

Optimizing Image Steganography using Particle Swarm Optimization Algorithm

Rafael Lima de Carvalho
Computer Science Department
University Federal of Tocantins
Palmas/Tocantins - Brazil

Warley Gramacho da Silva
Computer Science Department
University Federal of Tocantins
Palmas/Tocantins - Brazil

Ary Henrique Oliveira de Morais
Computer Science Department
University Federal of Tocantins
Palmas/Tocantins - Brazil

ABSTRACT

Image Steganography is the computing field of hiding information from a source into a target image in a way that it becomes almost imperceptible from one's eyes. Despite the high capacity of hiding information, the usual Least Significant Bit (LSB) techniques could be easily discovered. In order to hide information in more significant bits, the target image should be optimized. In this paper, it is proposed an optimization solution based on the Standard Particle Swarm Optimization 2011 (PSO), which has been compared with a previous Genetic Algorithm-based approach showing promising results. Specifically, it is shown an adaptation in the solution in order to keep the essence of PSO while remaining message hosted bits unchanged.

General Terms

Computational Intelligence, Particle Swarm Optimization

Keywords

Image Steganography, Particle Swarm Optimization, Genetic Algorithm, Image Processing, Optimization

1. INTRODUCTION

Steganography is the art of hiding digital information in order to prevent the detection of hidden messages, in the other words, steganography's goal is to avoid drawing suspicion to the transmission of a secret message. It includes techniques for hiding an image, a text file, an audio file, and even an executable program inside a cover image without distorting the cover image [12, 7]. Such as in [5] steganography can be formally defined as the embedding process that described a mapping $E : C \times M \rightarrow C$, where C is the set of possible covers and M the set of possible messages. The extraction process consists of a mapping $D : C \rightarrow M$, extracting the secret message out of a cover.

DEFINITION 1. The quadruple $\Psi = \langle C, M, D, E \rangle$, where C is the set of possible covers, M the set of secret messages with $|C| \geq |M|$, $E : C \times M \rightarrow C$ the embedding function and $D : C \rightarrow M$, the extraction function, with the property that $D(E(c, m)) = m$ for all $m \in M$ and $c \in C$ [5].

The embedding process is defined in a way that the cover and the corresponding stego-object are perceptually similar. The steganography scheme can be seen in the Figure 1.

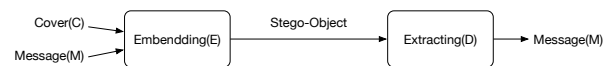


Fig. 1. Steganography scheme

According to [7] there are six categories of steganography namely: substitution system techniques, transform domain techniques, spread spectrum techniques, statistical method techniques, distortion techniques and cover generation techniques. The substitution system steganography replaces redundant or unneeded bits of a cover with the bits from the secret message using, for instance, the Least-Significant Bit (LSB) method of encoding the secret message. However, the LSB Steganography is easily perceptible by steganalysis techniques [10]. In order to avoid the steganography detection facility through the use of the LSB method, in [4] it was proposed a new strategy of substituting the bits. With the purpose of optimizing the performance of this proposed scheme, a Genetic Algorithm (GA) was used.

In this paper, it is proposed a solution based on the Particle Swarm Optimization (PSO) algorithm for optimization image steganography using a strategy of substituting the bits as proposed by [4]. Furthermore, the proposed PSO algorithm exhibits good properties and promising results when compared with those found by GA.

The organization of this paper is as follows. In the next section, the GA approach is presented from the literature, followed by the new proposed approach using PSO algorithm for optimization image steganography. Then, experimental results are presented in Section 3. This paper shows some possible future research directions and concludes in Section 4.

2. SOLUTION APPROACH

2.1 Genetic Algorithm

Genetic Algorithms (GA) are biologically inspired optimization heuristics, invented by John Holland in the 1960s [9]. GA is a search and optimization method that simulates the natural processes of evolution that consist basically of the

processes of crossover, mutation, phenotype mapping and fitness computation [8].

The basic idea of GA is shown in Algorithm 1 and works as follows. An initial population is generated. After this, the genetic algorithm loop begins with the crossover step, where two individuals are selected to be the parents of a new individual, this step is repeated until there are no more parents. Soon after, the mutation occurs, a low mutation rate is required in this step to not compromise the generation. And lastly the process of natural selection. To complete the loop, a stopping criterion is evaluated.

Algorithm 1 Basic Genetic Algorithm[8]

```

1: initialize population
2: repeat
3:   repeat
4:     crossover
5:     mutation
6:     phenotype mapping
7:     fitness computation
8:   until population complete
9: until termination condition

```

2.2 Particle Swarm Optimization Algorithm

Originally proposed in [6], the Particle Swarm Optimization is inspired on social behaviour and interaction among the members of a swarm (flock of birds, colony of bees, for example). The group behaviour is influenced by the combined experience of a single particle as well as the whole group. According to [11], the main inspiration of PSO comes from the simulation and analysis of social dynamics and the interactions among the members of organized colonies, therefore, it is categorized as a swarm intelligence algorithm.

Moreover, PSO is a versatile population-based optimization technique, similar to evolutionary algorithms, but with no crossover nor mutation operators. Basically, particles “fly” above the fitness landscape, while a particles movement is influenced by its attraction to its neighborhood best (the best solution found by members of the particles social network), and its personal best (the best solution the particle has found so far) [1].

In PSO, the population is formed by individuals called particles. For each particle, there are two main properties: the particle dynamics and the particle information network. In such algorithm, particles move over the search space using the following equations:

$$v(t+1) = v(t) + a(t+1) \quad (1)$$

$$x(t+1) = x(t) + v(t+1) \quad (2)$$

where a , v , x and t are acceleration, velocity, position and time, respectively.

Each particle’s acceleration parameter is updated following two attraction forces. The first one is the local best (indicated by $pbest$), which is a memory of the best positioning reached by the particle over time. Another factor composing the attraction force is the global best (or the neighborhood best, indicated by $gbest$). In this case, some network topologies have been tried in some research in order to find out the neighborhood impact over a particle in searching for solutions. According to [1], the fully connected network is a popular choice for unimodal problems and therefore there is a single $gbest$ representing the best location found by the whole swarm.

There is a class of PSO called Standard PSO (SPSO for short). There are three main versions of such standard which are the SPSO 2006 [2], the SPSO 2007 [3] and SPSO2011 [13]. Regardless the changed aspects, they follow a canonical basis, which are depicted in Algorithm 2. About the velocity update, the SPSO2011 uses the following equation to update the velocity:

$$v_i(t+1) = wv_i(t) + x'_i(t) - x_i(t) \quad (3)$$

where x'_i is a random point (not necessarily uniformly picked) that lies in the hypersphere $H_i(G_i, ||G_i - x_i||)$ for which $G_i = x_i + c \frac{p_i + l_i - 2x_i}{3}$

Algorithm 2 Canonical PSO [1]

```

1: for each particle  $i$  do
2:   Random choose  $v_i, x_i = p_i$ 
3:   Evaluate  $f(p_i)$ 
4:    $g = \arg \max(f(p_i))$ 
5: end for
6: repeat
7:   for each particle  $i$  do
8:     Update  $v_i, x_i$ 
9:     Evaluate  $f(x_i)$ 
10:    if  $f(x_i) > f(p_i)$  then  $\triangleright$  Updating the local best
11:       $p_i = x_i$ 
12:    end if
13:    if  $f(x_i) > f(p_g)$  then  $\triangleright$  Updating the global best
14:       $p_g = \arg \max f(p_i)$ 
15:    end if
16:  end for
17: until Termination criteria reached

```

2.3 Proposed Solution

The optimization principle used in this paper follows the same scheme presented in [4]. In their approach, each target byte should pass through a verification phase. In order to illustrate how this works, let’s take a byte as example, as shown in figure 2. In the figure, the first byte (61) is the original pixel value found in the target image. For this example, the message will be placed into the third and fourth position of the target pixel. By doing the substitution, the pixel value becomes 49, which gives a luminance difference of 12. After optimizing that pixel value, it could be seen that the value 64 is closer to the original value, while the message chunk to be hidden remains embedded.

The binary representation of the pixel gives a natural choice for approaches such as genetic algorithms. In this paper however it is shown how the particles of a PSO solution can be adapted in order to optimize the stego-image produced by the optimization scheme presented earlier. Looking at the acceleration and velocity update formulas employed by PSO algorithm, it is clear that the algorithm is naturally suited for continuous domain problems. When dealing with discrete problems though, the PSO literature considers the velocity as a vector of probabilities of changing a bit value. In this work, it is proposed the following methodology: use the velocity vector as continuous values, but after a position update of each particle, the hidden message values are restored to the result position. In this way, the experiment pretend to show whether the velocity update formulas can adapt to this update operator, and still can optimize the resulting image.

	Decimal representation	Binary representation
Original Pixel	6 1	0 0 1 1 1 1 0 1
Pixel after substitution	4 9	0 0 1 1 0 0 0 1
Optimized Pixel	6 4	0 1 0 0 0 0 0 0

Binary chunk of the message to be hidden

Fig. 2. Adopted optimization model. There is a chunk of data to be hidden into one pixel of the target image. After the optimization process, the target pixel value is closer to the original one, but carrying the hidden message.

The fitness function is a quadratic function that accounts the difference between the original pixels and the optimized ones. This function can be seen in equation 4.

$$\sum_i^N (pixel_i - optimizedPixel_i)^2 \quad (4)$$

As the optimal solution is the closest of the original set of pixels, the population of particles is generated taking random points around the original image. After that, the initial population is then updated to keep the bits of the hidden message and its fitness is calculated to proceed with the remainder of the PSO algorithm. The whole system is depicted in figure 3.

3. COMPUTATIONAL EXPERIMENT

In order to evaluate the proposed solution, the figure 6, a famous input image used along with image processing classes has been chosen along with a random text message to be hidden in the target picture. We ran the GA algorithm using the set of parameters defined in figure 4 as well as the PSO with configurations displayed on figure 5.

PopSize	Generations	CrossOver	Mutation
300	40	0.7	0.5

Fig. 4. GA parameters.

PopSize	Generations	Velocity	Inertia
20	4		default

Fig. 5. PSO parameters.

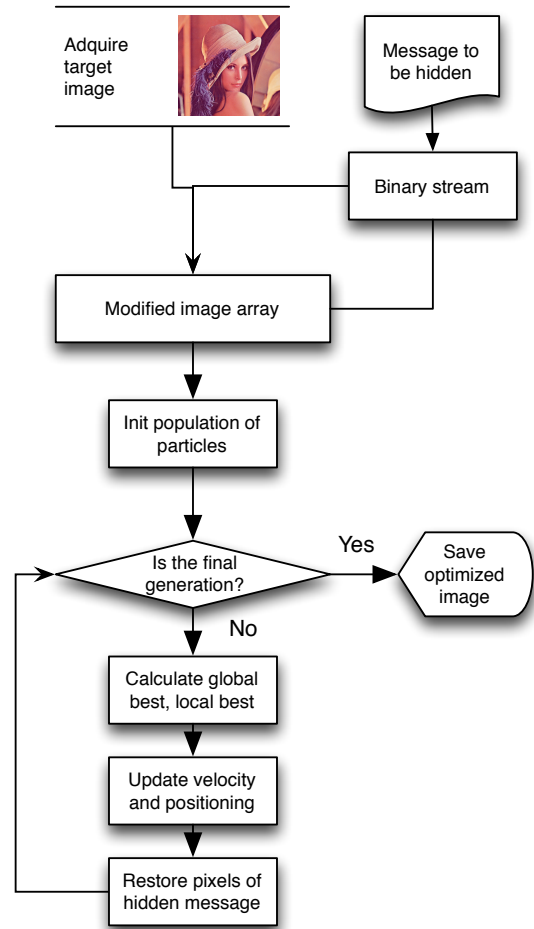


Fig. 3. Proposed system.

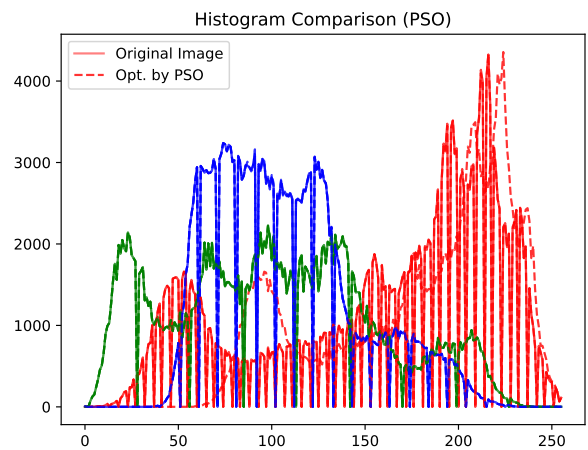


Fig. 7. Histogram comparison with the original image and the optimized image by PSO.



Fig. 6. A classic image used in image processing classes and used in this paper as input image.

It is important to highlight that as a comparison of the parameters of GA and the PSO, there are only a few generations (4 in PSO against 40 of GA). Another important fact is the number of individuals in the population. The PSO algorithm reached better results even with lesser computational resources. Neither GA parameters nor the PSO's have been optimized. Such optimization could produce better results for both GA and PSO.

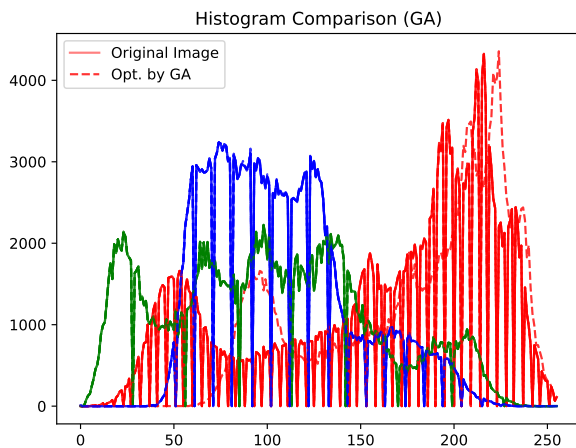


Fig. 8. Histogram comparison with the original image and the optimized image by GA.

In order to show the differences in the pixel density after the proposed optimization, figures 7 and 8 show the histogram of the original image and the optimized one, both for PSO and

GA solutions, respectively. As it could be seen in the images, the histogram of the optimized image is almost the same of the original one. Figure 9 summarizes the objective function values. The leftmost bar is the objective function value evaluated between the original image and the image after replacing the bits of the hidden message. As can be seen in the figure, the PSO reached the best results, when compared with the other approaches.

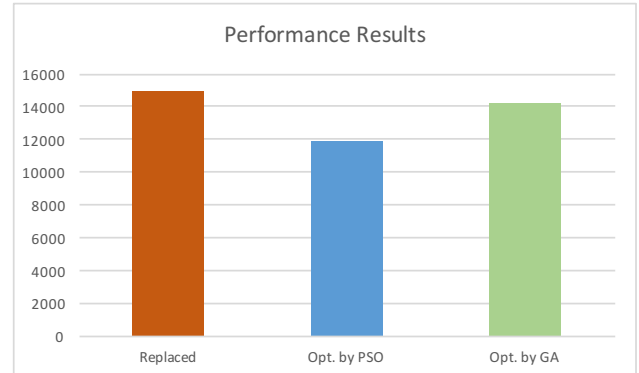


Fig. 9. Performance results according to equation 4.

4. CONCLUSIONS

This work presented an evaluation of the PSO2011 algorithm applied to the problem of optimizing a stego-image in Steganography application. As the LSB method is easy to be detected by steganalysis techniques, this work has shown that positioning the pixels in other positions is possible by the application of optimization techniques.

Furthermore, as could be seen by the experiment shown in the previous section, the PSO showed promising results being capable of producing better results than classical models such as genetic algorithms, when using the adopted optimization scheme. Moreover, the default position update function employed by the PSO2011 algorithm was capable of self adapting to the proposed changes, i.e., even after repositioning the message bits in the particle after position update.

The next steps for further improvements consist of finding optimal substitution matrices in order to approximate the hidden message with the target image. This could be combined with the proposed solution presented by this work, in order to approximate the original image, as a second stage of optimization.

5. REFERENCES

- [1] C. Blum and D. Merkle. *Swarm Intelligence: Introduction and Applications*. Natural Computing Series. Springer Berlin Heidelberg, 2008.
- [2] Maurice Clerc. *What is a Difficult Problem?* ISTE, 2010.
- [3] Maurice Clerc. *Standard particle swarm optimisation*, 2012. 15 pages.
- [4] Gangeshwar and James Attri. Optimizing image steganography using genetic algorithm. *International Journal of Engineering Trends and Technology (IJETT)*, 24(1):32–38, 2015.
- [5] S. Katzenbeisser and F.A.P. Petitcolas. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House computer security series. Artech House, 2000.

- [6] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995.
- [7] G. Kipper. *Investigator's Guide to Steganography*. CRC Press, 2003.
- [8] Oliver Kramer. *Genetic Algorithm Essentials*, volume 679 of *Studies in Computational Intelligence*. Springer, 2017.
- [9] Melanie Mitchell. *An Introduction to Genetic Algorithms*. Complex Adaptive Systems. The MIT Press, first printing, edition, 1996.
- [10] Sachin Mungmode, R.R. Sedamkar, and Niranjan Kulkarni. A modified high frequency adaptive security approach using steganography for region selection based on threshold value. *Procedia Computer Science*, 79:912 – 921, 2016.
- [11] Y.G. Petalas, K.E. Parsopoulos, and M.N. Vrahatis. Memetic particle swarm optimization. *Annals of Operations Research*, 156(1):99–127, 2007.
- [12] Frank Y. Shih. *Digital Watermarking and Steganography: Fundamentals and Techniques*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2007.
- [13] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas. Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 2337–2344, June 2013.