Using Gravitational Force in Terrain Optimization Problems

Erisa Terolli

Epoka University

Briland Hitaj

Epoka University

Oguz Altun

Epoka University

Abstract

Optimization problems represent algorithms designed for difficult problems which may require huge amount of space or computational time. Such kind of algorithms bring out solutions which are optimal and at the same time closer to the real life environments where not everything is precise and the possibility of errors is present at every instant.

Finding the minimal point in a terrain is a challenge of its own, especially when we are dealing with an unknown area. In order to tackle this problem, we thought of making use of gravitational force, since it is proportionally related to earth center proximity. In an unknown terrain, we spread our agents that are capable to communicate information to one another at randomly generated positions. Later on, each of these agents calculates the gravity variation with altitude at its respective position. Since, we were looking to find the optimal minimum point in the terrain, after the gravity variation with altitude is computed by each agent, the highest gravity is found. This is communicated to the other agents as well and they start moving toward the agent that is currently found at an area with high gravity variation. The agents move toward the high gravity agent with a certain heuristic coefficient. During their path they may encounter other terrain points where the gravitational force is stronger, which would cause a change in the path of other agents making them move toward the newly found position. This is done until an optimal minimum is found by the agents.

Our test results so far have been very promising. We aim to develop the algorithm furthermore in order to increase its efficiency and efficacy. We strongly believe that such an algorithm can be used to reach in the unexplored areas of ocean floor, or searching for minerals by minimizing the area of search in an optimal time.

Keywords: Optimization problems, Gravity variation, physics, agents, minimal point

1. Introduction

Using mathematical terms an optimization problem may be represented by the pair of S and f where set S contain all possible solutions and function f (f: S -> R) is the function to be optimized. Solving any optimization problem represent the process of finding one or more global optimal solution for function f (Talbi, 2009).

However there exits some hard optimization problems, which cannot be solved optimally using any exact methodology within an acceptable time period (Ilhem et. al. 2013). One well known methodology for solving this kind of hard optimization problems is Metaheuristics. It represent a set of optimization techniques used to give reasonable (but not always optimal) solutions to specific optimization problems (Talbi, 2009). Metaheuristic solutions generally are inspired by nature principles of physics, biology etc. and they involve stochastic components, components which involved usage of random variables (Ilhem et. al. 2013).

Two important optimization algorithms are Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) which uses evolutionary computation techniques and swarm intelligence techniques respectively (Liberti et.al. 2006). GA is developed by Holland et.al in 1970 at the University of Michigan and imitates the process of natural evolution (Mitchell, 1999). On the other hand PSO, developed by Kennedy and Eberhet in 1995, is a very simple optimization algorithm and is based on the movement behavior of bird and fish flocks (Bai, 2010).

In this work will be described a naturally inspired Metaheuristics optimization algorithm like GA and PSO named Earth Gravitational Optimization (EGO) algorithm. Making use of earth gravitational force, EGO attempts to find global minimums of island terrain in an effective and efficient approach.

The remaining part of this paper will be organized as follows. Section II will give a detailed description of the search space (island terrain) in which search will take place and the proposed Earth Gravitational Optimization (EGO) algorithm. In Section III we will make a comparison between GA, PSO and EGO based on the terrain described at Section II. Finally, conclusions and future works are given in Section IV.

2. Earth Gravitational Optimization (EGO) Algorithm

Generally in physics calculations we consider gravity as a constant value, but it varies in different heights of terrain, since greater height of terrain means greater distance from the Earth's center. Gravity of an object at height h above the Earth surface is calculated by Eq (1).

$$g_h = \frac{G * M_{Earth}}{(R_{Earth} + height)^2} \tag{1}$$

The Gravity Constant or g is calculated by the formula in Eq(2).

$$g_0 = \frac{G * M_{Earth}}{{R_{Earth}}^2} \tag{2}$$

So, a gravity variation at different heights can be simply calculated by subtracting from gh, g0 as shown below in Eq(3).

$$deltag = \frac{G * M_{Earth}}{(R_{Earth} + height)^2} - \frac{G * M_{Earth}}{{R_{Earth}}^2} \tag{3}$$

EGO algorithm is based on the fact that the higher the object is from Earth the lower will be its' gravity and the lower it is the hight of object from Earth the higher its' gravity is.

Identical agents which are capable to communicate information with each other are distributed randomly inside the island terrain. Each of them calculates the gravity variation with altitude at its respective position and communicates that value to the other agents. The agent which has the highest gravity variation meaning that has the highest gravity starts attracting the other remaining agents toward it.

The agents move toward the high gravity agent with a certain heuristic coefficient. This heuristic coefficint is calculated based on the distance on any agent from the current optimal agent. Coefficient will be indirectly proportional with the distance between two agents and it must be a number in]0,1[interval. For this reason it is suggested to be used the famous theorem that sum of two sides of a triangle is bigger than the other side, which means that ratio of sum of two triangle sides over the other side is always in the desired interval. If coordinates of an agent and the best optimal agent are (x1, y1) and (x2,y2) respectively the value of the coefficient is specified by Eq(4).

$$c = \frac{\sqrt{(x2 - x1)^2 + (y2 - y1)^2}}{abs(x2 - x1) + abs(y2 - y1)} \tag{4}$$

Agents, while travelling toward the agent with the highest gravity, in case of finding any position with gravity higher than the current best they force all other agents to start moving toward the new minimal position found. This procedure is repeated until optimal minimum is found by the agents.

The pseudocode of EGO algorithm.

```
Set Iteration = 0;
INITIALIZE
    Generate Agents at random positions
    Set OptimalGravityValue=0;
    Set OptimalAgent=-1;
WHILE(Optimum in NOT found)
        FOR(each agent)
            Evaluate gravity value following Eq (3);
            İf(currentGravityValue > LocalOptimalGravityValue)
                    OptimalGravityValue = currentGravityValue;
                    OptimalAgent = currentAgent;
        END FOR
```

```
7:                  FOR(each agent)
                        Change position toward LocalOptimalAgent following Eq (4);
                    END FOR
8:                  Return to Step 5
9:          END WHILE
```

3. Simulation Results

A Matlab simulation of EGO algorithm is made for finding the global minimum point in an island terrain generated by the pseudocode presented at below and visually displayed at Figure 1.

The pseudocode of island terrain generation.

```
1:      Set Iteration = 0;
2:      INITIALIZE
3:      FOR i = 0 to 500
4:              FOR j = 0 to 500
                        Set matrix element[i][j]=0;
                ENDFOR
        ENDFOR
5:      WHILE (Iteration<100)
                Genarate one peaks function[1];
                Put generated function randomly to matrix;
                Iterations = Iteration +1;
6:       END WHILE
```

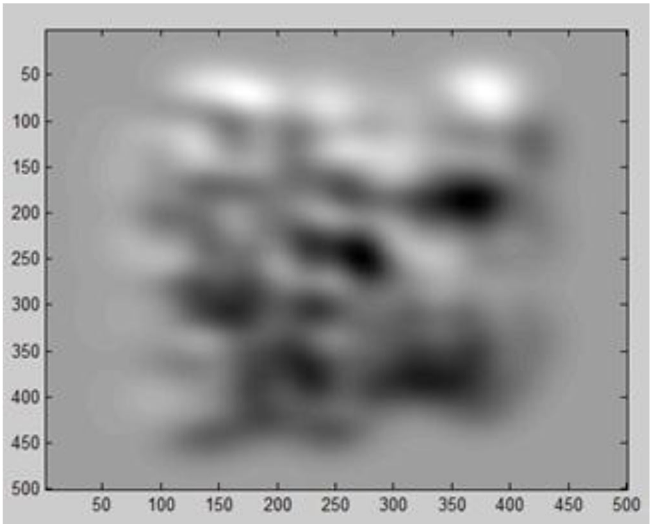

Figure 1 - Island Terrain Matlab Generation

In this simulation it is taken into consideration an 500x500 matrix where all points represent a specific terrain altitude which gets darker as altitude increases. The other parameters taken

[1] Peaks function is a Matlab function which returns a 49-by-49 matrix, generated by translating and scaling Gaussian distributions (Matlab Documentation Help)

consider for this simulation are shown at Table 1.

CONTROL PARAMETER	A
Number of Iterations	10
Number of agents	30
Global Minimum	-27.30565

Table 1- Control Parameters used in EGO algorithm simulation

EGO algorithm simulation is repeated 30 times with different random position for each agent. The simulation results in an average mean of -23.4222 with a Standard Deviation value of 2.755396. One of the best results of this simulation is visualized in Figure 2.

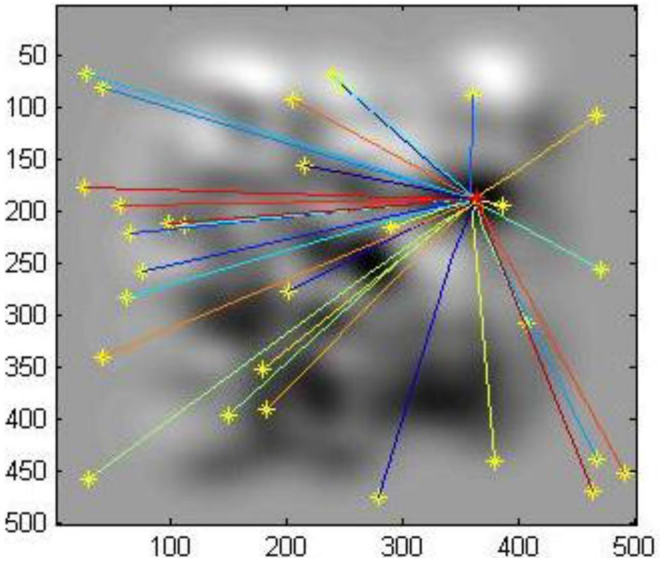

Figure 2 - Simulation Screenshot

4. Conclusion

A new nature-inspired optimization algorithm used in terrain optimization problems is proposed in this work. The new proposed algorithm is based on the gravitational force indirect proportionality with the earth center proximity. It is very simple since it uses only an ever-lasting force like gravitational force and some simple mathematical principles like triangle inequality theorem. The experimental results shows that this new algorithm works pretty well with terrain optimization problems. However the performance of this algorithm must be tested using a larger set of functions. In the near future it will be tested using the Benchmark functions, like Sphere function, Rosenbrock valley, Rastrigin function etc.

Acknowledgements: Assistance provided by Mr. Ibrahim Mesecan was greatly appreciated.

References:

Talbi, E. G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics.*Information Sciences*.

Liberti, L., & Maculan, N. (Eds.). (2006). *Global Optimization: Volume 84, From Theory to Implementation* (Vol. 84). Springer.

Mitchell, M. (1998). *An introduction to genetic algorithms* (complex adaptive systems).

Bai, Q. (2010). Analysis of particle swarm optimization algorithm. *Computer and information science*, *3*(1), P180.