# Problem Statement

Problem Statement

Non-valid Implicit Constraint

Lacking Progressive Evaluation

ROI Avoidance

Constant Optimization

Local Minimum

Losing Information

Less Qualitied Solution

Figure – the elements of the problem formulation

Objective 1

# Problem formulation

Assuming that we have an image which denotes the host image and it has the size of . Also, assuming that we have a message with the size of . The goal is to select subset of pixels from for inserting the bits of in the way of using the least four significant bits for embedding **LFSB** (Four Least significant Bit). Considering that the selected subset of pixels is a vector . Then the optimization problem is formulated as

Where

denotes the stegano image which is the image after embedding the message based on the solution inside

denotes the host image

denotes the mean square error between the host and the stegano and it is given by the equation

# Solution Space

The solution space is defined as a set of elements where each element has the length of the size of secrete message multiplied by the total number of pixels in the host image to encode the position of pixel for insertion and by a whole number ranging from 1 to 4 to encode the considered of bit for insertion. This means that the size of the solution space based on **the previous formulation**

for times

For a message with size , we can represent as

this allocates the size of bits.

this allocates the size of 2 bits (that is mean we need 2^2 for getting 1,2,3,4)

The total number of bytes that are needed to represent the value of is to store the message, i.e., stego key

كيف متغير واحد مثل | x | يشير الى قيمتين

مرة اجمالي (Total of bytes) و مرة عدد البايتات (number of bytes)

1. اجمالي البايتات التي احتاجها لتمثيل فضاء الحلول (solution space)الذي اسميناه |x|.

على فرض حجم الرسالة 6 بت، حجم الصورة (4\*3) سيكون

1. حيث | x | يشير إلى عدد البايتات اللازمة لتمثيل x. و ليكن 12 بايت من هو **x**
2. قيمة | x | يمثل عدد وحدات البايت اللازمة لتخزين الرسالة ، أي مفتاح stego.

Assuming S=”011011”

**x=(p1,0,p2,1,p3,1,p4,0,p5,1,p6,1). Is this correct representation?**

**How can determine the pixel position?**

**Log2(12)= 3.5850,**

**bytes**

## Example 1.1

**Problem**

.

**Solution**

this number allocates only 2 Bytes (is this solution default???)

(we need to 2 bytes, where s=2 bits as your assuming)

This solution means that the message of 2 bits will be inserted in the pixel 4 in the second least significant bit and in the pixel 9 in the third significant bit

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 10 | 11 | 12 |

## Example 1.2

Let us assume we have an image combined of 400 600 pixels. Also, we have a message of Message=” Heart is a repository of secrets”. We aim at using the previous representation to hide the message inside the image. How many bytes do we need for that?

635.9256

This means we need to allocate 636 Byte for only chromosome (We need to approximate the result)

## Example 1.3.

Assuming that we have an image combines of pixels and a message combined of bits. How much bits we need to represent its insertion according to the previous representation?

= bytes

(2^10)\*((log2(2^20)+2)/8)= 2816 (sometimes, no need to approximate the result)

The size of representation increases proportionally with the size of the message and the logarithmic of the size of the image

The authors such Kanan et al. (2014) who have used compact representation they fall in the issue of implicit constraint (reducing the number of candidate solutions), on the other side when using comprehensive representation, we fall in the problem of large size representation

Objective 2

# Jumping Based chromosome representation

Table –symbols used in the methodology and their interpretation

|  |  |
| --- | --- |
| Symbol | Meaning |
|  | Number of blocks in the height of the image |
|  | Number of blocks in the width of the image |
|  | Denotes the quantization level (input parameter) |
|  | The number of jumps for inserting the message |
|  | Number of bit changes in each block |
|  | Threshold |
|  | The direction of blocks |
|  | The number of used blocks |

## 4.1. Principle of working

1. We decompose the original image into image where

Where

denotes the quantization level (input parameter)

number of blocks in the height of the image

number of blocks in the width of the image

1. We define number of jumps for inserting the message , we use for amount of four bits which means that any message is at most is inserted in 16 diverse blocks (non-continuous).
2. We define a variable named number of bit changes within current location (set of connected blocks without jumps), which make track it and when it is more than pre-given threshold , we have to make a jump to continue inserting after the block is finished.
3. While inserting in any block, we use the same representation of Kanan et al. (2014).
4. Whenever we do a jump, we have to add to the chromosome the number of used blocks before jumping.
5. We add at the end of the chromosome the direction of connected blocks similar to the encoding of Kanan et al. (2014).

## Example 1.4.

Assuming we have an image combined of 600\*400 pixels and we have , message size is 200 bits. Assuming that we have the following solution

**Number of blocks**  or **amount of previous part of a message which it is stored**

4 bits

3 bits

Jumps bits

What can I call them?

Every part, metameric

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 0 | 1 | 1 |

Direction bits

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Notes:

If the number of jumps is different between two chromosomes, then their sizes are different.

The increase of the number of jumps implies an increase in the length of the chromosome.

The quantization that is considered implies allocation for the offset X, and offset Y in terms of a number of bits, while Kanan has allocated one byte for each of them we need a smaller number of bits according to the value of QL.

Assuming that we have the message ,

Assuming that =3

. Because the algorithm continues

Why added 2 bits only, is this a default?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 1 | 11011000 | 1100011 | 1000101 | 10000100 | 11110 | 10011111 |
| 2 | 11100001 | 1101001  1101011 | 1110101 | 1111011 | 101101 | 10101011 |
| 3 | 11010101 | 11010010 | 10001010 | 10000001 | 11001100 | 11101101 |
| 4 | 10001010 | 11011000 | 11000000 | 10010011 | 111001 | 11111100 |
| 5 | 10101 | 1010111 | 11011110 | 11110 | 110 | 10011111 |
| 6 | 110110 | 110110 | 11001111 | 11111111 | 11101010 | 11110110 |
| 7 | 1000101 | 10110111 | 100001 | 1101111 | 1001 | 10100010 |
| 8 | 110110 | 1100110 | 11110011 | 1000010 | 11010101 | 1101100 |
| 9 | 1111 | 11001001 | 11001001 | 10111010 | 10101000 | 1101001 |

## 4.2. Generating chromosome and Objective function

A role of the objective function is to evaluate the performance of the candidate solution. It works based on the following process

1. Generate a chromosome and use it for embedding inside the image and obtain the stego image
2. Use the original image and the stego image to find the value of MSE.
3. The chromosome is added at the beginning of the image and the MSE is calculated before insertion of a chromosome.

Pseudocode 1 of generating a chromosome (progressive evaluation objective 3)

|  |
| --- |
| **Input**  host\_image  message  QL  T\_nbc  b\_dir // block direction  p\_dir //pixel direction  // the same of Kanan  SB-Pole  SB-Dire  BP-Dire  Bit-Planes  **Output**  stego\_image  chromosome  MSE  **Start**  [m,n]=size(host\_image)  For example: m=9, n=6  **Where QL=2**  nb=ceil(n/ QL)  mb=ceil(m/ QL)  generate Random Block between 1 and nb, 1 and mb and assign it to the current location  and random values for, SB-Pole, SB-Dire , BP-Dire, Bit-Planes.  generate first part of chromosome  nbc=0;  while (message is remaining)  if(nbc<T\_nbc)  insert one bit in current location using (SB-Pole, SB-Dire ,BP-Dire, Bit-Planes)  update the value of nbc and used blocks    else    generate Random Block between 1 and nb, 1 and mb and assign it to current location  and random values for, SB-Pole, SB-Dire , BP-Dire, Bit-Planes and assign them to new part of chromosome  append number of used blocks and new part of chromosome to the current part  nbc=0;    end  end  calculate MSE (stego\_image,host\_image)  return stego\_image, chromosome, MSE    **End** |

## 4.3. Algorithm of extracting

We assume that the best chromosome is one that achieves minimum value of MSE is the one that will be used for embedding. We call it as It is stored in the beginning of the image. We point out that our algorithm is variable length genetic optimization, this means that the best chromosome is not fixed length. Hence, there is a need to insert the length of the chromosome at the beginning of the stego image before inserting the chromosome itself.

Pseudocode 2 of extracting a message

|  |
| --- |
| **Input**  stego\_image  **Output**  *message*  **Start**  *message* =[];  Extract the length of the chromosome from the beginning of the image  Extract from the beginning of the stego image based on the extracted length  Extract the number of jumps from first metameric  For from i=1 until take the corresponding metameric and number of blocks  For j=1 until number of    Use the metameric for extraction and them to the *message*  end  end  **End** |

## 4.4. Crossover

The goal of the crossover is to generate two chromosomes from two parents. Assuming that we have two parents , the crossover will conduct the following operation to generate the off-spring

What is , you said will explain it.

**Example please**

Pseudocode 3 of crossover operator

|  |
| --- |
| Input  two parents ,  Output  **C, D are mean a new of off-spring**  ,  Start  Create a set of all jumps as a concatenation of jumps of and jumps and store in  Give the number of jumps of to  Is  Give the number of jumps of to  For i=1 until number of jumps of  Generate random metameric from and add it to  End  For i=1 until number of jumps of  Generate random metameric from and add it to  End  End |

## 4.5. Mutation

The goal of the mutation is to generate one chromosome from an existing one. Assuming that we have chromosome the mutation will conduct the following operation to generate the mutated chromosome

Pseudocode 4 of mutation operator

|  |
| --- |
| Input    Output      Start  Extract the number of jumps in  Generate random index for metameric of from 1 until number of jumps  Change one of b\_dir, p\_dir , SB-Pole, SB-Dire , BP-Dire, Bit-Planes  End |

## 4.6. Environmental or elites ’selection and Population extraction

We use one of two methods for elites’s selection

1. Roulette wheel
2. K-tournament

## 4.8. Exploration and exploitation balance

Max is input parameter.

Current-Iteration is computed. How?

We use an adaptive mutation rate

Where

denotes the maximum number of iterations in the genetic

Generate random number r between 0 and 1

If (r <) perform mutation

Else

Do not do mutation

end

## 4.9. Constraint enabling validation

For each new jump, we check the amount block that is available without violating the constraint ROI.

## 4.10. general algorithm

|  |
| --- |
| **Input**  Host image  Message  Population size  QL  Max number of iterations  Mode = roulette wheel /k-tournament  Objective function =@MSE    **Output**  Best chromosome  **Start**  Read the size of the host image  Decompose the image into blocks based on size and QL  Generate first population based on population size and pseudocode 1  current= first population  For iteration=1 unit Max  off-spring=[];  While size of off-spring < Population size  Extract two elites based on mode (Objective function)  Generate two children and add them to off-spring  Perform mutation based on mutation rate  Update mutation rate  End  Candidate solutions=Combine(current, off-spring)  Select number of solutions equal to Population size using mode (Objective function) and  Add them to current population    End  Return best chromosome  End |

ROI definition

ROI is defined as a mask equal in size to the size of the host image, however, it represents a binary image where 1 represents forbidden region ROI and 0 allowed region NROI. This information comes from expert.

Before selecting any block for embedding we test if it is inside ROI we make jump

# Evaluation scenarios

**اسئلة**

1. عمل الخوارزمية، اخفاء النص في الصورة.
2. الكروموسوم يمثل stego-key الذي يحتاجه الطرف الاخر لاستخلاص الرسالة.
3. هل المقصود بكروموسوم متغير الطول لكل عملية تضمين بت من بتات الرسالة السرية في داخل الصورة المضيفة الواحدة ام من صورة مضيف لاخرى. (ارجو اعطائي مثال)
4. الباحث kanan استخدم 16 Direction متسلسلة، بينما طريقتنا سنستخدم، 16 diverse blocks (non-continuous) حيث يقصد بال direction هو بلوكات متسلسلة (kanan) او متفرقة (طريقتنا)
5. طريقة تضمين بتات الرسالة تعتمد على طريقة LSB التقليدية.
6. عدد بتات الصورة التي ستستضيف الرسالة السرية عددها متغيير اما 1 او 2 او 3 او 4. كيف سيتم تحديدها من خلال الخوارزمية الجينية.
7. الهدف من عملنا سيكون ضمن الاربعة بتات الاقل اهميه 4LSB للحفاظ على جودة الصورة المضيفة؟
8. انا كنت اتصور ان كل عملية اضافة بت من بتات الرسالة في مواقع بتات بكسل الصورة ستطبق عليها الخوارزمية الجينية حيث اخذ كل البتات و بالتالي هي ستحدد اي بتتين مناسبة لعملية التضمين بينما في عملنا على المقترح الجديد سيكون ...
9. اخذ فقط 4 بت 4LSB. سؤالي هل ستحدد اي منهم المناسبة للتضمين؟
10. قلت انه اذا تم تضمين بت من بتات الرسالة في بت من بكسلات الصورة المضيفة و اعطت تشوه كبير لا نستمر بتطبيقها على كافة بكسلات الصورة و بالتالي ناخذ بت اخرى من بكسلات الصورة. و بالتالي نقلل من وقت البحث. هل هذا سيحدد اي بت من بكسلات الصورة ستكون هي المرشحة لاعطاء افضل النتائج؟
11. اهداف البحث هل ستكون زيادة سعة التضمين و الجودة و تقليل وقت البحث؟
12. في النهاية الكرموسوم ليس له علاقة بتحديد افضل موقع للتضمين فقط للحصول على stego-key.
13. في منهجية البحث. مطلوب مني اتكلم عن فصل اجزاء الصورة عن بعضها اقصد (ROI and RONI) مالطريقة المستخدمة حتى اتكلم عنها.

Article of Hyifaa

<https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0170329&type=printable>