Discrete PSO with Simulated Annealing applied to improve multilevel phase holograms design

Ivan Cruz Aceves[1], Manuel Ornelas Rodríguez[1], and Arturo Hernández Aguirre[2]

[1] Instituto Tecnológico de León, León,Guanajuato, México.
[2] Centro de Investigaciones en Matemáticas, Guanajuato, Guanajuato, México.
{ivan.cruzac@gmail.com,mornelas67@yahoo.com.mx,artha@cimat.mx}

Abstract. This paper presents an efficient hybrid algorithm based on Particle Swarm Optimization (PSO) with discrete values using Simulated Annealing (SA), denomined (DPSOSA), for the optimum design of multilevel phase computer generated holograms (CGH's). This algorithm deals with discrete values using the periodicity of the Discrete Fourier Transform to improve the diffraction efficiency of CGH's. The experimental results indicate that DPSOSA can enhance the performance obtained with Genetic Algorithm (GA), Discrete Particle Swarm Optimization (DPSO) and Iterative Fourier Transform Algorithm (IFTA). The behaviour of DPSOSA is discussed.

Key words: Combinatorial Optimization, Computer Generated Holograms, Particle Swarm Optimization, Evolutionary Computation, Genetic Algorithms, Hybrid Algorithms, Simulated Annealing

1 Introduction

Computer generated holograms (CGH's) are attractive for a variety of applications, when reduced dimensions, light weight, and low replication cost are required. Some applications include optical testing, image displays, optical security and metrology, to mention but a few. A number of algorithms have been proposed for designing computer generated holograms [1][2]. The Iterative Fourier Transform Algorithm (IFTA) is one of the most popular. This algorithm was developed by Gerchberg and Saxton in 1972 [3]. The IFTA is efficient and can handle large amounts of data, however, the algorithm is sensitive to initial parameters and it is prone to stagnate in local minima. This study proposes the use of global optimization algorithms that do not depend on the starting parameters of IFTA, and that are effective in avoiding local minima. The paper is organized as follows; section 2 deals with the CGHs design using the IFTA, whose drawback is the inability to escape from local optima. Section 3 describes evolutionary and bio-inspired algorithms like Genetic Algorithm and Particle Swarm Optimization in discrete domain. In section 4 it is explained the proposed hybrid algorithm DPSOSA. Section 5 is concerned with the experimental setup, the results, their analysis and discussion. Finally section 6 presents some conclusions and future work.

2 CGH's design by using IFTA

The algorithm for hologram calculation with a standard version of IFTA [4], is shown in figure 1. The holograms were designed to work for transmission by modulating only the phase of the incident light and reconstructing the image of the desired object at the far field. Phase holograms are more attractive than amplitude holograms, because the former provide higher diffraction efficiency since ideally no energy is absorbed.

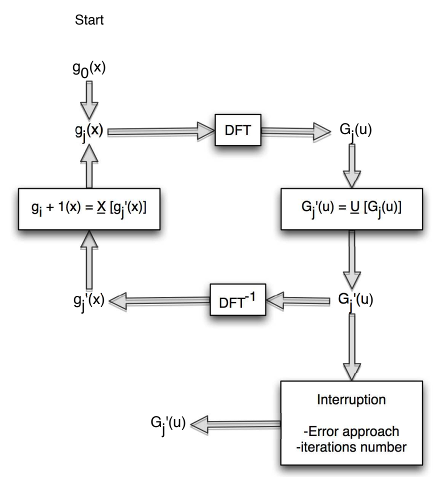

Fig. 1. Iterative Fourier Transform Algorithm Diagram (IFTA).

The first step is to calculate the transmission function of the hologram $(G_j(u))$ from the original object $g_o(x)$ by means of the Discrete Fourier Transform (DFT). The operators U and X are the rules of the Fourier plane (u) and the object plane (x), respectively. The U operator replaces the module of $G_j(u)$ by the unit value in the Fourier plane and retains its phase. The X operator maintains the phase of $g_j(x)$ and replaces its amplitude by the original signal $g_o(x)$. The common stop criterion of IFTA is by the approximation error between the original and the reconstructed signals, or by prefixed number of iterations. Due to limitations in the manufacturing methods of diffractive elements, the next step is to discretize the continuous transmission function obtained by IFTA in some number of phase levels (N). Then, the discrete phase function is encoded as an array of pixels in gray scale, thereby generating the hologram mask. IFTAs problem resides in the fact that it cannot achieve the established theoretical diffraction efficiency given by *equation* 1 [5], and shown in table 1. The efficiency problem is caused by stagnation at local minima.

$$\eta = \left[\left(\frac{N}{\pi} \right) * \sin \left(\frac{\pi}{N} \right) \right]^2 \tag{1}$$

Table 1. Theoretical diffraction efficiencies as a function of the number of phase levels.

2 phase levels	3 phase levels	4 phase levels	5 phase levels	6 phase levels
40.5%	68.3%	81%	87.5%	91.1%

The CGH's design is a common NP-hard combinatorial optimization problem. The objective is, applying IFTA, to obtain output holograms which will initialize the evolutive algorithms. These algorithms will search the optimal combination of discretized pixel values that maximizes the diffraction efficiency, given by *equation* 2. This ratio is the fitness function for the evolutive algorithms.

$$Diffraction\_Efficiency = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} ImgRecons(x,y)}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} Img(x,y)} * 100 \qquad (2)$$

Where ImgRecons (x, y) denotes reconstructed image and Img (x, y) is the original one.

3 Optimization Techniques

Evolutionary algorithms (EA's) [6] are bio-inspired metaheuristics that have proved very robust to solve hard or complex problems. These are population based algorithms, where every individual of the population encodes a candidate solution.

The mechanic of the algorithm is to evolve and improve the quality of the population. The repeated application of selection, reproduction and mutation operators produces variations with improved quality. A fitness function measures the goodness or quality of the solution carried by every individual.

3.1 Genetic Algorithms

Genetic algorithms (GA) have had a great deal of success in solving search and optimization problems [7][8][9]. The first step of a GA is to create the initial population, and to measure the quality of every individual with the fitness function. A selection operator, such as binary tournament or proportional, is applied and a temporal population with above average fitness is stored. Then, a reproduction operator, such as one point or two-point crossover, is applied to the pool of selected individuals and new individuals are spawned. After the application of the mutation operator, the new individuals populate the new generation.

3.2 Discrete Particle Swarm Optimization

Particle Swarm Optimization algorithm introduced by Kennedy and Eberhart [10], is applied in continuous spaces optimization.

Since its publication, many related works have appeared to adapt this technique for solving problems involving discrete spaces [11]. *Equations* 3 and 4 are preserved in modified algorithms in order to work in a discrete manner.

$$\mathbf{V}_i(t+1) = \mathbf{V}_i(t) + c1\omega * (Y_t - X_i(t)) + c2\omega * (G_t - X_i(t)) \tag{3}$$

This is the velocity equation where c1 and c2 are the inertia constants, w is learning factor, G is gbest i.e. the particle with the best fitness, Y is the best position of the present particle X. Once the velocity term has been computed the new particle position is calculated via *equation* 4.

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1) \tag{4}$$

The adaptation consists in the use of these canonical PSO equations, which are applied to the original population. Then the modified population is checked against the original one. Wherever a value is not repeated in corresponding individuals, they are marked as feasible for mutation. Finally, these individuals are replaced by feasible solutions by using a uniform random number generator. This is called random mutation.

3.3 Simulated Annealing

Simulated Annealing (SA) is a stochastic optimization algorithm that leads to a stationary state which is described by the Boltzmann distribution [12].

In essence this procedure resembles cooling a crystal down slowly such that defects and other lattice impurities can heal out and a pure crystalline structure (global minimum) is achieved at low temperature [13].

Its equilibrium properties at a temperature *Temp* are determined by the Boltzmann distribution with *equation* 5, and Δ is the difference between the current and the new solution.

$$Bolt = e^{-\Delta/Temp} \tag{5}$$

4 Hybrid Discrete Particle Swarm Optimization with Simulated Annealing Algorithm Proposed

A promising strategy has been the combination of different metaheuristics in order to exploit their strengths and thus obtaining better results for difficult problems, where no good approximations are achieved by using a single technique as shown in [14][15].

The related works with the fact of combining Particle Swarm Optimization and Simulated Annealing have demonstrated to obtain acceptable results due to the fact that velocity and position of PSO equations can be adapted to work in discrete environments as shown in [16]. Below the proposed algorithm DPSOSA is presented.

Algorithm 1 DPSOSA Algorithm

1. Initialize the control parameters,ω,$c1$, $c2$,γ,D,θ, *iBest*, *t=0*, *Temp*, α, ψ
2. Create and initialize the population Pop[κ,D] to apply Simulated Annealing,
3. **SimulatedAnnealing**(*Temp*, α, ψ, Pop)
   3.1 **for** *each individual* φ **do**
   3.2 **while**(*Temp* $> \psi$) **do**
   3.3 next = Permutation (current)
   3.4 $\Delta = f$(next) - f(current)
   3.5 **if**($\Delta > 0$)
   3.6 Assign next to current
   3.7 **else if**(φ Mod 2 == 1)
   3.8 Apply Boltzmann Operator
   3.9 Decrement *Temp* with α
   3.10 **end**//while **end**//for
   3.11 **return Pop**//to Initialization of DDE
4. **while** *stopping condition not true* **do**
   5. **for** *each individual* Xι (t) ϵ *Pop[t,D]* **do**
   6. **DiscreteMutation(Xι,θ)**
   7. Evaluate the fitness f(Xι)
   8. **if** f(Xι) is better than $f(X_{pbest})$
   9. Add **Xι** to X_{pbest}
   10. **if** f(Xι) is better than f(Pop(*iBest*))
   11. Add **Xι** to **Pop**(*iBest*)
   12. Update Velocity (Xι,ω,$c1$,$c2$)
   13. Update Position (Xι)
   14. **end**//for
15. **end**//while
16. **return Pop(*iBest*)**//as the solution

Where $c1$ y $c2$ are inertia weights, γ individuals in population, ω is learning factor, θ is mutation percentage, *iBest* index best solution and *Temp*, α, ψ are control parameters of Simulated Annealing.

The Simulated Annealing algorithm allows to initialize the population of DPSO, yielding greater variability to the information that the individual gets directly from IFTA.

5 Experiments

In this section, it is compared the DPSOSA model with the previously described algorithms. All the models have been tested with the same image and conditions; the image size is 128 x 128 pixels. It is analyzed the performance to compare these models in 30 executions.

The features of algorithms are described as follows:

- Genetic Algorithm with Proportional Selection (GAP):
  - 20 individuals
  - 100 generations
  - crossover = 80%
  - mutation = 5%
- Genetic Algorithm with Tournament Selection (GAT): same conditions GAP.
- Genetic Algorithm with Boltzmann Selection (GAB): same conditions GAP.
- Genetic Algorithm with Sigma Scaling Selection (GAS): same conditions GAP.
- Discrete Particle Swarm Optimization (DPSO):
  - 20 particles
  - 100 iterations
  - inertia = 0.7
  - learning factor = 2
  - mutation = 50%
- Discrete Particle Swarm Optimization with Simulated Annealing (DPSOSA):
  - same conditions DPSO
  - Temp = 10
  - alpha =0.999
  - epsilon=0.01

In table 2 and 3, it is showed the maximum and average diffraction efficiencies reported by the algorithms.

Table 2. Maximum diffraction efficiencies obtained by the implemented algorithms.

Phase levels / Method	2	3	4	5	6
IFTA	36.8832	61.3139	71.9660	76.9917	79.9846
GAP	38.2017	64.1379	75.0170	79.9976	82.6758
GAT	38.1199	63.9807	74.6681	79.7798	82.7991
GAB	38.0825	63.4223	74.5197	79.7394	82.7411
GAS	38.1085	63.3135	74.5224	79.8197	82.6829
DPSO	38.1853	63.3100	74.2949	79.2902	82.0156
DPSOSA	38.3106	63.4389	74.9432	80.1602	83.0577

Table 3. Average diffraction efficiencies reported by the algorithms.

Method \ Phase levels	2	3	4	5	6
IFTA	36.2821	60.7147	71.6853	76.7688	79.6112
GAP	37.8340	63.3993	74.4306	79.5346	82.5065
GAT	37.5643	63.2141	74.3967	79.4963	82.5386
GAB	37.5849	63.1036	74.3460	79.4334	82.4670
GAS	37.4874	63.0898	74.2884	79.4244	82.4182
DPSO	37.8008	62.8782	74.0260	79.0729	81.8170
DPSOSA	37.9287	63.3163	74.7011	79.9399	82.8017

Figure 2 shows the reconstructed images and their hologram of 256x256 pixels with 2,3 and 4 phase levels. In figure 3 it is showed the experimental image of 128x128 pixels used in the implemented algorithms. Figures 4, 5 and 6 show the diffraction efficiency comparisons among the algorithms.

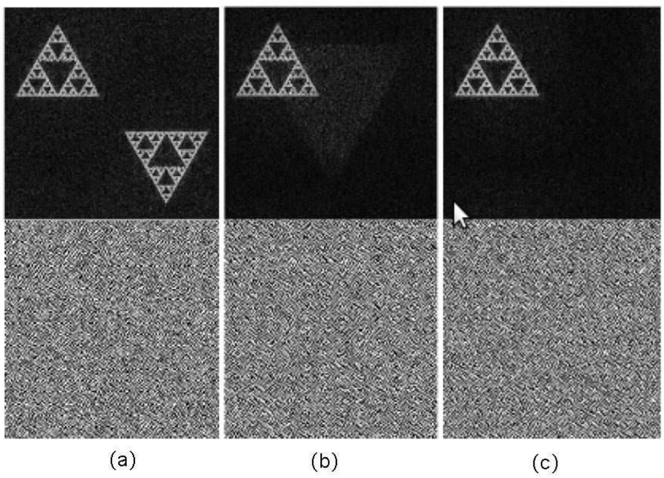

Fig. 2. Reconstructed image and their hologram with a) 2 phase levels, b) 3 phase levels, c) 4 phase levels.

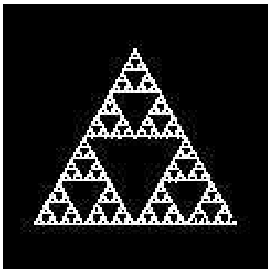

Fig. 3. Experimental Image.

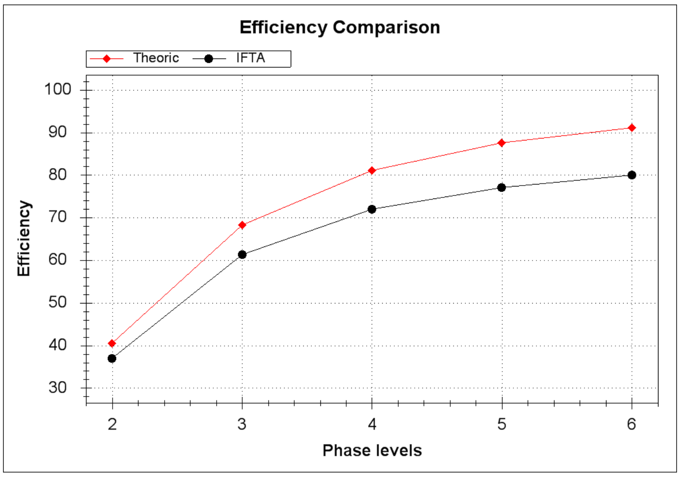

Fig. 4. Diffraction efficiency comparison between IFTA and theoretical values.

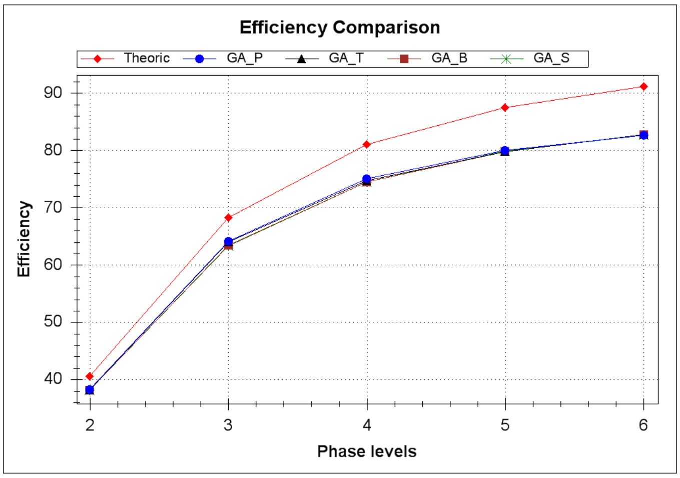

Fig. 5. Diffraction efficiency comparison between Genetic Algorithms and theoretical values.

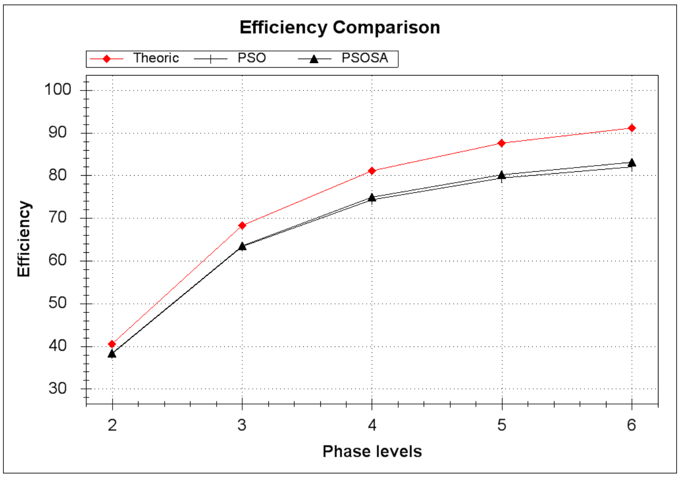

Fig. 6. Diffraction efficiency comparison among PSO, PSOSA and theoretical values.

6 Conclusions

In this work a hybrid algorithm design composed by robust global optimization techniques has been presented, demonstrating to obtain a better performance during the test. Discrete Particle Swarm Optimization with Simulated Annealing obtained the best solution for the CGH's design. Ideas for future work involve extending the proposed strategies to solve problems including another heuristics. It would be also interesting to work the approach for including not uniform domain size for every variable.

Acknowledgments. Authors thank to CONCYTEG and CONACYT for financial support.

References

1. M. Hsieh and C. Lin, Design of computer generated phase holograms by using iterative Fourier transform algorithm, IEEE, Lasers and Electro-Optics,ISBN:0-7803-7766-4, Vol 1, pag 293, (2003).
2. Hsieh, Mei Li, Computer generated holograms for creating three-dimensional images using a LCSLM, Proceedings of the SPIE, Volume 6314, pp. 63141E (2006).
3. R.W. Gerchberg and W.O. Saxton, A practical algorithm for the determination of phase from image and diffraction plane pictures, Optik 35, 237-246 (1972).
4. Frank Wyrowski, Diffractive optical elements: iterative calculation of quantized, blazed phase structures, Opt. Soc. Am. Vol 7,No. 6, 961-969 (1990).
5. E. Hasman, N. Davidson, and A. Friesem, Efficient multilevel phase holograms for CO2 lasers, Opt. Soc. Am. Vol 16, No. 6,423-425 (1991).
6. Back T., Evolutionary Algorithms in Theory and Practice, Oxford University Press, 1996.
7. P. Caamaño, F. Bellas, et al, Application domain study of evolutionary algorithms optimization problems, GECCO, ACM: 978-1-60558-130, 495-502 (2008).

8. B. Doerr, E. Happ, and C. Klein, Crossover can probably be useful in evolutionary computation, GECCO, ACM: 978-1-60558-130, 539-546 (2008).
9. M. Ornelas, F. Cuevas, et al, Algoritmos Genéticos y Recocido Simulado aplicados al diseño de hologramas de fase, XLIX Congreso nacional SMF/XIX Reunión anual AMO, San Luis Potosí, México, (2006).
10. J. Kennedy and R. Eberhart, Particle swarm optimization, In proceedings of IEEE International conference on Neural Networks, IEEE Press, 1942-1948, (1995).
11. M. Fatih, P. Suganthan and Q. Pan, A discrete particle swarm optimization algorithm for the generalized traveling salesman problem", GECCO, ACM: 978-1-59593-697-4, 158-167, (2007).
12. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by Simulated Annealing", SCIENCE, Vol. 220, Number 4598, (1983).
13. Alexander K. Hartmann, Heiko Rieger, "Optimization Algorithms in Physics", Wiley-VCH, Germany, (2002).
14. Z. Zapotecas and C. Coello, Hybridizing an evolutionary algorithm with mathematical programming techniques for multi-objective optimization, GECCO, ACM: 978-1-60558-130, 769 770, (2008).
15. A. Ochoa Ortiz, A. Muñoz and A. Hernández, A Hybrid system using PSO and Data Mining for determining the ranking of a new participant in Eurovision, GECCO, ACM: 978-1-60558-130, 1713-1714, (2008).
16. S. Sivanandam, P. Visalakshi, and A. Bhuvaneswari, Multiprocessor Scheduling using hybrid Particle Swarm Optimization with dynamically varying inertia, in International Journal of Computer Science & Applications, Vol. 4, Issue 3, pp 96-106, (2007).