Scientific Research Publishing

Journal of Applied Mathematics and Physics, 2025, 13(6), 2113-2127
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

Fano Load Redux: CFO with Negative Gravity

Richard A. Formato

Consulting Engineer & Registered Patent Attorney, Cataldo & Fisher LLC (Emeritus), Harwich, MA, USA
Email: rf2@ieee.org

How to cite this paper: Formato, R.A. (2025) *Fano Load* Redux: CFO with Negative Gravity. *Journal of Applied Mathematics and Physics*, **13**, 2113-2127.
https://doi.org/10.4236/jamp.2025.136118

Received: May 23, 2025
Accepted: June 27, 2025
Published: June 30, 2025

Abstract

This note describes optimally matching the seminal *Fano Load* using Central Force Optimization with Negative Gravity. This approach improves the best fitness by more than eighteen percent and suggests that some measure of Negative Gravity should be used in all CFO runs. CFO is a deterministic Global Search and Optimization metaheuristic based on an analogy to gravitational kinematics, the motion of bodies moving under the influence of gravity. Positive gravity causes objects to move towards each other, whereas negative gravity causes them to fly apart. A small amount of negative gravity in CFO improves the algorithm's exploration of the decision space by sampling regions that have been under-sampled or perhaps not sampled at all. The *Fano Load* problem illustrates this effect. While the possibility of Negative Gravity was mentioned in the original CFO paper, it was not used until recently when it was injected into optimization runs for Yagi-Uda antenna arrays. The results were compelling and led to the *Fano Load* problem being revisited.

Keywords

Central Force Optimization, *Fano Load*, Negative Gravity, Optimization Metaheuristic, Global Search and Optimization

1. Introduction

The seminal *Fano Load* was used as an example in the original Central Force Optimization (CFO) paper [1] in which CFO's results were compared to those from several other optimization algorithms. This note updates that discussion by applying CFO with *negative gravity*. CFO is a deterministic Global Search and Optimization (GSO) metaheuristic that searches a decision space (DS) by "flying" probes that sample its objective function's topology, which is unknown or unknowable. CFO locates local and global *maxima*, not minima as do most GSO's. It is based on an analogy to gravitational kinematics. With positive gravity, as in the real Universe, CFO's probes coalesce on the objective function's greatest fit-

ness(es), whereas negative gravity causes probes to fly apart [2], presumably into regions of DS that have been under-explored or perhaps not explored at all.

CFO has been applied to a wide variety of optimization problems, ranging from extensive benchmark suites [3] to, as examples, training neural networks [4], iris recognition [5], robot gait modeling [6], brain imaging [7], satellite imaging [8], arid region water distribution [9], UAV flight path calculation [10], creating Artificial Intelligence neural network ensembles [11], and a plethora of antenna and electromagnetic problems too extensive to list in this short note.

2. Matching the *Fano Load*

Figure 1 shows the network under consideration. The *Fano Load* is matched to a generator with internal resistance $R_g = 2.205\ \Omega$ by the three-element equalizer (C_1, L_2, C_3). The load comprises series inductor $L_{fano} = 2.3$ Hy (Henry) in series with the parallel combination of $R_{fano} = 1\ \Omega$ (Ohm) and $C_{fano} = 1.2$ Fd (Farad). Additional details are available in **Figure 2** in [12] and in Section 3A in [13]. The classic *Fano Load* has been used extensively in research on broad-band matching. As examples, in their work on optimizing broad-band matching Carlin and Amstutz [14] discuss three test cases that utilize the original *Fano Load* parameters. Dedieu *et al.* [15] propose a new method for broad-band matching using LC ladder equalizers based on Fano's original gain-bandwidth formulas that were published in his 1947 doctoral thesis [16]. More recently, Schwartz and Allen [17] present the first Helton approach to broad-band impedance matching using the *Fano Load* as a test case. The optimization objective is to maximize the power delivered to the load at all frequencies in the radian frequency range $0 \le \omega \le 1$. It is measured by the Transducer Power Gain (TPG), defined as the fraction of the maximum available power that is actually delivered to the load ($0 \le \text{TPG} \le 1$).

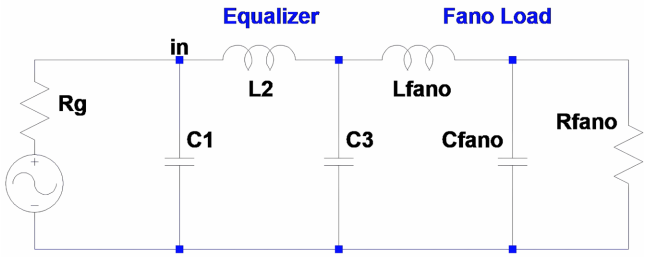

Figure 1. *Fano Load* problem.

CFO is a *deterministic* algorithm that maximizes the objective (fitness) function, unlike most algorithms that minimize it. If true randomness is not added, then every CFO run with the same setup produces exactly the same results. Multiple runs are not required to build statistics to measure the algorithm's performance. Nevertheless, if desired some measure of "randomness" can be introduced into CFO using pseudorandom variables (prv's). They can be injected into CFO in such a way as to preserve its determinism while providing some degree of randomness. A prv's value can be specified by enumeration or by calculation or by a

combination of both. True random variables by contrast must be calculated from a probability distribution and are consequently unknowable *a priori*. One prv enumeration approach is to use π-fractions [18]-[23], which is what was done here. π-fractions were used to inject negative gravity into CFO on a probe-by-probe basis at each step in the optimization run. For example, a run employing 150 probes at each of 100 steps with 15% negative gravity would ideally contain a total of 2250 π-fraction-assigned probes with $G < 0$, computed as 15% of the probe-step product of 15,000. G is CFO's "gravitational constant".

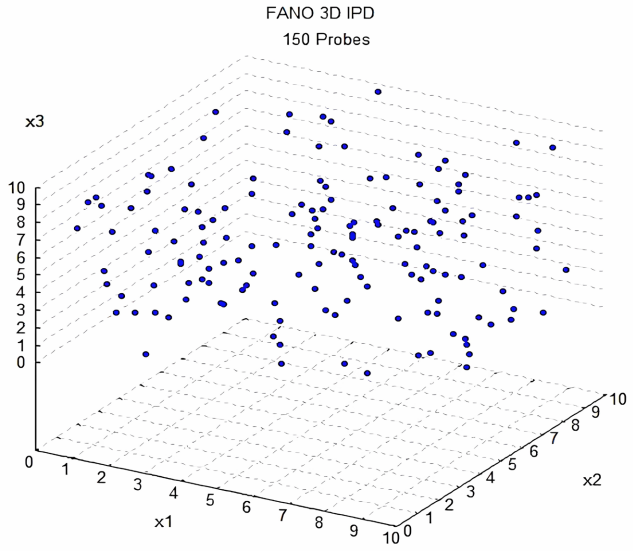

Figure 2. *Fano Load* π-fraction IPD, all runs.

The CFO runs reported here all consisted of 30 steps using 150 probes. The target values of $G < 0$ were 0%, 1%…, 45%, 50%, and 80%, with CFO setup parameters $G = \pm 2$, $\alpha = 2$, $\beta = 2$, $\Delta t = 1$ in a $[0\text{-}10]^3$ decision space. The same three-dimensional 150-probe π-fraction Initial Probe Distribution (IPD at step #0) was used for each CFO run as shown in **Figure 2**. Using this approach, the actual percentage of negative gravity versus the target value is plotted in **Figure 3** which reflects how uniformly distributed the π-fractions are on [0,1].

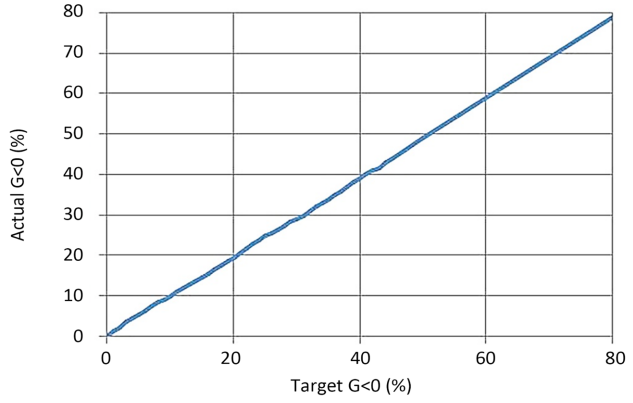

Figure 3. Actual $G < 0$ vs. Target $G < 0$.

Figure 4 plots the best fitness vs. the actual amount of negative gravity injected into CFO. The fitness (objective) function was the average value of TPG in $0 \le \omega \le 1.8$ whereas in [1] a different fitness function was used, *viz.* maximizing Min(TPG) over $0 \le \omega \le 1$. The fitness here is remarkably consistent as $G < 0$ increases from 3.4% all the way to 36.9% with an average value of 0.826. At 0% the best fitness is 0.721537. It increases to 0.853245 at 28.98% actual $G < 0$, a very substantial improvement of 18.25%. At 42% $G < 0$ the fitness drops back to its initial value of 0.721537, where it remains through 80% negative gravity (and presumably beyond).

The degree of fitness consistency over such a wide range of negative gravity values is rather surprising. In [18], for example, the beneficial range of $G < 0$ was much smaller, about 6% - 10%, for optimizing a six-element Yagi-Uda antenna array. The array problem, which is discussed in detail in [18], is another compelling example of the efficacy of adding a small amount of negative gravity to CFO. What is clear from these *Fano Load* data is that there is no way of knowing in advance how much negative gravity should be used in any given problem. Fortunately CFO's determinism and likewise pseudorandom π-fractions' or other prvs' make it relatively easy to quickly run test cases to answer this question. Another advantage is that the effects of using different objective functions can be seen by making only two CFO runs. Stochastic algorithms, on the other hand, make this very difficult if not, as a practical matter, impossible.

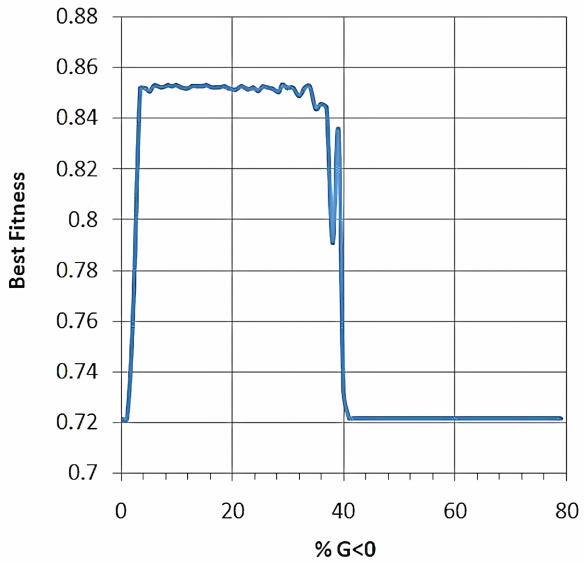

Figure 4. Best Fitness vs. Actual $G < 0$.

Figures 5-7 plot the CFO-optimized values for the equalizer components. Inspection of the TPG curves (not individually plotted because of their large number) shows that, arguably, the best performance is achieved at 33% target negative gravity (even though a very slightly higher fitness occurs at 30%). At 33% CFO-NG (name given the CFO algorithm with negative gravity) returned a best fitness

of 0.852765, and the corresponding optimized values for the equalizer components are $C_1 = 0.527$ Fd, $L_2 = 2.731$ Hy, and $C_3 = 1.053$ Fd.

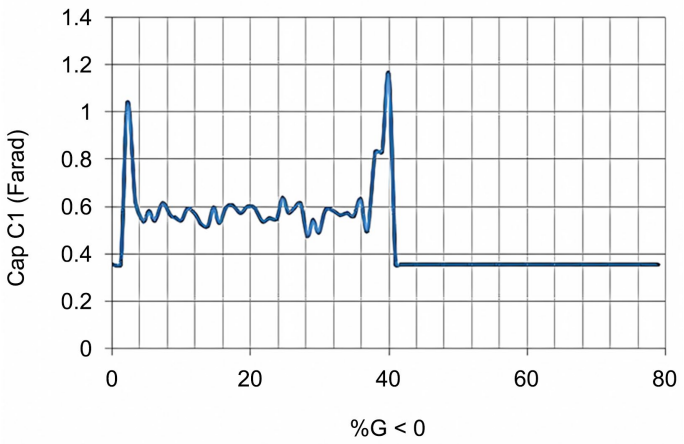

Figure 5. Equalizer component Capacitor C_1.

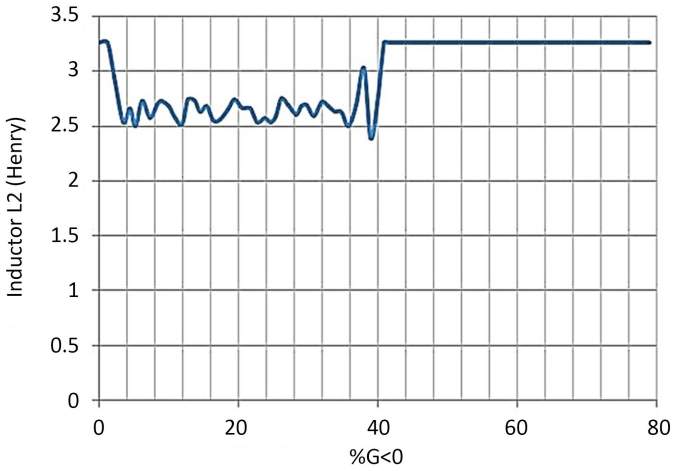

Figure 6. Equalizer component Inductor L_2.

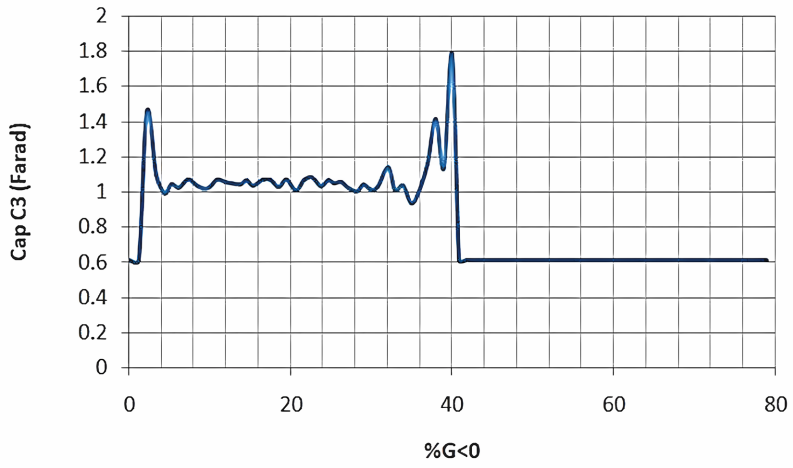

Figure 7. Equalizer component Capacitor C_3.

Figure 8 plots TPG using these component values. It is evident that the CFO-NG optimized equalizer does a very good job of transferring power to the *Fano Load*. Also plotted in the figure are the results from the four other optimizers discussed below.

The original *Fano Load* CFO implementation ("CFO2007") was compared to three other algorithms: i) GA-Simplex, ii) RFT (Real Frequency Technique), and iii) RSE (Recursive Stochastic Optimization) [12]-[17]. The optimized equalizer component values using these methods are summarized in Table 1 along with the original CFO2007 and this paper's CFO-NG results. All five methods converge on similar solutions. Figure 8 plots TPG for each set of component values. These results clearly show that CFO-NG solves the *Fano load* equalizer problem at least as well as other recognized optimization methods.

CFO-NG also compares favorably in terms of computational efficiency and rate of convergence. CFO-NG required 4,500 function evaluations to reach convergence in thirty steps. The GA-Simplex method employed a genetic algorithm with 100 individuals evolved for 100 generations, followed by a Nelder-Mead Simplex algorithm that ran for an additional 100 iterations using 170 function calls. GA-Simplex thus required a total of 10,170 function evaluations. RSE also comprised two calculation steps: 2000 iterations using a stochastic Gauss-Newton optimization routine to locate an approximate solution which was then refined using a 20,000 iteration random search algorithm. RSE thus required 22,000 function evaluations. The literature does not report run times or rates of convergence for GA-Simplex or RSE so that a direct comparison in that regard cannot be made to CFO-NG. As to RFT, it is not an evolutionary optimization algorithm. RFT is a quasi-analytical/graphical method in which the equalizer's resistance vs. frequency curve is approximated using piece-wise linear segments. RFT therefore cannot be described in terms of computational efficiency as measured by the number of required function calls. It is included because it specifically addresses the *Fano Load* problem and permits another comparison of the final results. Perhaps the single most useful measure of algorithmic efficiency is the total number of required function evaluations, and by that measure CFO-NG's performance is quite good.

Table 1. Comparison, Algorithms Solving the *Fano Load*.

----	RFT	RSE	GA-Simplex	CFO2007	CFO-NG
C_1 (Fd)	0.352	0.409	0.386	0.460	0.527
L_2 (Hy)	2.909	3.054	2.976	2.988	2.731
C_3 (Fd)	0.922	0.974	0.951	1.006	1.053

The evolution of CFO-NG's fitness appears in Figure 9. Note that 80 steps were used to provide a better resolution of the best fitness curve. CFO often converges very quickly [24], and this characteristic is quite apparent in Figure 9. D_{avg}, which

measures of how well CFO's probes have coalesced, is plotted in Figure 10. It shows that essentially all of CFO's 150 probes have converged on the optimal solution.

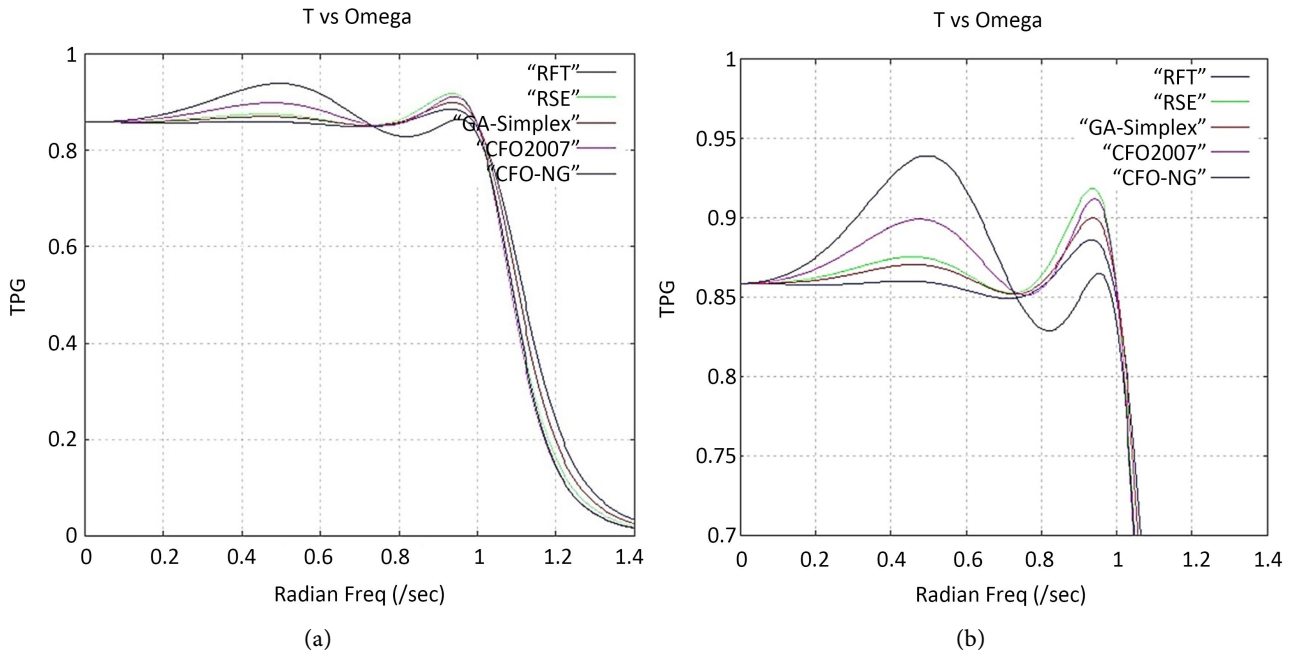

Figure 8. (a) TPG at 33% Target $G < 0$, 32.46% Actual. (b) TPG at 33% Target $G < 0$, 32.46% Actual.

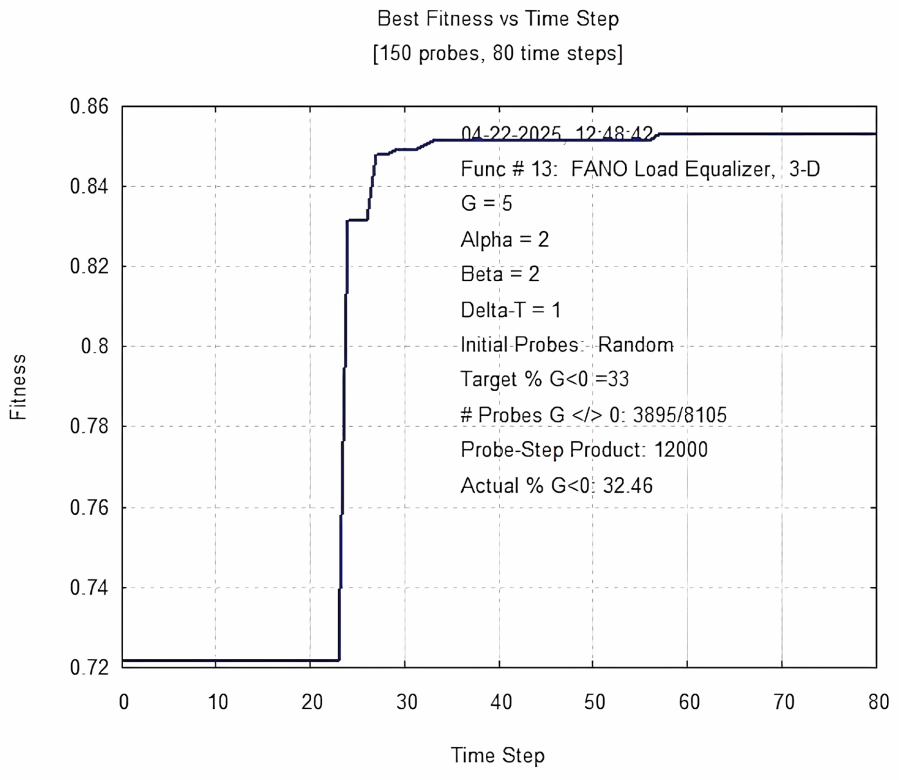

Figure 9. Fitness evolution at 33% Target $G < 0$, 32.46% Actual.

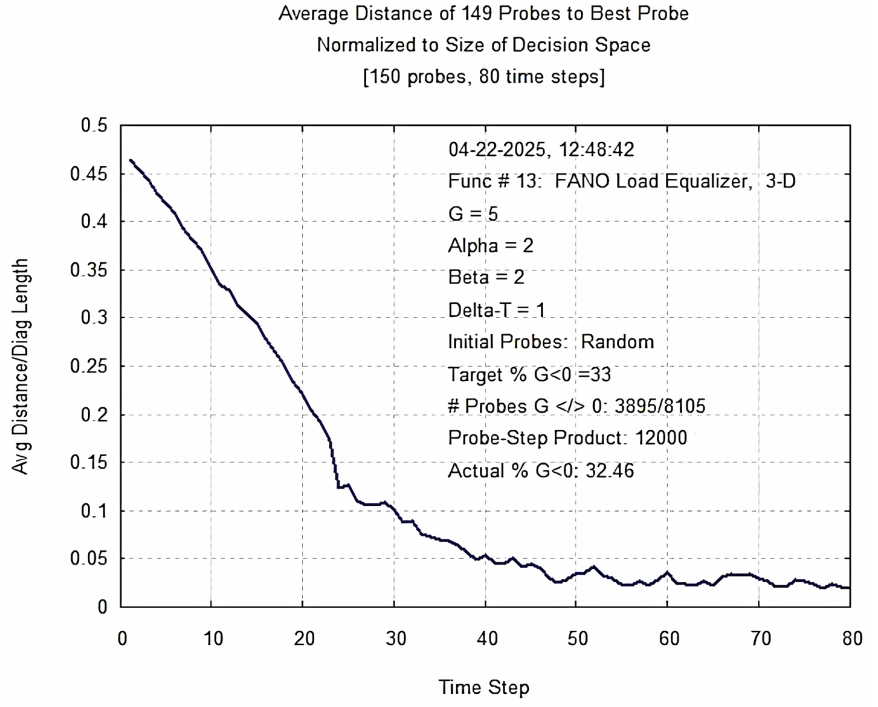

Figure 10. D_{avg} at 33% Target $G < 0$, 32.46% Actual.

3. Basic CFO Theory & Pseudocode

Central Force Optimization is a deterministic gradient-like GSO metaheuristic based on an analogy to gravitational kinematics. In the real Universe all material objects are attracted to each other by the force of gravity whose magnitude is [25].

$$F = \gamma \frac{m_1 m_2}{r^2} \quad (1),$$

in which $\gamma > 0$ is the "gravitational constant" and the m_i are the masses of the attracting objects. Gravity acts along the line connecting the masses' centers, making it by definition a *central force*. Hence, *Central Force Optimization* to describe this GSO.

CFO is a conceptual algorithmic framework for solving the following problem: In a DS defined by $x_i^{\min} \le x_i \le x_i^{\max}$, $i = 1, \cdots, N_d$, the x_i being *decision variables*, determine the locations of the global *maxima* of an objective function $f\left(x_1, x_2, \cdots, x_{N_d}\right)$. The value of $f\left(\vec{x}\right)$ at each point $\vec{x}$ is referred to as its "fitness." The objective function's topology is unknown or unknowable. It may be continuous or discontinuous, highly multimodal or unimodal, convex or non-convex, with one or more objectives, and possibly subject to constraints among the decision variables.

CFO is based on the metaphor of gravitational kinematics, the branch of physics that describes the behavior of masses moving solely under the influence of the force of gravity. The gravitational field is a conservative, linear, vector force field derivable from a scalar potential function. The "action at a distance" force of gravity thus may be computed directly as a vector quantity from Newton's universal

law of gravitation, or, alternatively, as the negative gradient of a scalar gravitational potential function. These distinct approaches highlight different conceptual aspects of the CFO analogy.

Gravity always attracts masses toward each other. Unlike the Coulomb electric force, which may be attractive or repulsive depending on the signs of the charges, real gravity never is repulsive. In the real Universe, gravity always accelerates masses toward each other. The vector acceleration experienced by mass m_1 due to mass m_2 is given by Newton's second law of motion as [25].

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

where $\hat{r}$ is a unit vector pointing towards mass m_1 from mass m_2 along the line joining m_1 and m_2.

A *constant* acceleration applied during the time interval t to $t+\Delta t$ changes a mass's location in three-dimensional space as follows (see Brand's detailed discussion in [26])

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

The mass's position is $\vec{R}\left(t+\Delta t\right)$ at time $t+\Delta t$, where $\vec{R}_0$ and $\vec{V}_0$, respectively, are the position and velocity vectors at time t. In a standard right-handed 3-D Cartesian coordinate system, the position vector is given by $\vec{R} = x\hat{\imath} + y\hat{\jmath} + z\hat{k}$, where $\hat{\imath}$, $\hat{\jmath}$, $\hat{k}$, are the unit vectors along the x, y and z axes, respectively, as traditionally denoted in classical physics.

Equation (2) and Equation (3) are referred to as the *equations of motion*. CFO generalizes them from 3 to N_d dimensions because CFO searches N_d-dimensional spaces for the extrema of an objective function to be maximized. Nevertheless, for the purpose of explaining the CFO analogy, the 3-dimensional decision space in **Figure 11** will be used to provide a concrete visualization.

CFO "flies" a group of probes through the decision space along trajectories determined by the generalized equations of motion at a set of discrete time steps. In this example, the position of each probe at each step is specified its position vector $\vec{R}_j^p$, where the indices p and j are the probe number and time step number, respectively. Probe p moves from position $\vec{R}_{j-1}^p$ at time step $j-1$ to position $\vec{R}_j^p$ at time step j, with the "time" interval between steps $j-1$ and j being Δt. In an N_d-dimensional decision space, the generalized 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 $\hat{e}_k$ is the unit vector along the x_k axis.

The objective function to be maximized is defined on the N_d-dimensional decision space. The value of the objective function at each point along a probe's trajectory is the "fitness" at that point. At time step $j-1$ at probe p's location, for example, the fitness is given by $M_{j-1}^p = f\left(x_1^{p,j-1}, x_2^{p,j-1}, \cdots, x_{N_d}^{p,j-1}\right)$. Every other

probe also has a fitness $M_{j-1}^{k}, k=1,\cdots,p-1,p+1,\cdots,N_p$, associated with it, where N_p is the *total* number of probes. In Figure 11 the fitness at each probe's location is represented by the *size* of the blackened circle at the tip of the position vector, the metaphorical correspondence being to the size of a "planet" in space. Ranked from largest to smallest, the fitnesses in Figure 11 are located at $\vec{R}_{j-1}^{s}$, $\vec{R}_{j}^{p}$, $\vec{R}_{j-1}^{n}$, and $\vec{R}_{j-1}^{p}$, respectively, the ranking being reflected in the relative size of the circles at the tip of each position vector. The probe number is $1\le p\le N_p$, while the time step number is $0\le j\le N_t$, where N_t is the *total* number of *time steps*.

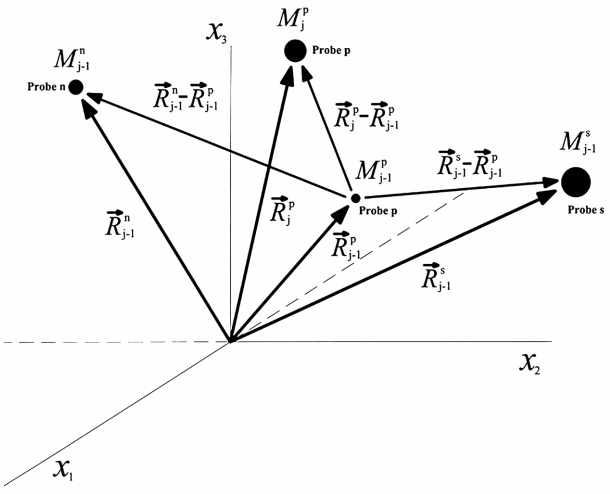

Figure 11. Typical 3D CFO decision space.

As time progresses the probes fly through the decision space along trajectories governed by the equations of motion. The force of gravity in CFO is created by a mass that is a *user-defined function* of the fitness at each of the other probes' locations. It is important to note that the symbol M_j^k used for probe k's fitness at time step j *must not* be confused with "mass" in CFO. While larger circles in Figure 11 do indeed correspond to greater fitness values, CFO's "mass" is *not* the same as the fitness for reasons discussed below.

How probe p moves from position $\vec{R}_{j-1}^{p}$ to $\vec{R}_{j}^{p}$ is determined by its initial position and by the *total* acceleration produced by the "masses" created by the user-defined function of the fitnesses at each of the other probes' locations.

The "acceleration" experienced by probe p due to probe n for the CFO implementation described here is given by

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

Similarly, probe p is subject to an acceleration due to probe s that is given by

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

Following standard notation, the vertical bars denote vector magnitude, $\left|\vec{A}\right| = \left(\sum_{i=1}^{N_d} a_i^2\right)^{\frac{1}{2}}$ where the a_i are the scalar components of vector $\vec{A}$.

G is a very important CFO parameter. It is CFO's "gravitational constant" (corresponding to γ in Equation (1) [note that the minus sign in Equation (2) has been absorbed into the order in which the differences in the acceleration expressions are taken]. Of course, most of the time $G > 0$, and masses are attracted towards each other as they are in the real Universe. But G can be less than zero, which is "negative gravity," and then masses repel, they are driven away from each other. Of course, the very point of this paper is to investigate the effect of adding a small amount of negative gravity to CFO, and the results show that it can be quite beneficial if not overdone.

Note that the terms in the numerator in the acceleration equation containing the objective function fitnesses, for example, $\left(M_{j-1}^{s} - M_{j-1}^{p}\right)^{\alpha}$, correspond *loosely* to the mass in (2). CFO's acceleration expression is quite different from what it is in physical space in three important ways: i) the definition of mass; ii) inclusion of the exponents $\alpha > 0$, $\beta > 0$; and iii) inclusion of the unit step $U(\cdot)$.

In this particular CFO implementation, the user-defined function that is CFO's "mass" begins with the *difference of fitnesses*, not the fitnesses values themselves. The algorithm designer is free to choose any function of the fitnesses, and different definitions may result in better performance against certain objective functions. One possibility, for example, might be a ratio of fitnesses or their differences, a notion reminiscent of the "reduced mass" concept in gravitational kinematics. But for this description of CFO theory only the difference of fitnesses definition will be used.

The use of the fitness difference $M_{j-1}^{s} - M_{j-1}^{p}$ here is intended to avoid excessive gravitational "pull" by other very close probes. It is likely that nearby probes in DS have similar fitness values, which may lead to an excessive gravitational force on the subject probe. The difference of fitnesses intuitively seems to be a better measure of how much gravitational influence there should be by the probe with a greater fitness on the probe with a smaller one.

The second difference is including the exponents α and β which in physical space take on the specific values 1 and 3, respectively [note that the CFO numerator does not contain a unit vector like Equation (2)]. The variation of gravitational acceleration with mass and distance in the Universe follows an immutable law. But in metaphorical "CFO space" the algorithm designer is free to assign a completely different variation with mass and distance, and the consequences can be quite dramatic. This flexibility is included in the free parameters α and β. CFO test runs reveal that the algorithm's convergence is sensitive to the exponent values, and that some values of these exponents are better than others.

The third difference is the inclusion of the unit step function $U(z) = \begin{cases} 1, & z \geq 0 \\ 0, & \text{otherwise} \end{cases}$. Because CFO space is metaphorical, it can be a strange

place in which physically unrealizable objects exist. Real mass must be positive, but not so mass in CFO space. In this CFO implementation, if the Unit Step were not included, then mass could be *positive* or *negative* depending on which fitness is greater in computing the fitness difference. *The Unit Step function eliminates the possibility of "negative" mass by creating only positive masses*. This attribute is quite important in any CFO implementation, because only positive mass is attractive in nature. If CFO is implemented with the possibility of negative mass, the corresponding accelerations are repulsive instead of attractive. The effect of a repulsive gravitational force is to fly probes away from large fitness values instead of toward them, just the opposite of what the algorithm is intended to do.

The expressions above are the accelerations experienced by probe p due *only* to the two probes n and s. But in fact probe p's trajectory is determined by the gravitational influence of *all* the other probes. The *total* acceleration experienced by p as it "flies" from position $\vec{R}_{j-1}^{\,p}$ to $\vec{R}_{j}^{\,p}$ thus is given by summing over all other probes:

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

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

$$\vec{R}_{j}^{\,p} = \vec{R}_{j-1}^{\,p} + \frac{1}{2}\vec{a}_{j-1}^{\,p}\,\Delta t^{2}, \quad j \geq 1 \tag{5}$$

(4) is analogous to (2), while (5) is analogous to (3). *These four equations embrace the gravitational kinematics metaphor on which CFO is based*.

Note that Equation (5) in [1] contains a "velocity" term $\vec{V}_{j-1}^{\,p}$ that intentionally has been omitted from (5) above. The reason for this change is that numerical experiments performed after that paper was published revealed that including the velocity term as written appeared to actually impeded CFO's convergence. This behavior is suspected to be a result of the difference between straight-line and curvilinear acceleration in the real world. For example, while the velocity vector of a point mass in circular motion is tangent to the circle, its acceleration vector is orthogonal and directed along the radius toward the circle's center. In reference [1] the velocity term already had been set to zero as a matter of convenience. Consequently the term $\vec{V}_{j-1}^{\,p}$ has been dropped from (5) in this formulation of CFO theory, which is an important change. Note, too, that in the velocity term, the coefficient $\frac{1}{2}$, and Δt in Equation (5) of [1] were retained *only* to preserve the analogy to gravitational kinematics. None of these variables is required (of course, Δt cannot be zero). A constant value of Δt and the factor $\frac{1}{2}$ simply should be absorbed into the gravitational constant G. Varying Δt, which changes the interval at which probes "report" their positions, likely will influence how CFO converges, but to the author's knowledge Δt's influence has not been investigated and consequently remains an open question.

The formulation of CFO theory presented here is based on the vector model of the gravitational force field. Another approach would be to compute the force of gravity as the negative gradient of a scalar gravitational potential see, for example [25] or [27] for an in-depth discussion. While the results are the same as the vector model, the gravitational potential approach emphasizes a different aspect of the CFO gravitational metaphor. Because the gradient is the derivative (slope) of the gravitational equipotential surface in the direction of maximum rate of change, conceptually CFO may be thought of as a form of deterministic gradient-based search. In the implementation described here, for example, the factor $\frac{\left(M_{j-1}^{s}-M_{j-1}^{p}\right)^{\alpha}}{\left|\vec{R}_{j-1}^{s}-\vec{R}_{j-1}^{p}\right|^{\beta}}$ looks very much like a directional derivative (difference of fitnesses divided by distance between evaluation points). But because α and β can take on any value assigned by the user, this ratio perhaps is best described as a "generalized directional derivative." Interpreting CFO as a generalized gradient-like methodology may be helpful in suggesting potentially useful analytical approaches and also for understanding how CFO works. This conceptual framework may be attractive to researchers more accustomed to thinking in terms of derivatives.

The CFO algorithm essentially comprises Equations (4) and (5). It is simple and easily implemented in a compact computer program. The basic pseudocode is:

i) compute initial probe positions, the corresponding objective function fitnesses, and assign initial accelerations.

ii) successively compute each probe's new position using (5) based on previously computed accelerations using (4).

iii) verify that each probe is located inside the decision space, making corrections as required.

iv) update the fitness at each new probe position.

v) compute accelerations for the next time step based on the new positions.

vi) loop over all time steps or until some termination criterion has been met.

Note that because CFO may "fly" a probe outside the DS domain into regions of unallowable solutions, such errant probes should be returned to the decision space or discarded. There are many possible retrieval schemes as discussed in the CFO references.

4. Conclusion

The *Fano Load* data illustrate the benefit of injecting some measure of negative gravity into the CFO algorithm, ostensibly because it improves CFO's ability to explore under-sampled regions of the decision space, regions that may contain important solutions [28]. It appears that any CFO run might benefit, possibly substantially, by injecting some measure of negative gravity. This is strongly recommended based on these results and other examples discussed in the references. One simple approach is to use π-fractions, which are available online along with *Fano Load* data [29].

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. (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

[3] Formato, R.A. (2011) Central Force Optimization with Variable Initial Probes and Adaptive Decision Space. *Applied Mathematics and Computation*, **217**, 8866-8872. https://doi.org/10.1016/j.amc.2011.03.151

[4] Green, R.C., Wang, L. and Alam, M. (2012) Training Neural Networks Using Central Force Optimization and Particle Swarm Optimization: Insights and Comparisons. *Expert Systems with Applications*, **39**, 555-563. https://doi.org/10.1016/j.eswa.2011.07.046

[5] Shaikh, N.F. and Doye, D.D. (2014) A Novel Iris Recognition System Based on Central Force Optimization. *International Journal of Tomography and Simulation*, **27**, 23-34

[6] Huan, T.T. and Anh, H.P.H. (2019) Optimal Stable Gait for Nonlinear Uncertain Humanoid Robot Using Central Force Optimization Algorithm. *Engineering Computations*, **36**, 599-621. https://doi.org/10.1108/ec-03-2018-0154

[7] El-Hoseny, H.M., Abd El-Rahman, W., El-Rabaie, E.M., Abd El-Samie, F.E. and Faragallah, O.S. (2018) An Efficient DT-CWT Medical Image Fusion System Based on Modified Central Force Optimization and Histogram Matching. *Infrared Physics & Technology*, **94**, 223-231. https://doi.org/10.1016/j.infrared.2018.09.003

[8] Talal, T.M., Attiya, G., Metwalli, M.R., Abd El-Samie, F.E. and Dessouky, M.I. (2020) Satellite Image Fusion Based on Modified Central Force Optimization. *Multimedia Tools and Applications*, **79**, 21129-21154. https://doi.org/10.1007/s11042-019-08471-7

[9] Haghighi, A. and Ramos, H.M. (2012) Detection of Leakage Freshwater and Friction Factor Calibration in Drinking Networks Using Central Force Optimization. *Water Resources Management*, **26**, 2347-2363. https://doi.org/10.1007/s11269-012-0020-6

[10] Chen, Y., Yu, J., Mei, Y., Wang, Y. and Su, X. (2016) Modified Central Force Optimization (MCFO) Algorithm for 3D UAV Path Planning. *Neurocomputing*, **171**, 878-888. https://doi.org/10.1016/j.neucom.2015.07.044

[11] Chao, M., Xin, S.Z. and Min, L.S. (2014) Neural Network Ensembles Based on Copula Methods and Distributed Multiobjective Central Force Optimization Algorithm. *Engineering Applications of Artificial Intelligence*, **32**, 203-212. https://doi.org/10.1016/j.engappai.2014.02.009

[12] Carlin, H. (1977) A New Approach to Gain-Bandwidth Problems. *IEEE Transactions on Circuits and Systems*, **24**, 170-175. https://doi.org/10.1109/tcs.1977.1084325

[13] Rodriguez, J.L., Garcia-Tunon, I., Taboada, J.M. and Basteiro, F.O. (2007) Broadband HF Antenna Matching Network Design Using a Real-Coded Genetic Algorithm. *IEEE Transactions on Antennas and Propagation*, **55**, 611-618. https://doi.org/10.1109/tap.2007.891546

[14] Carlin, H. and Amstutz, P. (1981) On Optimum Broad-Band Matching. *IEEE Transactions on Circuits and Systems*, **28**, 401-405. https://doi.org/10.1109/tcs.1981.1085001

[15] Dedieu, H., Dehollain, C., Neirynck, J. and Rhodes, G. (1994) A New Method for Solving Broadband Matching Problems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **41**, 561-571. https://doi.org/10.1109/81.317955

[16] Fano, R. (1947) Theoretical Limitations on the Broadband Matching of Arbitrary Impedances. Sc.D. Thesis, Massachusetts Institute of Technology. http://hdl.handle.net/1721.1/12909

[17] Schwartz, D.F. and Allen, J.C. (2004) Wide-Band Impedance Matching: H/Sup/spI Infin// Performance Bounds. *IEEE Transactions on Circuits and Systems II: Express Briefs*, **51**, 364-368. https://doi.org/10.1109/tcsii.2004.831427

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

[19] Borwein, J. and Bailey, D. (2008) Mathematics by Experiment. 2nd Edition, AK Peters. https://doi.org/10.1201/b10704

[20] 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

[21] Bailey, D.H. (2006) BBP Code Directory. https://www.experimentalmath.info

[22] 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. *Forum for Electromagnetic Research Methods and Application Technologies*, **19**, Article No. 10. http://www.e-fermat.org/

[23] Formato, R.A. (2017) Determinism in Electromagnetic Design & Optimization-Part I: Central Force Optimization. *Forum for Electromagnetic Research Methods and Application Technologies*, **19**, Article No. 9. http://www.e-fermat.org/

[24] Ding, D., Luo, X., Chen, J., Wang, X., Du, P., and Guo, Y. (2011) A Convergence Proof and Parameter Analysis of Central Force. *Journal of Convergence Information Technology*, **6**, 16-23. https://doi.org/10.4156/jcit.vol6.issue10.3

[25] Marion, J. (1970) Classical Dynamics of Particles and Systems. 2nd Edition, Harcourt Brace Jovanovich.

[26] Brand, L. (1966) Differential and Difference Equations. John Wiley & Sons, Inc.

[27] Goldstein, H. (1965) Classical Mechanics. Addison-Wesley Publishing Co., Inc.

[28] Formato, R.A. (2025) Six-Element Yagi-Uda Array: An Example of Using Negative Gravity in Central Force Optimization for Improving Decision Space Exploration. *Wireless Engineering and Technology*, **16**, 1-39. https://doi.org/10.4236/wet.2025.161001

[29] (2025) π-fractions #0-1,000,001. https://app.box.com/s/qdd8rzrhgaozne0ag1nes9jkm0bj6ark