

Gravitational Interactions Optimization

Juan J. Flores¹, Rodrigo López¹, and Julio Barrera²

¹ Universidad Michoacana de San Nicolás de Hidalgo
División de Estudios de Posgrado, Facultad de Ingeniería Eléctrica

² CINVESTAV-IPN

Departamento de Computación
Evolutionary Computation Group
Av. IPN No. 2508, Col. San Pedro Zacatenco
México, D.F. 07360, MEXICO

`juanf@umich.mx, rlopez@faraday.fie.umich.mx, julio.barrera@gmail.com`

Abstract. Evolutionary computation is inspired by nature in order to formulate metaheuristics capable to optimize several kinds of problems. A family of algorithms has emerged based on this idea; e.g. genetic algorithms, evolutionary strategies, particle swarm optimization (PSO), ant colony optimization (ACO), etc. In this paper we show a population-based metaheuristic inspired on the gravitational forces produced by the interaction of the masses of a set of bodies. We explored the physics knowledge in order to find useful analogies to design an optimization metaheuristic. The proposed algorithm is capable to find the optima of unimodal and multimodal functions commonly used to benchmark evolutionary algorithms. We show that the proposed algorithm (Gravitational Interactions Optimization - GIO) works and outperforms PSO with niches in both cases. Our algorithm does not depend on a radius parameter and does not need to use niches to solve multimodal problems. We compare GIO with other metaheuristics respect to the mean number of evaluations needed to find the optima.

Keywords: Optimization, gravitational interactions, evolutionary computation, metaheuristic

1 Introduction

Multimodal optimization problems deal with objective functions that commonly contain more than one global optima and several local optima. In order to find all the global optima in multimodal problems with classical methods, one typically runs a given method several times with different starting points, expecting to find all the global optima. However, these techniques do not guarantee the location of all optima. Therefore, this kind of techniques are not the best way to explore multimodal functions with complex and large search spaces. In the evolutionary computation literature exists a variety of metaheuristics challenging the typical problems of classical optimization. E.g. In particle swarm optimization with niches; the best particle makes a niche with all particles within a radius

r , until the niche is full; it then selects the next best no niched and its closets particles to form the second niche; the process until all particles are assigned to a niche. Objective function stretching, introduced by Parsopolous [8], [9] is another algorithm whose strategy is to modify the fitness landscape in order to remove local optima and avoid the premature convergence in PSO. In a minimization problem, a possible local minimum is stretched to overcome a local maximum allowing to explore other sections of the search space identifying new solutions. GSA introduced by Rashedi [10], is a gravitational memory-less (does not include a cognitive component in the model) metaheuristic capable to find only one global optima in unimodal and multimodal problems with more than one global optima, where a heavier mass means a better solution and the gravitational constant G is used to adjust the accuracy search. Section 2 compares GIO with GSA and CSS, two other metaheuristics very similar to our proposal. Section 3 reviews Newton's universal gravitation law. Section 4 proposes Gravitational Interactions Optimization. The main contribution of this paper. Section 5 presents the experimental framework and the obtained results. Finally, Section 6 presents our conclusions.

2 Review GSA GIO and CSS

GIO has similarities with another two algorithms inspired by the nature: GSA (Gravitational Search Algorithm), inspired on gravitational interactions and CSS Charge Search System (CSS) inspired on electrostatic dynamics laws. Our work is very similar to theirs, but since we have publications from around the same micai2009, micai2010, and have not had any personal communication with their respective authors, we can state that this work is independent from the works of CSS and GSA.

First GSA and CSS is very similar to GIO, assigning masses and charges respectively to the bodies, according to the fitness function in the place located, in order to determine the evolution.

The gravitational constant G in GSA decreases linearly with the time in GIO. We think that is not a good idea because convergence is regularly not linear. In an attempt to increase its exploration capabilities, when GSA is determining the total force exerted to a body, it weights each component (the exerted by each other body) by a random number. This situation, in the worst, destroy the underlying metaphor, i.e. the gravitational interaction.

Another point is that GSA uses Kbest agents in order to minimize computing time, although the complexity of the algorithm is not reduced. On the other hand, we allow all masses (agents) to interact with each other.

The CSS algorithm try to imitate the electrostatic dynamics in order to optimize unimodal and multimodal functions, assigning charge to the particles similarly as GSA and GIO assign masses to the bodies, We think that the high level of detail of the charges of the particles in CSS is unnecessary, and the parameter a is very large to functions; the estimation of this parameter could be arbitrary. We think that CSS lost sense of electromagnetism dynamics when they

assign binary flags to determine the direction of the attraction of the bodies. All charges are positive, and still attract each other, departing from the electrostatic metaphor and making it look more like a gravitational one.

Another important difference is that both CSS and GSA aim to locate the global optimum for multimodal function, while GIO's main interest is to determine all local and global optima for multimodal functions.

In our work we explore the properties of gravitational interactions in order to make an useful metaheuristic to find optima in unimodal and multimodal problems. In Section 2 addresses the main motivation of our work: The Newton's Law of Universal Gravitation. In Section 3 we define the Gravitational Interactions Optimization (GIO) metaheuristic for unimodal and multimodal functions. Section 4 presents to the GIO metaheuristic with different unimodal and multimodal problems. Section 5 presents the conclusions of this work.

3 Newton's Law of Universal Gravitation

The attraction force of two particles is proportional to their masses and inversely proportional to their distance. The Law of Universal Gravitation was proposed by Isaac Newton [7]. This law is stated in Definition 1.

DEFINITION 1 *The force between any two particles having masses m_1 and m_2 , separated by a distance r , is an attraction acting along the line joining the particles and has the magnitude. Shown in Equation (1).*

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

where G is a universal gravitational constant.

The forces between two particles with mass are an action-reaction pair. Two particles with masses m_1 and m_2 exert attracting forces F_{12} and F_{21} towards each other whose magnitudes are equal but their directions are opposed.

The gravitational constant G is an empirical physical constant involved in the computation of the gravitational attraction between particles with mass, which can be determined by the maximum deflection method [11].

$$G = 6.673 \times 10^{-11} N(m/kg)^2 \quad (2)$$

The gravitational force is extremely weak compared to other fundamental forces; e.g. the electromagnetic force is 39 orders of magnitude greater than the gravity force.

Newton's law of universal gravitation can be written in vectorial notation, which considers both: The force of the masses and the direction of each force. The vectorial notation is shown in Equation (3).

$$F_{12} = -G \frac{m_1 m_2}{|r_{12}|^2} \hat{r}_{12} \quad (3)$$

Where F_{12} is the force exerted by m_1 on m_2 , G is the gravitational constant, m_1 and m_2 are the masses of the particles, $|r_{12}|$ is the euclidean distance between particles m_1 and m_2 , and \hat{r}_{12} is the unit vector, defined as $\frac{r_2 - r_1}{|r_2 - r_1|}$, r_1 and r_2 are the locations of particles m_1 and m_2 . (See Figure 1).

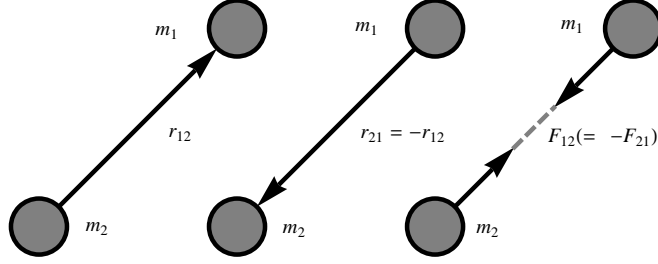


Fig. 1. The force exerted on m_2 (by m_1), F_{21} , is directed opposite to the displacement, r_{12} , of m_2 from m_1 . The force exerted on m_1 (by m_2), F_{12} , is directed opposite to the displacement, r_{21} , of m_1 from m_2 . $F_{21} = -F_{12}$, the forces being an action-reaction pair

4 Gravitational Interactions Optimization

In order to find one or more optima there exists a large variety of evolutionary algorithms, e.g. genetics algorithms (GA) [3], evolutionary strategies (ES) [4], ant colony optimization (ACO) [2], particle swarm optimization (PSO) [5], electrostatic PSO (EPSO) based on electrostatic interactions inspired upon PSO [1], etc. There exist works related to design metaheuristics that take into account the distance in order to determine the cluster membership of the particles computing and maximizing a ratio for all particles in the swarm with respect to the particle to be updated, e.g. FER-PSO [6]. We propose a Gravitational Interaction Optimization metaheuristic (GIO) capable of solving optimization problems. The motivation of the design of this metaheuristic is to find useful properties and analogies that can relate optimization problems with Newton's gravitational theory. In the approach presented in this paper, we abduct the interactions exhibited by a set of bodies and use them to guide the search for the global optimum in an optimization problem.

4.1 Gravitational Interactions for Unimodal Optimization

GIO is a population-based metaheuristic where a set of bodies are initially dispersed along the search space with a uniform random distribution. The fitness of bodies located on the search space are mapped as masses in a Gravitational field where the solutions are evolved. Each body stores its current position B and the best position so far B_{best} according to the fitness function. Bodies are

allowed to interact in a synchronous discrete manner for a number of epochs. The body interactions follow Newton's gravitational law and move each body to a new location in such way that whole population tends to reach the global optimum (or multiple local optima for multi-modal problems).

The fitness function is a mapping that transforms a vector $X = (x_1, x_2, \dots, x_n)$ to a scalar $f(X)$. This mapping associates the fitness value $f(X)$ to each location $X = (x_1 \dots x_n)$ of the search space. We assign a body B to every location X in the search space where an individual of the population is found. Body B is assigned a mass, whose magnitude is a function of the fitness of its location.

Newton's law of universal gravitation describes the attraction forces that exist between two punctual bodies with masses (described in vectorial form in 3). Substituting we obtain Equation (4).

$$\mathbf{F}_{ij} = \frac{M(f(B_i)) \cdot M(f(B_j))}{|B_i - B_j|^2} \hat{B}_{ij} \quad (4)$$

Where M is the mapping function that associates the fitness value f of domain $\{x : x \in \mathbb{R}\}$ a mass of codomain $\{y : y \in (0, 1]\}$ for each position of the body B_i . This mapping is computed using Equation (5).

$$M(f(B_i)) = \left(\frac{f(B_i) - \min f(B)}{\max f(B_{best}) - \min f(B)} (1 - \text{mapMin}) + \text{mapMin} \right)^2 \quad (5)$$

Where B_i is the position of the i th body and B_j is the j th body that contributes exerting a force on the mass B_i ; $|B_i - B_j|$ is the euclidean distance and \hat{B}_{ij} is the unit vector between bodies B_i and B_j ; $f(B_i)$ is the fitness of body B_i , $\min f(B)$ is the minimum fitness value of the current positions of the bodies, $\max f(B_{best})$ is the maximum fitness value of the best positions so far, mapMin is a constant with a small positive value near zero, such that $(1 - \text{mapMin})$ rescales the fitness value $f(B_i)$ to a mass between $[0, 1)$ values. The result is squared to emphasize the best and worst fitnesses.

One characteristic of the proposed method is the full interaction; i.e each body B_i interacts with every other body B_j through their masses. Interactions contribute to their displacement, according to the resultant force. Equation (6) computes the resultant force exerted on body B_i by the bodies B_j .

$$\mathbf{F}_i = \sum_{j=1}^n \frac{M(f(B_i)) \cdot M(f(B_{j,best}))}{|B_i - B_{j,best}|^2} B_i \hat{B}_{j,best} \quad (6)$$

Where \mathbf{F}_i is the resultant force of the sum of all vector forces between $M(B_i)$ and $M(B_{j,best})$, $|B_i - B_{j,best}|$ is the Euclidean distance between the current positions of body B_i and the best position so far of the body B_j . In order to avoid numerical errors we compute the force between masses $M(B_i)$ and $M(B_{j,best})$ only if $|B_i - B_j| \geq \times 10^{-5}$ (if the distance is smaller than that, we suppose both bodies collided already and are located in the same place; we are assuming punctual masses), $B_i \hat{B}_{j,best}$ is the unit vector that directs the force. In

order to estimate a displacement that could enhance the solution of particle B_i , it is necessary to solve Equation (4) for B_j . Assuming that we want to find a location of the body B with $M(f(B)) = 1$, B is computed using Equation (7).

$$B = \sqrt{\frac{M(f(B_i))}{|\mathbf{F}_i|}} \hat{\mathbf{F}}_i \quad (7)$$

To update the position of the bodies we use Equations (8) and (9).

$$V_{new} = \chi(V + R \cdot C \cdot B) \quad (8)$$

$$B_{t+1} = B + V_{new} \quad (9)$$

V is the current velocity of B_i , R is a random real number generated in the range of $[0, 1)$ and is multiplied by the gravitational interaction constant C , in order to expect random exploration distances with mean $\mu \approx 1$, we set $C = 2.01$, this displacement is constrained multiplying by a constant with a value of 0.86, in order to ensure the convergence. B is the main displacement computed by Equation (7).

Using Newton's law $F = ma$, we can compute velocity and from there displacement, as in GSA [10]. Using this scheme though, when bodies are far apart, forces are small, and the resulting displacement is also small; when the method is converging, bodies are closer to each other, producing larger forces, therefore larger displacements. This leads to a divergent or at least non-convergent behavior. GSA solves this problem by assuming G as a linearly decreasing function of time. As in classical mechanics, we consider G a constant and use a heuristic solution to this problem: where should a body of unitary mass be located to produce the same resulting force in B_i ? We use that location as B_i 's new location.

Using this heuristic, when bodies are far apart from each other, forces are small, as they would be produced by a unitary mass located far away. When the method is converging and masses are close together, resulting forces are larger, as produced by a close unitary mass, resulting in small displacements. This heuristic leads to very a convenient convergence scheme, where exploration takes place at the beginning and exploitation at end of the process.

The complete GIO algorithm is described the Algorithms 1, 2 and 3. Algorithm 1 computes the the total force exerted by the masses $M(f(B_j))$ mass $M(f(B_i))$; in order to prevent premature convergence and division by 0, we compute only those pairs of bodies with a distance greater than ϵ . Algorithm 2 computes the velocities of the bodies, receives the bodies and computes the resultant force that attracts the mass assigned to B_i . In order to prevent a division by 0 we compute the distance only if $|F_{total}| > 0$, the new velocity is computed by Equation (8), and finally we update the velocity associated to B_i . Algorithm 3 computes the new positions B of each iteration t , the algorithm take as parameters the search range, the number of bodies $nBodies$, and the

maximum number of iterations $tMax$. The algorithm computes the velocities with $computeVelocities(bodies)$ (Algorithm 2), and updates the their positions with $updatePosition()$, which implements Equation (9), $limitPositions()$ limits the positions of the bodies to the search space defined by the search range, $updateFitness()$ updates the fitness according to the new positions of the bodies and finally we update the best position so far with $updateB_{best}()$.

Algorithm 1 computeFtotal(index)

```

1:  $i \leftarrow index$ 
2:  $Ftotal \leftarrow 0$ 
3: for  $j \leftarrow 1$  to  $nBodies$  do
4:   if  $distance(B_i, B_{j,best}) > \epsilon$  then
5:      $Ftotal \leftarrow Ftotal + B_{i,j,best} M(f(B_i)) M(f(B_{j,best})) / distance(B_i, B_{j,best})^2$ 
6:   end if
7: end for
8: return  $Ftotal$ 

```

Algorithm 2 computeVelocities(bodies)

```

1: for  $i \leftarrow 1$  to  $nBodies$  do
2:    $Ftotal \leftarrow computeFtotal(i)$ 
3:   if  $|Ftotal| > 0$  then
4:      $distance \leftarrow \sqrt{M(f(B_i)) / |Ftotal|}$ 
5:   else
6:      $distance \leftarrow 0$ 
7:   end if
8:    $V_{new} \leftarrow \chi(V + R \cdot C \cdot distance \cdot \hat{Ftotal})$ 
9:    $updateVelocity(B_i, V_{new})$ 
10: end for
11: return  $Ftotal$ 

```

This scheme develops good results for unimodal problems. The results of the performace of the algorithm presented in this Section are presented in Section 4.

4.2 Gravitational Interactions for Multimodal Optimization

In the previous Subsection we showed the basic steps of the gravitational interactions metaheuristic. This scheme works well for unimodal problems. For multimodal problems it is necessary to add a cognitive component analogous to the one used in PSO [5]; the cognitive component is a constant that gives a weight to each body's memory. The new positions of the bodies are computed in order to find more than one optima with the Equations (10) and (11).

Algorithm 3 MainGravitationalInteraction(ranges, nBodies)

```

1: bodies  $\leftarrow$  initializeParticles(nBodies, ranges)
2: for  $t \leftarrow 0$  to maxIter do
3:   computeVelocities(bodies)
4:   limitVelocity()
5:   updatePosition()
6:   limitPosition()
7:   updateFitness()
8:   updatePbest()
9: end for

```

Adding the cognitive component to Equation (8) and using the constriction factor χ (Equation 12), makes the new Equation (10) capable to find more than one optimum in multimodal problems. The effect of this component is to make the local search more robust restricting to the bodies to local search, unless the gravitational forces of a cluster of masses overcome the force exerted by its cognitive component.

$$V_{new} = \chi(V + C_1 \cdot R_1 \cdot (B_{best} - B) + C_2 \cdot R_2 \cdot a) \quad (10)$$

$$B_{new} = B + V_{new} \quad (11)$$

where, analogous to PSO, C_1 and C_2 are the cognitive and the gravitational interaction constants, R_1 and R_2 are real random numbers variables in the $[0, 1]$ range and χ is the inertia constraint (Proposed by Clerk ([5])). The inertia constraint is used to avoid the bodies to explore out of the search space computed by Equation (12).

$$\chi = \frac{2\kappa}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \quad (12)$$

where $\phi = C_1 + C_2 > 4$, C_1 and C_2 ; κ is an arbitrary value in the range of $(0, 1]$. In our algorithm we set $C_1 = C_2 = 2.01$. The constriction factor in our algorithm helps to converge through the iterations.

To make multimodal Gravitational Interactions Algorithm (Algorithms 1, 2, and 3) described in the previous subsection, we replace line 8 in Algorithm 2 by Equation 10.

5 Experiments

In order to test the performance of the Gravitational Interactions Optimization algorithm for unimodal and multimodal functions, we tested both versions with some functions commonly used to measure the performance of different kinds of metaheuristics.

5.1 Test Functions

We show the performance of unimodal and multimodal Gravitational Interactions Optimization algorithm with 3 unimodal and 4 multimodal functions. The test functions tested are shown in the Table 1.

Unimodal Test Functions		
$U1$	$U1 = [1 + (1 + (x + y + 1)^2)(19 - 14x + 3y^2 + 6xy + 3y^2)] \cdot [(30 + (2x - 3y)^2)(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)]$	$-2 \leq x, y \leq 2$
$U2$	$U2 = (x + 2y - 7)^2 + (2x + y - 5)^2$	$-10 \leq x, y \leq 10$
$U3$	$U3 = -1100 \cdot (w^2 - x)^2 + (w - 1)^2 + (y - 1)^2 + 90 \cdot (y^2 - z)^2 + 10.1 \cdot ((x - 1)^2 + (z - 1)^2) + 19.8 \cdot (x^{-1}) \cdot (z - 1)$	$-10 \leq w, x, y, z \leq 10$
Multimodal Test Functions		
$M1$	$M1 = -\left(\left(y - \frac{5.1x^2}{4\pi^2} + \frac{5x}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x) + 10\right)$	$-5 \leq x \leq 10$ $0 \leq y \leq 15$
$M2$	$M2 = \sin(5\pi x)^6$	$-0 \leq x \leq 1$
$M3$	$M3 = -(x^2 + y - 11)^2 - (x + y^2 - 7)^2$	$-6 \leq x, y \leq 6$
$M4$	$M4 = -4\left((4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2\right)$	$-1.9 \leq x \leq 1.9$ $-1.1 \leq y \leq 1.1$

Table 1. Test functions used for our experiments

For unimodal optimization we used the functions in Figure 2: $U1$ is the Goldstein and Price function shown in Figure 2(a), $U2$ is the Booth function shown in Figure 2(b) and $U3$ is the 4 variable Colville Function. For multimodal optimization we used the functions of the Figure 3 $M1$ is the Branin's RCOS Function with 3 global optima (with no local optima) shown in Figure 3(a), $M2$ is the 6 global maximum univariable Deb's function shown in Figure 3(b), $M3$ is the Himmelblau's function with 4 global optima shown in Figure 3(c), $M4$ is the Six-Hump cammelback function with 2 global optima and 4 local optima shown in Figure 3(d).

5.2 Results

In our experiments we consider $\epsilon = 1 \times 10^{-3}$ to be an acceptable error to determine if the solution obtained had reached the optimum. We used 100 bodies for a maximum of 1000 iterations, we used as stop condition the inability of all the bodies to enhance their fitness memory solutions by 1×10^{-4} , or when the algorithm found all the optima. Each experiment was repeated 30 times. PSO with niches requires two extra parameters: the radius r , and the maximum number of particles on each niche $nMax$, we set $M1$ with $r = 0.5$ and $nMax = 50$, $M2$ with $r = 0.1$ and $nMax = 15$, $M3$ with $r = 0.5$ and $nMax = 30$, and $M4$ with $r = 0.5$ and $nMax = 25$.

The performance of Gravitational Interaction Optimization (GIO) is compared with Particle Swarm Optimization with niches (NPSO) in Table 2, considering the mean and the standard deviation of evaluations required to find

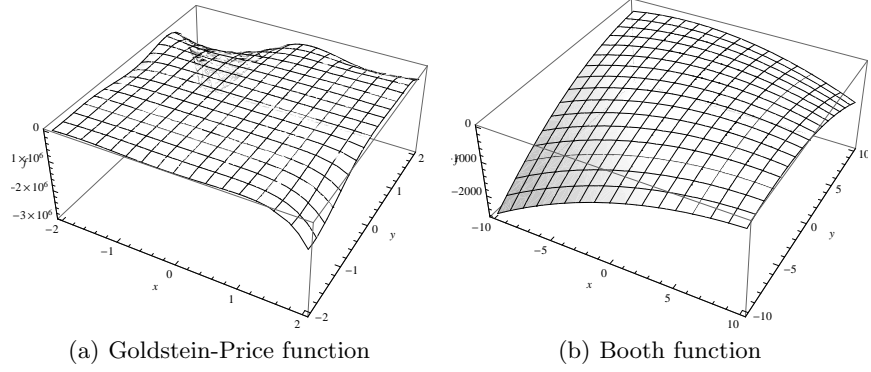


Fig. 2. Fitness landscape of two test functions with one optima used for measure the performance of Unimodal Gravitational Interactions

all the global optima (column **Evaluations**) and the percentage of successes (column **Success**) to finding all the optima.

Functions	PSO			GIO Unimodal		
	Evaluations		Success	Evaluations		Success
	μ	σ		μ	σ	
$U1$	1,394.44	399.22	20%	5,653.33	711.838	100%
$U2$	1,130.77	330.11	60%	6,057.55	3,984.54	70%
$U3$	764.00	777.75	83%	530.00	208.69	100%
	NPSO			GIO Multimodal		
	Evaluations			Evaluations		
	μ	σ		μ	σ	
$M1$	2,529.17	764.13	80%	2,803.33	972.90	100%
$M2$	276.66	81.72	100%	390.00	88.44	100%
$M3$	3,400.00	0.00	00.3%	2,323.33	288.496	100%
$M4$	1,136.67	303.41	100%	1,600.00	501.721	100%

Table 2. Results of our experiments

The obtained results show that Unimodal and Multimodal Gravitational Interactions have a higher probability to converge to global optima, avoiding premature convergence that PSO and PSO with niches with a similar number of evaluations required for the functions tested. We proved the GIO algorithm replacing the Equation 7 by the acceleration Equation proposed in [10] respect the Newton's law gravity multiplied by the gravity constant showed in Equation 2, the results were not better than GIO but we think that the Equation 7 could give us a more accurate behavior of gravitational constant G through the iterations.

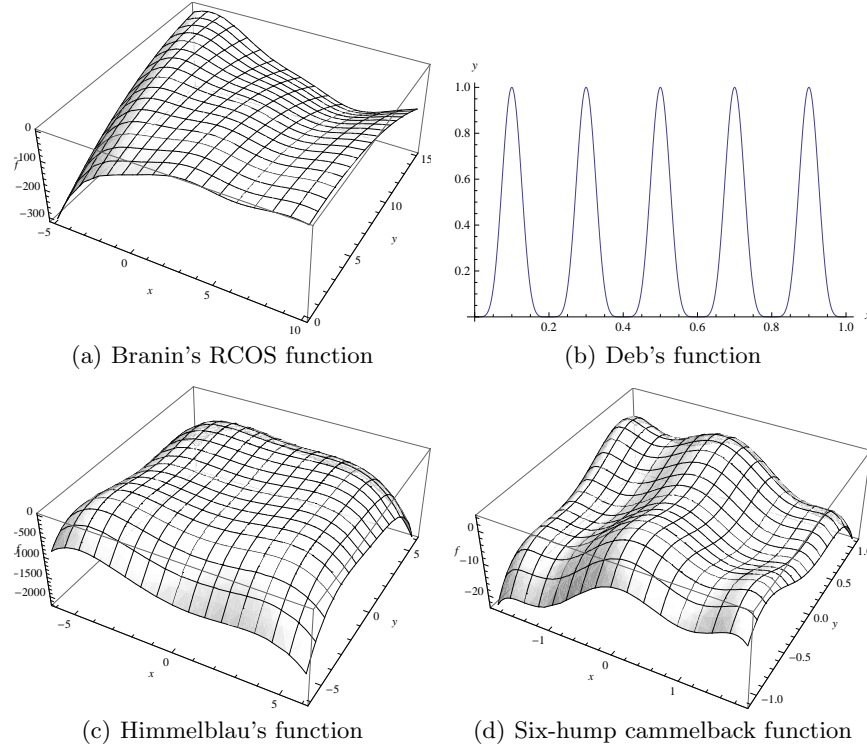


Fig. 3. Fitness landscape of multimodal test functions used in our experiments

6 Conclusions

We presented a new heuristic, more reliable than PSO, with no additional parameters like the radius and the maximum number of particles in a niche used in PSO with niches. In problems with high dimensions the radius is determined by trial and error, because we can not graph the objective function and make a visual analysis.

The same algorithm GIO, is used for unimodal and multimodal cases. When used in its general form. (i.e. including the cognitive component), GIO solves both cases without the need of any a-priori information. Adding the cognitive component allow us to solve both, unimodal and multimodal optimization problems, while GSA can only solve unimodal problems.

Furthermore, GIO has proven to find all optima in a multimodal problem, GSA can only determine one of them.

References

1. Julio Barrera and Carlos A. Coello Coello. A particle swarm optimization method for multimodal optimization based on electrostatic interaction. In *Proceedings of*

- the 8th Mexican International Conference on Artificial Intelligence (MICAI '09)*, pages 622–632, Berlin, Heidelberg, 2009. Springer-Verlag.
2. Alberto Coloni, Marco Dorigo, and Vittorio Maniezzo. Distributed optimization by ant colonies. In *Proceedings of the Parallel Problem Solving from Nature Conference*. Elsevier Publishing, 1992.
 3. David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1 edition, January 1989.
 4. Rechenberg Ingo. *Evolutionstrategie '94*. PhD thesis, Technische Universität Berlin, 1994.
 5. J. Kennedy and R. Eberhart. *Swarm Intelligence*. Evolutionary Computation. Morgan Kaufmann Publisher, 2001.
 6. Xiaodong Li. A multimodal particle swarm optimizer based on fitness euclidean-distance ratio. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation (GECCO '07)*, pages 78–85, New York, NY, USA, 2007. ACM.
 7. I. Newton. *Newtons Principia Mathematica*. Física. Ediciones Altaya, S.A., 21 edition, 1968.
 8. K. E. Parsopoulos, G.D. Magoulas, Uxbridge Ub Ph, M.N. Vrahatis, and V.P. Plagianakos. Stretching technique for obtaining global minimizers through particle swarm optimization. In *Proceedings of the Particle Swarm Optimization Workshop*, pages 22–29, 2001.
 9. K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis. Improving the particle swarm optimizer by function “stretching”. *Nonconvex Optimization and its Applications*, 54:445–458, 2001.
 10. Esmat Rashedi, Hossein Nezamabadi-pour, and Saeid Saryazdi. Gsa: A gravitational search algorithm. *Information Sciences*, 179(13):2232–2248, 2009.
 11. Hallyday. Robert, Resnick. David. *Physics Part I*. Physics. John Wiley and Sons Inc, 1966.