Expert Systems with Applications 40 (2013) 6693–6706

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Securing high resolution grayscale facial captures using a blockwise coevolutionary GA

Bassem S. Rabil *, Safa Tliba, Eric Granger, Robert Sabourin

Laboratoire d'Imagerie, de Vision, et d'Intelligence Artificielle, École de Technologie Supérieure, Université du Québec, Montreal, Canada

ARTICLE INFO

Keywords:
Biometrics
Intelligent watermarking
Evolutionary computation
Cooperative coevolution
Genetic algorithms
Grayscale texture masks

ABSTRACT

In biometric systems, reference facial images captured during enrollment are commonly secured using watermarking, where invisible watermark bits are embedded into these images. Evolutionary Computation (EC) is widely used to optimize embedding parameters in intelligent watermarking (IW) systems. Traditional IW methods represent all blocks of a cover image as candidate embedding solutions of EC algorithms, and suffer from premature convergence when dealing with high resolution grayscale facial images. For instance, the dimensionality of the optimization problem to process a 2048×1536 pixel grayscale facial image that embeds 1 bit per 8×8 pixel block involves 49k variables represented with 293k binary bits. Such Large-Scale Global Optimization problems cannot be decomposed into smaller independent ones because watermarking metrics are calculated for the entire image. In this paper, a Blockwise Coevolutionary Genetic Algorithm (BCGA) is proposed for high dimensional IW optimization of embedding parameters of high resolution images. BCGA is based on the cooperative coevolution between different candidate solutions at the block level, using a local Block Watermarking Metric (BWM). It is characterized by a novel elitism mechanism that is driven by local blockwise metrics, where the blocks with higher BWM values are selected to form higher global fitness candidate solutions. The crossover and mutation operators of BCGA are performed on block level. Experimental results on PUT face image database indicate a 17% improvement of fitness produced by BCGA compared to classical GA. Due to improved exploration capabilities, BCGA convergence is reached in fewer generations indicating an optimization speedup.

1. Introduction

In face recognition, high resolution facial capture images are commonly transferred and archived for various access control or surveillance applications. Since these images are vulnerable to unauthorized manipulations, watermarking is widely used to secure such images by embedding invisible watermark bits. The watermarking process should satisfy the trade-off between the distortion resulting from embedding the watermark (quality), and the resistance of the watermarked image to manipulations (robustness). The distortion resulting from the embedding is commonly measured using watermark quality metrics which is based on Human Vision System (HVS). The resistance of the watermarked image to manipulations is commonly measured using watermark robustness metrics, these metrics compare the original watermark with the extracted one from the manipulated watermarked image. Computational intelligence techniques are widely used to find the optimal watermark embedding parameters to satisfy the trade-off between watermark quality and robustness.

Different Evolutionary Computation (EC) techniques have been proposed in intelligent watermarking (IW) to find optimal embedding parameters. Authors have proposed using EC optimization techniques like Genetic Algorithms (GA) (Shieh, Huang, Wang, & Pan, 2004), Particle Swarm Optimization (PSO) (Wang, Sun, & Zhang, 2007), and hybrid GA-PSO (Lee, Lin, Su, & Lin, 2008) to find embedding parameters that maximize the fitness for both quality and robustness (Vellasques, Granger, & Sabourin, 2010). Most of these traditional methods represent all cover image using 8×8 pixels blocks in EC candidate solutions according to their positional order, and then iteratively improve the fitness until convergence is reached (Shieh et al., 2004). These methods often use single aggregated objective (Shieh et al., 2004; Vellasques, Sabourin, & Granger, 2012). To date, few authors have proposed mutli-objective formulation (Diaz & Romay, 2005; Rabil, Sabourin, & Granger, 2010), where the two objectives are optimized simultaneously, and multiple non-dominated solutions are located forming a Pareto front.

Traditional IW methods are efficient for low resolution images due to the limited search space corresponding to cover image blocks. Processing high resolution image captures results in

* Corresponding author. Tel.: +1 5143968800.
E-mail addresses: bguendy@livia.etsmtl.ca (B.S. Rabil), stliba@livia.etsmtl.ca (S. Tliba), Eric.Granger@etsmtl.ca (E. Granger), Robert.Sabourin@etsmtl.ca (R. Sabourin).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.06.043

complex optimization problems with a large search space. For instance, a facial image of resolution 2048×1536 and embedding capacity equal to 1 bit-per-block, the optimization problem has 49k real-valued variables represented by 293k binary bits. Most of EC optimization techniques suffer from convergence problems with this Large Scale Global Optimization (LSGO) problem due to representing all blocks in candidate solutions. LSGO is an optimization problem which is characterized by large number of decision variables resulting in huge search space for optimization solutions.

In an earlier work, Blockwise Multi-Resolution Clustering was proposed by the authors **(Rabil, Sabourin, & Granger, 2013) for rapid IW for streams of facial images. During the design phase of this framework, a limited number of face images are optimized, and the optimization results are stored in associative memory according to texture features. During generalization, streams of facial images are watermarked using the recalled optimal solutions from this associative memory. This framework provided a significant reduction of complexity up to 95.5% fitness evaluations compared to full optimization of all images in a homogeneous stream of high resolution facial images (Kasinski, Florek, & Schmidt, 2008). However the quality of solutions are dependent on the full optimization performed on images during design. Efficient optimization for high dimension problems would improve the quality of solutions produced by this proposed framework.

In this paper, specialized algorithm based on GA is proposed for LSGO applied on the optimization of embedding parameters in IW of high resolution images. To avoid embedding bits in smooth background that have low embedding capacity, only the textured blocks are considered as optimization candidate solution. The watermarking metrics are calculated globally on the whole image, however the local watermarking metrics for blocks can be used to guide the search through solutions. BCGA utilizes the cooperative coevolution between different solutions at the block level using the local watermarking metrics of these blocks. The embedding parameters of each block corresponding to higher local metrics are selected from different candidate solutions to construct new elite candidate solutions to be kept for next optimization generation. The crossover and mutation operators of BCGA are performed on block level rather than the entire candidate solution.

Proof of concept simulations are performed using the PUT database (Kasinski et al., 2008) of high resolution face images. This dataset provides homogeneous high resolution face images with smooth backgrounds. Reduced resolutions face images are experimented to identify the premature convergence limitation for traditional GA with higher resolution images. Then the original resolution of face images are used to evaluate the performance of the proposed approach. BCGA performance is compared against traditional GA and PBIL in terms of quality of solutions, and fitness evolution. Different fitness aggregating methods are evaluated to avoid anomalies with Pareto front for conflicting objectives of watermark quality and robustness. User-defined parameters of the proposed BCGA are tuned in the experimentation, and the impact of these parameters on BCGA fitness evolution is analyzed.

This paper is organized as follows. Section 2 presents a review of background on IW, LSGO, and texture metrics for grayscale images. Section 3 describes the proposed framework for IW of high resolution grayscale facial captures. The experimental methodology including PUT dataset (Kasinski et al., 2008) is described in Section 4. Finally results are presented and discussed in Section 5.

2. Intelligent watermarking of high resolution images

Digital watermarking is deployed in many domains to assure integrity and authenticity of the original signal via fragile and robust watermarking respectively (Vellasques et al., 2010). A fragile watermark is a type of watermark to ensure integrity, but it is broken if the watermarked image is manipulated or altered, while the robust watermark ensures authenticity and can be extracted after manipulating the watermarked image. Semi-fragile watermark considered in this paper is satisfying a trade-off between both the distortion introduced by the watermark and the watermark resistance to manipulations.

Most digital watermarking techniques proposed for grayscale images use different transform domains to embed a watermark that minimizes the visual impact, and to deal with the uncorrelated coefficients in the transform domain. The most commonly used transform domains in watermarking literature are Discrete Cosine Transform (DCT) (Shieh et al., 2004) and Discrete Wavelet Transform (DWT) (Lee et al., 2008). Using DCT transform inheriting robustness against JPEG compression which is based on DCT transform as well, the host image is divided into small blocks of pixels (8×8 pixels), transformed to frequency domain, and watermark bits are distributed among these blocks by changing frequency bands coefficients of these blocks according to the value of the watermark bit to be embedded. Few authors have considered other transforms based on DFT (Licks & Jordan, 2005) to improve robustness against geometric attacks since these transforms are more resistant to geometric manipulations.

2.1. Watermarking metrics

Digital watermarking system can be characterized using three main aspects: watermark quality, robustness, and capacity. Watermark quality measures the distortion resulting from watermark embedding. There are limits where the human vision cannot recognize the distortion resulting from the embedding (Voloshynovsky, Herrigel, & Baum, 1999). Watermark robustness seeks to assess the resistance to different manipulations and processing on the watermarked image. This can be measured by the correlation between the extracted watermark after the manipulations and the original watermark. Watermark capacity C measures the empirical number of embedded bits per block. This empirical capacity C is different than the maximum embedding bits calculated based on theoretical image processing calculations.

Watermark quality and robustness are commonly measured using Weighted Peak Signal-To-Noise Ratio (wPSNR) and Normalized Correlation (NC) respectively. Peak Signal-To-Noise Ratio (PSNR) is calculated between original image $X_c(w,h)$ and watermarked image $X_{cw}(w,h)$ of resolution $M_c \times N_c$ using the Mean Squared Error (MSE), where w, and h represents the index of pixels for width M_c and height N_c respectively:

$$MSE_c = \frac{1}{M_c.N_c}\sum_{w=1}^{M_c}\sum_{h=1}^{N_c}(X_c(w,h) - X_{cw}(w,h))^2$$
$$PSNR_c = 10log_{10}.\left(\frac{255^2}{MSE_c}\right)[dB] \quad (1)$$

Weighted PSNR uses an additional parameter called Noise Visibility Function (NVF) which is a texture masking function defined by Voloshynovsky et al. (1999). NVF arbitrarily uses a Gaussian model to estimate the amount of texture in any area. For smooth areas, NVF is equal to 1, and thus wPSNR has the same value of PSNR. For any other textured areas, wPSNR is higher than PSNR to reflect the fact that human eye will have less sensitivity to modifications in textured areas than smooth areas. Weighted PSNR shown in Eq. (2) is proposed in the latest benchmarking for watermarking systems introduced by Pereira, Voloshynovskiy, Madueno, Marchand-Maillet, and Pun (2001).

$$wPSNR_c = 10log_{10}.\left(\frac{255^2}{MSE_c \times NVF}\right)[dB] \quad (2)$$

The Normalized Correlation (NC) is calculated between embedded watermark $W(w,h)$ of resolution $M_W \times N_W$ where w and h represents the index of pixels for width and height respectively and the extracted watermark from the attacked image $W'(w,h)$:

$$NC = \frac{\sum_{w=1}^{M_W}\sum_{h=1}^{N_W}[W(w,h)W'(w,h)]}{\sum_{w=1}^{M_W}\sum_{h=1}^{N_W}[W(w,h)]^2} \tag{3}$$

2.2. Embedding and extracting watermarks

The watermark embedding/extracting algorithm considered in this paper is proposed by Shieh et al. (2004), where the original cover image is not required during extraction of the watermark. This reduces the memory required to store the original cover images. Using this algorithm, the cover image X_c of size M_cxN_c pixels to be watermarked is divided into 8×8 blocks and transformed into DCT domain where the resulting matrix $Y_{(m_c,n_c)}(a)$ for each image block at row m_c and column n_c of cover image blocks has the upper left corner as DC co-efficient. The rest of matrix are the AC coefficients, where the DCT coefficients index a ranging from 0 to 63 for 8×8 blocks are placed in zigzag order. The DCT transformed image $Y_{(m_c,n_c)}(a)$ is then used to get the ratio between DC and AC coefficients $R(a)$ using:

$$R(a) = \sum_{m_c=1}^{M_c/8}\sum_{n_c=1}^{N_c/8}\left(\frac{Y_{m_c,n_c}(0)}{Y_{m_c,n_c}(a)}\right), a \in [1,2,\ldots,63] \tag{4}$$

$$P_{(m_c,n_c)}(a) = \begin{cases} 1 & if(Y_{(m_c,n_c)}(a).R(a)) \geqslant Y_{(m_c,n_c)}(0) \\ & a \in \{eb_i\}, i = m_c \times (M_c/8) + n_c \\ 0 & \text{otherwise;} \end{cases} \tag{5}$$

Then polarities P are calculated using the Eq. (5). The indices of DCT coefficients modified belonging to $\{eb_i\}$ referred to as embedding bands for block b_i with i equals to $m_c \times (M_c/8) + n_c$.

Next, the watermarked DCT coefficient Y' is obtained using the Eq. (6). The embedding capacity for block b_i is defined as C_i in bits per block, and the watermark bits allocated for block at m_c row and n_c column $W_{(m_c,n_c)}(e)$, where e represents the index of set of embedding bands. The coefficient $Y_{(m_c,n_c)}(a)$ is not modified if the polarity $P_{(m_c,n_c)}(a)$ is equal to the watermark bit allocated $W_{(m_c,n_c)}(e)$, otherwise it is modified to values depending on the DC coefficient $Y_{(m_c,n_c)}(0)$ and the ratio of the coefficient a defined as $R(a)$. Finally the watermarked image X_{cw} is obtained using the inverse DCT for Y'. The watermark extraction follows the inverse operations of embedding algorithm.

$$Y'_{(m_c,n_c)}(a) = \begin{cases} Y_{(m_c,n_c)}(a) & \text{if } P_{(m_c,n_c)}(a) = W_{(m_c,n_c)}(e) \\ & a \in \{eb_i\},\ i = m_c \times (M_c/8) + n_c \\ (Y_{(m_c,n_c)}(0)/R(a)) + 1 & \text{if } P_{(m_c,n_c)}(a) = 0 \\ & W_{(m_c,n_c)}(e) = 1 \\ & a \in \{eb_i\}, i = m_c \times (M_c/8) + n_c \\ (Y_{(m_c,n_c)}(0)/R(a)) - 1 & \text{otherwise} \end{cases} \tag{6}$$

2.3. Optimization of embedding parameters

In traditional methods to optimize watermark embedding (Shieh et al., 2004), all cover image blocks are represented in optimization candidate solutions, and the selected embedding bands are altered iteratively during optimization to maximize the aggregated fitness $gf(EB_{X_c})$. This aggregated fitness includes watermark Quality Fitness (QF) and Robustness Fitness (RF) against different attacks. All cover image blocks have to be represented as optimization candidate solutions to allow distribution of watermark bits among cover image blocks. The optimization problem can be formalized as the following maximization problem:

$$\begin{aligned} &\max_{EB_{X_c}} \{gf(EB_{X_c})\} \\ &EB_{X_c} = \{eb_1, eb_2, \ldots, eb_i, \ldots, eb_{NB}\}, \quad \text{where } NB = (M_c/8) \times (N_c/8) \\ &eb_i = \{a_1, a_2, \ldots, a_e, \ldots, a_{C_i}\}, \quad \text{where } a_e \in [0,1,\ldots,63] \\ &\text{s.t. } a_e \neq 0, \quad \text{where } 1 < e < C_i, \text{ and } 1 < i < NB \\ &a_{e1} \neq a_{e2}, \quad \text{where } 1 < e1, e2 < C_i \end{aligned} \tag{7}$$

where b_i represents the 8×8 block in cover image of resolution M_c - $\times N_c$, the total number of blocks equals to NB, a_e represents the eth embedding band for block b_i, and the embedding capacity for block b_i is C_i. The first constraint considered ensures avoiding DC coefficient a_e for embedding, and the second constraint considered ensures avoiding using the same embedding bands for the same image block.

The flow chart of the IW optimization is shown in Fig. 1. Assume that watermark bits W are embedded in face image X_c using the embedding bands EB_{X_c} to produce the watermarked image X_{cw}. The quality fitness $QF(EB_{X_c})$ between the original face image X_c, and the watermarked image X_{cw} is assessed. Then different attacks are applied to X_{cw}, then watermark bits are extracted W'. The extracted watermark W' is compared against original watermark W to calculate robustness fitness $RF_1(EB_{X_c})$, $RF_2(EB_{X_c})$, and $RF_3(EB_{X_c})$ against different attacks. In this formulation quality fitness $QF(EB_{X_c})$ and different robustness fitness $RF_1(EB_{X_c}), RF_2(EB_{X_c})$, and $RF_3(EB_{X_c})$ are optimized as a single objective optimization using aggregation. The aggregated fitness is improved iteratively till a stopping criterion is reached, and the sets of optimal embedding bands eb_i for all blocks b_i are concluded EB_{X_c} for face image X_c.

2.4. Large-scale global optimization problems

Many real-world applications correspond to large scale global optimization (LSGO) problems to find optimal decision variable values in large search space. Authors in many domains have explored different strategies to handle such challenging optimization problems (Shan & Wang, 2010). Screening strategy involves removing noises and insignificant variables from the large problem by exploring the nature of the optimization problem. Mapping strategy reduces the search space by removing correlated variables and thus reducing the complexity of the large problem. And the most commonly used strategy used by authors is decomposing the large problem into smaller sub-problems, different decomposition and interaction methods have been proposed to be used among these sub-problems. In this paper, decomposing and screening strategies are considered for handling optimization problem corresponding to IW of high resolution facial images. These two strategies are more suitable for the nature of IW problem.

EC methods are not efficient for large scale problems, this attracted the attention of authors in EC literature to improve the capabilities of these methods for large dimension of search space. In recent years, there has been a growing interest in methods that learn the structure of a problem on the fly and use this information to ensure an adequate evolution of the different variables of the problem. One approach is based on probabilistic modeling of promising solutions to guide the exploration of the search space instead of using crossover and mutation, as in the case of simple GA (Pelikan, Goldberg, & Lobo, 2002).

In the Population Based Incremental Learning algorithm (PBIL) (Baluja, 1994), solutions are represented by binary strings of fixed length. The population of solutions is replaced by a probability vector which is initially set to the same probability 0.5 for all positions. After generating a number of solutions, the best solu-

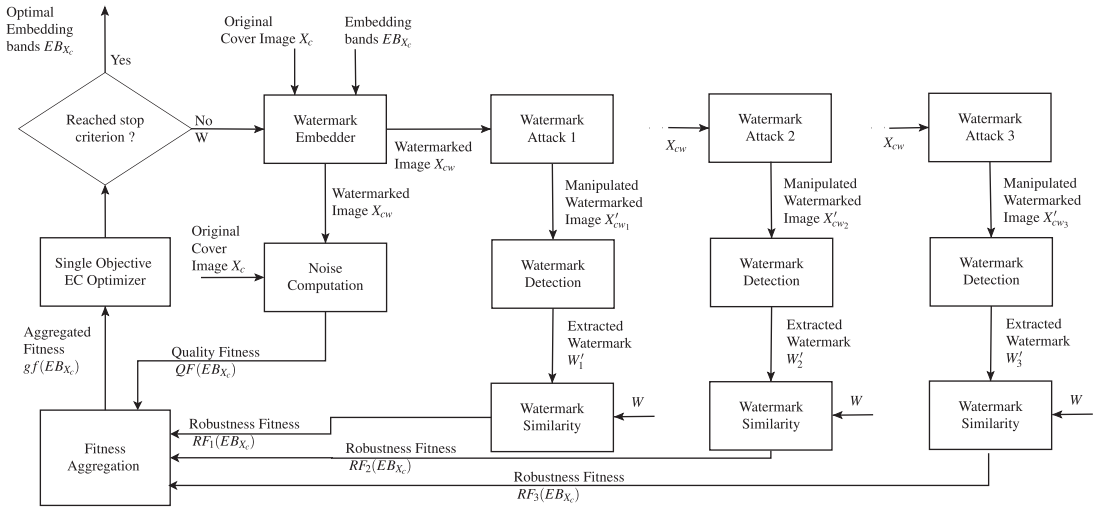


Fig. 1. Data flow diagram depicting the formulation for watermark embedding optimization to find optimal embedding bands EB_{X_c} for cover image X_c.

tions are selected, and the probability vector is shifted to the chosen solutions. The probability vector of PBIL evolves throughout generations using the previously calculated fitness. The Compact Genetic Algorithm (CGA) (Harik, Lobo, & Goldberg, 1999) replaces the population with a unique probability vector similar to PBIL. However, unlike the PBIL, it modifies the probability vector so that there is a direct correspondence between the population that is represented by the vector of probabilities and the probability vector itself. Each component of the vector is updated by replacing its value by the contribution of a single individual in the total frequency assuming a particular population size. Using this update rule, the theory of simple genetic algorithms can be used directly to estimate the parameters and behavior of the compact genetic algorithm.

Previous methods based on learning the structure of the problem is not efficient for problems of high dimension search space like the IW problem for high resolution face images. Thus such problems would involve other strategy for decomposing this large problem into smaller sub-components interacting together to find optimal overall solutions. Recently authors adopted coevolution strategy for handling such optimization problems.

In biology, coevolution is defined as a change of a biological object triggered by the change of a related object. In bio-inspired EC techniques this concept is adopted either this coevolution is cooperative or competitive. In cooperative coevolution algorithms (Potter & DeJong, 2000), the goal is to find individuals from which solutions can be better. The adaptability of the individual depends on its ability to cooperate with individuals of other species to solve a given problem. On the other side for competitive coevolution (Rosin & Belew, 1997), the adaptability of the individual depends on the competition with other individuals of other species, where each of these species compete with the rest of species.

Cooperative Coevolution (CC) adopts natural divide-and-conquer strategy, recently this approach is considered a promising solution for handling high-dimensional optimization problems. The main idea of cooperative coevolution based algorithms is to identify which decision variables, i.e, dimensions, of the search space interact. Non-interacting variables can be optimized as separate problems of lower dimensionality. Interacting variables must be grouped together and optimized jointly (Chen, Weise, Yang, & Tang, 2010). By cooperatively coevolving multiple EA subpopulations (each dealing with a subproblem of a lower dimensionality), we can obtain an overall solution derived from combinations of subsolutions, which are evolved from individual subpopulations.

Based on this principle, Potter and DeJong (2000) proposed a Cooperative Coevolutionary Genetic Algorithm (CCGA), which shares the search space by dividing the solution vector into smaller vectors. Potter and DeJong (2000) proved that CCGA has a significant performance improvement over traditional genetic algorithms for optimization problems of 30 variables. Subsequently, Sofge, Jong, and Schultz (2002) extended the model of Potter and DeJong (2000) to algorithms of evolution strategy (CCES).

den Bergh and Engelbrecht (2004) adopted the cooperative coevolution concept with PSO (CCPSO), this cooperative model was tested on functions of up to 30 variables (den Bergh & Engelbrecht, 2004) and 190 variables (den Bergh, 2002). All previous algorithms based on cooperative co-evolution adopts two simple strategies for problem decomposition. The first strategy decomposes a high-dimensional vector into single variables, this means an n-dimensional problem would be decomposed into n one-dimensional problems. This strategy is simple but it did not consider interdependencies among variables for non separable problems. The second strategy is splitting-in-half which decomposes a high-dimensional vector into two equal halves and thus reducing an n dimensional problem into two $\frac{n}{2}$-dimensional problems. This can be extended to recursively decompose into halves for larger dimension problems.

Yang, Tang, and Yao (2008) proposed grouping strategy and adaptive weighting for better capturing of the variable interdependencies for non separable problems. Li and Yao (2012) presented a new Cooperative Coevolving PSO (CCPSO) using random variable grouping technique. This proved efficiency with problems with dimensions up to 2000 real valued variables using the benchmarking functions provided in Congress on Evolutionary Computation (CEC'10) (Tang, Li, Suganthan, Yang, & Weise, 2009).

For optimization problems corresponding to IW of high resolution facial images considered in this paper, the number of variables is equal to 49k variables. These embedding variables for different blocks affect the overall watermarking fitness for the facial image, however the local watermarking metrics of these blocks can guide the search in such large search space. Such dimension of optimization problems was not considered before in literature, and thus an application specific algorithm is proposed in this paper to handle such problem using the characteristics of the IW problem.

2.5. Grayscale texture metrics

Adaptive watermarking systems proposed in literature model the Human Vision System to measure the level of tolerable distortion in an image. Accordingly the insertion of the mark is adjusted

to these levels of distortion in an image to maximize robustness and quality. These systems involve ranking cover image blocks according to texture. The most textured blocks are used for embedding the watermark, thus the embedding parameters and regions are adaptive to the image content. This section describes the most commonly used grayscale texture metrics which are considered in this paper. These metrics are used as perceptual masks to select embedding blocks.

Entropy is a scalar value of grayscale image X_c of resolution $M_c \times N_c$ representing the energy. Entropy is a statistical measure of randomness that can be used to characterize the texture of the whole image or blocks. For 8×8 blocks, the image X_c is divided into $\frac{M_c}{8} \times \frac{N_c}{8}$ blocks whose row and column indices defined as m_c, and n_c respectively. The entropy of a block b_i is defined as E_{b_i}:

$$E_{b_i} = \sum_{w,h \in b_i} SP(w,h).logSP(w,h) \tag{8}$$

where $SP(w,h)$ is the normalized spectrum of 8×8 pixels blocks with w and h representing the index of pixels for width and height respectively. Blocks of higher entropy value E_{b_i} represents more textured blocks, smooth textured blocks have lower entropy E_{b_i} value. Entropy considers only one aspect of the human visual system, and does not exploit other features such as contrast and structure elements of the image.

Noise Visibility Function (NVF) uses a general concept inspired by denoising. Voloshynovsky et al. (1999) have proposed different functions for the definition of NVF based on the function of the image denoising and the statistical properties of the image. In the field of image processing, the variance is used to gain information about the local activity in the image. Its small values indicate smooth regions, and large values indicate the presence of edges or highly textured regions. Using the local variance of pixels in block σ_x^2, NVF can be defined as:

$$NVF = \frac{1}{1+\sigma_x^2} \tag{9}$$

Visual models derived in compression techniques are perfectly adapted to the problem of watermarking. A common paradigm for perceptual masking is derived from the calculation of an image mask which depends on JND used in compression applications. Such a model can be directly extended to digital watermarking applications by supplying information on the ability of inserting a block of the image with respect to another, this ability is related to the degree of noise that can undergo this block that ensures transparency while providing a robust watermarking.

In an earlier paper by the authors Rabil et al. (2013), Robustness Scores (RS) was proposed as an adaptive texture metric for grayscale images. RS is well suited for full uneven embedding scheme, where it groups image blocks according to their texture features and use different embedding capacity for each group of blocks. RS is calculated for different number of blocks groups using robustness fitness against JPEG. It is calculated for embedding 1 bit-per-block in all blocks belonging to the same group measured using NC defined in Section 2.1.

3. Intelligent watermarking of high resolution grayscale facial images

The optimization of embedding parameters for high resolution images shown in Fig. 2 has high dimension of search space. It starts with representing only textured blocks TB in the candidate solution using perceptual texture masks, then an application specific GA algorithm called BCGA is used for embedding parameters optimization. It utilizes cooperative coevolution for local watermarking metrics of blocks, and performs GA operators crossover and mutation on block level. This coevolution on the block level improves GA algorithm capabilities for large optimization problems using cooperation of smaller problems.

The perceptual texture mask produces the mask for embedding ME_{X_c} for the face image X_c, where $ME_{X_c} = \{me_1, me_2, \ldots, me_i, \ldots, me_{NB}\}$, and $me_i \in \{0,1\}$. All face image X_c blocks b_i are represented in the mask for embedding, where $me_i = 1$ indicates that the block b_i is selected for embedding based on texture criteria defined by the perceptual texture mask. And thus the total number of blocks whose $me_i = 1$ is equal to the number of textured blocks selected for embedding TB, where $TB = \sum_{i=1}^{NB} me_i$. Only the blocks of me_i equals to 1 are represented in the candidate solution for optimization, this mapping ensures optimizing the embedding parameters in textured blocks only. For even embedding scheme with equal embedding capacity for all blocks, all blocks are represented in the candidate solutions for embedding. The proposed chromosome representation is shown in Fig. 3.

Using EC methods, the optimization process relies on a population of candidate solutions traversing search space to find the optimal solution. The population of candidate solutions S_p consists of pop solutions. Each candidate solution S_p consists of the sets of

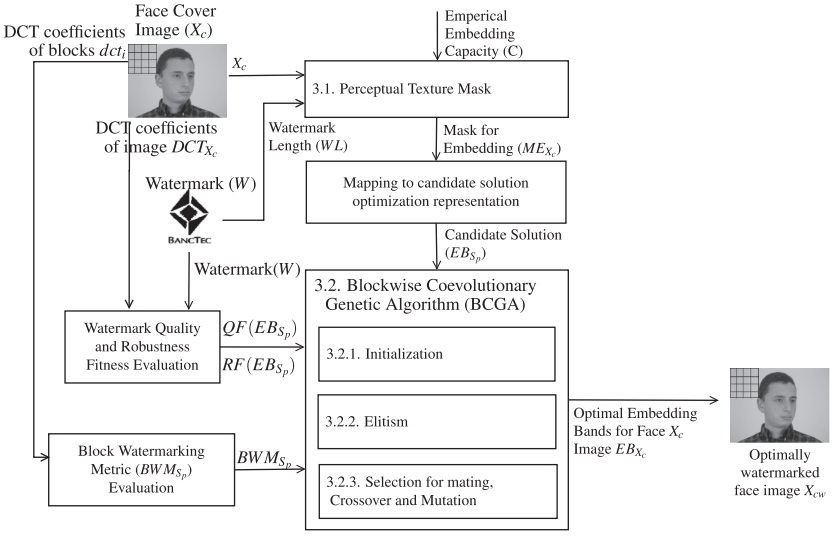


Fig. 2. Block diagram for the proposed watermarking system for high resolution face images.

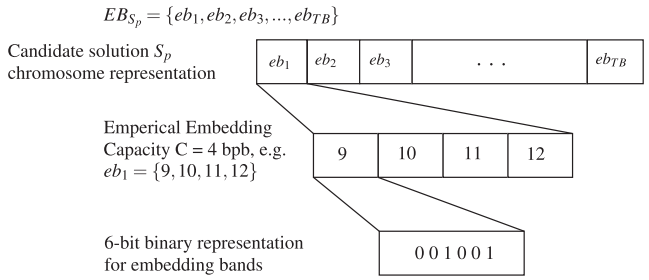


Fig. 3. Chromosome representation of the proposed BCGA.

embedding bands for different blocks b_i, where i ranges from 1 to TB, and thus $EB_{S_p} = \{eb_1, eb_2, eb_3, \ldots, eb_{TB}\}$. The size of the set eb_i equals to empirical embedding capacity C. And thus the dimension of the optimization problem to find the optimal embedding bands equals to $TB \times C \times 6$ bits, where TB is the number of textured blocks selected by perceptual texture mask, C is the empirical embedding capacity, and 6-bits are used to represent embedding bands ranging from 0 to 63 for blocks of size 8×8 pixels. The candidate solution of the best global fitness EB_{S_p} is selected as the optimal solution EB_{X_c} for face image X_c, representing the optimal embedding bands for this face image such that $gf(EB_{X_c}) = max_{p=1 \to pop} gf(EB_{S_p})$.

The global fitness to be optimized $gf(EB_{X_c})$ is the aggregated fitness for the whole face image X_c using both watermark quality and robustness against different attacks. For each candidate solution EB_{S_p}, block watermarking metric bwm_i are calculated for each block b_i using the embedding bands eb_i defined in EB_{S_p}. This bwm_i metric is a local aggregated fitness for metrics representing both watermark quality and robustness against different attacks using weighted sum aggregation. It is utilized in cooperative coevolution on block level for GA, where the blocks of higher bwm_i are assumed to form better candidate solutions EB_{S_p} having better global fitness $gf(EB_{S_p})$. The set of BWM for candidate solution S_p is defined as BWM_{S_p}, where $BWM_{S_p} = \{bwm_1, bwm_2, \ldots, bwm_{TB}\}$.

3.1. Perceptual texture masks

Biometric high resolution face images are characterized by having areas of smooth textures representing the background, and the face image for individuals with more textured areas. Embedding watermarks in smooth textured areas results in degradation for watermark fitness (Wu, 2001). Traditional IW methods (Shieh et al., 2004) relies on embedding bits in all cover image blocks, and thus these are not efficient with high resolution face images with smooth backgrounds. Using these traditional methods, handling watermarks of small lengths would involve padding these watermarks bits to represent all cover images in candidate solutions. And thus the minimum search space for these methods is equal to the number of cover image blocks.

The perceptual texture mask component selects the most textured blocks as blocks of interest for embedding the watermark. It utilizes texture metrics to find the most textured blocks TB. In this paper, the number of textured blocks TB is calculated based on the watermark length WL, and the empirical embedding capacity C. The embedding capacity is assumed to be equal for all textured blocks, and thus the number of textured blocks equals to WL/C.

3.2. Blockwise coevolutionary genetic algorithm (BCGA)

The proposed BCGA is an application specific optimization algorithm derived from GA. It utilizes cooperative coevolution for

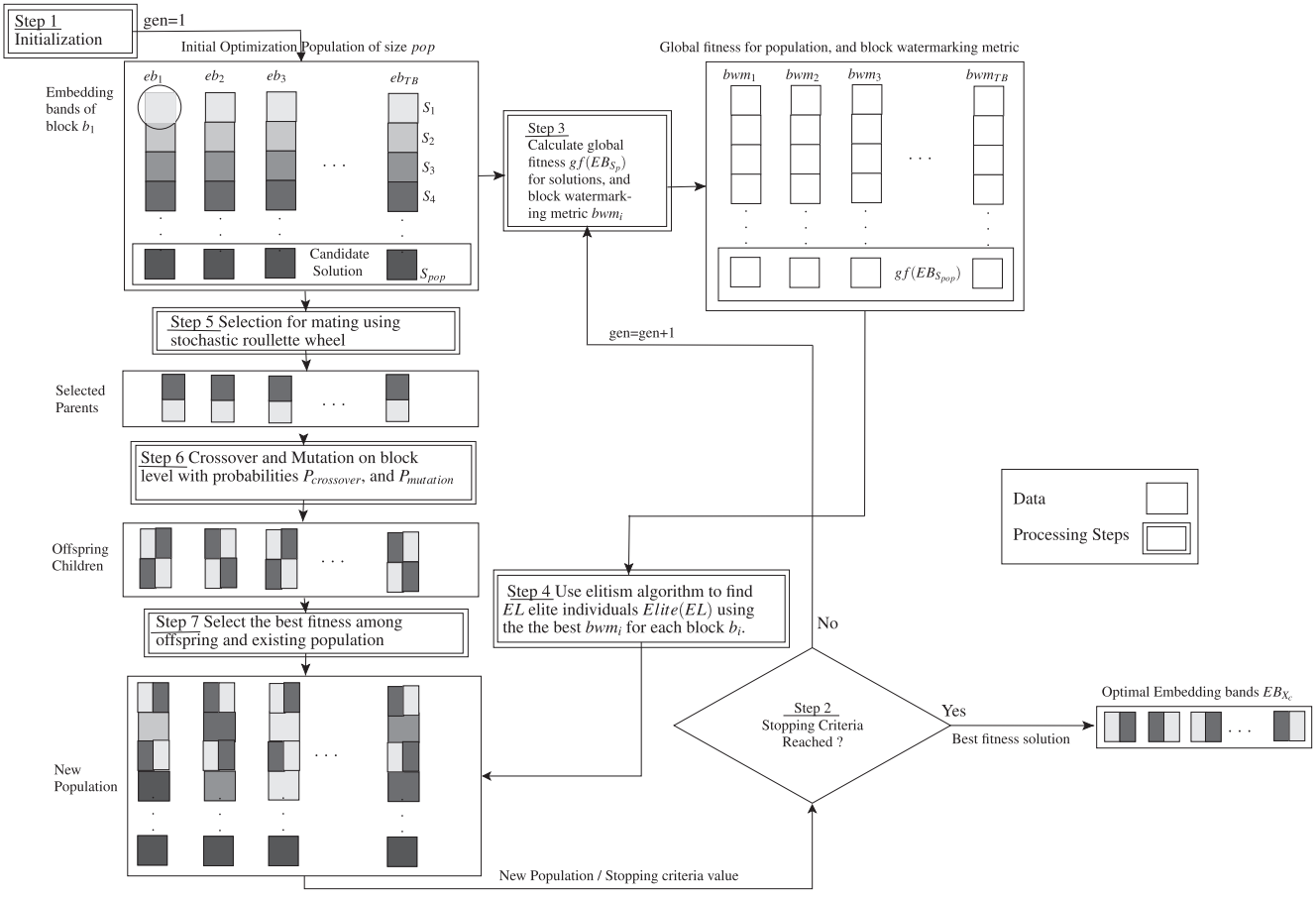


Fig. 4. Data flow diagram for the BCGA to find optimal embedding bands EB_{X_c} for image X_c.

subcomponents represented by 8×8 pixels blocks. Local watermarking metrics are used for these blocks to conclude the best fitness optimization candidate solutions using the best fitness subcomponent. The BWM of block b_i is defined as $bwm_i(eb_i)$, where eb_i are the embedding bands of block b_i, and the global fitness of the whole face image using embedding bands defined in candidate solution S_p is defined as $gf(EB_{S_p})$, where EB_{S_p} are the embedding bands of all blocks of X_c for this solution. The global fitness $gf(EB_{S_p})$ is the aggregated fitness for watermark quality fitness and watermark robustness fitness against different attacks.

All the candidate solutions in the population are used to find *Elite* solutions, the number of these elite solutions is defined as *EL*. These *Elite* solutions are defined as $Elite = \{EB_{S_{p1}}, EB_{S_{p2}}, \ldots, EB_{S_{pEL}}\}$, where *Elite*(*el*) refers to the elite candidate solution of index *el* in the elite set. This set of solutions *Elite* are used in elitism algorithm on the block level using bwm_i, where the embedding bands for blocks corresponding to the highest bwm_i are concatenated to generate the blockwise elite solutions. Crossover is performed on the selected parents of highest $gf(EB_{S_p})$ on the block level using two-points crossover with probability $P_{crossover}$, then mutation is applied on block level as well by changing random bits of probability $P_{mutation}$. The fitness of the new offspring generated is calculated, where the candidate solutions of least fitness are replaced with the most fit new offspring. The detailed data flow for BCGA to find optimal embedding bands EB_{X_c} for face image X_c is shown in Fig. 4 and Algorithm 1.

Algorithm 1. Finding optimal embedding bands EB_{X_c} for face image X_c using BCGA.

Input: Face image X_c blocks b_i, crossover probability $P_{crossover}$, mutation probability $P_{mutation}$, and number of elite solutions in population *EL*.

1: Initialize population of candidate solutions of size *pop*. The dimension of each candidate solution in this population equals to $TB \times C \times 6$ bits. Algorithm 2 shows the details of initialization process.
2: **while** Stopping criteria is not reached **do**
3: Evaluate global fitness $gf(EB_{S_p})$ for the whole candidate solution S_p for all solutions in the population, and BWM for blocks BWM_{S_p}.
4: The blocks are sorted using the metric $bwm_i(eb_i)$ as shown in Algorithm 3. The embedding bands corresponding to highest bwm_i are assumed to have good global fitness $gf(EB_{S_p})$ when concatenated to form new elite solution *Elite*(*el*).
5: Select best fitness individuals for reproduction using stochastic roulette wheel bias method.
6: Breed new candidate solutions through crossover and mutation on the block level with probabilities $P_{crossover}$, and $P_{mutation}$ respectively to give birth to offspring candidate solutions.
7: Evaluate the global fitness of new offspring candidate solutions and replace the least fitness candidate solutions of the population with the best fitness offspring, then insert *EL* elite solutions from step 4 in the population.
8: **end while**

Output: Optimal embedding bands EB_{X_c} for face image X_c

The objectives are proposed to be aggregated using Chebyshev weighted aggregation (Vellasques et al., 2012). This aggregation method is more robust to anomalies in the trade-off between the various fitness functions in a multi-objective optimization problem. In the Chebyshev approach, fitness values are aggregated according to their distances from reference points, under which the values of these fitness values are considered good (Collette & Siarry, 2008) as shown in Eq. (10).

$$\begin{aligned} gf(EB_{X_c}) = max_{l=1,2,3,4}\{&(1-w_1)(\gamma.QF(EB_{X_c}) - r_1), (1-w_2) \\ &\times (RF_1(EB_{X_c}) - r_2),\ (1-w_3)(RF_2(EB_{X_c}) - r_3), \\ &\times (1-w_4)(RF_3(EB_{X_c}) - r_4)\} \end{aligned} \tag{10}$$

where $(1-w_l)$ is the weight of the l^{th} objective with $(1-w_l) = \frac{1}{4} \forall l$, r_l is the reference point of objective l, and γ is a scaling weight for quality fitness $QF(EB_{X_c})$

The proposed local watermarking metrics are Peak Signal To Noise Ratio (PSNR) and Bit Correct Rate (BCR), where these metrics are the least complex metrics applicable to block level. The aggregated weighted sum $bwm_i(eb_i)$ for block b_i is defined in Eq. (11), where $PSNR_i(eb_i)$ is the peak signal to noise ratio for block b_i, $BCR_{i1}(eb_i)$ is the bit correct ratio for first attack on block b_i, $BCR_{i2}(eb_i)$ is the bit correct ratio for second attack on block b_i, and $BCR_{i3}(eb_i)$ is the bit correct ratio for third attack on block b_i.

$$\begin{aligned} bwm_i(eb_i) = PSNR_i(eb_i) + \lambda(&BCR_{i1}(eb_i) + BCR_{i2}(eb_i) \\ &+ BCR_{i3}(eb_i)) \end{aligned} \tag{11}$$

where PSNR is defined in Eq. (2) and BCR measures the ratio of correctly extracted bits to total embedded bits as shown in Eq. (12), where BC_i represents the number of correctly extracted watermark bits from block b_i, and C_i is the total number of watermark bits embedded in this block representing the empirical embedding capacity in bits per block.

$$BCR_i = BC_i / C_i \tag{12}$$

Algorithm 1 shows the main algorithm of BCGA to find optimal embedding bands EB_{X_c} for face image X_c. The population of candidate solutions is initialized (line 1) using Algorithm 2. Then the global fitness, and BWM are calculated for all candidate solution in the population (line 3). Using bwm_i, the blocks are sorted (line 4) as shown in Algorithm 3, where the blocks of higher local fitness from different candidate solutions are concatenated to generate new candidate solutions. The parent candidate solutions used for reproduction are selected using stochastic roulette scheme (line 5). Crossover and mutation operators use masks on block level to perform these operations to breed new candidate solutions (line 6) as shown in Algorithm 4. The global fitness of the new offspring is calculated (line 6), and the candidate solutions of least fitness from the population are replaced with the new offspring, then elite solutions *Elite*(*el*) are inserted in the population (line 7). Until stopping criteria is reached (line 2), global fitness is improving iteratively along generations.

3.2.1. Initialization

Algorithm 2 shows the initialization process for the population of candidate solutions. The embedding bands for all blocks in the same candidate solution S_p are the same at initialization time. For the first method (lines 1–7) applied on the first portion of the population of candidate solutions, the initial bands are chosen based on the index of candidate solution *p*, index of embedding bands *j*, and embedding capacity *C*. The embedding bands set is initialized to an empty set (line 2), then it is concatenated with different initial embedding bands (line 3–5) until the embedding capacity *C* is reached. This set of embedding bands eb_i is replicated *TB* times (line 6) to form the initial candidate solution S_p. For the second method (lines 9–16) applied on the second portion of the population, the initial bands are chosen using random integers ranging from 1 to 63-*C*, and index of embedding bands *j*. For populations of smaller size, only the first method of initialization is used. For example when using embedding capacity $C = 4$ for a population of size $pop = 24$, the first method of initialization is applied on candidate solutions S_p, where $p = 1, 2, \ldots, 14$, and the second method is applied on $p = 15, 16, \ldots, 24$. This initialization algorithm ensures high diversity in the initial population of candidate solutions.

Algorithm 2. Initialization of the proposed BCGA, where *repeat*(*V*,*r*) is a function that construct a vector by replicating *r* times the vector *V*.

Input: Candidate solution index *p*, population size *pop*, number of textured blocks for embedding *TB*, index of embedding bands *j* in embedding bands set eb_i for block b_i, and embedding capacity *C*

1: **for** $p = 1$ to $(64/C)$-1 **do**
2: Initialize empty set for embedding bands $eb_i = \{\}$.
3: **for** $j = 1$ to C **do**
4: $eb_i = \{eb_i(p-1)*C + j\}$ #uniform empirical embedding bands based on *p*, and *C* for block b_i.
5: **end for**
6: $EB_{S_p} = repeat(eb_i, TB)$;
7: **end for**
8: **if** $pop > (64/C) - 1$ **then**
9: **for** $p = (64/C) - 1$ to *pop* **do**
10: Generate random variable *rand* of integer value between [1 63-*C*]
11: Initialize empty set for embedding bands $eb_i = \{\}$.
12: **for** $j = 1$ to C **do**
13: $eb_i = \{eb_i\ rand + j\}$ #random consecutive embedding bands initialization for block b_i.
14: **end for**
15: $EB_{S_p} = repeat(eb_i, TB)$;
16: **end for**
17: **end if**

Output: Population of initial candidate solutions EB_{S_p}, where $p = 1,2,..,pop$

3.2.2. Elitism

As shown in Fig. 5, the sets BWM_{S_p} are calculated for all candidate solutions in the optimization population of size *pop*. For each block b_i, the corresponding bwm_i are sorted descendingly. The blocks of highest bwm_i are concatenated together to generate the elite solution *Elite*(1), and the second best blocks are concatenated together to generate the second elite solution *Elite*(2).

Algorithm 3. Elitism based on local fitness of BCGA.

Input: Candidate solutions S_p, where *p* is the index of candidate solution, BWM_{S_p} set of BWM for candidate solution S_p, where it consists of bwm_i for block b_i, textured blocks *TB*, index of elite candidate solution *el*, and number of elite solutions *EL*.

1: **for** $p = 1$ to *pop* **do**
2: **for** $i = 1$ to *TB* **do**
3: Calculate BWM_{S_p} of candidate solution S_p of index *p*, where each BWM_{S_p} consists of bwm_i for all blocks b_i.
4: **end for**
5: **end for**
6: **for** $i = 1$ to *TB* **do**
7: Sort the blocks descendingly for each block b_i to find the indices *p* of the top values of bwm_i in BWM_{S_p}.
8: **end for**
9: **for** $el = 1$ to *EL* **do**
10: Concatenate the embedding bands eb_i of all blocks corresponding to the top values bwm_i from BWM_{S_p} of different candidate solutions S_p to generate elite candidate solutions *Elite*(*el*) using the indices in Step 7.
11: **end for**

Output: *EL* elite candidate solutions of index *el* defined as *Elite*(*el*).

Algorithm 3 shows the proposed algorithm of elitism. The metric bwm_i is calculated for the blocks of all candidate solutions (line 1–5), where this metric represents a measure to rank blocks based on watermarking fitness. For each block b_i, the values of bwm_i are sorted to find the indices *p* of top values of bwm_i (line 6–8). The embedding bands corresponding to the top values of bwm_i are concatenated together to generate elite individuals *Elite*(*el*) (line 9–11), such that the embedding bands corresponding to the best values are used to generate *Elite*(1), and the second best values used to generate *Elite*(2).

3.2.3. Selection for mating, crossover and mutation

The proposed selection for BCGA to find candidate solutions for crossover and mating is based on roulette wheel method. The global fitness of candidate solutions EB_{S_p} is normalized, then the cumulative normalized fitness values are stored in an array where the last fitness in this array is equal to 1. This method is used to select the parent candidate solutions $EB_{S_{p11}}$, and $EB_{S_{p22}}$ for crossover and mutation to generate the new offspring.

As shown in Fig. 6, the crossover is performed on block level using 2-point crossover. The embedding capacity $C = 4$, yielding to a size of 24 binary bits for embedding bands for each block eb_i. For the example shown in the figure the 2-point crossover results in 3 equal portions of 8 bits to be crossovered. A mask for crossover of each block cm_i is generated to be concatenated to form the crossover mask for the whole solution *CM*. The whole crossover mask is applied on the whole solution to obtain the crossovered solutions. Then mutation is applied similarly on block level to obtain mutated crossovered solutions.

Algorithm 4. Proposed crossover and mutation operators on the block level of BCGA.

Input: Candidate solution indices *p*11 and *p*22 for selected parents, number of textured blocks for embedding *TB*, embedding capacity *C*, probabilities of crossover and mutation defined as $P_{crossover}$ and $P_{mutation}$ respectively, and number of crossover points *cp*.

1: **for** i = 1 to *TB* **do**
2: Generate crossover mask cm_i for block b_i for selected parents candidate solutions $EB_{S_{p11}}$, and $EB_{S_{p22}}$ with probability $P_{crossover}$ based on *cp*.
3: **end for**
4: Concatenate block masks cm_i to obtain candidate solution crossover mask *CM*
5: Apply crossover mask *CM* on the parent candidate solutions $Eb_{S_{p11}}$, and $EB_{S_{p22}}$ to obtain crossovered solutions S_{p21}, and S_{p12}
6: **for** i = 1 to *TB* **do**
7: Generate mutation mask mm_i for block b_i for the crossovered solutions $EB_{S_{p21}}$, and $EB_{S_{p12}}$ with probability $P_{mutation}$.
8: **end for**
9: Concatenate block masks mm_i to obtain candidate solution mutation mask *MM*
10: Apply mutation mask *MM* on the crossovered solutions $EB_{S_{p21}}$, and $EB_{S_{p12}}$ to obtain mutated crossovered solutions of indices *p*12*m*, and *p*21*m*.

Output: New offspring candidate solutions of indices *p*12*m*, and *p*21*m*

Algorithm 4 shows the crossover and mutation operators used to generate new offspring candidate solutions using selected parent solutions $EB_{S_{p11}}$, and $EB_{S_{p22}}$. The crossover is performed on block

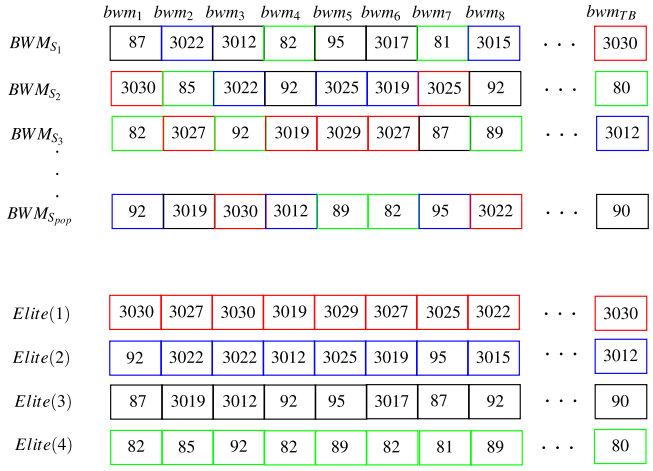


Fig. 5. Proposed elitism on block level with $EL = 4$ using block watermarking metric bwm_i.

level (line 1–5), where the crossover mask cm_i is generated for blocks (line 1–3) using the number of crossover points cp with probability $P_{crossover}$. The block crossover masks cm_i are concatenated to form the whole candidate solution crossover mask CM (line 4). This mask CM is applied to the selected parents to obtain crossovered solutions $EB_{S_{p21}}$, and $EB_{S_{p12}}$ (line 5). Also mutation is performed on block level (line 6–10), where this starts with generating mutation masks for blocks mm_i (line 6–8) with probability $P_{mutation}$ to change bit values. These block masks are concatenated to generate the whole candidate solution mask MM (line 9). This mask MM is applied to crossovered solutions to obtain the mutated crossovered solutions $EB_{S_{p21m}}$, and $EB_{S_{p12m}}$ (line 10).

4. Experimental methodology

The database for face images used in experiments is the PUT (Kasinski et al., 2008) face database which consists of 100 individuals where images feature 100 poses for each individual. Sample face images of the first pose of individuals are shown in Fig. 7. The face images of this database are characterized by a smooth background for all individuals, and different light conditions and clothes. The percentage of background and foreground blocks are nearly the same for each individual, where the different poses represent head rotation and facial expression changes only. This ensures texture homogenity for facial images to be used as cover images in watermarking application. Color face images of resolution 2048×1536 are converted to grayscale level. The first pose of all individual has the name pattern IIII1001.JPG where IIII is

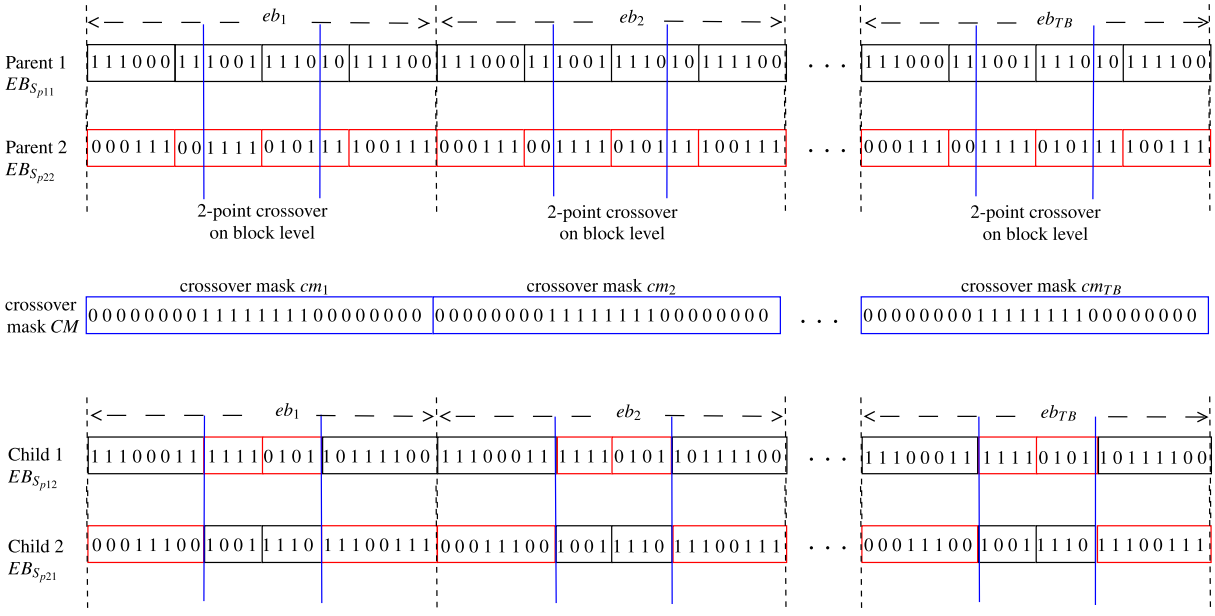


Fig. 6. Proposed crossover on block level with $cp = 2$, embedding capacity $C = 4$, and 6-bit representation for embedding bands.

Fig. 7. Sample face images from first pose of the PUT database (Kasinski et al., 2008), where grayscale images have a resolution 2048×1536 pixels.

Fig. 8. BancTec binary logo used as watermark to be embedded in face images.

the individual number in 4 digits. Face images of this pose are used for proposed system development. The tuning set of face images consists of the first pose of the individuals starting from 0001 till 0040. This set is used for developing and tuning user-defined parameters of the proposed BCGA.

The watermark to be embedded is BancTec binary logo with different resolutions shown in Fig. 8. The watermark embedding/extracting algorithm used in experiments is proposed by Shieh et al. (2004) as illustrated in Section 2.2. The metrics used in experimentation for measuring watermark quality and robustness for the whole face image to find optimal bands are wPSNR and NC respectively as defined in Section 2.1. BWM utilizes weighted sum of BCR and PSNR defined in Section 3.2. The attacks considered in the experimentation are JPEG compression with quality factor equals to 80%, median filtering, and low pass filtering. Entropy is used to select the most textured blocks.

The experiments are executed on gentoo linux based server of 24 GB memory size and 8 cores Intel Xeon CPU X5650 of speed 2.67 GHz. The implementation utilizes Graphical Processing Units (GPU) for DCT transform using 8 NVIDIA Tesla C2050/C2070 GPU cards. The proposed methodology consists of experiments to compare BCGA to baseline system, then tuning aggregation weights, and finally sensitivity analysis on the key algorithm parameters.

Traditional GA is evaluated for different watermark lengths *WL* embedded in different resolutions facial images to study the convergence properties for different search space size. The default watermark length is 48.8k, and the empirical embedding capacity for both BCGA and baseline is set to 1 bit-per-block. The weight λ used in aggregating bwm_i is equal to 30. The population size *pop* equals to 24, and the number of elite individuals *EL* equals to 6. The probabilities for crossover and mutation is set to 0.9 and 0.04 respectively.

The first experiment uses only the first face image to evaluate the convergence properties of the traditional GA. The embedding capacity is fixed to 1 bit per block, and the face image is resized to different reduced resolutions. The first face image is reduced to 20%, 40%, 60%, 80% to characterize the evolution using different dimensions of the search space. Then the evolution of traditional GA is compared to the proposed BCGA evolution using the original size of the first face image which is 2048×1536. In this experiment both GA and BCGA uses simple weighted sum aggregation with $\lambda = 20$, where $gf = QF + 20(RF_1 + RF_2 + RF_3)$. Using traditional GA with weighted sum aggregation represents the baseline system proposed by Shieh et al. (2004). Two methods of crossover are experimented with the baseline GA to measure the impact of increasing crossover points on the performance of baseline GA. Single point crossover is compared to scattered crossover which uses random number of crossover points using randomly generated crossover masks. The baseline GA with the two crossover methods are compared to PBIL and the proposed BCGA using the same population size of candidate solutions and the same weighted sum aggregation.

The next experiment use 40 face images to develop and tune different parameters of the proposed BCGA, where the watermark bits are embedded in the most textured 25% of the blocks using 4 bits-per-block. The second experiment is a sensitivity analysis for

Table 1
Optimization of embedding 1 bit-per-block in reduced size of face image 0001 from PUT database (Kasinski et al., 2008) using baseline method proposed by Shieh et al. (2004), PBIL and BCGA with $gf = QF + 20(RF_1 + RF_2 + RF_3)$.

Resize %-Dimension	Optim. Method	wPSNR [dB]	NC1 (JPEG QF = 80)	NC2 (Median Filtering)	NC3 (LPF)	Conv. Gen.
20%-12k	GA-scattered	57.8835	0.9292	0.7250	0.7875	90
	BCGA	63.0753	0.9340	0.9303	0.9830	26
40%-47k	GA-scattered	57.7174	0.9184	0.6133	0.7585	87
	BCGA	62.3770	0.9287	0.9212	0.9761	26
60%-106k	GA-scattered	56.9713	0.9042	0.5582	0.7075	77
	BCGA	61.0249	0.9546	0.9593	0.9862	12
80%-187k	GA-scattered	57.0923	0.9068	0.5472	0.6859	78
	BCGA	62.2147	0.9509	0.9192	0.9756	48
100%-293k	GA-single	56.9532	0.9023	0.5411	0.6949	10
	GA-scattered	57.3236	0.9138	0.5337	0.6947	43
	PBIL	62.2491	0.9167	0.5535	0.7196	89
	BCGA	59.0047	0.9721	0.9269	0.9749	20

different aggregation methods. The aggregation methods evaluated are the simple weighted sum, and Chebyshev aggregation. The baseline GA fitness are considered reference fitness levels used in Chebyshev aggregation. These Chebyshev reference fitness values represents the minimum fitness values which are considered acceptable for decision maker.

The third experiment utilizes the tuned weights from the previous experiments to measure the impact of user defined parameters on the aggregated fitness evolution using Chebyshev aggregation. The evolution of fitness is measured for the first image for the most important parameters of BCGA. These parameters include the probability of both crossover and mutation defined as $P_{crossover}$ and $P_{mutation}$ respectively. Also the number of elite candidate solutions defined as *EL* is considered in this sensitivity analysis.

The fourth experiment evaluates the impact of using different texture metrics for grayscale images on the aggregated fitness evolution using Chebyshev aggregation. The fitness evolution is measured for the first image using Entropy, JND, NVF, and RS. These texture metrics are used to select the most textured blocks for embedding. The embedding capacity considered in this experiment is 4 bits-per-block, where the most textured 25% of the blocks are used for embedding 4 bits-per-block. For RS texture metric, the

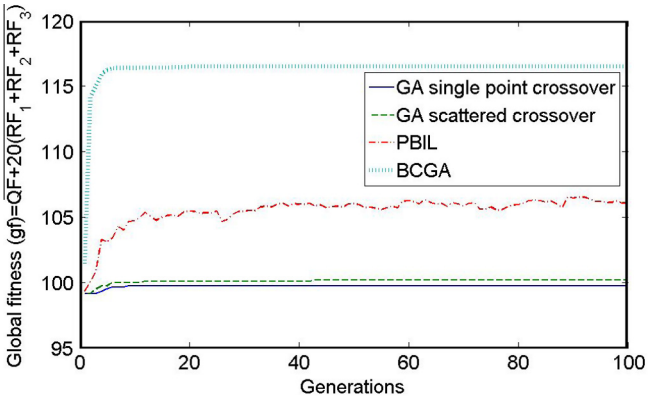


Fig. 9. Evolution of baseline system using GA with two crossover methods, and PBIL compared to the proposed BCGA using embedding capacity of 1 bits-per-block for face image 0001.

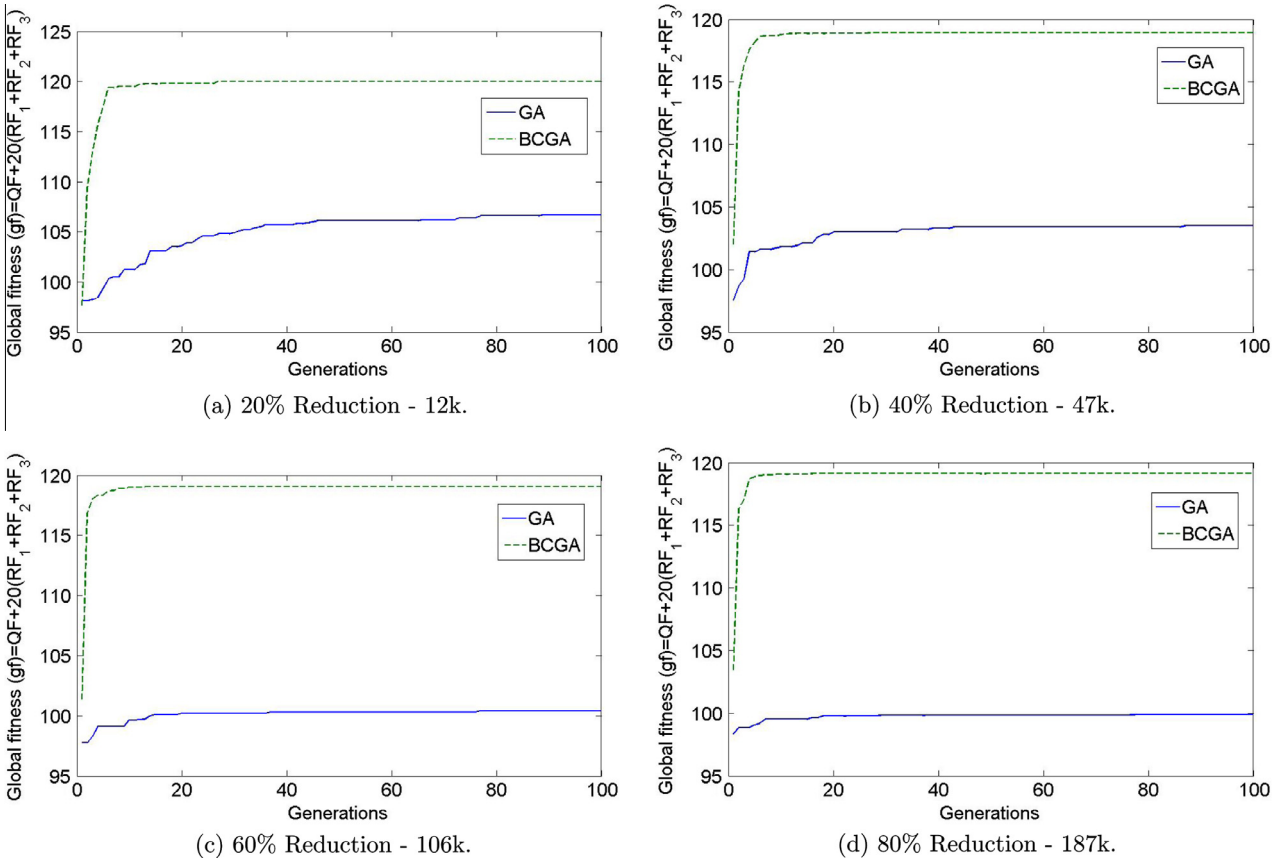


Fig. 10. Evolution of reduced size face image 0001 using baseline system using GA proposed by Shieh et al. (2004) with embedding capacity 1 bit-per-block using different sized watermarks.

number of blocks groups is determined such that the number of blocks belonging to the most textured groups equals or slightly exceeding to 25% of the cover image blocks.

5. Results and discussion

Table 1 shows the fitness produced, and the generation where convergence is assumed using traditional GA and PBIL compared to proposed BCGA for the first face image using its original size. The fitness produced by BCGA is significantly higher robustness compared to GA. The evolution of fitness for methods is shown in Fig. 9, where traditional GA is suffering from a premature convergence at a low fitness starting the generation 43. Using scattered crossover improves the baseline performance using single point crossover, however the traditional GA still suffers from premature convergence.

PBIL provides faster convergence than baseline GA with the two crossover methods. For PBIL, the intrinsic probability vector is evolving rather than candidate solutions. The best fitness candidate solution is used to update the evolving probability vector. This explains the fluctuation of the fitness of the global fitness *gf* using weighted sum aggregation for PBIL.

The average time complexity of BCGA generation is equal to 322 CPU seconds for original size of face image using GPU implementation for watermark fitness evaluations. BCGA has increasing fitness until the generation 20 on a weighted sum fitness which is 17% better compared to GA. The proposed elitism mechanism searches for the best embedding parameters for the same block across the population of candidate solutions, and thus it provides better exploration capabilities for BCGA. This indicates that few generations would be sufficient to produce fitness of satisfactory level.

Also Table 1 shows the fitness produced for different dimensions of the search space using BCGA compared to the baseline GA with scattered crossover points. The fitness produced is degrading for dimensions larger than 47k bits. The generation where convergence is assumed is to decline as image dimension grows. This concludes the efficiency of the baseline system proposed by Shieh et al. (2004) for dimension 47k bits or less. Optimization problem for IW of biometric facial capture images of resolution higher than

Table 2
BCGA quality of solutions compared to traditional GA Shieh et al., 2004 using weighted sum aggregation for tuning set of 40 facial images.

Optim. Method	wPSNR [dB]	NC1 (JPEG QF = 80)	NC2 (Median filtering)	NC3 (LPF)
Baseline GA	57.27 ± 1.59	0.9184 ± 0.0063	0.5285 ± 0.0085	0.6640 ± 0. 0325
Proposed BCGA	57.40 ± 1.82	0.9472 ± 0.0139	0.9223 ± 0.0087	0.9691 ± 0. 0030

Table 3
Chebyshev aggregation using different quality fitness weights γ for embedding optimization using BCGA with embedding capacity equals to 4 bits-per-block for tuning set of 40 face images.

Weight γ	wPSNR [dB]	NC1 (JPEG QF = 80)	NC2 (Median filtering)	NC3 (LPF)
1/20	68.81 ± 1.48	0.6286 ± 0.0399	0.5020 ± 0.0042	0.4789 ± 0. 0105
1/70	57.38 ± 1.88	0.9452 ± 0.0153	0.9265 ± 0.0070	0.9694 ± 0. 0018
1/120	57.31 ± 1.81	0.9475 ± 0.0125	0.9271 ± 0.0066	0.9698 ± 0. 0021
1/200	57.19 ± 1.78	0.9525 ± 0.0130	0.9270 ± 0.0071	0.9698 ± 0. 0022

608 × 816 representing 40% reduction of PUT face images is suffering from premature convergence as shown in Fig. 10. For cover images of lower resolution represented by 20% reduction of PUT face image original size, BCGA outperforms also traditional GA. However for such lower resolutions the traditional GA is not suffering from premature convergence. This proves the efficiency of the baseline system proposed by Shieh et al. (2004) only for low resolution grayscale images with no smooth textured areas.

Table 2 shows the performance of both the baseline GA compared to the proposed BCGA for tuning set of face images including 40 face images. Both baseline GA and the proposed BCGA use weighted sum aggregation for quality and robustness fitness. The mean fitness for the baseline GA method are used in the next experiments as reference fitness values for Chebyshev aggregation defined as r_1, r_2, r_3, and r_4 in Eq. (10).

Table 3 shows the performance of the proposed BCGA using Chebyshev aggregation with different weights for quality fitness γ as shown in Eq. (10). The resulting robustness fitness against different attacks are improved with decreasing the value of γ, on the other side quality fitness produced is degrading. Starting at the value of $\gamma = 1/200$, the robustness fitness against LPF and MF is no more improved, while the robustness against JPEG is slightly improved. The minimum quality fitness is over 42 dB which is considered acceptable level of watermark quality.

The next experiments deal with sensitivity analysis of different parameters on the BCGA evolution, where only the face image 0001 is used in these experiments. Fig. 11(a) shows the fitness evolution in log scale of BCGA using Chebyshev aggregation shown in Eq. (10) for different number of elite solutions *EL*. Table 4 shows the optimal fitness after 100 generations. The best fitness corresponds to using number of elite candidate solutions equals to 4 solutions. This value for *EL* is used in subsequent sensitivity experiments.

Fig. 11(b) shows the fitness evolution in log scale of BCGA using Chebyshev aggregation for different probabilities of crossover on block level $P_{crossover}$. Table 5 shows the optimal fitness after 100 generations. The best fitness corresponds to using the probability $P_{crossover}$ equals to 90%. For probabilities less than this till 0.1% the evolution is nearly the same. The values of *EL* and $P_{crossover}$ producing the best aggregated fitness *gf* are used in subsequent sensitivity experiment. This shows minimal impact of $P_{crossover}$ and *EL* on the fitness evolution of BCGA.

Fig. 12 shows the evolution of BCGA using Chebyshev aggregation for different probabilities of mutation on block level $P_{mutation}$. Table 6 shows the optimal fitness after 100 generations. The best fitness corresponds to using the probability $P_{mutation}$ equals to 4% which corresponds to (1/gene length). BCGA evolution is sensitive to probability of mutation $P_{mutation}$ more than *EL* and $P_{crossover}$ tunable parameters.

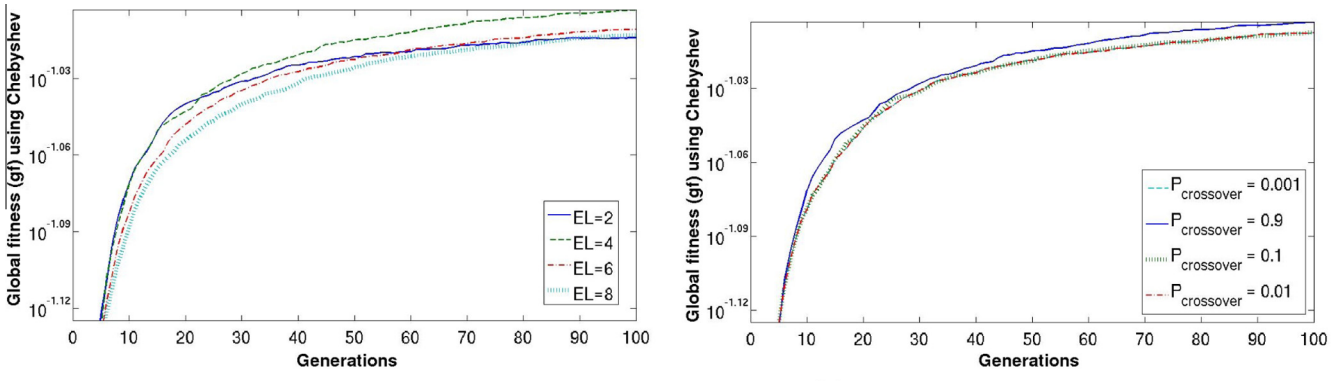
Fig. 13 shows the evolution of BCGA using Chebyshev aggregation for different perceptual texture masks. Table 7 shows the optimal fitness produced after 100 generations. The best global fitness is produced using entropy masks. Both JND and RS (Rabil et al., 2013) texture metrics are next to entropy, and finally NVF has the least global fitness however it produces the best quality fitness. The quality fitness measured in wPSNR of values over 42 dB is considered acceptable. All texture metrics produce acceptable quality fitness. The texture metric RS (Rabil et al., 2013) is well suited for full uneven embedding scheme, where it classifies blocks into different clusters and assign different embedding capacity for each cluster of blocks. However for simple uneven embedding scheme used in this paper, it produces similar evolution to JND.

6. Conclusions and future work

Using traditional IW methods to secure facial captures, all the cover image blocks have to be represented in the population of candidate solutions. For high resolution facial captures, these methods involves handling an optimization problem with huge search space. This huge search space results in premature convergence for traditional EC algorithms.

In this paper, an approach is presented to handle optimizing watermark embedding parameters for high resolution facial captures. Perceptual texture masks are used to select only textured blocks to be represented in EC candidate solutions. Then an application specific GA algorithm based on co-evolution called BCGA is proposed to address this high dimensionality optimization problem. BCGA utilizes local watermarking metric BWM on block level to improve the convergence properties of GA. It performs the traditional GA operators on block level rather than the whole candidate solutions. The proposed elitism mechanism assumes that embedding parameters of blocks corresponding to higher local metric are assumed to have better global fitness for the whole image when concatenated.

Simulation results on PUT face images database (Kasinski et al., 2008) suggest that the proposed BCGA provides solutions (embedding parameters) with 17% better fitness compared to that of traditional GA. Indeed, BCGA avoids the premature convergence issues of traditional GA. The convergence for BCGA is assumed in fewer iterations, and thus speeds up the optimization. Chebyshev aggregation provides better quality of solutions compared to weighted sum aggregation. Sensitivity analysis is performed for user defined



(a) Number of Elite EL. (b) Crossover Probability $P_{crossover}$.

Fig. 11. Impact of user-defined parameters of BCGA on fitness evolution using Chebyshev aggregation for *gf* as shown in Eq. (10). Log scale is used for *gf* due to the high similarity of fitness evolution for *EL* and $P_{crossover}$.

Table 4
Impact of the number of elite individuals *EL* on BCGA performance using Chebyshev aggregation for *gf* as shown in Eq. (10).

EL	wPSNR [dB]	NC1 (JPEG QF = 80)	NC2 (Median	NC3 (LPF) Filtering)	*gf*
2	57.3322	0.9328	0.9172	0.9730	0.0968
4	57.3123	0.9367	0.9270	0.9722	0.0993
6	57.1159	0.9419	0.9202	0.9685	0.0975
8	57.4569	0.9369	0.9181	0.9673	0.0970

Table 5
Impact of the probability of crossover $P_{crossover}$ on BCGA performance using Chebyshev aggregation for *gf* as shown in Eq. (10).

$P_{crossover}$	wPSNR [dB]	NC1 (JPEG QF = 80)	NC2 (Median	NC3 (LPF) Filtering)	*gf*
0.9	57.3123	0.9367	0.9270	0.9722	0.0993
0.1	57.3711	0.9497	0.9232	0.9711	0.0983
0.01	57.2814	0.9482	0.9231	0.9712	0.0983
0.001	57.2814	0.9482	0.9231	0.9712	0.0983

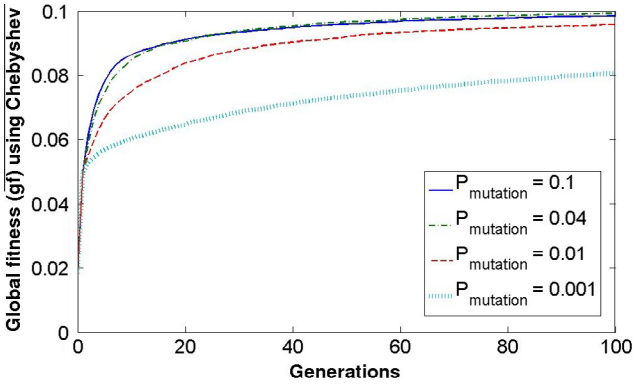


Fig. 12. Impact of $P_{mutation}$ on BCGA evolution using Chebyshev aggregation for *gf* as shown in Eq. (10).

Table 6
Impact of the probability of mutation $P_{mutation}$ on BCGA performance using Chebyshev aggregation for *gf* as shown in Eq. (10).

$P_{mutation}$	wPSNR [dB]	NC1 (JPEG QF = 80)	NC2 (Median Filtering)	NC3 (LPF)	*gf*
0.1	57.3989	0.9438	0.9238	0.9674	0.0985
0.04	57.3123	0.9367	0.9270	0.9722	0.0993
0.01	57.3031	0.9284	0.9133	0.9637	0.0958
0.001	55.5735	0.9216	0.8525	0.9186	0.0806

parameters of BCGA, and different texture metrics used in perceptual texture masks. BCGA evolution is more sensitive to probability of mutation and the texture metrics used for selecting embedding blocks compared to to other user-defined parameters.

The approach presented in this paper can be generalized on high dimensional optimization problems, where local metrics of subcomponents of this high dimension problem is affecting the overall optimization problem. BCGA elitism algorithm can be re-used in optimization algorithms to provide better exploration capabilities for search space, where it implements cooperative coevolution approach at subcomponent level. This can decrease the number of generations required to reach convergence even for lower dimensional problems.

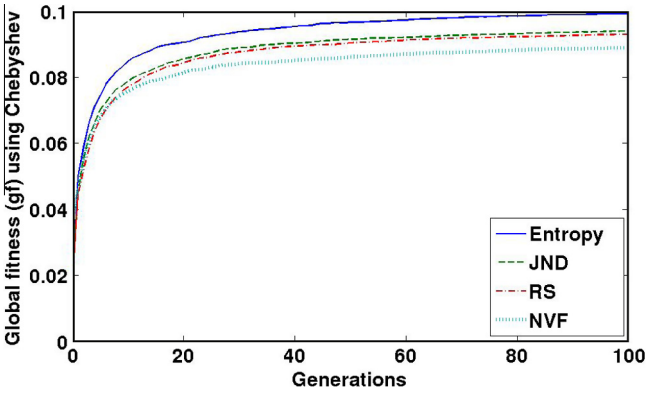


Fig. 13. Impact of using different grayscale texture metrics as perceptual masks on BCGA evolution using Chebyshev aggregation for *gf* as shown in Eq. (10).

Table 7
Impact of the perceptual texture mask on BCGA performance using Chebyshev aggregation for *gf* as shown in Eq. (10).

Texture Measure	wPSNR [dB]	NC1 (JPEG QF = 80)	NC2 (Median Filtering)	NC3 (LPF)	*gf*
Entropy	57.3123	0.9367	0.9270	0.9722	0.0993
JND	57.5036	0.9569	0.9060	0.9659	0.0940
RS(Rabil et al. (2013))	57.2707	0.9538	0.9018	0.9636	0.0930
NVF	63.2379	0.9546	0.8859	0.9643	0.0890

In a future work, standard LSGO algorithms which are tested against optimization problems of 2k variables will be experimented on the IW of facial images using different resolutions. This identifies the search space size limitation for these methods for IW optimization problem. Also more detailed sensitivity analysis will be performed for the impact of BCGA user-defined parameters on the quality of solutions produced using long streams of high resolution facial image captures.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada and BancTec Canada Inc.

References

Baluja, S. (1994). Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report, CMU.

den Bergh, F. V. (2002). An analysis of particle swarm optimizers. Ph.D. Thesis, Dept. Computer Science, Univ. Pretoria, Pretoria, South Africa.

den Bergh, F. V., & Engelbrecht, A. (2004). A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*, 225–239.

Chen, W., Weise, T., Yang, Z., & Tang, K. (2010). Large-scale global optimization using cooperative coevolution with variable interaction learning. In *Parallel problem solving from nature PPSN XI* (pp. 300–309). Berlin Heidelberg, Krakw, Poland: Springer.

Collette, Y., & Siarry, P. (2008). On the sensitivity of aggregative multiobjective optimization methods. *Journal of Computing and Information Technology, 16*, 1–13.

Diaz, D. S., & Romay, M. G. (2005). Introducing a watermarking with a multi-objective genetic algorithm. In *Genetic and evolutionary computation conference, Washington DC, USA* (pp. 2219–2220).

Harik, G. R., Lobo, F. G., & Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transaction on Evolutionary Computation, 3*, 287–297.

Kasinski, A., Florek, A., & Schmidt, A. (2008). The PUT face database. *Image Processing & Communications, 13*, 59–64.

Lee, Z. J., Lin, S. W., Su, S. F., & Lin, C. Y. (2008). A hybrid watermarking technique applied to digital images. *Applied Soft Computing, 8*, 798–808.

Li, X., & Yao, X. (2012). Cooperatively coevolving particle swarms for large scale optimization. *IEEE Transactions On Evolutionary Computation, 16*, 210–224.

Licks, V., & Jordan, R. (2005). *Geometric attacks on image watermarking systems, IEEE multimedia*. IEEE Computer Society.

Pelikan, M., Goldberg, D. E., & Lobo, F. G. (2002). A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications, 21*, 5–20.

Pereira, S., Voloshynovskiy, S., Madueno, M., Marchand-Maillet, S., & Pun, T. (2001). Second generation benchmarking and application oriented evaluation. In *Information hiding workshop III, Pittsburgh, PA, USA* (pp. 340–353).

Potter, M. A., & DeJong, K. A. (2000). Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation, 8*, 1–29.

Rabil, B. S., Sabourin, R., & Granger, E. (2010). Intelligent watermarking with multi-objective population based incremental learning. In *IEEE International conference on intelligent information hiding and multimedia signal processing 2010, Darmstadt, Germany* (pp. 131–134).

Rabil, B. S., Sabourin, R., & Granger, E. (2013). Rapid blockwise multi-resolution clustering of facial images for intelligent watermarking. *Machine Vision and Applications*. http://dx.doi.org/10.1007/s00138-013-0493-1.

Rosin, C. D., & Belew, R. K. (1997). New methods for competitive coevolution. *Evolutionary Computation, 5*, 1–29.

Shan, S., & Wang, G. G. (2010). Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization, 41*, 219–241.

Shieh, C. S., Huang, H. C., Wang, F. H., & Pan, J. S. (2004). Genetic watermarking based on transform-domain techniques. *Pattern Recognition, 37*, 555–565.

Sofge, D., Jong, K. D., & Schultz, A. (2002). A blended population approach to cooperative coevolution for decomposition of complex problems. In *Proceedings of the congress on evolutionary computation, Honolulu, Hawaii* (Vol. 1, pp. 413–418).

Tang, K., Li, X., Suganthan, P., Yang, Z., & Weise, T. (2009). Benchmark functions for the CEC2010 special session and competition on large scale global optimization. Technical Report, Nature Inspired Computation Application Lab., Univ. Sci. Technol. China, Hefei, China.

Vellasques, E., Granger, E., & Sabourin, R. (2010). Intelligent watermarking systems: A survey. *Handbook of pattern recognition and computer vision* (4th ed., pp. 1–40).

Vellasques, E., Sabourin, R., & Granger, E. (2012). gaussian mixture modeling for dynamic particle swarm optimization of recurrent problems. In *Genetic and evolutionary computation conference, Philadelphia, USA* (pp. 73–80).

Voloshynovsky, S., Herrigel, A., & Baum, N. (1999). A stochastic approach to content adaptive digital image watermarking. In *Proceedings of the 3rd international workshop on information hiding, Dresden, Germany* (pp. 211–236).

Wang, Z., Sun, X., & Zhang, D. (2007). A novel watermarking scheme based on PSO algorithm. *Bio-Inspired Computational Intelligence and Applications, 4688*, 307–314.

Wu, M. (2001). Multimedia data hiding. Ph.D. Thesis, Princeton University, 2001.

Yang, Z., Tang, K., & Yao, X. (2008). Large scale evolutionary optimization using cooperative coevolution. *Information Sciences, 178*, 2985–2999.