A Compact Genetic Algorithm with Elitism and Mutation Applied to Image Recognition

Rafael R. Silva, Heitor S. Lopes*, and Carlos R. Erig Lima

Bioinformatics Laboratory
Federal University of Technology Paraná (UTFPR),
Av. 7 de setembro, 3165 80230-901, Curitiba (PR), Brazil
rafael.rsi@gmail.com, hslopes@pesquisador.cnpq.br, erig@utfpr.edu.br

Abstract. The problem of object recognition in images is a hard problem frequently found in industrial and academic application. This work presents the application of an extension of the Compact Genetic Algorithm (emCGA) to three problems of object recognition in real images. Results are compared with an exhaustive search algorithm and another CGA. Results suggested the efficiency of emCGA for this problem and encourages future developments.

Keywords: Compact Genetic Algorithm; object recognition.

1 Introduction

The Genetic Algorithm (GA) is an efficient tool for optimization problems and has been applied on several engineering problems. However, in some applications, the computational cost of the GA can be too high, demanding a prohibitive execution time or excessive hardware resources. A possible alternative is the use of a genetic algorithm with smaller computational complexity, which can run in less powerful systems. Alternatively, it can be implemented in parallel architectures using reconfigurable logical devices [2]. A possible limitation of the GA implementation is the amount of memory required to store the population. This is particularly true for hardware implementations. A GA evolves a population, not a single point, thus requiring memory space to store such information.

On the other hand, the CGA (Compact Genetic Algorithm) can achieve the same level of quality of a SGA (Simple Genetic Algorithm) with uniform crossover, but using less memory to store the population. This is possible because the CGA works with a probability vector instead of the whole population [5].

Another feature of CGA is the use of techniques to evolve the probability vector, imitating the behavior of a SGA. Due to the simplicity of these techniques and the small memory requirements, some works proposed software and hardware implementations of the CGA, showing good results with significant resources reduction for its construction.

* This work was partially supported by the Brazilian National Research Council – CNPq, under research grant no. 309262/2007-0 to H.S. Lopes.

D.-S. Huang et al. (Eds.): ICIC 2008, LNAI 5227, pp. 1109–1116, 2008.

This work presents the application of a new extension of a CGA, the emCGA [8], to object recognition in images. This approach uses elitism and introduces a new mutation operator. Differently from other mutation operators introduced for CGA, this one is applied to new individual generation, thus imitating the crossover operator of a SGA. This operator does not increase significantly computational cost or memory consumption and increases the overall performance, when compared with similar works.

2 The Compact Genetic Algorithm

The CGA is an Estimation of Distribution Algorithm (EDA), first proposed by Harik et al. [5], that generates descendants using a statistical population model, instead of the traditional recombination operators and mutation. [4] estimated the convergence time for a special class of GA problems without iterations among building blocks (BB). The idea of CGA was to simulate an independent random walk model for each bit of the chromosome. As result, the population is reduced to a vector of probabilities that occupies only $L * log_2(N)$ bits of memory, in comparison with $L * N$ for a SGA (where L is the chromosome length and N the size of the population). The CGA can imitate the behavior of a SGA with uniform crossover using a reduced amount of memory. The importance of the size of the population in GA performance has been focused in other works, such as [4]. Usually, a large population size results in better quality of the solution, but it increases the computation cost and memory use.

Since CGA was introduced by [5], several extensions were proposed aiming to improve its performance. Their main focus are the introduction of techniques to allow the CGA to overcome the performance reduction in problems with higher order building blocks. The extensions of CGA that proposed some elitist technique were those that showed the best balance between demand of resources and performance. Elitism allows to increase the selective pressure and to reduce the genetic drift.

Despite of the competitive performance of CGA with SGA for low order BBs, it does not achieves the same performance when high order BBs are present in the problem. In this case, with the compact representation of the population in CGA, the information regarding high order relationships between the chromosome bits does not survive throughout generations, differently that what happens in SGA. For real-world problems, these disturbances generated by the uniform crossover should be overcome, since many problems presents local optima (i.e. multimodal) and interdependent genes (high order BB). However, increasing the selective pressure in CGA tends to decrease these noisy effects, because it increases the probability of high order BBs to survive throughout generations [4],[1]. Some works proposed modifications in the basic CGA to improve performance by increasing the selective pressure, such as: neCGA [1], mCGA [3] and emCGA [8]. Among them, the emCGA showed the best tradeoff between solution quality (fitness value) and convergence speed (number of evaluations).

2.1 The emCGA

The previously mentioned works (neCGA and mCGA) perform better than the original CGA. Both works use elitism, but mCGA uses the mutation operator with success. Another work has also proposed a mutation operator [9], called MBBCGA, but it is different from the mCGA mutation operator. This work presents the application of an extension operator, aimed at improving even more the quality of solutions but, also, keeping a reasonable convergence speed. The previously mentioned works that use mutation operators do not focus on population diversity control, but on local search. In the neCGA the mutation is applied to the elite individual of the current generation to generate another individual. After a tournament over these individuals, the best one will be the elite for the next generation. In the mCGA, mutation is applied in the first individual to substitute the second generation of the conventional CGA.

The new proposed operator allows a more efficient control of the selective pressure, adjusting the population diversity as the consequence of the manipulation of the probability vector. Comparing the proposed mutation operator with that of mCGA, the new operator decreases the number of tournaments per generation and, consequently, the total number of fitness evaluations per generation. The consequence is a significant improvement in the convergence speed of the algorithm [8]. The new CGA resulting from the use of the proposed mutation is named emCGA (elitism with mutation CGA). Basically, the proposed mutation operator changes the random generation phase, by changing the chromosome just-generated with the probability vector.

3 The emCGA for the Object Recognition Problem

The problem of object recognition in images is frequently found in industrial and academic application. However, the recognition of objects using traditional search algorithms is computationally expensive. Particularly, this effort increases when it is present translation, rotation, scale variation or a partial object obstruction [6]. Therefore, the implementation of fast search algorithms for this problem is of great interest. In most cases, applications using these algorithms should be executed in real time, thus requiring fast algorithms instead of exhaustive search algorithms. Algorithms based on metaheuristics can offer reasonable performance and quality of solutions [7]. This fact motivated the use of emCGA to the object recognition problem in real images. In this work, a technique of digital processing of images is explored to extrapolate significant properties of an object in an image and to create a computational model. The proposed model, called of Light Intensity (LInt) model, is based on the intensity of the light in three channels of the image of the object to be detected, Red, Green and Blue (RGB). The RGB channels are weighted with 0.3, 0.59 and 0.11, respectively. Converting every pixel of the object image yields a generic matrix M_{ilum}, where each element is a value of 256 gray levels, corresponding to the light intensity of the pixel in the same image position. When the value of the element is 0 means that the light intensity is 0%, and 255 is 100%.

The fitness function of the emCGA measures the similarity between two images. The object is defined by a reference image or, simply, reference. The input images are converted previously to the LInt model. Therefore, it is possible to reduce the computational cost and the memory demands in the fitness function execution. Based on [7], the fitness function computes the percentage of similarity between two images using the absolute error between the reference and input image models, described by equation 1.

$$S(k) = 100\%. \frac{E_{max} - \{\sum_{i=1}^{n} \sum_{j=1}^{m} |F(I,k,i,j) - M_{ref}(i,j)|\}}{E_{max}} . \tag{1}$$

Where I is LInt matrix of the input image, F is function that transforms an image in a LInt model, returning its (i,j) element,M_{ref} is reference image LInt matrix, $E_{max}$is absolute maximum error, n are number of lines of reference image LInt matrix, m are number of columns of reference image LInt matrix and k is parameter vector of F transform.

This fitness function returns a value in the range 0..100% representing the similarity between the object and the input images. The higher this value, the higher the probability of finding the object in the input image, according to parameters vector k. Therefore, the objective is to find a vector k that yields the higher similarity measured by the fitness function.

Objects studied in this work are three-dimensional. Consequently, they can be found displaced horizontally (x axis), vertically (y axis) and in depth (z axis), as well as rotated in any of the three possible planes (angles θ, α and β). Matrix M_{ref} has n lines and m columns and all their elements are compared for computing similarity. In our experiments we considered only translations in the (x, y) plane and rotations in all planes. The F transform applies a translation and rotation operation to point (i, j) and returns the corresponding element of a input matrix I, according to equation 2 (for simplicity, this equation considers only translation and rotation in the (x, y) plane). As the values of light intensity vary from 0 to 255, the absolute maximum error of an element is 255. Therefore, the absolute maximum error for all elements (E_{max}) is defined as $E_{max} = 255.n.m$.

$$F(k,i,j) = I(x', y') . \tag{2}$$

The parameters vector k is composed by the translation position (x, y) and the rotation angle θ. The position (x', y') is the transformed position of (i, j), where $x' = x + (i. \cos\theta + j. \sin\theta)$ and $y' = y + (i. \cos\theta - j. \sin\theta)$. However, as this transformation uses trigonometrical functions that return a real number, the returned element of the matrix I is the value of the closest integer of this position, by rounding up x' and y'.

The translation position is a position in the input image and it is also the central position of the possible object location. When a position is evaluated close to the border, some positions (x', y') may be invalid. In this case the error in these points will be the maximum. This procedure allows the identification of an object partially clipped close to the borders. A chromosome in the emCGA is then defined by the parameter vector k and coded according to the following ranges: $0 \leq x \leq c$, $0 \leq y \leq l$ and $0 \leq \theta < 360$ degrees, where c and l are the number of columns and lines of the input image. Displacements are measured in pixels (steps of 1) and angle in degrees (steps of 0.7). The overall length of the chromosome is given as $L = log_2(c) + log_2(l) + 9$.

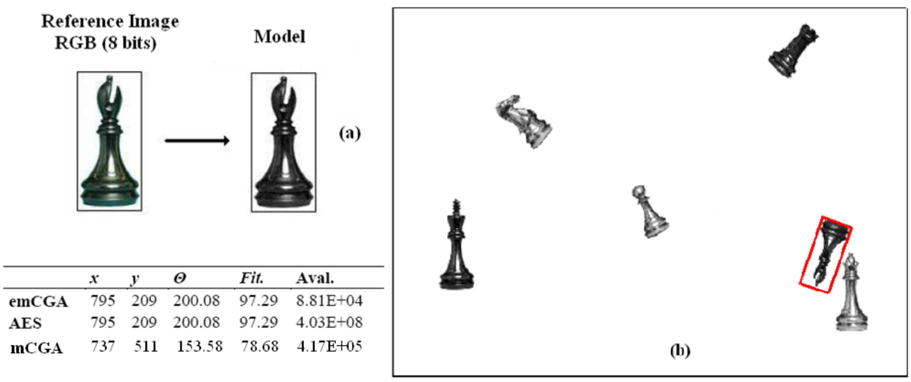
The emCGA uses two parameters that need to be previously defined: the population size and the mutation rate. The search of an object in an image is a complex problem, since it is a multimodal problem, with many local maxima randomly distributed. Besides, it is important to note that the genes in the chromosome, that is, the parameters being optimized, are strongly interconnected. Consequently, the recommended mutation rate is approximately $1/L$, and the size of the population should be large enough [8].

The input images used in our experiments were 1024 x 768 bits large. In this case, parameters x and y will need 10 bits for encoding, plus 9 bits for the rotation angle. Consequently, the recommended mutation rate is approximately 3.4%. The size of the population that yielded a reasonable tradeoff between performance and computational cost was 16384 individuals. The GAs, as well as the emCGA, have their stochastic nature implemented through a pseudo-random number generator. So, it is also necessary to define a seed for this generator. For each independent run, a different random seed was chosen.

4 Experiments and Results

Three experiments were run to evaluate emCGA for the object recognition task in images. Results are presented in figures showing the reference image(s), the input image and the object (standing out in this input image using red lines). We opted to show the input in gray levels to facilitate the visualization.

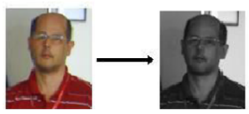
Our experiments use images from digital pictures. For the sake of having a golden standard, that is, the optimal solution, for a given problem, we devised



	x	y	Θ	Fit.	Aval.
emCGA	795	209	200.08	97.29	8.81E+04
AES	795	209	200.08	97.29	4.03E+08
mCGA	737	511	153.58	78.68	4.17E+05

Fig. 1. Chess Bishop Search. a) Reference Image and LInt Model. b) Input Image and object recognized.

Reference Image RGB (8 bits) — Model



(a) (b)

	x	y	θ	*Fit.*	Aval.
emCGA	792	428	0	100	1.08E+05
AES	792	428	0	100	4.03E+08
mCGA	185	293	3.52	84.88	3.84E+05

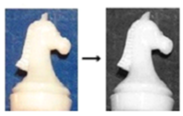
Fig. 2. Faces Search. a) Reference Image and the associated LInt model. b) Input image and the recognized face.

an Exhaustive Search Algorithm (AES). AES evaluates the same fitness function (equation 1) for all possible combinations of the parameters vector k. The results of the application of emCGA are compared with those obtained by AES and with those obtained by mCGA[3], so as to evaluate its accuracy and computational cost. Values for the parameters vectors are shown in each experiment as well as the fitness value (Fit.) and the number of fitness evaluations (Eval.).

In the first experiment, the object to be found in the input image is a chess bishop piece. The reference image is used to generate LInt model, according to figure 1a. This model is used by the emCGA to find the object in the input image. The object found is presented in figure 1b inside a rectangle to stand it out. This procedure will be also used in the remaining experiments in this work.

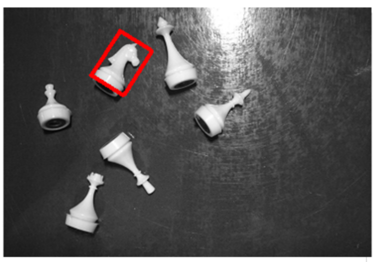
	x	y	θ	*Fit.*	Aval.
emCGA	327	552	32.41	92.86	9.58E+04
AES	327	552	32.41	92.86	4.03E+08
mCGA	452	501	225.44	80.63	3.59E+05

Reference — Model



	x	y	θ	*Fit.*	Aval.
emCGA	292	455	345.91	85.04	8.07E+04
ABE	292	455	345.91	85.04	4.03E+08
mCGA	292	455	345.91	85.04	3.84E+05

(a)



(b) (c)

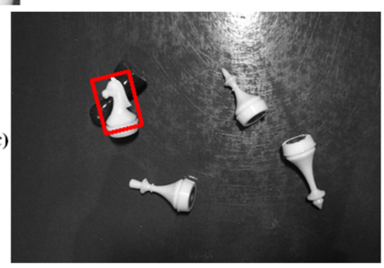


Fig. 3. Chess Knight Search. a) Reference Image and Model. b) First Input image and object recognized. c) Second Input image and object recognized.

Notice that the bishop in the input image is translated and rotated relative to the reference image.

In the second experiment, a human face is recognized in a digital picture in which there are several people. This is a difficult problem for a computer-based vision system, since many local optima can be easily realized. The face used is shown in figure 2a and input image and the recognized face are shown in figure 2b . The reference image is cut out of the input image.

In the last experiment, the object to be recognized is a chess knight piece (figure 3a). In the first input image, shown in figure 3b, the object is translated and rotated in the image plane (x, y). The main distortion in this image that makes the problem somewhat difficult is due to a different illumination angle, shedding a bright spot to the object. In the second input image (figure 3c), the object is also rotated in both (z, y) and (x, z) axes. These rotations impose a more challenging task than the previous image. Besides these distortions, in the two input images other similar pieces are added, increasing the difficulty of the problem by injecting more local maxima.

5 Discussion and Conclusions

The object recognition problem for computational vision easily falls into an exhaustive search. It is expected that the use of a heuristic search algorithm, such as the emCGA, can achieve similar results to an AES, but with smaller computational cost. The computational costs of the algorithms are a function of the number of fitness evaluations.

The experiments were developed to evaluate the efficiency of emCGA in a real-world problem. Complex input images were used to show the robustness of the object detection method using the emCGA. In the first experiment, the chess pieces introduce several local maxima in the search space. Results shown that the emCGA reaches the global maximum with a computational cost of approximately 0.022% of the AES and 21.1% of the mCGA. However, mCGA achieved a performance around 20% worse than emCGA. In the second experiment no distortion was added, since the reference image is extracted from the input image. However, the input image has many local maxima, since there are a lot of similar faces. The emCGA was able to find the global maximum with a computational cost of approximately 0.027% of the AES. Also, emCGA was 4 times faster than mCGA, which again performed worse than emCGA. This results show that the emCGA, using the LInt model, has a good discriminatory power, capable of capturing small details in the images. The last experiment was especially devised to verify how the method behaves in the presence of significant distortions in the images. In the first input image the object to be recognized was rotated in two axes, and for the second input image, in three axes. For both cases, emCGA reached the global maximum with a computational cost of approximately 0.02% of the AES, and emCGA converged 4 to 5 times faster than mCGA. Notice that for the first input image mCGA did not found an acceptable solution for the problem, since θ is rotated around 180 degrees of the expected orientation.

Recall that, in our experiments, the chromosome of emCGA encoded only rotation in plane (x, y). Notwithstanding, emCGA was able achieve a good result even in the presence of unexpected distortions in the 3D space.

Finally, through this work it is possible to conclude that the emCGA using the LInt model is appropriate for applications that require a compact, fast and efficient search algorithm, with limited computational resources. Therefore, we conclude that the proposed method can be efficiently applied to real-world problems without increasing significantly the implementation complexity or its computational cost.

Future work will focus in evaluating the limits of emCGA to other real problems, as well as comparing it with other similar approaches, and re-implementing the system using reconfigurable logic for real-time image processing.

References

1. Ahn, C., Ramakrishna, R.: Elitism-Based Compact Genetic Algorithms. IEEE Trans. Evol. Comput. 7, 367–385 (2003)
2. Ferlin, E.P., Lopes, H.S., Lima, C.R.E., Chichaczewski, E.: Reconfigurable Parallel Architecture for Genetic Algorithms: Application to the Synthesis of Digital Circuits. In: Diniz, P.C., Marques, E., Bertels, K., Fernandes, M.M., Cardoso, J.M.P. (eds.) ARCS 2007. LNCS, vol. 4419, pp. 326–336. Springer, Heidelberg (2007)
3. Gallagher, J., Vigraham, S., Kramer, G.: A Family of Compact Genetic Algorithms for Intrinsic Evolvable Hardware. IEEE Trans. Evol. Comput. 8, 111–126 (2004)
4. Harik, G., Cantu-Paz, E., Goldberg, D.E., Miller, B.L.: The Gambler's Ruin Problem, Genetic Algorithms, and the Sizing of Populations. Evol. Comput. 7, 231–253 (1999)
5. Harik, G., Lobo, F., Goldberg, D.E.: The Compact Genetic Algorithm. IEEE Trans. Evol. Comput. 3, 287–297 (1999)
6. Jain, R., Kasturi, R., Schunck, B.G.: Machine Vision. McGraw-Hill, New York (1995)
7. Perlin, H.A., Lopes, H.S., Mezzadri, T.C.: Particle Swarm Optimization for Object Recognition in Computer Vision. In: Nguyen, N.T., Borzemski, L., Grzech, A., Ali, M. (eds.) IEA/AIE 2008. LNCS (LNAI), vol. 5027, pp. 11–21. Springer, Heidelberg (2008)
8. Silva, R.R., Lopes, H.S., Erig Lima, C.R.: A New Mutation Operator for the Elitism-Based Compact Genetic Algorithm. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 159–166. Springer, Heidelberg (2007)
9. Zhou, C., Meng, K., Qiu, Z.: Compact Genetic Algorithm Mutated by Bit. In: 4th World Congress on Intelligent Control and Automation, vol. 4, pp. 1836–1839. IEEE Press, New York (2002)