EE553 OPTIMIZATION TERM PROJECT
A Comparison of Particle Swarm Optimization, Genetic Algorithm and Hill-Climbing Algorithm on Optimization of Substitution Permutations for a Novel Image Steganographic Method based on Integer Wavelet Transformation

Okyanus Oral 2305134

1 Introduction

During a secret communication, in order to prevent unwanted access to a secret message, image steganography can be used to hide secret data into a cover image. The aim of steganography is to hide the message so that the embedded data is undetected until retrieval by the desired group. Since communication is secret, stego image is not expected to experience any harsh attacks [1]. Although there are multiple criteria on the evaluation of image steganography methods, due the its aim of secrecy, perceptual transparency is the crucial metric.

A novel IWT (integer wavelet transform) based image steganography presented in [2] uses PSO (particle swarm optimization) to optimize its parameters such that the obtained stego image has low distortion. In this project, a comparison of PSO, GA (genetic algorithm) and HCA (hill-climbing algorithm) is given in terms of their optimization performances on the parameters of the IWT based image steganography.

2 Image Steganography

The science of hiding secret information such that only the intended receiver can detect and extract the the secret message is steganography. In steganography, secret information is embedded into a cover object such as audio, video, text and image. Images as cover objects are considered as good carriers due to their high degree of pixel value redundancy and the imperfect visual perception of humans [2].

Image steganographic techniques are categorized into two; spatial domain and transform domain image steganography. In spatial domain steganography, the secret message is embedded directly into pixel values. Several methods such as k least-significant bit substitution or quantization index modulation can be used for the embedding process. In transform domain steganography, the cover object is first transformed into another domain and then the data is embedded later. For the transformation of the cover image, DFT (Discrete Fourier Transform) DCT (Discrete Cosine Transform), DWT (Discrete Wavelet Transform), IWT (Integer Wavelet Transform) can be used. After the transformation, the secret data can be embedded into the coefficients of the cover/host image. Transform domain methods allow the embedding of data to different spectral channels of the image. Therefore, transform domain image steganography generally provides more robustness to attacks (such as low pass filtering) and higher perceptual visual quality [2]. The image after the embedding, which contains the secret message, is called stego image.

2.1 IWT Based Image Steganography

The following subsections describe the IWT Based Image Steganography presented in [2].

2.1.1 Integer Haar Wavelet Transform

A major disadvantage of some transform domain image steganography techniques is that the embedded information can be lost during the inverse transform due to finite precission or rounding errors. An elevation of this problem is the use of integer based methods.

The embedding scheme in [2] uses Integer Haar Wavelet transform domain, Eqn. 1, to embed the information. As the human visual perception of high-frequency content/details is flawed, the secret message is embedded to the high frequency content of the host image in IWT domain.

One Dimensional Integer Haar Wavelet Transform:

$$S_{1,n} = \lfloor \frac{(S_{0,2n} + S_{0,2n+1})}{2} \rfloor ; D_{1,n} = S_{0,2n+1} - S_{0,2n} \tag{1}$$

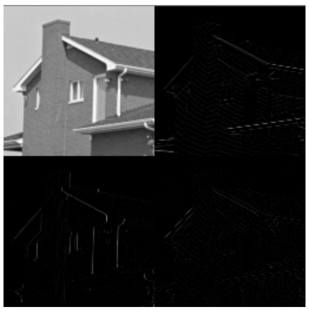
One Dimensional Inverse Integer Haar Wavelet Transform:

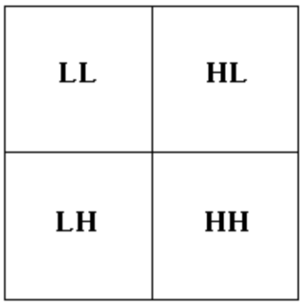
$$S_{0,2n} = S_{1,n} - \lfloor \frac{D_{1,n}}{2} \rfloor ; S_{0,2n+1} = \lfloor \frac{D_{1,n} + 1}{2} \rfloor \tag{2}$$

On the equations above $S_{0,2n}$ and $S_{0,2n+1}$ are the consecutive pixel values on the cover image.

A picture of a house and its Integer Haar Wavelet Transform is given in Fig. 1. The high frequency content of the house image corresponds to first, third and fourth quadrants in Fig. 9c, that are HL, LH and HH in Fig. 2d.

(a) House


(b) IWT(House)



(c) IWT Level 1, Frequency Content

Figure 1: Picture of a House and its Integer Haar Wavelet Transform (depth 1)

2.1.2 Embedding

Embedding of the secret message for any given image starts from the first column of LH and proceeds column by column excluding entries in LL. Sequentially, the k least significant bits of the host image's IWT coefficients are overwritten with OPAP (Optimal Pixel Adjustment Procedure) from the bit-stream of the secret message.

For k least significant bit substitution, the utilized OPAP method minimizes the error on the stego and host image IWT coefficients. Therefore, it can be considered as a one dimensional search algorithm during optimization of best visual quality stego image. The method minimizes the error by changing the unsubstituted bits of the host coefficient to match with the host value.

For example, given a host coefficient, $c = (164)_{10} = (10100100)_2$ and a substitution message of 2 bits with value $(3)_{10} = (11)_2$, if the 2 least significant bits of the host coefficient c is directly substituted without any other adjustment the error, d, would be $d = c - c' = (10100100)_2 - (10100111)_2 = (164)_{10} - (167)_{10} = -3$. However, if the first 6 most significant bits of c' are adjusted such that $c' = (163)_{10} = (10100011)_2$ then the error decreases to $d = c - c' = (164)_{10} - (163)_{10} = 1$ [3]. For substitution of k least significant bits of B bit binary numbers, the procedure of adjustment is as follows,

$$c' = \begin{cases} c' + 2^k, & d > 2^{k-1} \quad \text{and} \quad 0 \leq c' + 2^k \leq 2^B - 1 \\ c' - 2^k & d < -2^{k-1} \quad \text{and} \quad 0 \leq c' - 2^k \leq 2^B - 1 \\ c' & \text{otherwise} \end{cases} \tag{3}$$

Furthermore, before the embedding, the encryption-permutation, which is needed to decode the secret information, is appended to the start of the secret message. Also, k, bits per pixel information, is embedded to the last HH coefficient of the stego image.

Embedding capacity of the system is directly proportional to k, bits per pixel, which is set to take values from 1 to 4.

2.1.3 Encryption

For any given message, its content is first encrypted with a substitution-matrix/substitution-permutation. A substitution-permutation describes a cipher, it is a mapping between the actual values of message to its encrypted ones. Therefore, one can encrypt/decrypt the message with the knowledge of the substitution-permutation.

For k bits per pixel embedding, a substitution-permutation defines a one-to-one mapping for each of 2^k numbers to the same range. A sample substitution-permutation, M, for $k = 3$ is given below.

$$M = \begin{bmatrix} {}_0 3 & {}_1 2 & {}_2 5 & {}_3 7 & {}_4 4 & {}_5 6 & {}_6 1 & {}_7 0 \end{bmatrix}$$

The preceding subscripts are the corresponding indices to be substituted from.

Let the message bit stream be,

$$m_B = 011111010100101$$

to encrypt the data, first split the message bit stream left to right with packages of $k = 3$ bits,

$$m_{B,k=3} = 011 - 111 - 010 - 100 - 101$$

.

then convert each package to its corresponding decimal value,

$$m_{B,k=3} \rightarrow m_{D,k=3} = 3 - 7 - 2 - 4 - 5$$

for each decimal, d, in $m_{D,k=3}$, replace it with the corresponding decimal in substitution-permutation on the index of d. The encoded secret decimal message, $s_{D,k=3}$ becomes,

$$m_{D,k=3} \rightarrow s_{D,k=3} = 7 - 0 - 5 - 4 - 1$$

At the last step convert the secret decimal message to secret bit-stream, s_B, which is embedded to the host image.

$$s_{D,k=3} \rightarrow m_{B,k=3} = 111 - 000 - 101 - 100 - 001 \rightarrow s_B = 111000101100001$$

2.1.4 Visual Quality

The visual quality of the stego images depends on the number of embedded bits-per-pixel, total number of occupied pixels and substitution-permutation. The first two variables are fixed for a given setting. Therefore, only tunable parameter is the substitution-permutations. To obtain the best visual quality, the the best substitution-permutation should be selected for a given setting (host image, message, bits-per-pixel). This is done with the PSO algorithm.

PSNR (Peak Signal to Noise Ratio) is selected as the metric for visual quality.

$$PSNR = 10log_{10}(\frac{P_{max}^2}{MSE}) \tag{4}$$

In the Eqn. 4, P_{max}^2 stands for the square of maximum pixel value in the host image and MSE stands for mean squared error between stego and host images.

3 Particle Swarm Optimization

PSO is nature a inspired population based meta-heuristic algorithm which can be utilized to optimize continuous multimodal (functions that have multiple local minima or maxima) functions. The algorithm only requires function evaluations and does not require gradient information.

The algorithm mimics the movement of flock of birds to find the optimal solution of the given function. It focuses on the collective search for food of flocking animals to profit from the experience of all other members [4].

The algorithm initializes predefined number of particles as solution candidates in the search space. At each iteration of the algorithm, the particles' locations, $x[t]$ are updated. For the i^{th} particle, the updates on the particle's location $x_i[t]$ and velocity $v_i[t]$ are based on the particle's personal best location $x_i^*[t]$, the explored global best location $x_:^*[t]$ and the speed $v_i[t]$ of the particle at iteration t, see Eqn. 5.

$$x_i[t+1] = x_i[t] + v_i[t] \tag{5}$$
$$v_i[t+1] = \chi \cdot (v_i[t] + c_1 \cdot r_1 \cdot (x_i^*[t] - x_i[t]) + c_2 \cdot r_2 \cdot (x_:^*[t] - x_i[t]))$$

In Eqn. 5 c_1 and c_2 are personal and global best acceleration coefficients, they define the magnitude of displacement along the direction of personal best location and global best location. The parameter $\psi = \frac{2\omega}{|2-\phi-\sqrt{\phi(\phi-4)}|}$ is the constriction coefficient, where $\phi = c_1 \cdot r_1 + c_2 \cdot r_2$. The magnitude of the constriction coefficient is proportional to the inertia, $\omega \in [0,1]$, which is the proportional amount of velocity passed to the next iteration. r_1 and r_2 are Uniform random variables, making the algorithm stochastic. It is the algorithm's stochastic nature that allows it to be utilized on multimodal optimization. Overview of the algorithm is given in Alg. 1.

Algorithm 1: Overview of Particle Swarm Optimiztaion

```
1  $X \leftarrow$ Instantiate an initial swarm;
2  $V \leftarrow$ Instantiate initial velocities to 0;
3  $F \leftarrow$ Compute fitness for each individual using $X$;
4  $X^* \leftarrow X$ Set the personal bests to the initial values $(X, F)$
5  $(x^*_:, f^*_:) \leftarrow$Set the global best $max((F^*, X^*)\{i\})$;
6  while Termination criteria are not satisfied do
7      $(X, V) \leftarrow$ Update the particles using the Eqn. 5;
8      $(X^*, F^*)\{i\} \leftarrow$ Update the personal bests of each particle, $i$ if $(x_i, f_i) > (X^*, F^*)\{i\}$;
9      $(x^*_:, f^*_:) \leftarrow max((f^*_:, x^*_:), (F^*, X^*)\{i\})$;
10 end
11 return $x^*_:$
```

Termination criteria is set as number of iterations.

However, on the image steganography problem, the search space is discrete in nature for the optimization of permutations. Therefore PSO algorithm's location update must contain a fixing step to return to feasible solutions. To do so, particles' locations, substitution-permutations, are corrected by assigning the ascending order of their entries. For example for embedding with 2 bits-per-pixel, let a resulting location after position update step be $x_i[t+1] = [-0.6, 5, 20]$, then the corrected feasible location becomes $x_i[t+1] = [0, 3, 2, 1]$.

3.1 Remarks

The steganography results presented in [2] were compared with the steganography methods that do not use an optimization scheme. Therefore, although the stego image visual quality outperforms other methods, effectiveness of using PSO to find substitution-permutations are in question [2].

Obtaining the best visual quality stego image with optimum substitution-permutations is a mixed integer non-linear programming problem (MINLP). The PSO algorithm used is not well suited for MINLP and requires a correcting scheme as described on the previous section. Therefore other optimization algorithms that are suited for MINLP can outperform PSO in optimization of substitutions.

In this project GA and HCA are implemented to optimize permutations. Parameters of PSO, GA and HCA are than optimized for a test image and a test message, via a meta-optimizer. Meta optimizer is selected as Bayesian-Optimizer to account for the stochastic nature of GA, PSO and HCA. Then the results of GA, PSO and HCA with their optimum parameters for the given task is compared.

4 Genetic Algorithm

Genetic algorithm (GA) is one of the first population based stochastic optimization algorithms. It is a heuristic method and can be used to solve multimodal, MINLPs. It also does not require gradient information of the fitness-function (objective function) [5].

Evolution of species under selective pressure and survival of the fittest is the motivating idea behind evolutionary/genetic algorithms. The candidate solutions to a given problem is tried to evolve to a better, optimum, solution throughout generations.

The algorithm, starts by initiating a candidate solution population. At each iteration of the algorithm, the fitness of individuals from the population is evaluated and recorded. The individuals with appealing fitness are then used to create offsprings and/or transferred to the next generation.

Algorithm 2: Overview of Genetic Algorithm

```
1  $P \leftarrow$ Instantiate initial population;
2  $F \leftarrow$ Compute fitness for each individual using $P$;
3  while Termination criteria are not satisfied do
4      $\hat{P} \leftarrow$ Select individuals from $P$, using $F$ and apply crossover;
5      $P \leftarrow$ Apply mutation to individuals of $\hat{P}$;
6      $F \leftarrow$ Compute fitness for each individual using $P$;
7  end
8  $p \leftarrow$ The individual from $P$ with maximum fitness;
9  return $p$
```

An overview of GA is given in Alg. 2.

Selection of parents can be performed by different methods such as roulette wheel sampling, rank based roulette wheel selection, stochastic universal sampling. Considering that the PSNR values with each substitution-permutation does not have high variance, in order to preserve selective pressure, ranked based roulette wheel is selected as the operator. In rank based roulette wheel selection, every individual's probability of being selected is inversely proportional to their rank. That is, for a population with N members, the m^{th} ranking individual has a selection probability of $\frac{m}{\sum_{i=1}^{N} i}$.

In order to create offsprings, crossover is applied. The ratio of population that is subjected to crossover is set by the user before training. For a pair of parents, the crossover operation creates a new candidate solution. The operation is dependent on the application and has a wide variety [6]. For example, if the candidate solutions are in a continuous space, then crossover operator could be defined as convex combination of the two parents; or if the solutions are discrete in nature, single point crossover can be defined where the gene pairs are exchanged to create offsprings.

Convex Combination	Single Point Crossover	Position Preserving Permutative Crossover
parent$_1$ = $[+1.0, -2.0, +3.0, +1.50]$	parent$_1$ $= [A, B, 3, 1, 4]$	parent$_1$ $= [1, 3, 5, 2, 4] \rightarrow [1, -, -, 2, -]$
parent$_2$ = $[+0.0, +5.0, +2.0, -2.30]$	parent$_2$ $= [A, C, 5, 4, 3]$	parent$_2$ $= [2, 5, 1, 3, 4]$ passes: {5,3,4}
offspring= $[+0.2, +3.6, +2.2, -1.54]$	crossover point $\uparrow$	offspring $= [1, 5, 3, 2, 4]$
	offspring$_1$ $= [A, B, 3, 4, 3]$	
	offspring$_2$ $= [A, C, 5, 1, 4]$	

Furthermore, for the problem of finding substitution-permutation, the crossover operation need to be permutative. There are many permuative crossover operations, such as cyclic crossover, position preserving crossover, order preserving crossover [6]. For this problem, position preserving crossover operator is selected since neither the order nor the cyclic permuatitions are of interest. The only consideration is partially preserving encoding structure during crossovers. The preserved genes from the first parent is selected randomly using the inheritance probability parameter set by the user. Given that inheritance probability is p, any gene from the parent preserves its position on the offspring with probability p.

After applying crossover operations to selected individuals, elites are selected. The ratio of elites is again set by the user before training. Elites are the solution candidates that have the highest fitness. It is important to preserve elites in order not to forget the best solutions and to exploit the vicinity of the best solutions.

The remainder number of individuals for the next population is again selected with rank based roulette wheel selection.

Also in order to continue exploring the search space, mutation operation is applied to non-elite individuals. Similar to crossover operation, mutation operator is problem dependent. The mutated individuals still need to be feasible solutions to the problem. Although there are many other mutation methods, gene shuffling is selected since it preserves permutations. The amount of shuffling is determined by the user with probability of mutation. Given that probability of mutation is p, a gene is selected to be shuffled with probability p.

Termination criteria is set as number of iterations.

5 Hill Climbing Algorithm

Hill Climbing Algorithm (HCA) is a meta-heuristic local search method which, similar to GA, does not require computation of objective function gradients [7].

The algorithm starts by initiation of a random feasible solution point, it then stochastically searches for the better solution near the vicinity of the current one. It accepts the new location if the new location has a higher fitness than the current position [7]. The overview of the algorithm is given in Alg. 3

Algorithm 3: Overview of Hill Climbing Algorithm

```
1  $x \leftarrow$ Create a random solution;
2  $f \leftarrow$ Compute fitness of $x$;
3  while Termination criteria are not satisfied do
4      $\hat{x} \leftarrow$ Mutate the solution $x$;
5      $\hat{f} \leftarrow$ Compute fitness of $\hat{x}$;
6      if $\hat{f} > f$ then
7          $x \leftarrow \hat{x}$;
8          $f \leftarrow \hat{f}$;
9      end
10 end
11 return $x$
```

The same mutation operator in GA is used for creating candidate solutions around the current solution. The mutation probability is set by the user before training.

The termination criterion is the number of iterations.

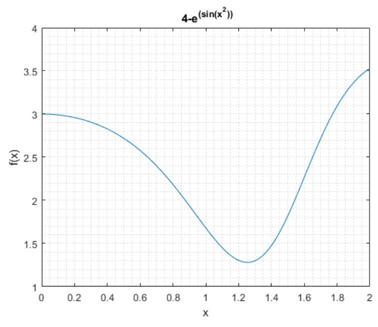
6 Bayesian Optimization

Bayesian optimization is, in general, utilized for optimization of objective functions that take long time to evaluate. It is well suited with optimization problems that have stochastic evaluations. The optimizer can tolerate stochastic noise since parameters are optimized by utilizing the estimated posterior and prior probability densities in Bayesian Learning [8].

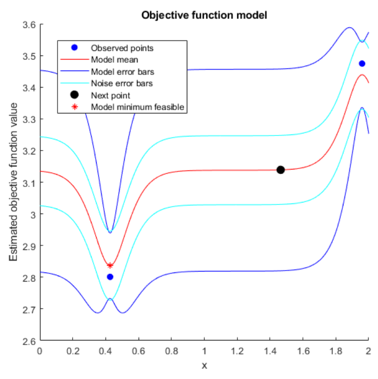
At each iteration, the algorithm quantifies the uncertainty of the objective values on the search space and decides where to sample next with an acquisition scheme. The posterior distribution of objective function are then updated with each observed data. To do so the optimizer uses Gaussian Processes and Gaussian Regression to model and fit to the objective function [8].

The algorithm only requires function evaluations and therefore it can be used on optimizing the parameters of other algorithms as a meta-optimizer.

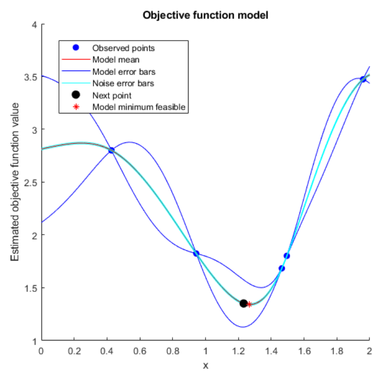
An example of sampling procedure, uncertainty estimates, the function approximations at different iterations and the ground truth function is given in Fig. 5, where the optimizer tries to find the minimum while building confidence on its function approximations.



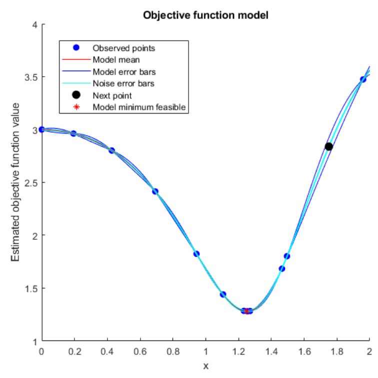
(a) Ground Truth Function



(b) Bayesian Optimizer - Iteration : 2



(c) Bayesian Optimizer - Iteration : 5



(d) Bayesian Optimizer - Iteration : 11

Figure 2: Bayesian Optimization Through Iterations

7 Testing Conditions

As it was mentioned in the previous sections, parameter search for GA, HCA and PSO are carried out with Bayesian Optimization. For each optimizer, Bayesian Meta-Optimizer had sampled the respective parameter space with 30 samples. The optimum parameter estimations are then outputted along with the corresponding maximum PSNR values, see Tab. 1.

The parameters of GA, HCA and PSO cannot take arbitrary values therefore feasible regions need to be considered for the meta-optimization process. Furthermore, for the unbounded parameters, search space is narrowed down to acceptable bounds before running the Bayesian Optimizer.

The feasible and the regions of interest for the optimizer parameters are set as follows:

PSO:

- Inertia Weight, $\omega \in [0, 1]$. Inertia weight should not exceed 1. If it exceeds 1 the particles would geometrically increase and preserve their velocity at each iteration. Which would not allow the algorithm to converge.

- Personal and Global Best Acceleration Factor, $c_1, c_2 \in [1, 5]$. If the acceleration factors are in $[0, 1]$ then the position updates would only be the convex combination of previous points. That would make the search collapse into the convex hull of the initial population. Therefore they should be greater than 1. Although these parameters do not have an upper bound, it is not desirable for particles to drastically skid away. Therefore, their upper bound is selected as 5 by experience and intuition from previous trials.

GA:

- Crossover Ratio of Population, Elitist Ratio of Population, Inheritance Probability, Mutation Probability $r_c, r_e, p_i, p_m \in [0, 1], 0 \leq r_c + r_e \leq 1$. All the probabilities and ratios should be between 0 and 1, also the sum of elitist ratio and crossover ratio of population should be at maximum 1.

HCA:

- Mutation Probability $p_m \in [0, 1]$, the probability should be between 0 and 1.

PSO and GA are run 50 iterations with swarm and populations sizes of 30, meanwhile HCA is run $50 \cdot 30 = 1500$ iterations. Therefore, for each trial, the number of function evaluations are constant for all optimizers and is fair. The number of iterations and swarm/population size is selected as the best value reported in [2]. Although, if the algorithms are run indefinitely, all of them will presumably converge to the optimum value. Therefore, limiting the run time also evaluates efficiency. Furthermore, for the majority of trials with PSO, the improvement reaches a plateau after the 50^{th} iteration, see Fig. 3. Hence, total iterations of 50 for GA, PSO and 1500 should provide meaningful comparisons for the algorithm performances.

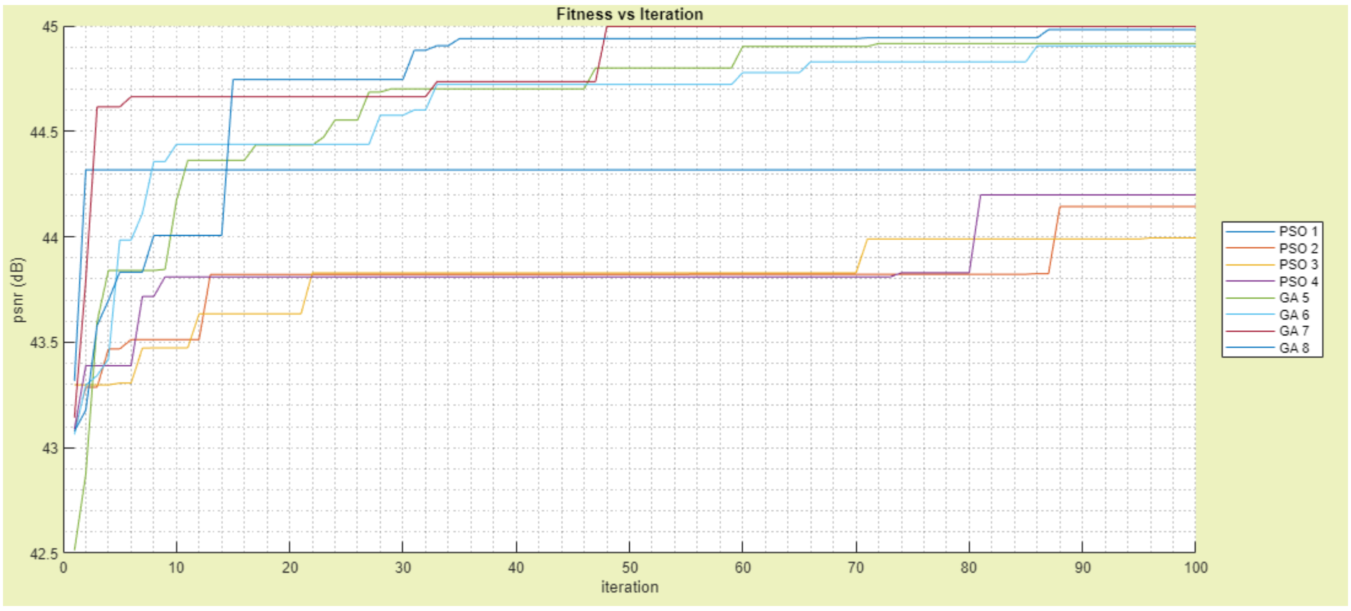


Figure 3: Stego Image PSNR versus Iterations for Trials with PSO ($\omega = 0.864, c_1 = 2.05, c_2 = 2.05$ [2]) and GA ($r_c = 0.7, r_e = 0.1, p_i = 0.5, p_m = 0.6$)

The visual quality of the stego image intrinsically depends on the host image and the embedded message. Therefore, optimization of substitution-permutations depends on the host image and the selected message. In order to asses the performance of the optimization algorithms, GA, PSO and HCA are run on the same host image with the same message.

The host image is selected as the cameraman image, see Fig. 4 and the test message is generated to occupy roughly half of the pixels of the host image. The test message can be found in Appendix.

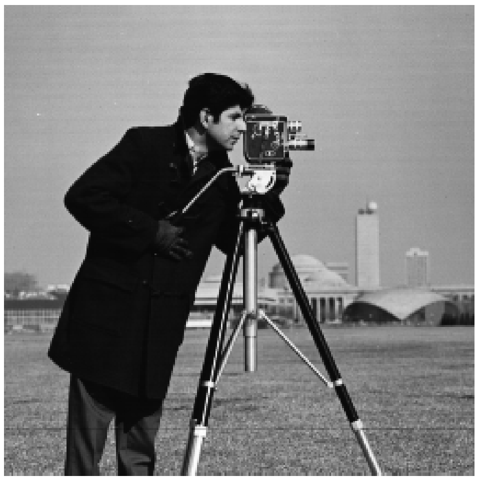


Figure 4: Cameraman Image - Host Image

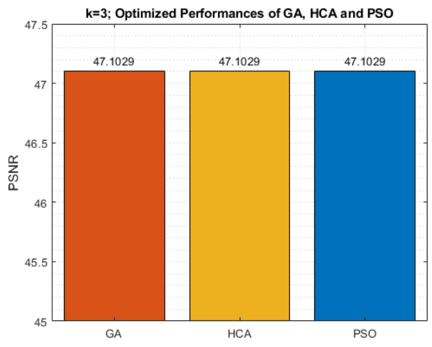
Tests are carried out for $k = 4$ and $k = 3$ bits-per-pixel embedding. The tests with $k = 1$ and $k = 2$ are excluded because the total number of possible permutation-substitutions are $2^1! = 2$ and $2^2! = 24$ respectively, which can easily be found via grid-search.

8 Results and Conclusions

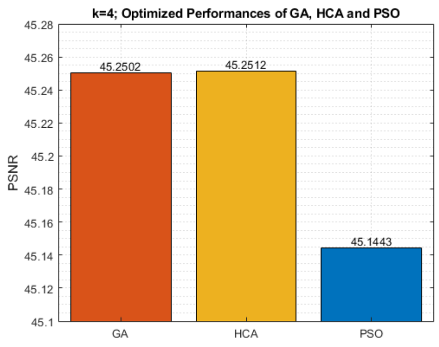
Optimum parameter estimates, using the Bayesian Meta-optimizer, of PSO, GA and HCA for $k = 3$ and $k = 4$ bits-per-pixel embedding are summarized in Tab. 1.

Table 1: Optimum Parameters and Performances of PSO, GA and HCA

k=3:					**PSNR**
PSO	$\omega = 0.31404$	$c_1 = 2.1193$	$c_2 = 4.7589$		**47.1029 dB**
GA	$r_c = 0.75911$	$r_e = 0.089913$	$p_i = 0.063172$	$p_m = 0.38941$	**47.1029 dB**
HCA	$p_m = 0.72032$				**47.1029 dB**
k=4:					**PSNR**
PSO	$\omega = 0.13436$	$c_1 = 4.7646$	$c_2 = 3.2043$		**45.1443 dB**
GA	$r_c = 0.79733$	$r_e = 0.041043$	$p_i = 0.61298$	$p_m = 0.26129$	**45.2502 dB**
HCA	$p_m = 0.18859$				**45.2512 dB**



(a) Bayesian Meta-Optimization Results, k=3



(b) Bayesian Meta-Optimization Results, k=4

Figure 5: Bayesian Meta-Optimization Results

It is quite evident that for $k = 3$ all of the optimizers with their optimum parameters were able to find the same solution. For $k = 3$ there are $2^3! = 8! = 40320$ possible substitution-permutations, meanwhile number of function evaluations is constant, 1500. That is, all of the optimizers are working efficient enough.

For $k = 4$ PSO is outperformed by GA and GA is outperformed by HCA. However, all of the optimal results are quite close to each other, Nevertheless, degreded performance of PSO compared to proposed GA was expected.

HCA outperforming PSO and GA is an important result for the problem at hand. It shows that improvements on solutions are more efficiently carried out randomly rather than combination of other solution candidates. That is, the fitness function is not necessarily locally correlated. This is further supported by the optimal parameters of the algorithms. For example for GA, optimal ratio of elites in the population, r_e, is nearly equal to 0 for $k = 3$ and for $k = 4$, which shows that the algorithm works better if exploration is higher than exploitation. Furthermore for $k = 3$ and for $k = 4$ acceleration coefficients, c_1, c_2 of PSO are not conservative, again exploration is favoured in the search space.

The optimal mutation ratio p_m of HCA with $k = 3, p_m \approx 0.72$ and with $k = 4, p_m \approx 0.18$ are considerably different. This can be explained with the number of occupied pixels. For the message with fixed length, number of pixels occupied when $k = 3$ is $\frac{4}{3}$ times as large when $k = 4$. Increased number of pixel occupancy decreases the dependency of each pixel with a favourable substitution value. To better explain the importance of number of occupied pixels an extreme scenario can be given as an example. If the image has infinite number of occupied pixels where the image has irregular patterns, the choice of substitution-permutation would not have any significance and all solutions would yield equal visual quality. Therefore, when $k = 3$, due to increased number of pixel occupancy, the uncorrelatedness of candidate solutions increases and therefore a higher mutation ratio is favoured.

9 MATLAB Application

General overview of the proper execution of the application consists of 6 steps:

1. Select a host image using Image Panel,
2. Specify the number of bits to embed, k, via Message Options Panel,
3. Write or select a message to embed via Message Options Panel,
4. From Optimizer Options Panel Select the desired Optimizer and set its parameters accordingly,
5. Specify the number of iteration and start the optimization from Optimization Panel,
6. Using the show results button on Image Panel, display the stego, error image and extracted/recovered message.

In order to adhere to the execution steps described above, user interface of the application is explained in detail. Application starts with the main screen shown in Fig. 6.

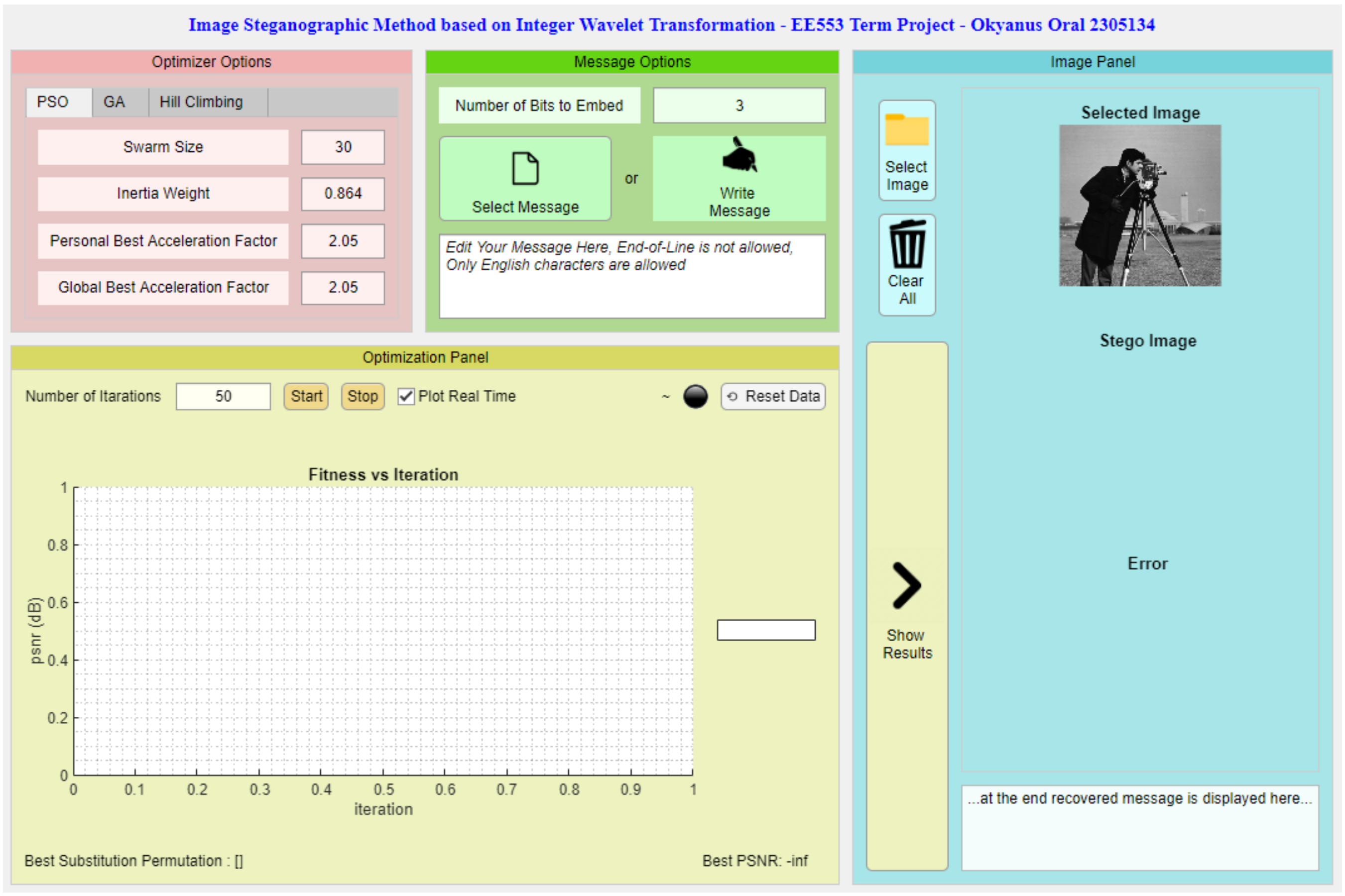


Image Steganographic Method based on Integer Wavelet Transformation - EE553 Term Project - Okyanus Oral 2305134

Swarm Size 30
Inertia Weight 0.864
Personal Best Acceleration Factor 2.05
Global Best Acceleration Factor 2.05

Number of Bits to Embed 3

Edit Your Message Here, End-of-Line is not allowed,
Only English characters are allowed

Number of Itarations 50
Plot Real Time ✓

Best Substitution Permutation : []
Best PSNR: -inf

...at the end recovered message is displayed here...

Figure 6: Starting Main Screen

Main screen is composed of 4 panels: Image Panel, Message Options Panel, Optimizer Options Panel and Optimization Panel. Apart from their basic functionality mentioned in the general overview, panels also have other extra features.

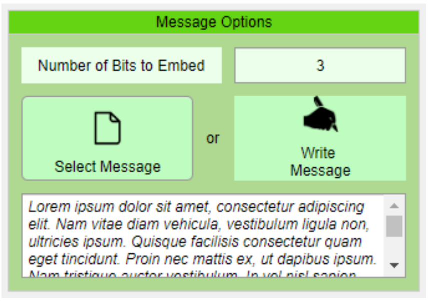
Image Panel is shown in Fig. 8. It is the panel where images are displayed.

From the select Image Button on the Image Panel a host image can be selected for further processing.

After optimization is completed, results can also displayed on the Image Panel by pressing the show results button. Stego Image and the Error Image (Host-Stego Image) is displayed to further demonstrate the embedding process. Furthermore, extracted/recovered message is displayed at its bottom.

One can also reset all the data held in the application using the clear all button.

Message Options Panel is shown in Fig 7. In this panel the message to be embedded is inputted. From the Select Message Button a text message can be selected from a *.txt* file or a new message can be written at the edit-field. Also, number of bits to be embeded, bits-per-pixel value k can be be inputted here.



Message Options

Number of Bits to Embed: 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam vitae diam vehicula, vestibulum ligula non, ultricies ipsum. Quisque facilisis consectetur quam eget tincidunt. Proin nec mattis ex, ut dapibus ipsum.

Figure 7: Message Options Panel

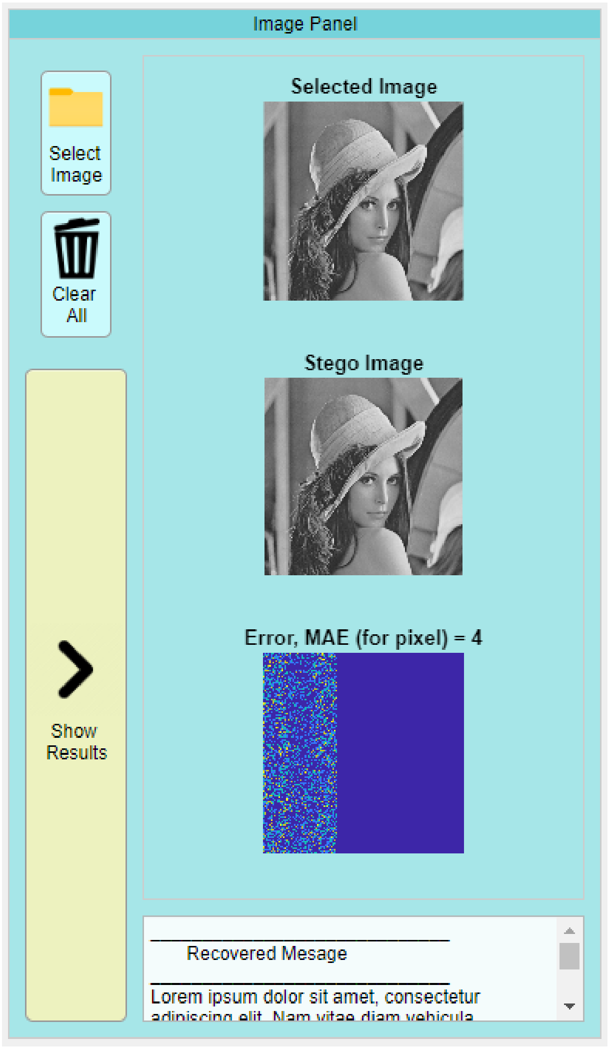


Image Panel

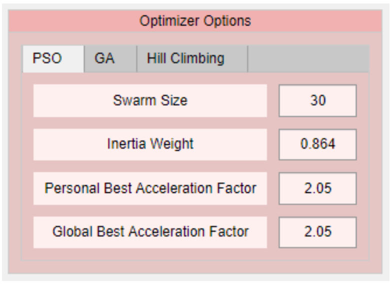
Error, MAE (for pixel) = 4

Recovered Mesage

Lorem ipsum dolor sit amet, consectetur

Figure 8: Image Panel

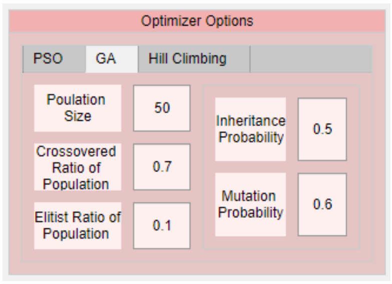
Optimizer Options Panel shown in Fig. 9. Optimizers can be selected from PSA, GA and Hill Climbing Tabs. Their parameters are then can be inputted to respective edit-fields.



Optimizer Options

Swarm Size	30
Inertia Weight	0.864
Personal Best Acceleration Factor	2.05
Global Best Acceleration Factor	2.05

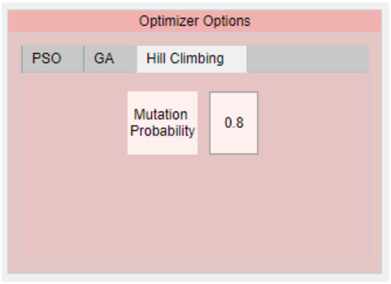
(a) PSO Options



Optimizer Options

Poulation Size	50
Crossovered Ratio of Population	0.7
Elitist Ratio of Population	0.1
Inheritance Probability	0.5
Mutation Probability	0.6

(b) GA Options



Optimizer Options

Mutation Probability	0.8

(c) HCA Options

Figure 9: Optimizer Options Panel

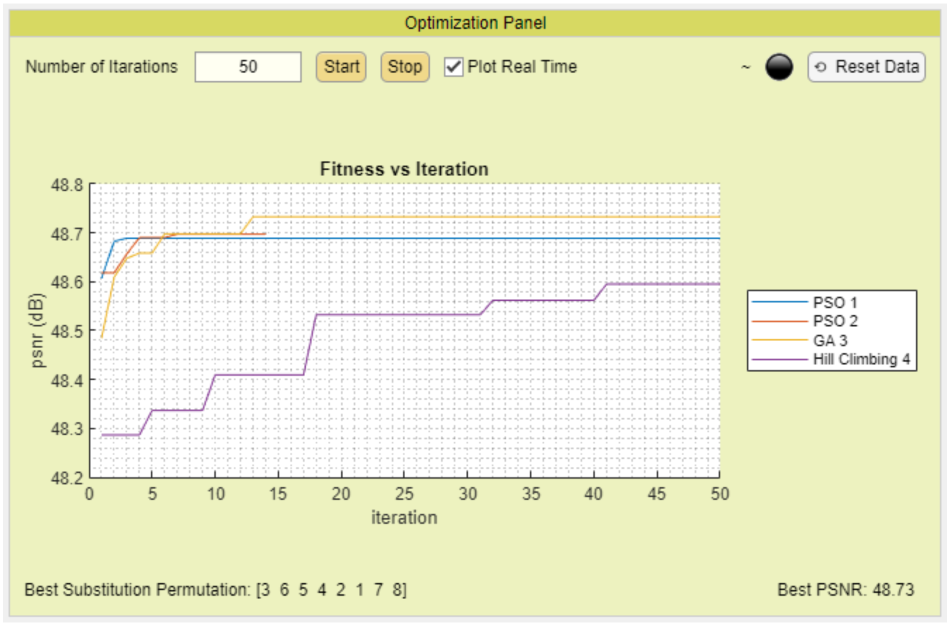
Optimization Panel is shown in Fig. 10. From the Optimization Panel one can specify number of iterations for the optimization, and start/stop the optimization of substitution permutations.

The best substitution permutation and the corresponding PSNR value is displayed at the bottom of the panel. (Note: Substitution-permutation is displayed with its minimum value starting from 1 instead of 0, and does not indicate incorrect solution. It is adjusted for MATLAB array indexing.)

If Plot Real time option is selected, fitness plot is updated at each iteration. It can be disabled for faster execution.

Furthermore, when optimization is started the lamp located at the top-right corner of the panel flickers, indicating that the optimization process has not finished.

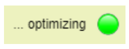
At each start, fitness history and the label of the algorithm is added to the application so that the optimization curves of different setting can be compared. Nevertheless, from the reset data button, all the optimization history can be reset.



Best Substitution Permutation: [3 6 5 4 2 1 7 8]

Best PSNR: 48.73

(a) Optimization Panel with Fitness Curves obtained via Different Optimizers. PSO 2 is intentionally stopped at 14^{th} iteration to demonstrate the Stop Bbutton.



(b) Indicator Lamp

Figure 10: Optimization Panel

All the codes can be found in Appendix.

Appendix

Test Message:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam vitae diam vehicula, vestibulum ligula non, ultricies ipsum. Quisque facilisis consectetur quam eget tincidunt. Proin nec mattis ex, ut dapibus ipsum. Nam tristique auctor vestibulum. In vel nisl sapien. Nam sit amet ante nec metus mattis feugiat vel sit amet lectus. Pellentesque at tempus lectus.Aenean ac imperdiet libero. Duis ornare ipsum at facilisis consectetur. Aenean quam felis, pretium eu est vitae, ultricies tempor ipsum. Curabitur sollicitudin, ex sit amet posuere vestibulum, ipsum sem eleifend nibh, sit amet eleifend quam neque vel nulla. In a augue imperdiet velit lacinia tempus id placerat elit. Duis eros felis, fermentum ac neque ut, elementum maximus ipsum. Etiam a finibus lacus. Nulla vel ultrices massa. Proin non auctor leo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Maecenas risus justo, consectetur nec eros tristique, sodales aliquam risus. Quisque a augue euismod, ornare magna sed, viverra felis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.Etiam dolor ex, blandit nec massa eget, auctor vulputate urna. Praesent ullamcorper consectetur nisi, id vulputate leo semper eget. Fusce nulla nibh, faucibus sed tortor quis, accumsan blandit libero. Vestibulum a ex aliquam, ultricies urna ac, efficitur massa. Duis sit amet purus nibh. Phasellus nec euismod mi. Suspendisse dapibus ex orci, ut faucibus tellus tincidunt sit amet. Integer efficitur mollis lacus, a tristique purus euismod et. Curabitur rutrum pretium hendrerit. Sed a tristique elit.Proin non venenatis est, a tempor nisl. Maecenas iaculis tellus a tellus convallis, eu lobortis lorem pharetra. Nunc vitae nisl et augue venenatis mollis non et elit. Nullam viverra pharetra neque. Vestibulum pharetra ultrices justo at vulputate. Vestibulum quis maximus metus. Duis ornare fermentum purus in scelerisque. Duis et magna arcu.Nunc nec lectus fermentum, ultrices tortor eu, convallis velit. Praesent rhoncus enim nec turpis suscipit, eget pharetra libero lobortis. In mollis purus ac erat porta efficitur. Nam sed leo turpis. Sed posuere velit sit amet augue tincidunt dapibus. Praesent non nisi condimentum, imperdiet dolor at, dapibus augue. Vestibulum feugiat quam mauris, ac vehicula nibh sagittis ac. Suspendisse rutrum non arcu quis finibus. Donec suscipit massa ultricies nisi sodales, sit amet tincidunt tellus dignissim. Vestibulum odio erat, porta a orci quis, interdum bibendum elit. Suspendisse ipsum ante, ultrices vel odio ac, ultricies sagittis ipsum. Cras ac est ac enim vestibulum tincidunt vitae ut diam. Vestibulum vel sollicitudin lectus. Nunc vel feugiat nibh. Maecenas quis accumsan elit. Etiam pretium turpis viverra faucibus sollicitudin.Sed condimentum consectetur est vitae faucibus. Duis efficitur, nisl ac ornare fringilla, mauris turpis lacinia mi, eget hendrerit quam arcu nec ipsum. Ut porta viverra elit, eget porttitor.

Link for Codes: `https://www.dropbox.com/sh/qrstj8o9gzdsaor/AAC7mvLEVopvILOM9FN9BEQwa?dl=0`

References

[1] A. S. Monal Mehta, “Digital watermarking and steganography,” *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT)*, vol. 02, 2013.

[2] P. K. Muhuri, Z. Ashraf, and S. Goel, “A novel image steganographic method based on integer wavelet transformation and particle swarm optimization,” *Applied Soft Computing*, vol. 92, p. 106257, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620301976

[3] J. PejaŚ and Łukasz Cierocki, “Reversible data hiding scheme for images using gray code pixel value optimization,” *Procedia Computer Science*, vol. 192, pp. 328–337, 2021, knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 25th International Conference KES2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921015210

[4] “Particle swarm optimization,” *In Proceedings of the International Conference on Neural Networks; Institute of Electrical and Electronics Engineers*, vol. 4, pp. 1942–1948, 1995.

[5] M. S., *Genetic Algorithm. In: Evolutionary Algorithms and Neural Networks.* Springer, Cham, 27 June 2018, vol. 780.

[6] P. S. A.J. Umbarkar1, “Crossover operators in genetic algorithms: A review,” *ICTACT JOURNAL ON SOFT COMPUTING*, vol. 6, 2015.

[7] U. Halıcı, “Computational intelligance meta-heuristics and local search,” 2020, . Middle East Technical University, Ankara,Turkey.

[8] P. I. Frazier, “A tutorial on bayesian optimization,” 2018, 1807.02811, arXiv, stat.ML.