Heurísticas para Matriz de Substituição do LSB utilizando Algoritmos Genéticos e Path Relinking

Marcus V.M. Vieira, André L. Brazil, Aura Conci, Célio V.N. Albuquerque[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
CEP: 24210-240 – Niterói – RJ – Brazil

{mvieira,abrazil,aconci,celio}@ic.uff.br

Abstract. *With the increase of data traffic the need for transmitting secure information gains another perspective. One way to protect the data sent over the web is to embed the relevant information within an unsuspected image. This paper proposes the protection of secret data by combining the genetic algorithm with a path relinking process to optimize the substitution matrix in order to improve the stego image quality. Computational results show that the proposed combination outperforms the LSB substitution technique, the genetic algorithm technique and other variations, concerning both security and stego image quality.*

Resumo. *Com o aumento do tráfego de dados, a necessidade de transmitir informações de modo seguro ganhou outro perspectiva. Um modo de proteger os dados enviados na Internet é inserir a informação relevante dentro de uma imagem comum. Este trabalho propõem a proteção da informação secreta, combinando o algoritmo genético com o path relinking para otimizar a matriz de substituição, a fim de melhorar a qualidade da estego imagem gerada. Resultados computacionais mostraram que o método proposto supera a técnica de substituição do LSB, do algoritmo genético e outras variações, tanto em termos de segurança e qualidade da estego imagem.*

1. Introdução

Um grande esforço já foi feito para fornecer proteção aos dados. O maneira mais comum e fácil de proteger os dados é através da utilização das técnicas criptográficas. Estas técnicas são eficazes contra ataques de recuperação da informação. No entanto, enfrentam algumas dificuldades e vulnerabilidades, especificamente relacionados com a distribuição de chaves [Bellare and Rogaway 1993].

A criptografia transforma os dados em uma informação aparentemente sem sentido. Dessa forma protege o conteúdo relevante, mesmo que ele se seja capturado durante o processo de transmissão [Jan and Tseng 1996, Bourbakis and Alexopoulos 1992, Highland 1997, Rhee 1993]. Existem diversos algoritmos de criptografia, entretanto a maioria deles tem uma desvantagem significativa: a simples evidência de que existe informação ocultada pode ser suficiente para que os atacantes corrompam os dados ocutlados, impossibilitando a extração da mensagem por parte daquele para quem a mensagem foi endereçada.

Com a esteganografia este problema não ocorre, pois a informação é ocultada em imagens aparentemente sem importância e de forma imperceptível [Johnson and Jajodia 1998, Anderson and Petitcolas 1998, Petitcolas et al. 1999,

Provos and Honeyman 2003, Julio et al. 2007]. Assim, evita-se a detecção por atacantes que provavelmente não analisarão as imagens para verificar se há algo escondido [Bender et al. 1996, Duric et al. 2005]. Assim, as técnicas mais recentes buscam formas de deixar as estego-imagens com um pequeno grau de distorção em relação as imagens de cobertura [Wang et al. 2001, Thien and Lin 2003, Chang et al. 2003, Wang 2005, Yu et al. 2007, Hsu and Tu 2010].

Para aumentar a segurança, pode-se combinar os dois métodos, de forma que a informação secreta seja primeiro criptografada e depois ocultada. Desta maneira, por exemplo, uma imagem interceptada em que se desconfie a existência de algo oculto, permanece como informação protegida, pois ainda tem que ser descriptografada.

Este trabalho é uma extensão da solução utilizando algoritmos genéticos propostos por Wang et al.(2001). Portanto, a nossa contribuição está na combinação dos algoritmos genéticos com um processo de refinamento conhecido como *Path Relinking* [Glover and Laguna 1997, Glover 1998] para alcançar imagens com distorções indetectáveis. Então, similarmente ao trabalho de Wang, este trabalho foca em não degradar a qualidade da estego imagem.

Este artigo é organizado da seguinte maneira. Na Seção 2 são introduzidos os conceitos dos algoritmos genéticos e do *path relinking*. Na Seção 3, estamos propondo um algoritmo esteganográfico que utiliza a combinação dos algoritmos genéticos com o *path relinking*. Já na Seção 4 são apresentados os resultados dos algoritmos propostos e dos métodos de ocultamento existentes. Finalmente, na Seção 5, a conclusão desse estudo e os trabalhos futuros são apresentados.

2. Conceitos

Nesta seção, a técnica mais popular para a esteganografia é apresentada: a Substituição do Bit Menos Significativo (LSB - *Least Significant Bit*) [Johnson and Jajodia 1998]. Além disso, são introduzidos os conceitos necessários para entender a classe de algoritmo *GAPR* (Algoritmos Genéticos com *Path Relinking*) proposta na Seção 3 e alguns trabalhos relacionados também.

Imagens digitais são representadas por uma matriz de pixels, na qual cada pixel é composto por bytes. A técnica de substituição do LSB utiliza os bits menos significativos para armazenar a informação secreta na imagem de cobertura. Ao modificar o LSB de cada byte, os olhos humanos não conseguem perceber a diferença. Isto acontece devido a baixa influência do LSB na tonalidade e intensidade dos pixels. Por isso, a alteração não afeta a qualidade da estego imagem gerada.

Entretanto, a carga de informação que o bit menos significativo suporta é muito baixa. Então, para ser mais eficiente, técnicas recentes propuseram a substituição de mais bit por pixel [Wang et al. 2001, Thien and Lin 2003, Chang et al. 2003, Wang 2005, Yu et al. 2007]. Porém, o desafio é selecionar corretamente a forma como serão substituídos para não distorcer significamente a estego imagem. Sendo esta a principal motivação deste trabalho.

2.1. Matriz de Substituição

A matriz de substituição é uma técnica esteganográfica que mapeia em uma matriz os bits menos significativos substituídos pelos ocultados. Dessa forma, ela produz estego ima-

gens mais semelhantes à imagem de cobertura quando comparado à simples substituição do LSB. O processo de mapeamento da matriz de substituição é mostrado na Figura 1.

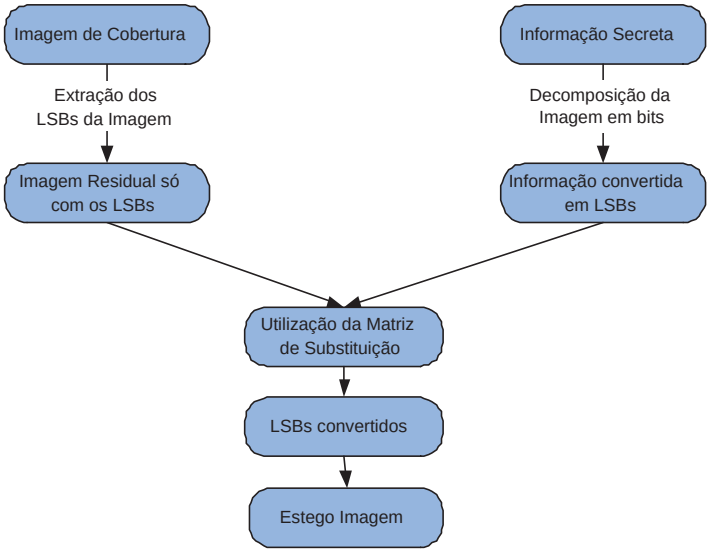


Figura 1. Ocultamento da imagem usando a matriz de substituição

Para que a matriz de substituição possa ser usada no ocultamento da informação, ela deve possuir as seguintes características:

1. Cada linha representa a sequência de bits que serão substituídos;
2. Cada coluna representa a sequência de bits que serão convertidos;
3. Cada valor, na coordenada [linha, coluna] da matriz só pode ser 0 ou 1, onde o valor 1 indica que a linha será convertida na coluna 0. Então, o valor 1 só pode aparecer uma única vez em cada linha ou coluna. (Isto ocorre porque uma sequência de bit não pode ser convertida em duas sequências de bits diferentes).

Já a matriz de substituição será uma matriz quadrática de tamanho 2^k, onde k é o número de bits menos significativos substituídos no processo. Então, quando três bits menos significativos forem usados $(k = 3)$, a matriz de substituição possuirá o tamanho de 2^3, então uma matriz 8 x 8. A Figura 2 mostra um exemplo de matriz de substituição.

Sequência de Bits Convertidos

Sequência de Bits Substituídos	0	1	2	3	4	5	6	7
0	0	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	0	0	0	1
5	0	0	0	1	0	0	0	0
6	0	0	0	0	0	1	0	0
7	0	1	0	0	0	0	0	0

Figura 2. Exemplo de uma matriz de substituição

Entretanto, a matriz de substituição deve ser incorporada à imagem de cobertura, ou informada de alguma forma, pois ela é necessária para codificar e decodificar os valores do LSB. Apenas com ela o receptor será capaz de reverter o processo de ocultamento da informação na estego imagem e obter a informação desejada. Assim, ela possui a mesma funcionalidade das chaves criptográficas.

Encontrar a matriz de substituição perfeita é um grande desafio, devido ao grande número de possibilidade existentes [Wang et al. 2001]. Para ser mais exato, o número total de possibilidades da matriz de substituição é $2^k!$. Então, quando $k > 3$, o número total de matrizes que serão processadas se torna muito grande. Por exemplo, quando $k = 4$, o total é maior que 20 trilhões de possibilidades.

Entretanto, para aumentar a qualidade da estego imagem gerada, pode-se utilizar uma matriz de substituição para cada canal de cor dos pixels da imagem. Com isso, o número total de possibilidades aumentará para $(2^k!)^n$, onde n é quantidade de matrizes de substituição utilizadas. Então, substituído 3 bits menos significativos $(k = 3)$ e utilizando 3 matriz de substituição, uma para cada canal de cor do RGB $(n = 3)$, o total de possibilidades passará a ser maior que 65 trilhões. Porém, utilizando uma matriz com $k = 3$ e $n = 1$, o total de possibilidade é de aproximadamente 40 mil.

Então, o uso de algoritmos heurísticos para encontrar soluções quase ótimas é de extrema importância. Uma das mais promissoras técnicas são os algoritmos genéticos apresentados na próxima seção.

2.2. Algoritmos Genéticos

Os algoritmos genéticos (AG) são uma classe particular de algoritmos evolutivos inspirados na idéia darwiniana de seleção natural. Estes algoritmos são métodos de busca e otimização que simulam os processos naturais de evolução, nomeados: Crossover; Mutação e Seleção Natural.

O crossover é a etapa que mais favorece a evolução, pois ela combina as melhores características dos indivíduos. Esta combinação é feita selecionando dois indivíduos da população para serem divididos e recombinados em dois novos indivíduos. Já o processo de mutação consiste em alterar algum cromossomo do individuo. Esta etapa é importante por contribuir para evolução de indivíduos similares, enquanto o crossover não.

Na etapa de seleção natural, é necessário aferir a cada indivíduo uma pontuação, para que apenas os mais aptos sobrevivam, convergindo para a solução do problema proposto.

Estas etapas são utilizadas para assegurar que a nova geração seja renovada, apresentando as melhores características e adaptações de seus ancestrais. Estes indivíduos mais bem adaptados tendem a não desaparecer da população, uma vez que a seleção natural os escolherá a partir do conjunto inicial e dos gerados no crossover.

O algoritmo genético básico esta representado na Figura 3 e funciona da seguinte forma. Uma população inicial de cromossomos é gerada. Após isto, o ciclo do algoritmo genético começa com a etapa de crossover, onde dois indivíduos são selecionados para serem os pais de um novo individuo, esta etapa se repete até que não haja mais pais. Nos algoritmos originais de Holland (1992), um deles era elegido de acordo com o seu valor de aptidão, enquanto o outro pai era elegido aleatoriamente. Logo após ocorre a mutação,

nesta etapa, é necessário uma taxa de mutação baixa para não comprometer a geração. E por último o processo de seleção natural. Para terminar o ciclo, pode ser utilizado o número de gerações ou algum critério de convergência.

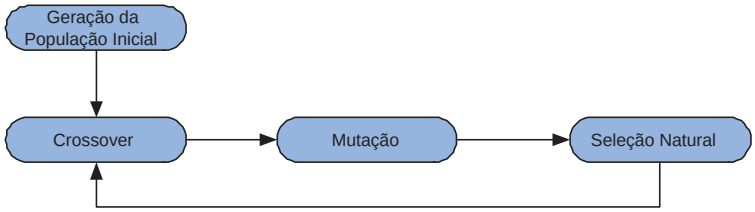


Figura 3. Os Algoritmos Genéticos

Mesmo com a mutação, os algoritmos genéticos após alguns ciclos passam a gerar indivíduos semelhantes aos da geração passada, uma vez que os indivíduos com as melhores características seriam escolhidos pela seleção natural. Assim, determinados genes serão mais presentes nas gerações futuras. Então, com o *path relinking*, indivíduos que não estariam presentes na geração passariam a ser testado. Além disso, ao executar *path relinking* no fim de cada ciclo do algoritmo genético, ele contribuirá para o próximo ciclo.

2.3. *Path Relinking*

Path Relinking é um método interessante de busca local que pode ser aplicado em uma grande variedade de aplicações [Glover and Laguna 1997, Glover 1998]. Quase todos os métodos que trabalham com um conjunto elite de candidatos ou uma seleção de melhores objetos podem ser refinados com esta etapa.

Pode-se utilizar em duas estratégias: aplicado como um pós-otimizador em todos os pares do conjunto elite; e agindo como um intensificador de cada solução ótima local obtida do conjunto elite, adotando-a como solução inicial.

Aplicar o *path relinking* como uma estratégia de intensificação de cada solução ótima local parece ser mais eficaz do que simplesmente utilizá-lo como uma etapa de pós-otimização. [Resende and Ribeiro 2007]

Algumas variantes surgiram a partir destas abordagens, como o *backward path relinking* e o *reverse path relinking*. O *backward path relinking* é um intensificador da solução ótima local, sendo a solução ótima, a solução guia, diferentemente do intensificador normal. Já o *reverse path relinking* é executado após o path relinking, trocando apenas a solução inicial pela solução guia e vice-versa.

O processo *path relinking* é representado na Figura 4 e inicia escolhendo dois objetos entre os presentes no conjunto elite. Um dos objetos é eleito como ponto de partida (S1) e o outro como objeto guia (S2). Para que o processo gere uma grande quantidade de soluções, os dois objetos devem ser bastantes diferentes.

Após a escolha dos objetos, o próximo passo é verificar a diferença entre S1 e S2, para que parte da solução inicial seja alterada, tornado-o cada vez mais similar a S2. Essas pequenas alterações são mapeadas em novos objetos, como objetos intermediários. Como em cada etapa do processo mais de uma solução é gerada, então é necessário escolher o

melhor objeto que soluciona o problema para que o processo possa continuar a partir dele, como novo ponto de partida. Este processo é repetido até que solução guia S2 é alcançada.

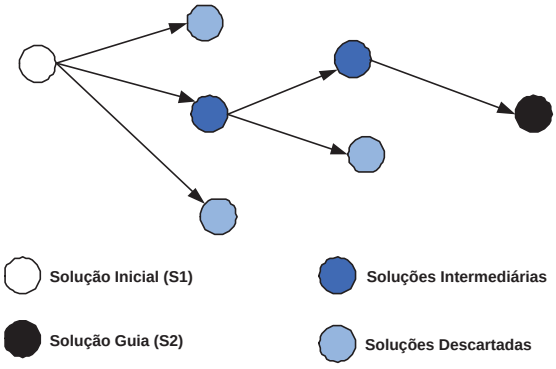


Figura 4. *Path Relinking*

Finalmente, se algum objeto intermediário gerado apresentar um valor maior ou igual ao melhor objeto presente no conjunto elite, este objeto é adicionado a esta lista. O *path relinking* termina quando não houver mais objetos no conjunto elites para serem processados.

Resultados interessantes podem ser encontrados, aplicando *path relinking* de maneira inversa (chamado *Reverse Path Relinking*).

3. Classe de Algoritmos *GAPR*

A descoberta da matriz de substituição ótima pode ser muito custosa, especialmente se $k > 3$. Então, uma alternativa é encontrar uma solução quase-ótima em um tempo razoável. Para alcançar isto, o uso dos algoritmos genéticos foi proposto [Wang et al. 2001]. Wang et al.(2001) comprovaram o uso dos algoritmos genéticos como uma técnica promissora. Para tentar melhorar a qualidade das estego imagens geradas, estamos propondo a combinação do *path relinking* com os algoritmos genéticos, criando a classe de algoritmo *GAPR*.

Como os algoritmos genéticos são evolucionários e o *path relinking* trabalha com um conjunto elite de soluções, é de extrema importância a utilização de função para avaliar os objetos gerados. Para isto, foi escolhido o PSNR (*Peak Signal-to-Noise Ratio*), dada pela equação 1. PSNR é um método simples de avaliar a perda de qualidade da imagem comparando duas instâncias da mesma imagem.

$$PSNR = 20 \times \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right). \tag{1}$$

Na equação 1, MAX_I é o máximo valor possível de intensidade da cor. Em uma imagem 8-bits por pixel, MAX_I é 255. Entretanto, antes de calcular o PSNR é necessário obter o MSE (*Mean Square Error*) das duas imagens. O MSE pode ser calculado pela equação 2.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||I(i,j) - K(i,j)||^2, \tag{2}$$

onde *I* e *K* são as imagens na comparação, *i* e *j* são as coordenadas de cada pixel e *m* e *n* são o número total de pixels em cada direção.

A classe de algoritmo *GAPR* utiliza as matrizes de substituições para melhorar a qualidade das estego imagens geradas. Entretanto, os algoritmos genéticos utilizam vetores, sendo necessário transformar a matriz de substituição em um vetor [Wang et al. 2001]. Esta transformação é simples, pois o valor 1 na matriz só deve aparecer uma vez por linha e coluna, então cada posição do vetor corresponde a linha da matriz e os valores são os índices das colunas. A Figura 5 mostra a conversão da matriz em um vetor.

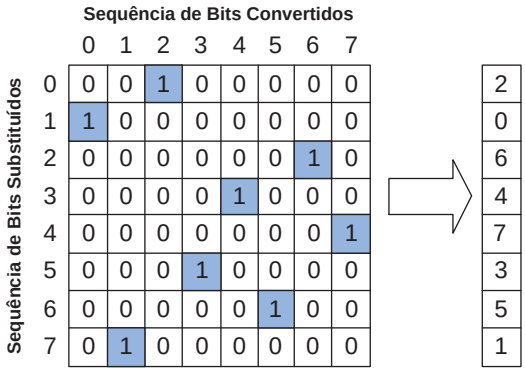


Figura 5. Conversão da Matriz de Substituição em Vetor

Uma vez os conceitos já estabelecidos na Seção 2 e as etapas compartilhadas pelos algoritmos da classe, já pode-se definir cada um deles.

3.1. *GAPR Basic*

GAPR Basic é a combinação dos algoritmos genético com o *path relinking*. Essa combinação pode ser feita, pois o *path relinking* utiliza um conjunto elite de soluções e os algoritmos genéticos podem prover isso a cada geração. O *GAPR Basic* utiliza um ciclo similar ao do algoritmo genético. Entretanto, ao final do ciclo do AG, é adicionado o *path relinking*. Assim, outras soluções podem ser testadas e podem contribuir para o próximo ciclo.

Como a idéia básica é gerar estego imagens com o menor grau de distorção, o *GAPR Round Trip* foi proposto. Este algoritmo testa mais soluções que o *GAPR Basic*.

3.2. *GAPR Round Trip*

GAPR Round Trip adiciona uma nova etapa ao algoritmo do *GAPR Basic*: O *reverse path relinking*. Ao terminar o *path relinking* com a solução inicial (s1) e a solução guia (s2), as soluções são trocadas, ou seja, a solução inicial passa ser a s2 e a solução guia s1. Com isso, outras soluções que não foram testadas pelo *GAPR Basic* passarão a ser testadas, porém uma grande quantidade já terá sido testada.

Como a utilização do *path relinking* , o tempo de processamento aumentou consideravelmente, por isso, estamos propondo um algoritmo mais simples e rápido, chamado *GAPR Fast*.

3.3. *GAPR Fast*

O *GAPR Fast* não testa todas as soluções possíveis que o *path relinking* gera. Ao invés disso, ele escolhe a primeira solução intermediária e a utiliza como novo ponto de partida, como mostra a Figura 6.

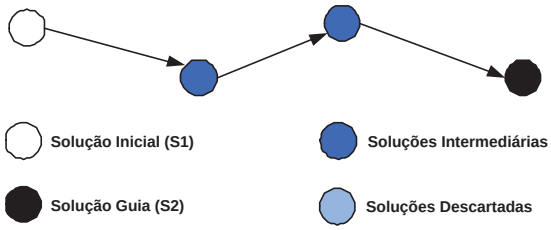
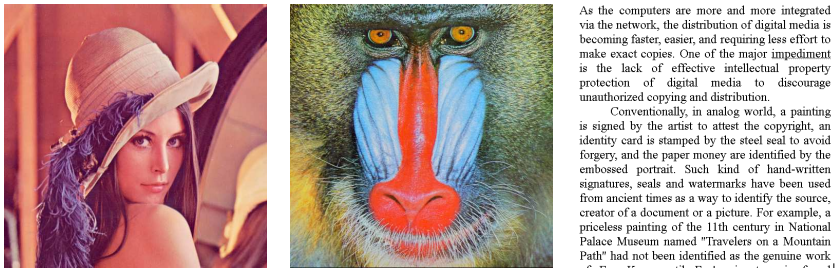


Figura 6. O processo do path relinking fast

Além disso, *GAPR Fast* é menos rigoroso na escolha das soluções que podem fazer parte do conjunto elite, já que ele aceita qualquer solução intermediária que seja melhor que a pior solução presente nesta geração, ao invés de aceitar apenas soluções intermediárias iguais ou melhores que a melhor solução, como é utilizado no *GAPR Basic* e *GAPR Round Trip*.

4. Resultados

Os testes foram executados em um computador pessoal com processador Intel Q6600 2.40 GHz e 2GB de memória RAM, utilizando sistema operacional Microsoft Windows Seven 32bits. O código fonte foi desenvolvido em C++ para ser executado em apenas um processador.


As the computers are more and more integrated via the network, the distribution of digital media is becoming faster, easier, and requiring less effort to make exact copies. One of the major impediment is the lack of effective intellectual property protection of digital media to discourage unauthorized copying and distribution.

Conventionally, in analog world, a painting is signed by the artist to attest the copyright, an identity card is stamped by the steel seal to avoid forgery, and the paper money are identified by the embossed portrait. Such kind of hand-written signatures, seals and watermarks have been used from ancient times as a way to identify the source, creator of a document or a picture. For example, a priceless painting of the 11th century in National Palace Museum named "Travelers on a Mountain Path" had not been identified as the genuine work

(a) Lena (b) Babuíno (c) Texto

Figura 7. Imagens de Cobertura

(a) Avião (b) Barco (c) Tyffane

Figura 8. Informação Secreta

Foram realizados 9 testes para avaliar os algoritmos pertencentes a classe *GAPR*. Estes testes consistem na combinação entre as imagens de cobertura apresentada na Figura

7 e as informações secretas da Figura 8. Estas imagens são as mesmas figuras utilizadas por Wang et al.(2001).

O processo de *path relinking* e os algoritmos genéticos utilizam um conjunto de parâmetros que definem o tempo gasto e a qualidade das estego imagens geradas. Então, os parâmetros foram setados de forma que a distorção seja a menor possível, em um tempo razoável. Para que isso fosse alcançado, utilizou-se os mesmos valores de Wang et al. Os parâmetros são os seguintes:

1. Taxa de mutação de 10% por gene;
2. Número de gerações igual a 8, pois apresenta o melhor custo benefício em relação ao ganho de qualidade e tempo de processamento nos teste realizados com 2, 5, 8, 11 e 14 gerações;
3. A seleção natural escolhe apenas as 10 soluções mais aptas e
4. O processo de *path relinking* só pode adicionar 20 soluções ao conjunto elite.

Com os parâmetros definidos, cada teste foi realizado 10 vezes devido ao não-determinismo dos algoritmos genéticos e *path relinking*. Então, calculou-se a média e o desvio padrão dos resultados de cada execução, gerando as Tabelas 1.

A partir das Tabelas 1, plotou-se o gráfico apresentado na Figura 9. Este gráfico facilita a comparação das técnicas, mostrando a diferença de qualidade obtida entre os algoritmos propostos e a técnica de LSB.

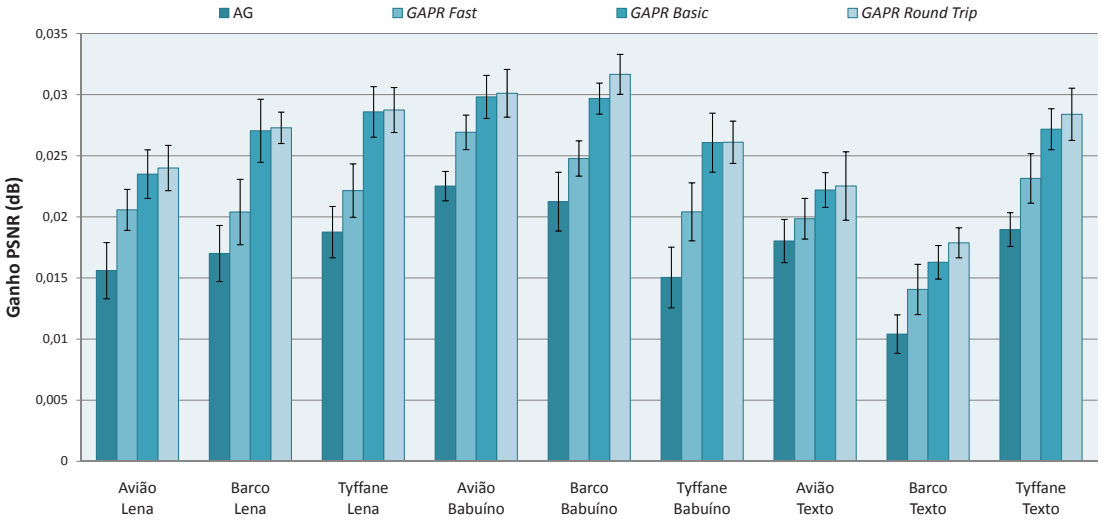


Figura 9. Comparando o ganho obtido pelo algoritmo da classe GAPR

Como pode ser visto no gráfico da Figura 9, em todos os testes realizados, os algoritmos da classe *GAPR* geraram estego imagens com nível de distorção menor do que os AG. Além disso, os piores resultados encontrados nos algoritmos *GAPR* apresentaram melhoria em relação aos AG.

Algumas observações são importantes salientar. Primeiro, todas as técnicas apresentam melhoria em relação a simples substituição do LSB. Segundo, o desvio-padrão de quase todos os testes realizados pelos algoritmos não determinísticos são bem pequenos, indicando uma convergência consistente para soluções quase ótimas em apenas uma execução.

(a) Imagem de Cobertura: Lena

Informação Secreta		Avião	Barco	Tyffany
LSB	Média	34,798	31,954	33,448
AG	Média	34,814	31,971	33,467
	σ	0,002	0,002	0,002
GAPR Fast	Média	34,819	31,975	33,471
	σ	0,002	0,003	0,002
GAPR Basic	Média	34,822	31,981	33,477
	σ	0,002	0,003	0,002
GAPR Round Trip	Média	34,822	31,981	33,477
	σ	0,002	0,001	0,002

(b) Imagem de Cobertura: Babuíno

Informação Secreta		Avião	Barco	Tyffany
LSB	Média	34,875	31,986	33,500
AG	Média	34,898	32,007	33,515
	σ	0,001	0,002	0,002
GAPR Fast	Média	34,902	32,011	33,521
	σ	0,001	0,001	0,002
GAPR Basic	Média	34,905	32,016	33,526
	σ	0,002	0,001	0,002
GAPR Round Trip	Média	34,905	32,018	33,526
	σ	0,002	0,002	0,002

(c) Imagem de Cobertura: Texto

Informação Secreta		Avião	Barco	Tyffany
LSB	Média	32,414	29,406	30,945
AG	Média	32,432	29,417	30,964
	σ	0,002	0,002	0,001
GAPR Fast	Média	32,434	29,420	30,968
	σ	0,002	0,002	0,002
GAPR Basic	Média	32,437	29,422	30,973
	σ	0,001	0,001	0,002
GAPR Round Trip	Média	32,437	29,424	30,974
	σ	0,003	0,001	0,002

Tabela 1. Comparação da Qualidade (Valores PSNR)

O uso do *path relinking* resultou em um leve aumento de qualidade em relação aos algoritmos genéticos. O aumento médio da qualidade foi de aproximadamente 50,6% comparado com o ganho do AG em relação a técnica de substituição do LSB.

GAPR Round Trip gerou os melhores resultados. O ganho de qualidade é aproximadamente 53,3%. Entretanto, comparando com o *GAPR Basic*, o aumento foi em média de apenas 1,5% melhor.

Além de analisar o nível de distorção das estego imagens em comparação às imagens de cobertura, é interessante avaliar o tempo de processamento de cada imagem gerada. Assim, é possível analisar qual algoritmo é melhor para cada situação. O gráfico da Figura 10 mostra a média do tempo gasto por cada técnica, além do desvio padrão do mesmo.

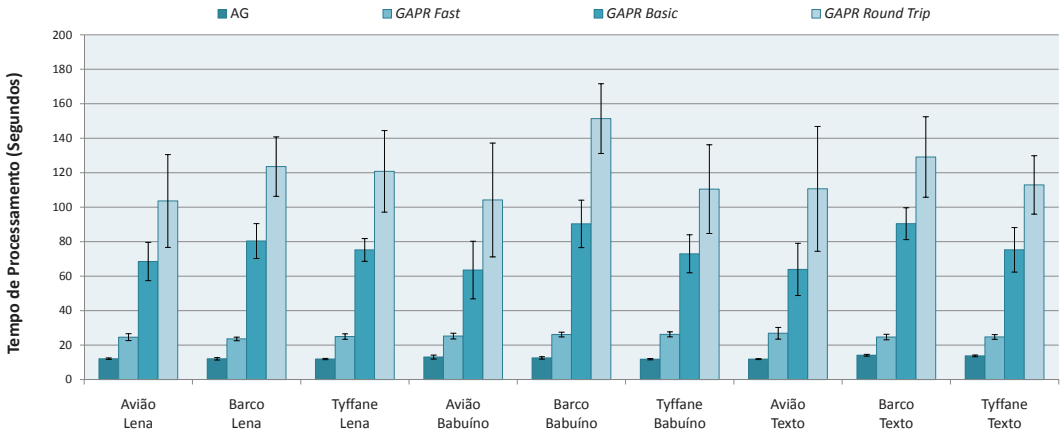


Figura 10. Comparando o tempo de processamento dos algoritmos *GAPR*

Mesmo com a melhora de qualidade apresentada pelos algoritmos da classe *GAPR*, o tempo de processamento aumentou consideravelmente, devido ao aumento da quantidade de soluções intermediárias geradas pelo *path relinking*. Uma alternativa é o uso do algoritmo rápido, *GAPR Fast*, que apresenta resultados em um tempo menor do que os demais e ainda aumenta a qualidade em relação ao AG.

O gráfico da Figura 11 apresenta o comportamento do ganho em relação a técnicas de substituição do LSB, assim como o tempo de processamento gasto por cada técnica. A partir desses gráficos, percebe-se que o ganho de qualidade segue uma tendência linear. Além disso, comprova-se o pouco acréscimo do *reverse path relinking*. Entretanto, o tempo gasto segue uma curva exponencial. Por isso, o *GAPR Round Trip* apresentou a pior razão $qualidade/tempo$ dentre os algoritmos da classe *GAPR*.

Além das análise de tempo e qualidade das técnicas, avaliou-se através do gráfico da Figura 12 o tempo de processamento de cada passo do algoritmo *GAPR Round Trip*. Fica claro que o grande causador do aumento do tempo do *GAPR Round Trip* é o *path relinking*, ocupando mais de 80% do tempo de processamento.

Outro ponto que vale salientar é que o *path relinking* não gasta o mesmo tempo, nem gera o mesmo número de soluções que o *reverse path relinking*. Essa diferença de valores ocorre devido ao parâmetro que limita o número de soluções que podem ser

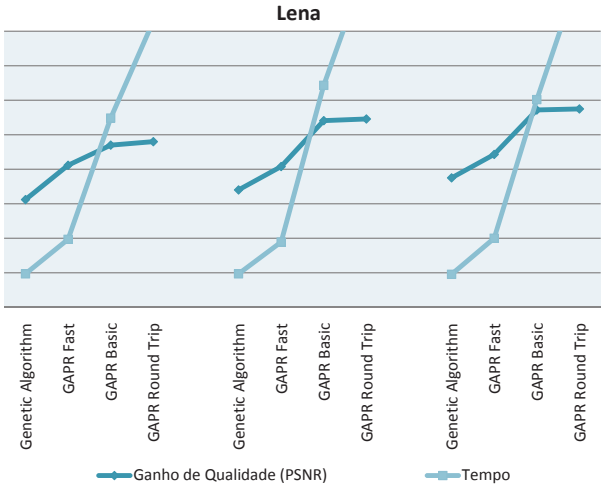


Figura 11. Comparando o ganho de qualidade e o tempo de processamento

adicionadas ao conjunto elite. Por isso que o *reverse path relinking* gasta menos tempo que o *path relinking*. Embora, o tempo seja menor, o reverse path relinking não apresenta a mesma eficiência no ganho de qualidade. Assim, o seu uso pode não ser compensatório.

5. Conclusão

O uso do *path relinking* e do *reverse path relinking* resultou em uma melhora sobre a heurística baseada nos algoritmos genéticos e na técnica de LSB. Entretanto, ela apresentou um custo elevado em relação ao tempo de processamento. Porém, como o principal objetivo era manter a informação secreta "invisível"para os atacantes, a qualidade das estego imagens foi priorizada.

Em todos os testes, a classe de algoritmos *GAPR* apresentou uma melhora sobre os algoritmos genéticos. Entretanto o ganho de qualidade foi similar a uma função linear, enquanto o tempo de processamento cresceu de maneira exponencial. Na verdade, a qualidade das soluções finais obtidas através do *path relinking* é fortemente influenciada pela capacidade do método principal de gerar soluções boas e diversas.

Um aspecto interessante do *GAPR* é a flexibilidade, permitindo diversas combinações dependendo da abordagem. Assim, o usuário pode decidir que algoritmo melhor se aplica no que ele deseja. Além disso, a classe de algoritmos *GAPR* pode ser utilizada em outros tipos de mídias.

Algumas configurações do algoritmo genético combinado com o *path relinking* foram testadas. Uma delas é a execução do *path relinking* ao final do ciclo do algoritmo genético. Esta combinação resultou em um aumento de qualidade, apesar do longo tempo de processamento. Já o *GAPR Fast* gerou estego imagens em um tempo de processamento um pouco mais longo que o algoritmo genético, porém o ganho de qualidade não foi tão grande quanto o *GAPR Round Trip*.

No futuro, pretendemos investigar a aplicabilidade de outras combinações utilizadas para resolver outros problemas [Vallada and Ruiz 2010, Zhang and Lai 2006]. Além de estudar os parâmetros para entender como eles determinam os efeitos sobre a solução

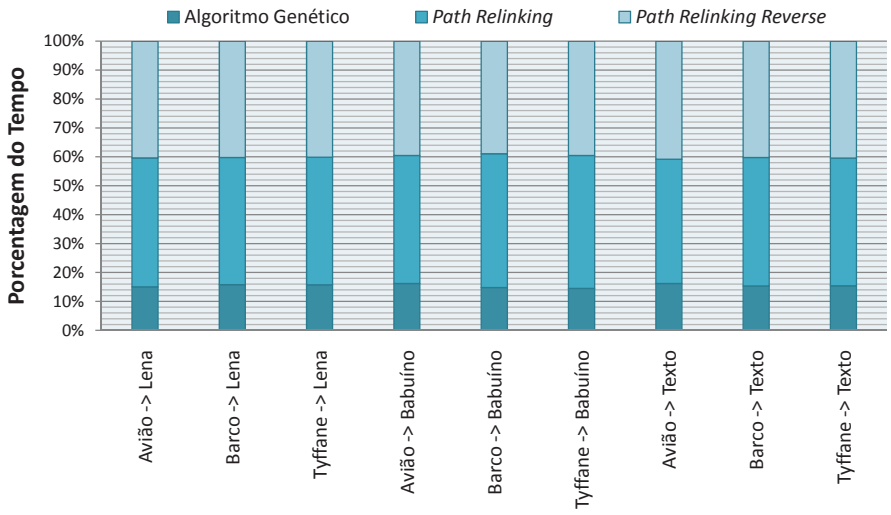


Figura 12. Percentual do total de soluções geradas em cada etapa do *GAPR Round Trip*

encontrada.

Outro trabalho futuro é a comparação dos resultados aqui obtidos com a técnica *Ant Colony Optimization* [Hsu and Tu 2010] que já está em andamento.

Referências

Anderson, R. and Petitcolas, F. (1998). On the limits of steganography. *Selected Areas in Communications, IEEE Journal on*, 16(4):474–481.

Bellare, M. and Rogaway, P. (1993). Entity authentication and key distribution. In *Advances in Cryptology-CRYPTO'93*, pages 232–249. Springer.

Bender, W., Gruhl, D., Morimoto, N., and Lu, A. (1996). Techniques for data hiding. *IBM Syst. J.*, 35(3-4):313–336.

Bourbakis, N. and Alexopoulos, C. (1992). Picture data encryption using scan patterns. *Pattern Recognition*, 25(6):567–581.

Chang, C.-C., Hsiao, J.-Y., and Chan, C.-S. (2003). Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy. *Pattern Recognition*, 36(7):1583 – 1595.

Duric, Z., Jacobs, M., and Jajodia, S. (2005). Information hiding: Steganography and steganoanalysis. *Pattern Recognition*, 24(6):171–187.

Glover, F. (1998). A template for scatter search and path relinking. In *AE '97: Selected Papers from the Third European Conference on Artificial Evolution*, pages 3–54, London, UK. Springer-Verlag.

Glover, F. and Laguna, F. (1997). *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA.

Highland, H. J. (1997). Data encryption: A non-mathematical approach. *Computers & Security*, 16(5):369 – 386.

Hsu, C. and Tu, S. (2010). Finding Optimal LSB Substitution Using Ant Colony Optimization Algorithm. In *2010 Second International Conference on Communication Software and Networks*, pages 293–297. IEEE.

Jan, J.-K. and Tseng, Y.-M. (1996). On the security of image encryption method. *Inf. Process. Lett.*, 60(5):261–265.

Johnson, N. and Jajodia, S. (1998). Exploring steganography: Seeing the unseen. *Computer*, 31(2):26–34.

Julio, E., Brazil, W., and Neves, C. (2007). Esteganografia e suas Aplicações. *Universidade Federal do Rio de Janeiro.(Org.). Livro Texto dos Minicursos-SBSEG 2007*, pages 54–102.

Petitcolas, F., Anderson, R., and Kuhn, M. (1999). Information hiding-a survey. *Proceedings of the IEEE*, 87(7):1062–1078.

Provos, N. and Honeyman, P. (2003). Hide and seek: an introduction to steganography. *Security & Privacy, IEEE*, 1(3):32–44.

Resende, M. and Ribeiro, C. (2007). An Introduction to GRASP. *XXXIX Simpósio Brasileiro de Pesquisa Operacional*.

Rhee, M. Y. (1993). *Cryptography and Secure Communications*. McGraw-Hill, Inc., New York, NY, USA.

Thien, C.-C. and Lin, J.-C. (2003). A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function. *Pattern Recognition*, 36(12):2875 – 2881.

Vallada, E. and Ruiz, R. (2010). Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem. *Omega*, 38(1-2):57 – 67.

Wang, R.-Z., Lin, C.-F., and Lin, J.-C. (2001). Image hiding by optimal lsb substitution and genetic algorithm. *Pattern Recognition*, 34(3):671 – 683.

Wang, S.-J. (2005). Steganography of capacity required using modulo operator for embedding secret image. *Applied Mathematics and Computation*, 164(1):99–116.

Yu, Y.-H., Chang, C.-C., and Lin, I.-C. (2007). A new steganographic method for color and grayscale image hiding. *Computer Vision and Image Understanding*, 107(3):183–194.

Zhang, G. and Lai, K. (2006). Combining path relinking and genetic algorithms for the multiple-level warehouse layout problem. *European Journal of Operational Research*, 169(2):413 – 425.