1155 | 1300 | 3125 | 3800 | 330 |
| 5a | 72 | 1220 | 1700 | nr ⁽²⁾ | 1050 |
| 5b | 144 | 3480 | 5695 | nr | 1770 |

Notes: ⁽¹⁾ Different initial probe distributions; ⁽²⁾ Not reported in [6].

CFO therefore more than holds its own in terms of computational efficiency, especially for an algorithm that is not as highly developed as those reported in [6]. Out of six problems, CFO performed the best on three, out-performing the other algorithms by a very wide margin. On problem #3, CFO turned in the second best

result, but it was close to the best efficiency. On problem #4, CFO was the second best performer. And on problem #2 it was the third most efficient algorithm. Across the six problems, CFO was best on three and second best on two. *PSO* turned in the next best overall performance, placing first on two problems and second on three. On the one problem where CFO did not finish in the top two, it was the third best. Thus, CFO has clearly proven to be very competitive in terms of efficiency.

4. A Pseudorandom CFO Using π Fractions

An important consideration in global search and optimization (GSO) is uniformity in randomly generated sample points because how well the decision space is sampled hangs in the balance. Sample points should be generated using a truly uniformly distributed random variable calculated from a probability distribution, but most pseudo random sequence generators fall short because their points in fact are not uniformly distributed (see the detailed discussion in [15]).

Design and optimization (D&O) problems in applied EM, in particular antenna D&O, often rely on global search and optimization metaheuristics based on Nature-inspired metaphors. Because these algorithms are inherently stochastic each run produces different results. These uncertain outcomes can be mitigated by pseudo randomly sampling the decision space. The π fractions provide a method for creating deterministic uniformly distributed decision space sample points.

4.1. π Fractions

The π fractions comprise a set of random numbers extracted from the mathematical constant π that are uniformly distributed on the interval [0,1). They are generated by the Bailey, Borwein, Plouffe (BBP) hexadecimal digit extraction algorithm [16]. In an inherently deterministic algorithm like CFO, the π fractions can improve DS exploration by adding a measure of pseudorandomness [17]. Used in an inherently stochastic algorithm like π GASR [15] the π fractions render an otherwise stochastic algorithm effectively deterministic.

A major advantage of determinism is that every optimization run with the same setup yields precisely the same results. This characteristic allows the algorithm designer to quickly and with certainty evaluate the effects of changes such as different run parameters or different fitness functions. By contrast, such evaluations using a stochastic algorithm require many independent runs, often hundreds or even thousands, to generate adequate statistics, and even then the results are imprecise because of their statistical nature. Having to make so many runs can be a very serious limitation in real-world optimization problems, especially ones that require relatively long-running external modeling engines such as NEC with highly segmented antennas [17].

Alternatives to the π fraction approach include using a compiler's built-in random number generator or a separately coded routine employing the same "seed" value on successive runs in order to generate a repeatable sequence of ostensibly

“random” numbers. These approaches, however, run the risk of creating undesirable bi-dimensional correlations in high dimensionality decision spaces, an effect seen, for example, in Halton and van der Corput low-discrepancy sequences [18] [19]. In many cases, the correlations are visually striking as seen in [20] [21]. In contrast, undesirable correlations in the π fractions are readily avoided by not using them in their order of occurrence [15].

In the present work, algorithms π CFO and π GASR [15] both employ the following π fraction pseudocode instead of using a compiler’s built-in random number generator. This procedure generates random real numbers $a \leq r_i < b$ and mitigates the correlation problems discussed above (note that initialization values are completely arbitrary). Algorithm π GASR is based on the modified GA described in [22] to which π fractions have been added.

$\pi_i : \{\pi \text{ fractions in order of occurrence; } i=1, \dots, N_\pi\}$

Initialize: $N_\pi \leftarrow 215,830$: $init_1 \leftarrow 17$
 $init_2 \leftarrow 22$: $inc \leftarrow 5$: $i \leftarrow init_1$

Procedure R_π : generates a random number
in $[a,b)$ using π fractions

$r_i = a + (b-a) \pi_i$
 $i \leftarrow i + inc$
if $i > N_\pi$ $\therefore i \leftarrow init_2$

End

Pseudocode for π Fraction Random Numbers used
in Algorithms π CFO and π GASR

4.2. Utility of π Fractions in CFO

Under certain conditions, CFO is prone to trapping at local maxima or global maxima (for functions like PBM #3 that have multiple global maxima). When this happens oscillation in D_{avg} (average distance, CFO probe with the best fitness and all other probes, see §7.2) appears to be a sufficient but not necessary indicator. Typically, the oscillation in D_{avg} is on a plateau-like region, often flat but not necessarily. An example of this effect is seen in the 30-dimensional ($N_d = 30$) Step function which is a standard benchmark defined as $f(x) = -\sum_{i=1}^{N_d} (\lfloor x_i - x_o + 0.5 \rfloor)^2$, $-100 \leq x_i \leq 100$, $i = 1, \dots, N_d$. The Step is unimodal and highly discontinuous. In this paper, the usual global maximum of zero has been offset from the origin to the point (x_o, \dots, x_o) where $x_o = 75.123$. Perspective plots of the 30-D step appear in **Figure 1(a)** and **Figure 1(b)** (entire domain and region of global maximum, respectively).

Using a π fraction IPD appears to defeat local trapping as shown in the D_{avg} plots in **Figure 2(a)** and **Figure 2(b)** using, respectively, a uniform on-axis probe IPD and a random π fraction IPD. The oscillation associated with trapping is quite apparent in **Figure 2(a)**, but none is evident in **Figure 2(b)**. The first plot comprises two oscillatory plateaus connected by a substantial jump in D_{avg} . But in contrast no such oscillation is evident in **Figure 2(b)**.

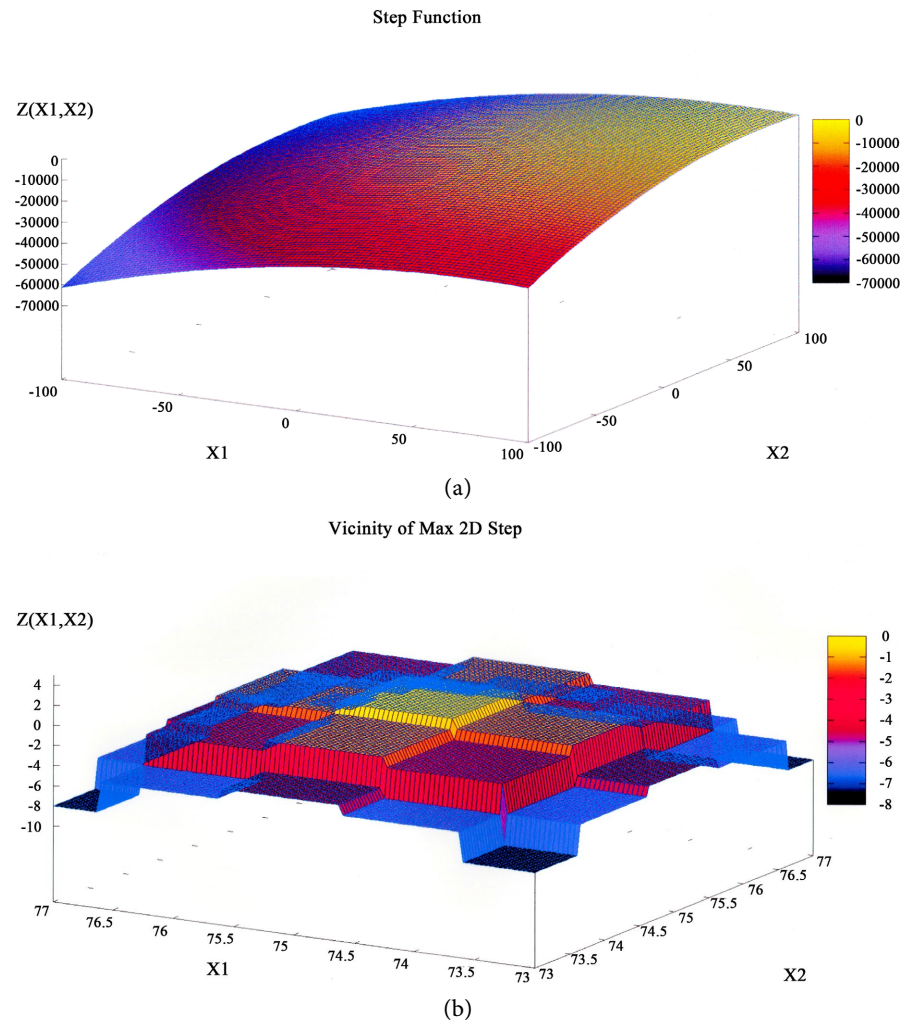


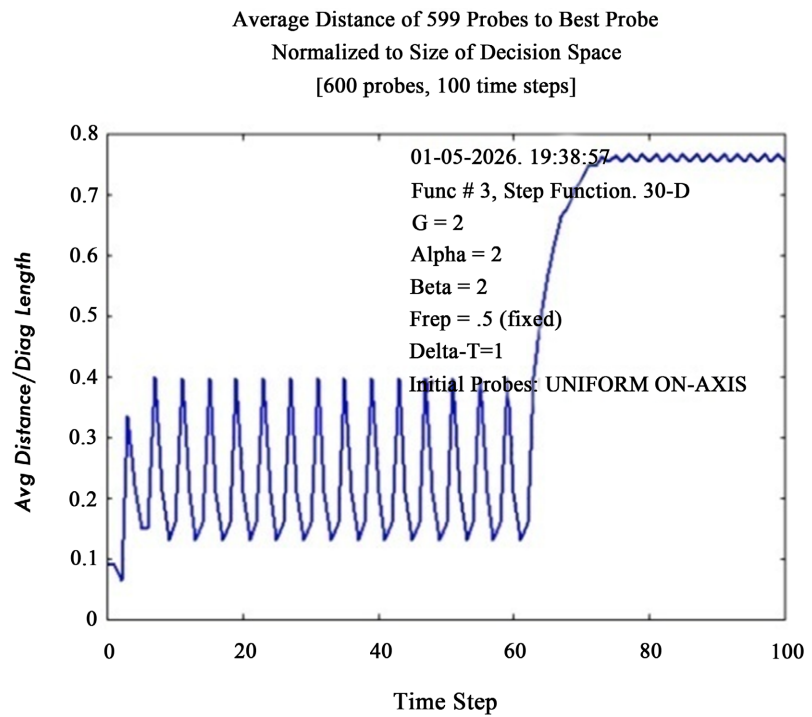
Figure 1. (a) 2-D Step over domain; (b) 2-D Step near global max.

5. Algorithms Compared

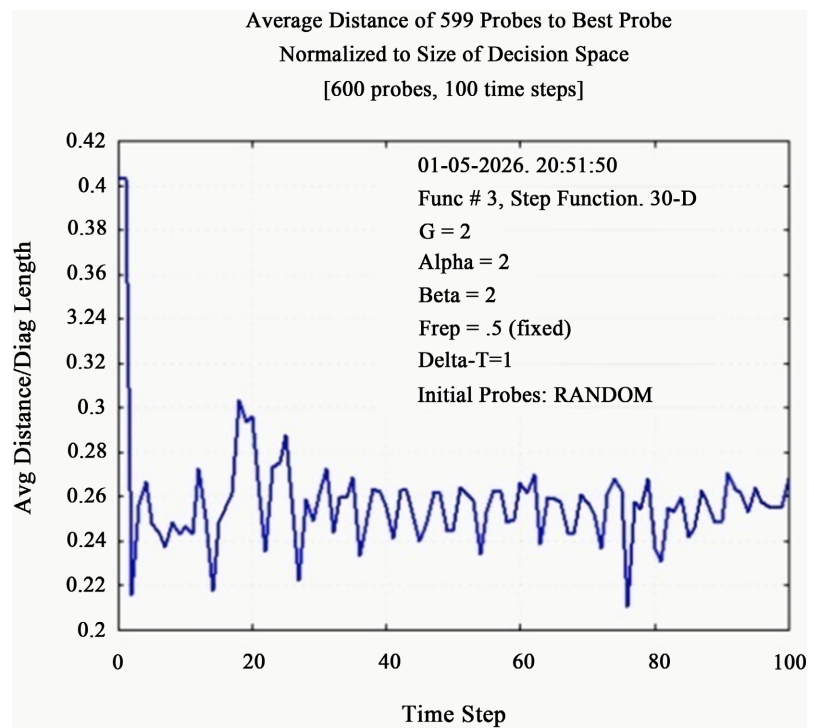
Data from three algorithms, CFO, π CFO, and π GASR are compared to the published PBM data. The PBM suite was initially studied using completely deterministic CFO without π fraction pseudorandomness, although a measure of pseudorandomness was otherwise introduced as described in [23]. Initial probe accelerations were set to zero, the Repositioning Factor (F_{rep}) was initialized and incremented deterministically, and a deterministic “Probe Line” Initial Probe Distribution (IPD) was used (for details see [23]). By contrast, the present version, π CFO, is pseudorandomized using π fractions to compute the initial probe accelerations, F_{rep} , and the IPD. Run parameters were the same as in [15] with DS shrinking and early termination checking. Only *one* π CFO run was made for each benchmark because with π fractions algorithm remains completely deterministic.

The second randomized algorithm applied to PBM is π GASR (Genetic Algorithm with Sibling Rivalry, discussed in detail in [15]). Because the original GASR is inherently stochastic, calls to the compiler’s built-in random number generator were replaced by the π fraction procedure described above (run parameters oth-

erwise the same as in [15]). The best π GASR fitness over multiple runs is reported here because the very purpose of using π fractions is to eliminate true randomness by replacing it with deterministic pseudorandomness.



(a)



(b)

Figure 2. (a) Uniform IPD; (b) π fraction IPD.

6. Comparison Results

Table 6 compares the best fitness results for algorithms CFO, π CFO and π GASR to the published PBM data. While there are slight differences, overall the maximum directivity data are quite similar for all five problems. Consistency is very high on PBM #5 and somewhat lower on the others. In all cases, the best fitness is close to the reported PBM data, but, as pointed out above, some (minor) questions arise concerning the accuracy of some of the published PBM results. A fair reading of these data is that all the tested algorithms provide maximum directivities close to the actual maxima, and that the π fraction approach is effective in making stochastic π GASR deterministic so that only a single run is actually required instead of many.

Table 6. Best fitness.

| PBM Benchmark | Maximum Directivity | | | |
|------------------|-----------------------------|---------|-----------|------------|
| | PBM | CFO | π CFO | π GASR |
| 1 | 3.32 | 3.20627 | 3.24340 | 3.25837 |
| 2a (no noise) | 18.3 ⁽¹⁾ | 18.3654 | 18.2810 | 17.9473 |
| 2b (noisy) | <i>nr</i> ⁽²⁾ | 18.6880 | 19.7609 | 18.8314 |
| 3 | 7.05 ⁽¹⁾ | 6.48634 | 6.57766 | 6.57658 |
| 4 | 5.8 ⁽¹⁾ | 5.71479 | 5.29663 | 5.29663 |
| 5 (6 el) | \sim 11.25 ⁽³⁾ | 11.2202 | 11.2202 | 11.2202 |
| 5 (7 el) | <i>nr</i> | 13.1826 | 13.0918 | 13.1826 |
| 5 (10 el) | \sim 19 ⁽²⁾ | 19.0985 | 19.0985 | 19.0985 |
| 5 (13 el) | <i>nr</i> | 25.0611 | 25.0035 | 25.0035 |
| 5 (16 el) | \sim 31 ⁽²⁾ | 30.9742 | 30.9742 | 30.9742 |
| 5 (24 el) | \sim 47 ⁽²⁾ | 46.8813 | 46.8813 | 46.8813 |

Notes: ⁽¹⁾ Values estimated from the figures in [6]; ⁽²⁾ *nr* – not reported in [6]; ⁽³⁾ Values marked with \sim are estimated from Figure 13 in [6].

Coordinates for the best fitnesses are shown in **Table 7**. These data also show a high degree of consistency between the tested algorithms except for π GASR on PBM #2a where the x_1 coordinate is different from the other two algorithms'. This disagreement also shows up to a lesser degree in π GASR's maximum directivity in **Table 2**. On PBM #5 the dipole element separations are all close to 0.99λ with the greatest variability returned by algorithm π CFO. The differences in d_i , however, have no effect on the best fitness as seen in the remarkably consistent data in **Table 6**.

Table 7. Best fitness coordinates.

| PBM Benchmark | PBM | | CFO | | π CFO | | π GASR | |
|---|--------------------|--------|-------------------|--------|-----------------------|--------|-------------------|--------|
| | x_1 | x_2 | x_1 | x_2 | x_1 | x_2 | x_1 | x_2 |
| 1 | 2.58λ | 0.63 | 2.5509λ | 0.6181 | 2.5896 | 0.6195 | 2.5845 | 0.6198 |
| 2a (no noise) | $\sim 5.85\lambda$ | 1.5730 | 5.9236λ | 1.5569 | 5.9246 | 1.5554 | 6.9270 | 1.5467 |
| 2b (noisy) | nr ⁽¹⁾ | nr | 6.9360λ | 1.5472 | 5.8877 | 1.5560 | 9.8907 | 1.5230 |
| 3 | 0.5 | 1.5730 | 0.4802 | 1.5733 | 2.4806 | 1.5611 | 1.5201 | 1.5704 |
| 4 | 1.5λ | 0.834 | 1.4952λ | 0.7110 | 1.4913 | 0.7176 | 1.4942 | 0.7317 |
| Min/Max $d_i, i = 1, \dots, N_{el} - 1$ | | | | | | | | |
| 5 | 0.99 λ | | 0.983/1 λ | | 0.974/1.199 λ | | 0.987/1 λ | |

Notes: ⁽¹⁾ nr – not reported in [6].

π fractions have been shown to be an effective approach to mitigating the inconsistency inherent in stochastic global search and optimization. Here, the case for using π fractions is made by way of example using the PBM antenna benchmarks as representative real-world problems requiring an external modeling engine for fitness calculation. Stochastic algorithms require multiple runs to build meaningful statistics, often runs numbering in the hundreds or more. This requirement usually is not a limitation when optimizing analytical benchmarks because fast built-in routines are used, but it easily becomes a very significant impediment when a long-running external modeling engine is required as is often the case for real-world problems. π fractions address this concern by rendering deterministic (pseudorandom) an otherwise probabilistic optimizer so that only a single run is required. π fractions also are useful in inherently deterministic algorithms by injecting a measure of pseudorandomness that may improve an algorithm's ability to explore the decision space.

7. The CFO Metaphor

There are many nature-inspired search and optimization EAs, as examples metaheuristics based on animal and insect behaviour, human population behaviour, or physical processes such as water drop pooling. Most of these algorithms are fundamentally stochastic because the processes they analogize are inherently random. In fact, their ability to locate maxima derives from this inherent randomness. Without it, these EAs cannot function, they fail completely.

CFO, by contrast, is completely deterministic because it is based on the metaphor of gravitational kinematics, the branch of physics that computes the motion of masses moving solely under the influence of gravity. Because gravity is deterministic, so too is CFO. This feature is a significant improvement over stochastic algorithms. Every CFO run with the same setup parameters returns precisely the same results.

But, of course, as with all deterministic methodologies, the downside is a tendency for CFO to become trapped at local maxima. In fact, the two major issues in CFO's further development appear to be (1) finding effective ways of mitigating trapping and (2) establishing a precise methodology for specifying run parameters. This section describes CFO's basic equations and discusses these issues, in particular oscillation in D_{avg} (defined below) as a harbinger of trapping and recent results in the astronomy literature that may be useful in understanding this phenomenon.

Even though CFO is fundamentally deterministic, the algorithm designer is free to inject some randomness if doing so improves performance. When local trapping occurs, pseudorandom perturbation of CFO's "probe" distribution may mitigate the trapping. The judicious injection of some measure of randomness has not been investigated here, but it has been used in other CFO implementations with very good results [23]. The author emphasizes CFO's deterministic nature because CFO is one of very few algorithms that provide true reproducibility and control over computed results. This characteristic may be especially useful in parameter-tuning implementations in which CFO's run parameters are adjusted using real-time feedback such as discussed in [24]. Some CFO implementations may benefit from a pseudorandom component, and that possibility certainly merits consideration. In fact, in [23] it is recommended for all CFO implementations.

7.1. CFO, NEOs and Energy

This section asks an interesting question, but does not answer it: Is there a connection between the orbital trajectories of Near Earth Objects (NEOs) and CFO? The basis for this question is an obvious similarity between the required velocity curve to avoid Earth collision with an NEO and CFO's "average distance" plot. The possibility of a connection is entirely speculative, and it is raised to hopefully encourage the research that may resolve it.

An NEO, for example a large asteroid, approaching planet Earth can become gravitationally "trapped" for a while. As a general rule, in the absence of friction or some other energy dissipation mechanism, an object's trajectory will depend only upon its (constant) total energy. Recent work on gravitational "resonant returns" [25], however, shows that certain objects closely approaching the Earth can exchange energy with the planet without dissipation and, for some time at least, become gravitationally trapped in a tight orbit. If the orbit is such that the object passes through a small region of space known as a "keyhole", then a collision with the Earth may occur. NEO's not entering the keyhole cannot collide.

Many NEO's are asteroids that fall into this category and can be quite large, for example Apophis with approximate dimensions 340×450 meters. If an object of solid rock with those dimensions impacts the Earth the result likely is catastrophic, possibly cataclysmic. One way to avoid such an encounter is to "nudge" the object into a different orbit by slightly changing its velocity. Extending the model in [25] Schweikart *et al.* [26] calculate the required velocity change for asteroid Apophis to avoid the keyhole. **Figure 2** in [26], reproduced below with permission, plots

the velocity change Δv required to modify the Apophis' trajectory to avoid a collision with Earth as a function of when the encounter occurs (**Figure 3**).

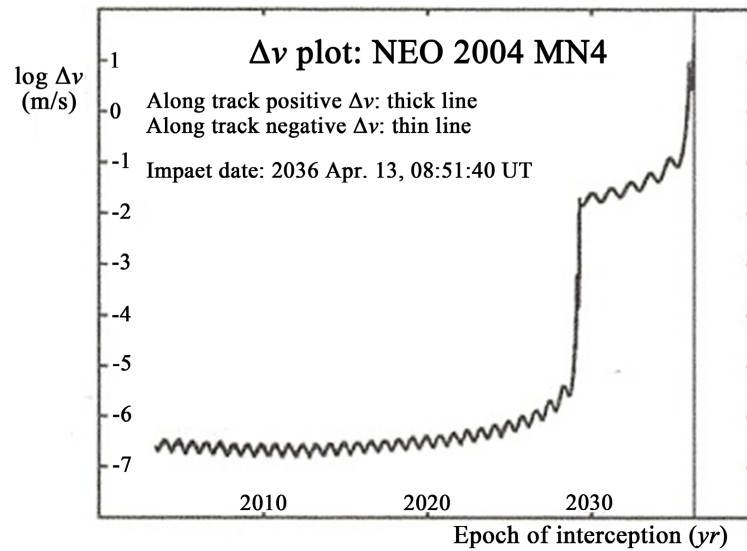


Figure 3. Reproduction of **Figure 2** in Schweickart [16].

The Apophis Δv plot is particularly interesting because of its unusual structure, two oscillatory plateau-like regions separated by a rapid rise in Δv . The oscillatory behaviour is a direct result of actual gravitational trapping in the physical Universe. An example is the MOID (Minimum Orbital Intersection Distance) plot in **Figure 4** in [25] which shows a similar oscillatory behavior. When MOID is zero impact occurs.

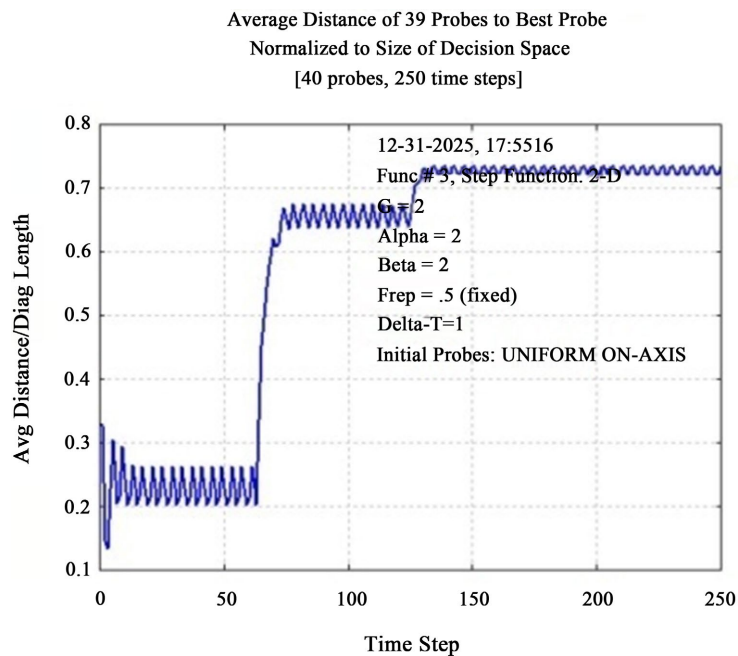


Figure 4. D_{avg} , 2D step function.

The CFO-NEO nexus is a strikingly similar appearance of some of CFO's D_{avg} plots. An example is provided by the two-dimensional Step function

$f(x) = -\sum_{i=1}^2 (\lfloor x_i - x_0 + 0.5 \rfloor)^2$, $x_0 = 75.123$ whose D_{avg} curve (average distance, CFO probe with the best fitness and all other probes) along with perspective views of the 2D Step.

Not only is the structural similarity between the Apophis Δv plot and CFO's D_{avg} self-evident, it is quite remarkable and not limited only to the curves' appearances. The Apophis Δv and the resulting distance nudge Δd are proportional, that is $\Delta d = \Delta v \cdot t$, t being the time, the epoch of encounter, while the Step's D_{avg} is essentially the same quantity where the CFO run step number is in effect the time. Thus, the relationship between gravitational trapping in the real Universe and trapping near a maximum in metaphorical CFO space appears to be more than simply visual similarity. In the author's opinion it is difficult to imagine that the Apophis curve, which is based on actual NEO gravitational trapping, and the CFO D_{avg} curve, which is based on the metaphor of gravitational kinematics, are *not* manifestations of the same or, at a minimum, very similar underlying phenomena. Although admittedly speculative, the author believes that this similarity is compelling evidence of the validity of CFO's gravitational metaphor, and that it might point the way to techniques for mitigating trapping. In particular, the analysis of Valsecchi *et al.* [25] may be directly applicable to improving CFO so that it avoids local trapping.

Another possible application of the conservative energy exchange between a small object and a larger gravitating mass in a trapping encounter is using the object's total energy, E_{tot} , as a measure of trapping. This suggestion, too, is speculative in nature, but it reflects what appears to be a strong correlation between real gravitational kinematics and CFO's metaphorical search methodology. If the effect of gravitational trapping is to modify E_{tot} , then with a suitable definition of energy in CFO space E_{tot} may become a key factor in signalling trapping and possibly providing a mitigation mechanism simply by increasing its value. It may also be possible to re-formulate the basic CFO equations in terms of an energy model. Whether or not this approach has any merit is a question for researchers far more capable than the author, one that he hopes will be pursued.

7.2. The CFO Equations

CFO addresses the following problem: Locate the global maxima of the objective function $f(x_1, x_2, \dots, x_{N_d})$ defined on the decision space (DS) $x_i^{\min} \leq x_i \leq x_i^{\max}$, $1 \leq i \leq N_d$, the x_i being *decision variables*, where i the coordinate number, and N_d the space's dimensionality (number of coordinates). *Fitness* refers to the value of $f(\bar{x})$ at the point \bar{x} . The objective function's topology, which is unknown, may be continuous or discontinuous, highly multimodal or unimodal, and possibly subject to a set of constraints Ω among the decision variables. While many optimization algorithms search for minima, CFO searches for maxima, that is, minimizing $-f(\bar{x})$.

CFO “flies” a group of “probes” through DS. Their trajectories are computed from the *equations of motion* for each probe’s acceleration and position vector. These equations are derived by generalizing Newton’s universal law of gravitation and law of motion from three-dimensional physical space to metaphorical N_d -dimensional “CFO space.” The resulting equations are [1]-[4]:

$$\bar{a}_{j-1}^p = G \sum_{\substack{k=1 \\ k \neq p}}^{N_p} U(M_{j-1}^k - M_{j-1}^p) \cdot (M_{j-1}^k - M_{j-1}^p)^\alpha \times \frac{(\bar{R}_{j-1}^k - \bar{R}_{j-1}^p)}{|\bar{R}_{j-1}^k - \bar{R}_{j-1}^p|^\beta} \tag{1}$$

$$\bar{R}_j^p = \bar{R}_{j-1}^p + \frac{1}{2} \bar{a}_{j-1}^p \Delta t^2, \quad j \geq 1 \tag{2}$$

The probe number is $1 \leq p \leq N_p$, and the time step (iteration) number is $0 \leq j \leq N_t$, where N_p and N_t are the *total* number of probes and the *total* number of time steps, respectively. \bar{a}_{j-1}^p is “probe” p ’s acceleration at time step (iteration) $j-1$. $\bar{R}_j^p = \sum_{k=1}^{N_d} x_k^{p,j} \hat{e}_k$ is its position vector at step j , where the $x_k^{p,j}$ are probe p ’s coordinates at step j , and \hat{e}_k is the unit vector along the x_k axis. Note that the version of eq. (2) in [1] contains a “velocity” term not included in the CFO formulation in [3]. The velocity term was dropped because including it actually impeded convergence (reasons unknown). In [1], however, this term already had been set to zero as a matter of convenience, so that from the outset it never was required, and consequently is no longer included.

CFO “flies” its N_p probes through the decision space as a function of time (iteration number) along trajectories calculated using equations (1) and (2). A new probe distribution is computed at each step, and at the location of each probe the objective function’s fitness is computed. For example, at time step $j-1$ at probe p ’s location, the fitness is $M_{j-1}^p = f(x_1^{p,j-1}, x_2^{p,j-1}, \dots, x_{N_d}^{p,j-1})$. Note that M does *not* mean “mass” (see [1] for a discussion of the symbology). Each of the other probes has a fitness $M_{j-1}^k, k = 1, \dots, p-1, p+1, \dots, N_p$, associated with it at step $j-1$.

The function $U(\cdot)$ in eq. (1) is the Unit Step function, $U(z) = \begin{cases} 1, & z \geq 0 \\ 0, & \text{otherwise} \end{cases}$.

Following standard notation, the magnitude of vector \bar{B} is $|\bar{B}| = \left(\sum_{i=1}^{N_d} b_i^2 \right)^{\frac{1}{2}}$,

where the b_i are its scalar components. G is CFO’s “gravitational constant,” and Δt the “time” interval between steps during which the acceleration is constant. It is important to remember that this terminology is chosen solely to reflect CFO’s gravitational metaphor, as is the factor $\frac{1}{2}$ in eq. (2), and beyond that there is no significance to the nomenclature. α and β are the CFO exponents. While G and Δt have direct analogs in the equations of motion for gravitationally controlled real masses moving through physical space, there is no analog in Nature for α and β . They are included to provide added flexibility in how CFO is implemented by giving the algorithm designer the freedom to change how

CFO's "gravity" varies with mass or distance, or both, in order to achieve a more effective algorithm. Of course, α and β are perfectly acceptable parameters in metaphorical "CFO space," because there gravity can be whatever the designer wants it to be.

The Unit Step function in eq. (1) is extremely important because it creates positive-definite CFO mass. Under the gravitational analogy on which CFO is based objects in CFO space have "mass" just as they do in the physical Universe. But unlike real objects, mass in CFO space is a *user-defined* function of the objective function's fitness (not necessarily the fitness value itself). The problem is, depending on how the user defines that function, the mass can be positive or negative. And negative mass leads to serious problems!

In this paper CFO "mass," by definition, is $U(M_{j-1}^k - M_{j-1}^p) \cdot (M_{j-1}^k - M_{j-1}^p)^\alpha$, that is, the difference in fitness values raised to the α power multiplied by the Unit Step. The *difference* of fitnesses intuitively seems to be a good measure of how much "gravitational" influence the value of the objective function at one point in the decision space should have on a probe at another point in the space, but other choices certainly are possible. The exponent α is optional, giving the algorithm designer added flexibility as discussed above. In marked contrast, with the difference of fitnesses definition, the Unit Step is an essential element, because without it the mass could be negative depending on which fitness is greater. In the real world, gravity always is attractive because mass is positive. Negative mass creates a repulsive gravitational force that flies probes away from maxima, instead of the positive gravitational force required to fly probes toward them. This is just the opposite of what CFO is intended to do. The Unit Step solves the negative mass problem by creating positive-definite CFO mass and, consequently, a positive gravitational force that is always attractive.

A frequently used measure of how well CFO's probes have clustered at a maximum is the average distance between the probe with the best fitness and all the other probes, normalized to the size of the decision space (length of the principal diagonal), that is,

$$D_{avg} = \frac{1}{L \cdot (N_p - 1)} \sum_{p=1}^{N_p} \sqrt{\sum_{i=1}^{N_d} (x_i^{p,j} - x_i^{p^*,j})^2}$$

where p^* is the number of the probe with the best fitness, $L = \sqrt{\sum_{i=1}^{N_d} (x_i^{\max} - x_i^{\min})^2}$ is the diagonal

length, and $\vec{R}_j^p = \sum_{k=1}^{N_d} x_k^{p,j} \hat{e}_k$ is probe p 's position vector at step j . Plotting D_{avg}

as a function of the time step is helpful in showing how the probe distribution evolves. As pointed out in §4.1, oscillation in D_{avg} appears to be a sufficient but not necessary condition for local trapping of CFO's probe with the best fitness.

7.3. A New "Hyperspace Directional Derivative"?

The Unit Step introduced in the definition of mass in §7.2 may be important in another, purely mathematical way. It may be the basis for defining a new type of "hyperspace directional derivative." This idea is entirely speculative because no

such mathematical operator currently exists, but the author believes it merits consideration if it is shown to be widely useful in addressing high-dimensionality optimization problems generally. In Nature, gravity is a conservative vector force field that produces “action at a distance”. While its fundamental equations are vector in nature, gravity also can be calculated as the gradient of a scalar potential function, just as the electric field is derived from an electric potential. The general notion of a “gradient” also applies in CFO space, but to some degree its utility depends on how mass is defined. For example, in this CFO implementation with $\alpha = \beta = 2$, the difference in fitness values divided by the distance between probes is essentially the square of the average slope, that is, the square of a derivative. When the Unit Step is included, this term, in a general sense, may be thought of as a sort of “generalized derivative”. However, with entirely different CFO mass definitions, or the present definition with different exponent values, this generalization may not be as apparent or compelling; but in this case it seems both clear and appropriate. This observation suggests the possibility of defining a new mathematical construct, a “hyperspace directional derivative”, that includes the Unit Step so that it always points towards an objective function’s maxima. The author makes this suggestion entirely on a speculative basis. As with the “energy” notions discussed in §7.1, whether or not this suggestion makes any sense is a question left to the mathematicians who hopefully will pursue CFO’s further development.

7.4. CFO Parameters

The CFO implementation described in this paper requires seven user-specified run parameters: α , β , G , Δt , N_t , N_p , and F_{rep} . They are all defined above, except F_{rep} , the “repositioning factor.” The probe initial acceleration could be used as an eighth parameter, but it is zero for all runs reported here. F_{rep} is an algorithmic element and has nothing to do with fundamental CFO theory. For the specific CFO implementation used in this paper, F_{rep} addresses the problem of errant probes. When equations (1) and (2) in §7.2 are used to compute a probe distribution, some “errant” probes may end up outside DS. When that happens the question is, how to get them back? There are many possibilities. For example, the decision space boundary could be made “reflecting” (probes bouncing back into the decision space) or “absorbing” (probes extinguished and replaced with a new ones), and so on. Another possibility is randomly re-initializing the errant probes’ locations, and most of these were tried during CFO’s early development. After evaluating various approaches a simple entirely empirical scheme was developed and is used here “ F_{rep} repositioning” where F_{rep} is the “repositioning parameter”.

On a coordinate-by-coordinate basis, probes flying out of DS are placed a fraction, F_{rep} , of the distance between the probe’s starting coordinate and the corresponding boundary coordinate. But because there is no obvious way to assign a value to $0 \leq F_{rep} \leq 1$, the following simple, deterministic, perhaps clumsy, but workable approach was taken. At each step, the current and previous four fitness

values were stored in a 5-element array. F_{rep} started at a value of 0.5 and was incremented by 0.005 whenever the absolute value of the difference between 5th array element and elements 3, 4, and 5 differed by less than 0.0005. If incrementing F_{rep} in this manner results in $F_{rep} \geq 1$, then it was reset to the starting value, and this procedure repeated with the then current probe distribution. If the data reported here are used for validation, it is important to note that the five saved fitness values are not necessarily sequential because of how the array index was computed (see pseudocode in Appendix B has details), which then affects when F_{rep} is incremented. F_{rep} 's starting value and increment, and the fitness tolerance were determined empirically, that is, on a trial and error basis, as are all CFO parameters used in this and the other CFO papers.

Of all the CFO parameters, F_{rep} and N_p seem to have the greatest effect by far on the algorithm's performance. The sensitivity to F_{rep} is what led to the empirical scheme described above. Changing F_{rep} generally improves convergence, because doing so avoids or terminates local trapping (this effect discovered empirically). The number of probes, N_p , determines how well the decision space is sampled, that is, how much information CFO has about the objective function at the start of a run. Numerical testing shows that CFO's convergence frequently is a sensitive function of how many initial probes are deployed. Another related non-numerical factor is where the initial probes are placed. As some of the cases in §6 show, changing the initial probe distribution may make the difference between capturing the global maximum or missing it.

The question of how best to assign values to CFO's run parameters is very important, yet it remains unanswered. At this point, there is no methodology to do that. Like most EAs, actually using the algorithm for a specific problem or class of problems, that is, setting up and making many runs, results in the user's learning how to make best use of the algorithm, so specifying run parameters is done empirically. Some parameter values work better than others, and the experience gained by making many runs is reflected in the user's choice of parameters in subsequent ones.

Of course, the result is that the user *de facto* becomes a part of the algorithm because every run includes a degree of subjectivity. Needless to say, it would be better to implement an EA using an objective methodology to specify run parameters, but there presently is none for CFO, or for that matter for most EAs. One significant advantage that CFO offers is its determinism. Results that are reproducible from one run to the next coupled with CFO's frequently rapid convergence, even if to a local maximum, lend CFO to implementations that perform real-time parameter adjustments in response to a feedback mechanism. This sort of *Reactive Search* [24] may be particularly appropriate for CFO because the algorithm is fast and deterministic. It consequently is an extension that deserves consideration.

8. Why a Deterministic Optimizer?

When it comes to selecting an optimization program there are countless algo-

rithms to choose from, almost all of which are stochastic in nature. The number of effective fundamentally deterministic algorithms is actually quite limited (see for example [27] for a list). In addition, an important distinction must be drawn between benchmark test functions and real-world problem objective functions, the kind of fitness functions that practicing engineers routinely face.

Real-world problems usually require a *two-step* solution whereas benchmark problems require only one. Optimization algorithms are almost always evaluated against benchmark suites for the very reason that the solutions are known in advance. An algorithm's effectiveness is measured by the quality of its solutions (how close it comes to the known answers) and by the required computational effort (typically number of function evaluations). *At no point is the fitness function itself an unknown, but this is almost always the case in real-world problems.* As a consequence, the *user*, that is, the practicing engineer, must define a suitable objective function before any optimization can be done, and this added requirement makes solving the problem much harder because not all objective functions are well-behaved or for that matter yield good results even if they are. This dichotomy can be illustrated with a straight-forward example problem.

When the very form of the objective function is unknown the entire optimization process must be repeated for *every* candidate function. While this is not a fundamental limitation, as a practical matter in many real-world cases it turns out to be one that simply cannot be overcome because, again as a practical matter, computation times frequently become too long. Evaluating a benchmark test function like the Step is straightforward and quick because it requires only built-in functions (summation, floor, exponentiation), but real problems usually are quite different.

Real-world problems often require an external "modeling engine" that is frequently far more computationally intensive than using built-in functions. In antenna D&O, for example, Finite Element and Method of Moments codes are popular with their attendant long run times. And this limitation renders statistical evaluations of candidate fitness functions completely impractical. As an example, the run time for the 7303-segment vehicle-mounted antenna model shown in **Figure 5** was 369 seconds on a 2.30GHz 8-core Core i7-3610QM machine using a multithreaded version of the Numerical Electromagnetics Code (NEC4/MP, [14]) under 64-bit Windows 7 [17]. An optimization run requiring, say, 1000 evaluations would take more than 102 hours, well over 4 days! And this must be repeated for *each candidate* fitness function!

Knowing that an algorithm is effective against the Step Function tells the practitioner absolutely nothing about how to formulate useful fitness functions and whether or not the algorithm will work well against those functions whatever form they may take. Stochastic algorithms make answers to this and related questions very difficult and consequently are frequently not well-suited to solving the real-world problems routinely encountered by practicing engineers. CFO solves addresses this dilemma, and π fractions offer the potential of turning purely stochastic algorithms into pseudorandom, hence deterministic, ones.

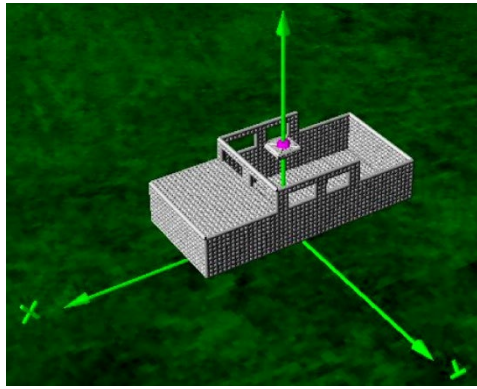


Figure 5. Vehicle-mounted antenna.

9. Conclusions

This paper describes Central Force Optimization, a deterministic, Nature-inspired evolutionary search and optimization metaheuristic that embraces the metaphor of gravitational kinematics. It reports the results of testing CFO against the PBM suite of antenna benchmarks. CFO acquits itself quite well, in many respects better than the more established EAs analyzed in the PBM paper [6]. CFO located all global maxima, even in a noisy decision landscape, and it did so with excellent efficiency. CFO generally performed better than the algorithms reported in [6].

This paper also highlights the striking similarity between velocity curves for Near Earth Objects in close encounters with planet Earth and CFO's average distance curves in the presence of local trapping. The nature of these plots suggests that oscillation is a sufficient but not necessary condition for local trapping, and that the theory of gravitational "resonant returns" might be of use in better understanding and perhaps mitigating trapping.

The concept of "energy" in the gravitational metaphor was discussed, and suggestions made for its use as a test for local trapping and possibly a trapping mitigation mechanism. Also discussed was the possibility that CFO could be re-formulated on the basis of a "total energy" model.

Finally, this paper suggests the possibility of a new mathematical construct, a hyperspace "directional derivative" based on the Unit Step function that always points towards a function's maxima. This idea is admittedly entirely speculative in nature because no such mathematical operator currently exists, yet the author believes it does merit consideration by mathematicians far more capable than he.

Acknowledgements

The author thanks the reviewers for their helpful comments that have improved this exposition. The author also wishes to acknowledge the following individuals who in various ways at various times have helped bring CFO to the attention of the optimization community. If CFO continues to realize what appears to be its considerable potential in no small measure it will be due to their interest and encouragement: Emeritus Professor Kusum Deep, Department of Mathematics, In-

dian Institute of Technology, Roorkee; Professor Mario Pavone, Department of Mathematics, University of Catania, Sicily, Italy; Professor Zhihua Cui, Taiyuan University of Science and Technology, Division of System Simulation and Computer Application, Taiyuan, Shanxi, China; Professor Nihad Dib, Department of Electrical Engineering, Jordan University of Science and Technology, Irbid, Jordan. If, on the other hand, CFO fails to meet this expectation then the responsibility is entirely the author's.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Formato, R.A. (2007) Central Force Optimization: A New Metaheuristic with Applications in Applied Electromagnetics. *Progress In Electromagnetics Research*, **77**, 425-491. <https://doi.org/10.2528/pier07082403>
- [2] Formato, R.A. (2008) Central Force Optimization: A New Nature Inspired Computational Framework for Multidimensional Search and Optimization. In: Krasnogor, N., Nicosia, G., Pavone, M. and Pelta, D., Eds., *Studies in Computational Intelligence*, Springer Berlin Heidelberg, 221-238. https://doi.org/10.1007/978-3-540-78987-1_21
- [3] Formato, R.A. (2009) Central Force Optimisation: A New Gradient-Like Metaheuristic for Multidimensional Search and Optimisation. *International Journal of Bio-Inspired Computation*, **1**, 217-238. <https://doi.org/10.1504/ijbic.2009.024721>
- [4] Formato, R.A. (2010) Central Force Optimization and NEOs—First Cousins? *Journal of Multiple-Valued Logic and Soft Computing*, **16**, 547-565.
- [5] Qubati, G.M., Formato, R.A. and Dib, N.I. (2010) Antenna Benchmark Performance and Array Synthesis Using Central Force Optimisation. *IET Microwaves, Antennas & Propagation*, **4**, 583-592. <https://doi.org/10.1049/iet-map.2009.0147>
- [6] Pantoja, M.F., Bretones, A.R. and Martin, R.G. (2007) Benchmark Antenna Problems for Evolutionary Optimization Algorithms. *IEEE Transactions on Antennas and Propagation*, **55**, 1111-1121. <https://doi.org/10.1109/tap.2007.893396>
- [7] Li, W., Shi, X. and Hei, Y. (2008) An Improved Particle Swarm Optimization Algorithm for Pattern Synthesis of Phased Arrays. *Progress in Electromagnetics Research*, **82**, 319-332. <https://doi.org/10.2528/pier08030904>
- [8] Ghaffari-Miab, M., Farmahini-Farahani, A., Faraji-Dana, R. and Lucas, C. (2007) An Efficient Hybrid Swarm Intelligence-Gradient Optimization Method for Complex Time Green's Functions of Multilayer Media. *Progress In Electromagnetics Research*, **77**, 181-192. <https://doi.org/10.2528/pier07072504>
- [9] Sijher, T.S. and Kishk, A.A. (2005) Antenna Modeling by Infinitesimal Dipoles Using Genetic Algorithms. *Progress in Electromagnetics Research*, **52**, 225-254. <https://doi.org/10.2528/pier04081801>
- [10] Cengiz, Y. and Tokat, H. (2008) Linear Antenna Array Design with Use of Genetic, Memetic and Tabu Search Optimization Algorithms. *Progress in Electromagnetics Research C*, **1**, 63-72. <https://doi.org/10.2528/pierc08010205>
- [11] Zainud-Deen, S.H., Hassen, W.M. and Awadalla, K.H. (2009) Crack Detection Using a Hybrid Finite Difference Frequency Domain and Particle Swarm Optimization

- Techniques. *Progress in Electromagnetics Research M*, **6**, 47-58.
<https://doi.org/10.2528/pierm09012404>
- [12] Mangaraj, B.B., Misra, I.S. and Barisal, A.K. (2008) Optimizing Included Angle of Symmetrical V-Dipoles for Higher Directivity Using Bacteria Foraging Optimization Algorithm. *Progress in Electromagnetics Research B*, **3**, 295-314.
<https://doi.org/10.2528/pierb07121005>
- [13] Yau, D. and Crozier, S. (2003) A Genetic Algorithm/Method of Moments Approach to the Optimization of an RF Coil for MRI Applications—Theoretical Considerations. *Progress in Electromagnetics Research*, **39**, 177-192.
<https://doi.org/10.2528/pier02062602>
- [14] Burke, G.J. (1992) Numerical Electromagnetics Code—NEC-4, Method of Moments, Part I: User's Manual and Part II: Program Description—Theory. UCRL-MA-109338, Lawrence Livermore Nat'l Lab.
http://kirkmcd.princeton.edu/examples/NEC_Manuals/NEC4UsersMan.pdf
- [15] Formato, R.A. (2017) Determinism in Electromagnetic Design & Optimization—Part II: BBP-Derived π fractions for Generating Uniformly Distributed Sampling Points in Global Search and Optimization Algorithms. *FERMAT*, **19**, Article 10.
<https://efermat.github.io/articles/formato-art-2017-vol19-jan-feb-010/>
- [16] Bailey, D., Borwein, P. and Plouffe, S. (1997) On the Rapid Computation of Various Polylogarithmic Constants. *Mathematics of Computation*, **66**, 903-913.
<https://doi.org/10.1090/s0025-5718-97-00856-9>
- [17] Formato, R.A. (2017) Determinism in Electromagnetic Design & Optimization—Part I: Central Force Optimization. *FERMAT*, **19**, Article 9.
<https://efermat.github.io/articles/formato-art-2017-vol19-jan-feb-009/>
- [18] Halton, J.H. (1960) On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals. *Numerische Mathematik*, **2**, 84-90.
<https://doi.org/10.1007/bf01386213>
- [19] van der Corput, J.G. (1935) Verteilungsfunktionen. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen*, **38**, 813-821.
- [20] Krykova, I. (2003) Evaluating of Path-Dependent Securities with Low Discrepancy Methods. Ph.D. Thesis, Worcester Polytechnic Institute.
- [21] Jørgensen, K. (2008) Himalaya Options: An Analysis of Pricing Himalaya Options in a Monte Carlo and Quasi-Random Monte Carlo framework. Master's Thesis, School of Business, Aarhus University.
- [22] Li, W.T., Shi, X.W., Hei, Y.Q., Liu, S.F. and Zhu, J. (2010) A Hybrid Optimization Algorithm and Its Application for Conformal Array Pattern Synthesis. *IEEE Transactions on Antennas and Propagation*, **58**, 3401-3406.
<https://doi.org/10.1109/tap.2010.2050425>
- [23] Formato, R.A. (2025) Pseudorandomness in Central Force Optimization. *Journal of Applied Mathematics and Physics*, **13**, 4567-4597.
<https://doi.org/10.4236/jamp.2025.1312250>
- [24] Brunato, M. and Battiti, R. (2007) Reactive Search: Machine Learning for Memory-Based Heuristics. In: Gonzalez, T.F., Ed., *Chapman & Hall/CRC Computer & Information Science Series*, Chapman and Hall/CRC, 21-1-21-17.
<https://doi.org/10.1201/9781420010749.ch21>
- [25] Valsecchi, G.B., Milani, A., Gronchi, G.F. and Chesley, S.R. (2003) Resonant Returns to Close Approaches: Analytical Theory. *Astronomy & Astrophysics*, **408**, 1179-1196.
<https://doi.org/10.1051/0004-6361:20031039>

- [26] Schweickart, R., Chapman, C., Durda, D., Hut, P., Bottke, B. and Nesvorny, D. (2006) Threat Characterization: Trajectory Dynamics. arXiv: physics/0608155.
<http://arxiv.org/abs/physics/0608155>
- [27] Luke, S. (2016) Essentials of Metaheuristics, Online Version 2.3.
<https://people.cs.gmu.edu/~sean/book/metaheuristics/>
- [28] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1986) Numerical Recipes: The Art of Scientific Computing. Cambridge University Press.
- [29] Shinzato, T. (2007) Box Muller Method.
<https://www.scribd.com/document/399483818/Box-Muller>
- [30] (2012) Power BASIC for Windows, ver. 10.04.0108. PowerBASIC, Inc.

Appendix A: PBM Benchmarks—Geometries, Landscapes & NEC4 Files

This section describes each PBM benchmark antenna and, for the 2D cases, its decision space using a perspective view of the topology. Additional perspective plots and projections onto the decision space's principal planes as well as samples of the NEC4 input/output files used to generate the plots and to validate NEC4 are available online at the URL cited below.

A.1. Benchmark #1: Variable Length Center-Fed Dipole

The antenna geometry for Problem #1 is shown in **Figure A1**. The objective is to maximize a center-fed dipole's directivity, D , as a function of its total length, L , and the polar angle, θ . A perspective view of the 2D landscape appears in **Figure A2** with additional plots available online at <https://app.box.com/s/j6km9sy4udstujy5fx9e40p6ci70qs85> (the online data also include sample NEC4.1D input/output files). The topology is smoothly varying with a single global maximum and two local maxima of similar amplitude.

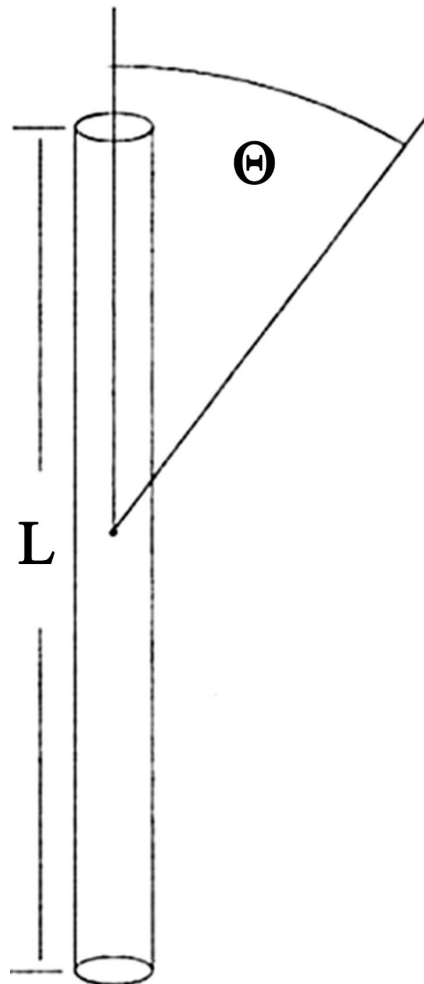


Figure A1. Dipole.

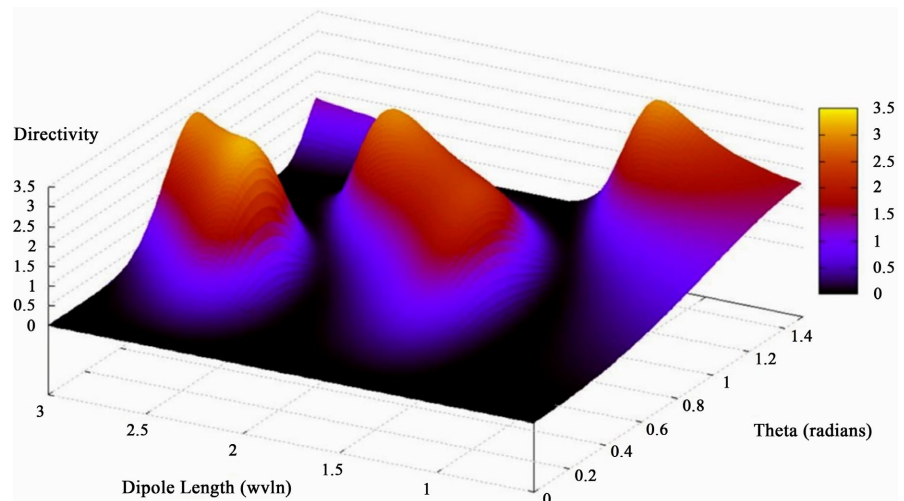


Figure A2. Benchmark #1 topology, perspective view.

A.2. Benchmark #2: Uniform Dipole Array

The problem #2 antenna is the uniform array of half-wave dipoles shown in **Figure A3**. All elements are center-fed with in-phase equal amplitude sources. The standard right-handed Cartesian coordinate system used by NEC4 is also shown, as are the polar angle θ and azimuth angle ϕ . The objective is to maximize directivity $D(d, \theta)$ in the plane $\phi = 90^\circ$ as a function of element separation d and polar angle θ without and with the presence of additive Gaussian noise. **Figure A4(a)** shows the landscape without noise, and **Figure A4(b)** with it. Following [6], noise is generated by adding to the NEC4-computed directivity a normally distributed zero-mean, 0.2-variance random variable z , here computed using the Box Muller method [28] [29] as: $z = \mu + \sigma \sqrt{-2 \ln(s)} \cos(2\pi t)$, where μ and σ , respectively, are the mean (zero) and standard deviation (0.4472), and s and t are random variables uniformly distributed on $[0, 1]$. s and t are generated using the compiler's internal RND function [30] seeded with the CFO run's start time.

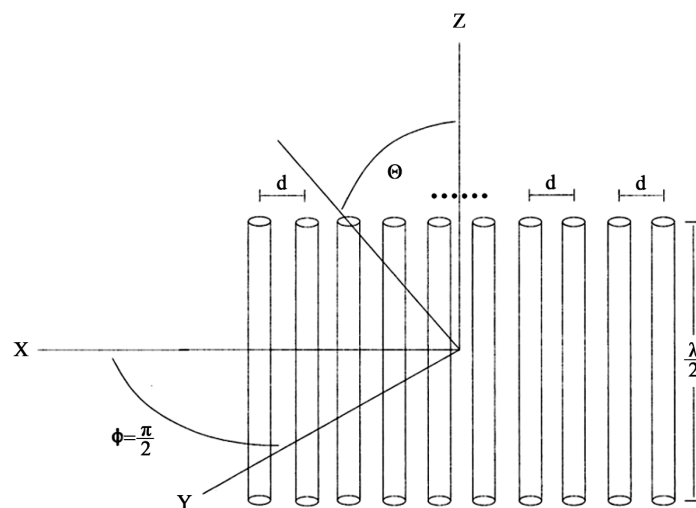


Figure A3. Uniform array of half-wave center-fed in-phase dipoles.

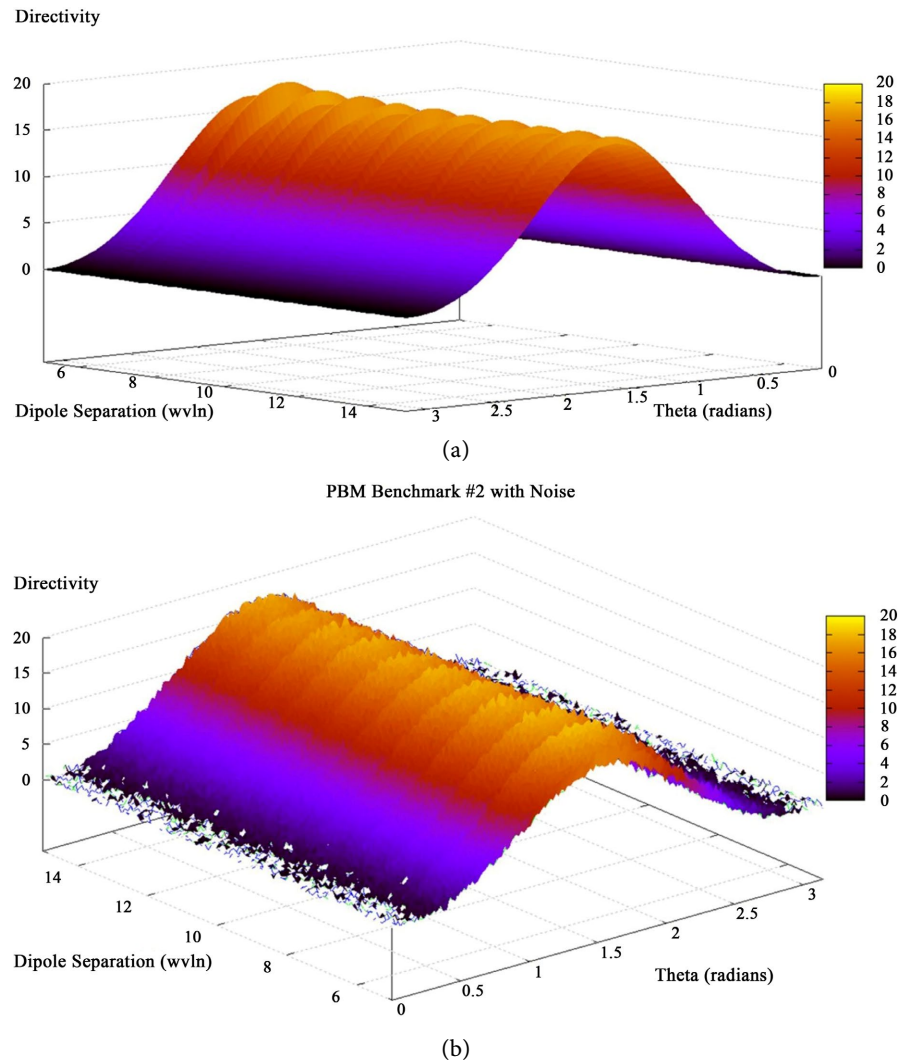


Figure A4. (a). Uniform half-wave dipole array without noise, perspective view; (b) Uniform dipole array with additive gaussian noise, perspective view.

A.3. Benchmark #3: Circular Array of Half-Wave Dipoles

The antenna for problem #3 is the circular array of half-wave dipoles shown in **Figure A5**. The array comprises eight dipoles parallel to the z-axis uniformly deployed on a one-wavelength radius circle. All elements are center-fed by equal-amplitude sources; but, following [6], the phase varies as $\alpha_n = -\cos[2\pi\beta(n-1)]$, $n = 1, \dots, 8$. The unit-amplitude excitation therefore is $V_n = \cos \alpha_n + j \sin \alpha_n$. The objective is to maximize the directivity $D(\beta, \theta)$ in the plane $\phi = 0^\circ$ as a function of the dimensionless phase parameter $0 \leq \beta \leq 4$ and the polar angle θ . The range for β produces the four global maxima at $\left(\beta_i = i - 0.5, i = 1, \dots, 4; \theta = \frac{\pi}{2}\right)$ seen in the perspective topology plot in **Figure A6**.

A.4. Benchmark #4: Vee Dipole

The Vee-dipole antenna for benchmark #4 is shown in **Figure A7**. The antenna

comprises two arms of length L_{arm} with inner angle 2α connected by a feed segment of length $2L_{feed}$ fed at its midpoint. The objective is to maximize the directivity $D(L_{total}, \alpha)$ along the +X-axis as a function of the total dipole length $0.5\lambda \leq L_{total} = 2L_{arm} + 2L_{feed} \leq 1.5\lambda$ and the inner half-angle $\frac{\pi}{18} \leq \alpha \leq \frac{\pi}{2}$ with $L_{feed} = 0.01\lambda$. Topology of benchmark #4's decision space appears in **Figure A8**. This objective function is unimodal with a single global maximum at $D(L_{total}, \alpha) = (1.5\lambda, 0.834)$. The surface is smoothly varying without pronounced local maxima.

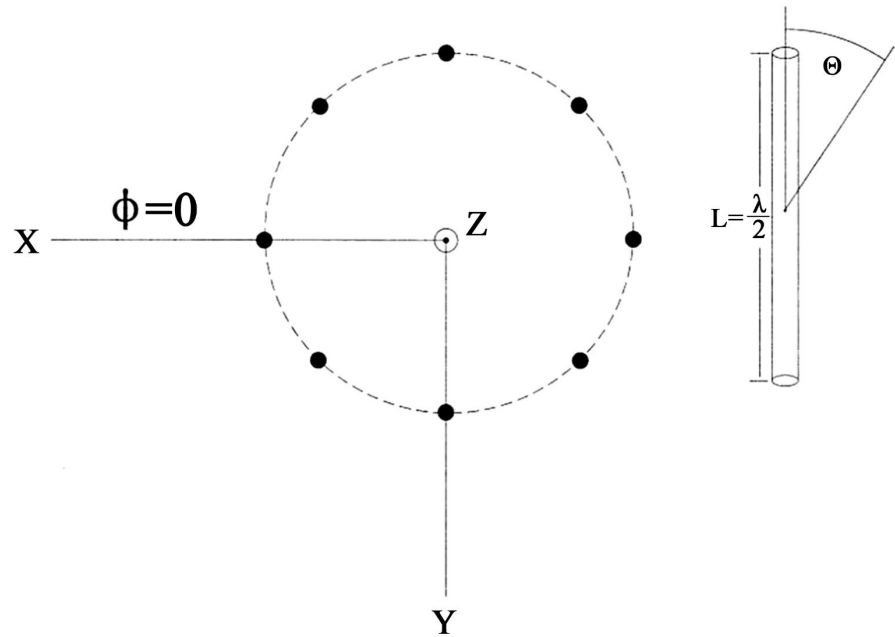


Figure A5. Circular array of half-wave dipoles (1λ radius).

PBM Benchmark #3

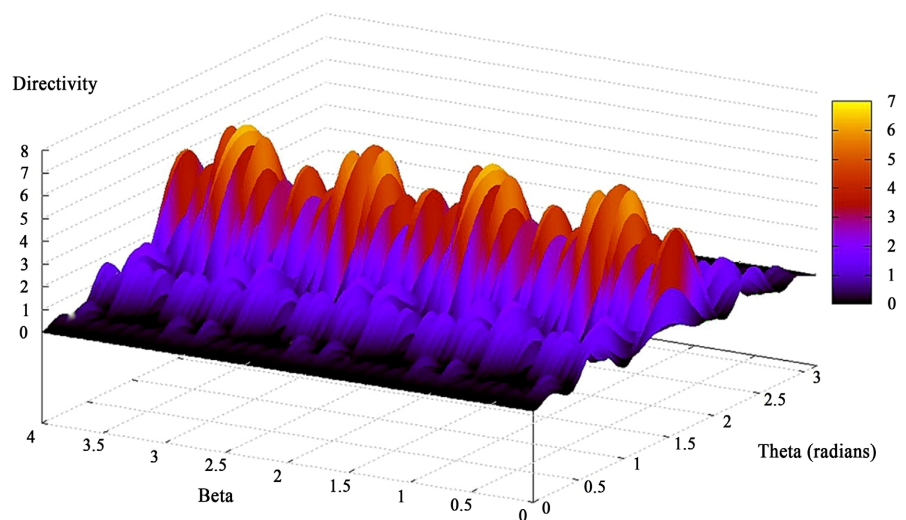


Figure A6. Circular array landscape, perspective view.

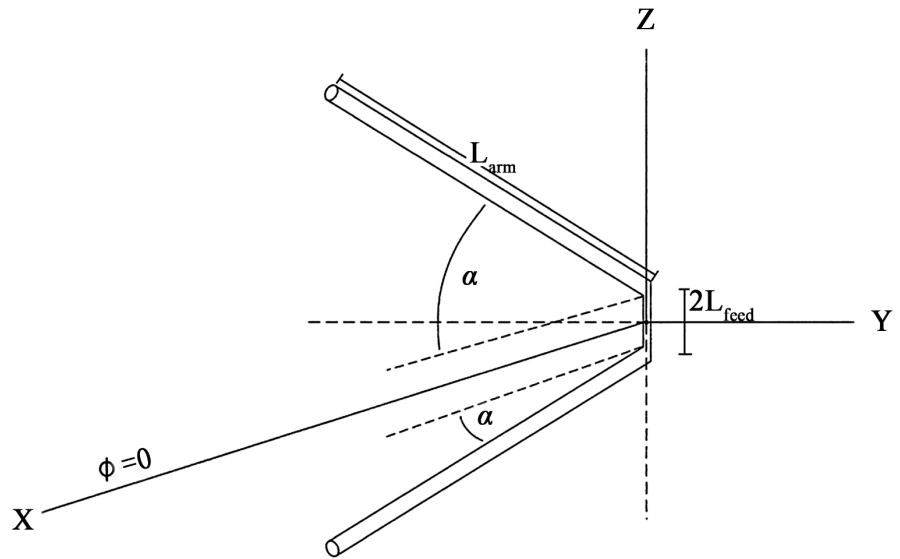


Figure A7. Vee dipole.

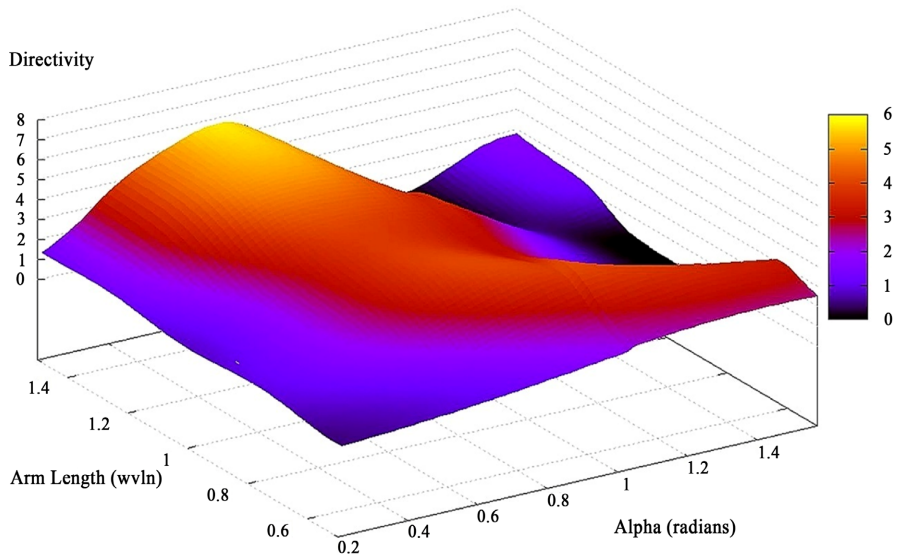


Figure A8. Vee dipole decision space topology, perspective view.

A.5. Benchmark #5: N-element Collinear Dipole Array

Benchmark #5 is a collinear array of N_{el} half-wave dipoles as shown in **Figure A9**. All elements are center-fed in-phase with equal amplitudes sources. The objective is to maximize directivity $D(d_i, i = 1, \dots, N_{el} - 1)$ in the plane $\phi = 0^\circ$ as a function of the element center-to-center spacings $0.5\lambda \leq d_i \leq 1.5\lambda$. Because there are $N_{el} - 1$ spacings in an N_{el} array, the dimensionality of this problem is $(N_{el} - 1)D$, unlike the previous four benchmarks each of which is 2D. As discussed at length in [6], maximum directivity occurs at $d_i = 0.99\lambda, \forall i$ independent of the number of elements, that is, with all dipoles spaced 0.99λ regardless of the array size. Of course, the value of the directivity does depend on the array size, increasing approximately in proportion to the length.

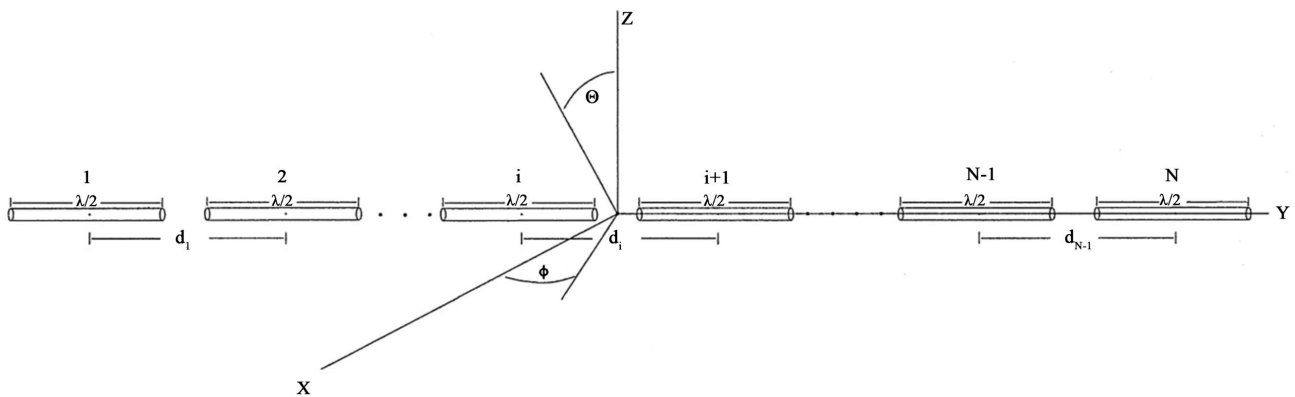


Figure A9. N_e -element collinear dipole array.

Appendix B: CFO Pseudocode

As discussed in §7.4 specifying run parameters is an important consideration in setting up a CFO run. Some of that discussion is consequently repeated here. A CFO run requires seven user-specified parameters: N_t , N_p , G , α , β , Δt , and F_{rep} , some of which appear to be more important than others. A possible eighth parameter is the initial probe acceleration, which is set to zero for all runs reported here and consequently is not included in this list. Constant values for G and Δt combine multiplicatively in a single coefficient, but of course these parameters could be separated and varied during a run. For the runs reported here, $\alpha = \beta = 2$ because this value seems to provide good results for many test functions and may be good default values. In this paper, there is no change in fitness value or average probe distance between steps 0 and 1 because the initial acceleration is set to zero.

CFO is a simple algorithm that is easily implemented. These are the basic steps:

- (a) compute initial probe positions and corresponding objective function fitnesses and assign initial accelerations
- (b) successively compute each probe's new position using equation (2) §7.2 based on previously computed accelerations using equation (1) §7.2
- (c) verify that each probe is located inside the decision space, making corrections as required
- (d) update fitness at each new probe position
- (e) compute accelerations for the next time step using new positions
- (f) loop over time steps or until termination criterion is met.

While the initial acceleration is zero for runs reported here, a non-zero value would “fly” probes in the directions of the accelerations at step 0 which may or may not improve performance. Because CFO may fly probes outside the DS domain into regions of unallowable solutions, any errant probes should be returned to the decision space. There are many possible retrieval schemes, and the one used here is simple and deterministic as described below and in §7.4. The repositioning factor, F_{rep} , which is used to return errant probes, in fact is one of CFO's most important parameters. When and how it should be changed are unanswered ques-

tions. The simple approach below seems to provide good results, and it was chosen solely for that reason.

Note that all CFO parameters presently are chosen on an empirical basis, that is, by “trial and error” because there is no methodology at this point for specifying their values. This question remains a pressing one in CFO’s further development. The parameter values that have been used provide good results for the purpose of demonstrating CFO’s effectiveness and efficiency as an optimization metaheuristic against the PBM suite, but these are not necessarily the best values.

CFO pseudocode is as follows with values for N_{saved} , TOL_{fit} , F_{rep}^{init} and ΔF_{rep} are those used in this paper as shown:

(1) Data Structures

(A) Create arrays $R(p, i, j)$ [position vectors], $A(p, i, j)$ [accelerations], $1 \leq p \leq N_p$, $1 \leq i \leq N_d$, $0 \leq j \leq N_t$ [probe #, coordinate #, time step #]

(B) Create fitness array $M(p, j) = f(R(p, i, j))$

(C) Create array of last N_{saved} best fitnesses $M_{best}(q), 1 \leq q \leq N_{saved}$

(2) Initialization

At time step $j = 0$:

(A) (i) Uniform probes on each coordinate axis (note: $\frac{N_p}{N_d}$, # probes/dimension, user-specified).

For $i = 1$ to N_d , $n = 1$ to $\frac{N_p}{N_d}$:

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

(A) (ii) Probes slightly off decision space diagonal

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

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

(A) (iii) Uniform 2D grid

$$\Delta x_1 = \frac{x_1^{\max} - x_1^{\min}}{\frac{N_p}{N_d} - 1}, \Delta x_2 = \frac{x_2^{\max} - x_2^{\min}}{\frac{N_p}{N_d} - 1}; \text{ For } k = 1 \text{ to } \frac{N_p}{N_d}, m = 1 \text{ to } \frac{N_p}{N_d}:$$

$$p = \frac{N_p}{N_d}(k-1) + m, R(p, 1, 0) = x_1^{\min} + (k-1)\Delta x_1,$$

$$R(p, 2, 0) = x_2^{\min} + (m-1)\Delta x_2$$

(A) (iv) Other user-specified probe distribution [**Note—This option used for this paper**]

(B) Initial acceleration

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

(C) Initial fitness

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

$$\text{Best Fitness} = \text{MAX}(M(p, 0), 1 \leq p \leq N_p)$$

(D) Saved Fitnesses & Fitness Tolerance

$$N_{\text{saved}} = 5, N_{\text{sat}} = 3, \text{TOL}_{\text{fit}} = 0.0005$$

(E) Initial Reposition Factor & Increment

$$F_{\text{rep}}^{\text{init}} = 0.5, \Delta F_{\text{rep}} = 0.005, F_{\text{rep}} = F_{\text{rep}}^{\text{init}}$$

(3) Time Loop ($0 \leq j \leq N_t$)

(A) Saved Fitness Array Index: $s = j \text{ MOD } N_{\text{saved}}$; If $s = 0$ then $s = N_{\text{saved}}$

Important Note: Saved fitnesses are not necessarily sequential as a result of how s is calculated, which affects when F_{rep} is incremented See discussion in §4.4.

New Probe Positions

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

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

(b) *Retrieve errant probes as required* (Note: For Repositioning Factor, $0 \leq F_{\text{rep}} \leq 1$, see Formato [2]-[4])

$$\text{If } R(p, i, j) < x_i^{\min} \text{ then } R(p, i, j) = x_i^{\min} + F_{\text{rep}} \cdot (R(p, i, j-1) - x_i^{\min})$$

$$\text{If } R(p, i, j) > x_i^{\max} \text{ then } R(p, i, j) = x_i^{\max} - F_{\text{rep}} \cdot (x_i^{\max} - R(p, i, j-1))$$

(B) Update Fitness Matrix

For $p=1$ to N_p :

$$M(p, j) = f(R(p, i, j))$$

If $M(p, j) \geq \text{Best Fitness}$ then $\text{Best Fitness} = M(p, j)$ &

$$M_{\text{best}}(s) = M(p, j)$$

(C) Update Reposition Factor

$$\text{If } \left| M_{\text{best}}(N_{\text{saved}}) - \frac{1}{N_{\text{sat}}} \sum_{q=s}^{N_{\text{saved}}} M_{\text{best}}(q) \right| \leq \text{TOL}_{\text{fit}}, \quad s = N_{\text{saved}} - N_{\text{sat}} + 1, \text{ then}$$

$$F_{\text{rep}} = F_{\text{rep}} + \Delta F_{\text{rep}}$$

If $F_{\text{rep}} \geq 1$ then $F_{\text{rep}} = F_{\text{rep}}^{\text{init}}$

(D) Update Accelerations

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

$$A(p, i, j) = G \sum_{\substack{k=1 \\ k \neq p}}^{N_p} U(M(k, j) - M(p, j)) \\ \times (M(k, j) - M(p, j))^\alpha \times \frac{R(k, i, j) - R(p, i, j)}{|\bar{R}_j^k - \bar{R}_j^p|^\beta},$$

$$\text{where } |\bar{R}_j^k - \bar{R}_j^p| = \sqrt{\sum_{m=1}^{N_d} (R(k, m, j) - R(p, m, j))^2}$$

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