Halftone Image Generation with Improved Multiobjective Genetic Algorithm

Hernán E. Aguirre, Kiyoshi Tanaka, Tatsuo Sugimura and Shinjiro Oshita

Dept. of Electrical and Electronic Engineering, Faculty of Engineering,
Shinshu University, 4-17-1 Wakasato, Nagano, 380-8553 Japan
{{ahernan, ktanaka, tsugimu}@gipwc,oshita@oshan}.shinshu-u.ac.jp

Abstract. A halftoning technique that uses a simple GA has proven to be very effective to generate high quality halftone images. Recently, the two major drawbacks of this conventional halftoning technique with GAs, i.e. it uses a substantial amount of computer memory and processing time, have been overcome by using an improved GA (GA-SRM) that applies genetic operators in parallel putting them in a cooperative-competitive stand with each other. The halftoning problem is a true multiobjective optimization problem. However, so far, the GA based halftoning techniques have treated the problem as a single objective optimization problem. In this work, the improved GA-SRM is extended to a multiobjective optimization GA to generate simultaneously halftone images with various combinations of gray level and spatial resolution. Simulation results verify that the proposed scheme can effectively generate several high quality images simultaneously in a single run reducing even further the overall processing time.
Keywords: multiobjective genetic algorithm, multiobjective optimization, halftoning problem, cooperative-competitive genetic operators.

1 Introduction

The multiobjective nature of most real-world problems makes multiobjective optimization (MO) a very important research topic. Evolutionary algorithms (EAs) seem particularly desirable to solve MO problems because they evolve simultaneously a population of potential solutions to the problem in hand, which allows to search for a set of Pareto optimal solutions concurrently in a single run of the algorithm. Many authors have been increasingly investigating MO using EAs in recent years and the number of applications has been rapidly growing [1–4]. In the signal processing area, application methods using EAs, especially genetic algorithms (GAs), are also steadily being developed[5].

In this work, we especially focus on the image halftoning technique using GAs. Kobayashi et al.[6, 7] use a GA to generate bi-level halftone images with quality higher than conventional techniques such as ordered dithering, error diffusion and so on[8]. However, it uses a substantial amount of computer memory and processing time[6, 7]. Recently, Aguirre et al.[9, 10] have proposed an improved GA (GA-SRM) to overcome these two drawbacks of the conventional halftoning technique with GAs. GA-SRM is based on an empirical model of GA that applies genetic operators in parallel putting them in a cooperative-competitive stand with each other[11–14]. The improved GA-SRM, extended to the halftoning problem, can generate high quality images achieving a 98% reduction in the population size and an 85%-70% reduction in processing time.

The halftoning problem is a true MO problem in which high gray level and high spatial resolution must be sought to achieve high quality images. The GA based halftoning techniques mentioned above, however, treat the problem as a single objective optimization problem and can generate only one image at a time.

In this work, the improved GA-SRM[9, 10] is extended to a multiobjective optimization GA to generate simultaneously halftone images with various combinations of gray level and spatial resolution. The simulations results show that the proposed scheme can effectively generate several images in a single run reducing even further the overall processing time.

2 Halftoning Problem with GAs

Digital halftoning, a key component of an image display preprocessor, is the method that creates the illusion of continuous tone pictures on printing and displaying devices that are capable of producing only binary picture elements. The fast growing computer and information industry requires each time higher image quality and demands higher resolution devices. The halftoning algorithms capable of delivering the appropriate image quality for such devices are also needed.

Kobayashi et al.[6, 7] use a GA to generates bi-level halftone images with quality higher than traditional techniques such as ordered dithering, error diffusion and so on[8]. An input gray tone image of R gray levels is divided into non-overlapping blocks of $n \times n$ pixels, and then the 2-dimensional optimum binary pattern for each image block is searched using a GA[6, 7]. The GA uses a $n \times n$ 2-dimensional binary representation for the individuals. Crossover interchanges either sets of adjacent rows or columns between two individuals and mutation inverts bits with a very small probability per bit after crossover similar to canonical GA[15, 16]. Individuals are evaluated for two factors required to obtain visually high quality halftone images. (i) One is high gray level resolution (local mean gray levels close to the original image), and (ii) the other is high spatial resolution (appropriate contrast near edges)[6, 7]. The gray level resolutions error is calculated by

$$E_m(\boldsymbol{x}_i^{(t)}) = \frac{1}{n^2} \sum_{(j,k) \in block} \mid p(j,k) - \hat{p}_b(j,k) \mid \qquad (1)$$

where $\boldsymbol{x}_i^{(t)}$ is i-th individual at t-th generation, $p(j,k)$ is the gray level of the (j,k)-th pixel in the original image block, and $\hat{p}_b(j,k)$ is the estimated gray level associated to the (j,k)-th pixel from the generated binary block. To obtain $\hat{p}_b(j,k)$, a reference region around the (j,k)-th binary pixel (for example 5×5 pixels) is convoluted by a gaussian filter that models the correlation among pixels. On the other hand, the spatial resolution error is calculated by

$$E_c(\boldsymbol{x}_i^{(t)}) = \frac{1}{n^2} \sum_{(j,k) \in block} |\ (p(j,k) - \bar{p}_s(j,k)) - (q(j,k) - \frac{1}{2})R\ | \tag{2}$$

where $\bar{p}_s(j,k)$ is the local mean gray level around the (j,k)-th pixel (within a reference region) in the original image block, and $q(j,k)$ is the binary level of the (j,k)-th pixel in the generated image block. These two errors are combined into one single objective function as

$$e(\boldsymbol{x}_i^{(t)}) = \omega_m E_m(\boldsymbol{x}_i^{(t)}) + \omega_c E_c(\boldsymbol{x}_i^{(t)}) \tag{3}$$

where ω_m and ω_c are the weighting parameters for gray level and spatial resolution errors, respectively. The individuals' fitness is assigned by

$$f(\boldsymbol{x}_i^{(t)}) = e(\boldsymbol{x}_W^{(t)}) - e(\boldsymbol{x}_i^{(t)}) \tag{4}$$

where $e(\boldsymbol{x}_W^{(t)})$ is the combined error of the worst individual at t-th generation. The high image quality that can be achieved is the method's major strength. However, it uses a substantial amount of computer memory and processing time. High quality, visually satisfactory, halftone images are obtained with 200 individuals and 200 generations (totally 40,000 evaluations) per image block[6, 7].

Recently, Aguirre et al.[9, 10] have proposed an improved GA (GA-SRM) to overcome these two drawbacks of the conventional halftoning technique with GAs. GA-SRM is based on an empirical model of GA that applies genetic operators in parallel putting them in a cooperative-competitive stand with each other[11–14]. GA-SRM is applied to the halftoning image problem using genetic operators properly modified for this kind of problem(see **4.3**). GA-SRM with parallel adaptive dynamic block (ADB) mutation impressively reduces processing time and computer memory to generate high quality images. For example, GA-SRM with qualitative ADB using a 2 parent 4 offspring configuration needs about 6,000-12,000 evaluations per image block, depending on the input image, to obtain results similar to those achieved by the conventional image halftoning technique using GAs. These data represent a 98% reduction in the population size and an 85%-70% reduction in processing time.

3 Multiobjective Optimization (MO)

MO methods deal with finding optimal solutions to problems having multiple objectives. Let us consider, without loss of generality, a minimization multiobjective problem with M objectives:

$$minimize\ \ \boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), \cdots, g_M(\boldsymbol{x})) \tag{5}$$

where $\boldsymbol{x} \in \boldsymbol{X}$ is a solution vector in the solution space $\boldsymbol{X}$, and $g_1(\cdot), \cdots, g_M(\cdot)$ the M objectives to be minimized. Key concepts used in determining a set of solutions for multiobjective problems are dominance, Pareto optimality, Pareto set, and Pareto front. These concepts can be defined as follows.

A solution vector $\boldsymbol{y} \in \boldsymbol{X}$ is said to *dominate* a solution vector $\boldsymbol{z} \in \boldsymbol{X}$, denoted by $\boldsymbol{g}(\boldsymbol{y}) \preceq \boldsymbol{g}(\boldsymbol{z})$, if and only if $\boldsymbol{y}$ is partially less than $\boldsymbol{z}$, i.e., $\forall j \in \{1, \cdots, M\},\ g_j(\boldsymbol{y}) \leq g_j(\boldsymbol{z}) \wedge \exists j \in \{1, \cdots, M\} : g_j(\boldsymbol{y}) < g_j(\boldsymbol{z})$.

A solution vector $\boldsymbol{x} \in \boldsymbol{X}$ is said to be *Pareto optimal* with respect to $\boldsymbol{X}$ if it is not dominated by any other solution vector, i.e., $\neg\exists\ \boldsymbol{x'} \in \boldsymbol{X} : \boldsymbol{g}(\boldsymbol{x'}) \preceq \boldsymbol{g}(\boldsymbol{x})$. The presence of multiple objectives, usually conflicting among them, gives rise to a set of optimal solutions. The Pareto optimal set is defined as:

$$P = \{\boldsymbol{x} \in \boldsymbol{X} | \neg\exists\ \boldsymbol{x'} \in \boldsymbol{X} : \boldsymbol{g}(\boldsymbol{x'}) \preceq \boldsymbol{g}(\boldsymbol{x})\} \tag{6}$$

and the Pareto front is defined as:

$$PF = \{\boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), \cdots, g_M(\boldsymbol{x}))\ | \boldsymbol{x} \in P\} \tag{7}$$

The multiobjective nature of most real-world problems makes MO a very important research topic. The presence of various objectives, however, implies trade-off solutions and makes these problems complex and difficult to solve. EAs seem particularly desirable to solve MO problems because they evolve simultaneously a population of potential solutions to the problem in hand, which allows to search for a set of Pareto optimal solutions concurrently in a single run of the algorithm.

Many authors have been increasingly investigating MO using EAs (MOEA) and the number of applications has been rapidly growing. The list of contributors to the field is extensive and comprehensive reviews can be found in [1–4]. Fonseca and Fleming[1] and Horn[2] examine major MOEA techniques, Coello [3] presented a MOEA review classifying implementations from a detailed algorithmic standpoint, discussing the strengths and weaknesses of each technique. Recently, Van Veldhuizen and Lamont[4] expand upon these reviews.

4 GA-SRM extension to MO

4.1 Concept of GA-SRM

We have presented an empirical model of GA that puts parallel genetic operators in a cooperative-competitive stand with each other pursuing better balances for crossover and mutation over the course of a run[11–14]. The main features of the model are (i) two genetic operators with complementary roles applied in parallel to create offspring: Self-Reproduction with Mutation (SRM) that put emphasis on mutation, and Crossover and Mutation (CM) that put emphasis on recombination (ii) an extinctive selection mechanism, and (iii) an adaptive mutation schedule that varies SRM's mutation rates from high to low values based on SRM's own contribution to the population.

The parallel formulation of genetic operators allows the combination of crossover with high mutation rates avoiding operators' interferences, i.e. beneficial recombinations produced by crossover are not lost due to the high disruption introduced by parallel mutation and similarly the survivability of beneficial mutations are not affected by ineffective crossing over operations. The parallel application of genetic operators implicitly increases the levels of cooperation between them to introduce and propagate beneficial mutations. It also sets the stage for competition between operators' offspring.

Although the parallel formulation of genetic operators can avoid interferences between operators, it does not prevent SRM from creating deleterious mutations or CM from producing ineffective crossing over operations. To cope with these cases we also incorporate in the model the concept of extinctive selection that has been widely used in Evolutionary Strategies[17]. Through extinctive selection the offspring created by CM and SRM coexist competing for survival and reproduction as well. The poor performing individuals created by CM and SRM are eliminated. The parallel formulation of genetic operators tied to extinctive selection creates a cooperative-competitive environment for the offspring created by CM and SRM. GA-SRM based on this model remarkably improves the search performance of GA[10, 14, 18].

4.2 Multiobjective GA-SRM for Halftoning Problem

To extend GA-SRM to MO for halftoning image generation we follow a cooperative population search with aggregation selection[2, 19–22]. The population is monitored for non-dominated solutions; however, Pareto based fitness assignment[23, 24] is not directly used. A predetermined set of weights $\boldsymbol{W}$, which ponder the multiple objectives, defines the directions that the algorithm will search simultaneously in the combined space of the multiple objectives. $\boldsymbol{W}$ is specified by

$$\boldsymbol{W} = \{\boldsymbol{\omega}^1, \boldsymbol{\omega}^2, \cdots, \boldsymbol{\omega}^N\} \tag{8}$$

where N indicates the number of search directions. The k-th search direction $\boldsymbol{\omega^k}$ is a vector of nonnegative weights specified by

$$\boldsymbol{\omega}^k = (\omega_1^k, \cdots, \omega_M^k) \tag{9}$$

where M indicates the number of objectives and its components satisfy the following conditions

$$\omega_j^k \geq 0 \ (j = 1, \cdots, M) \tag{10}$$

$$\sum_{j=1}^{M} \omega_j^k = 1 \tag{11}$$

We evaluate individuals for the same two factors indicated in **2**, (number of objectives $M = 2$): (i) high gray level resolution and, (ii) high spatial resolution. Here we use the same evaluation functions E_m and E_c, respectively, proposed in [6, 7] to calculate objective values and assign its normalized values to each individual as indicated by

$$g_1(\boldsymbol{x}_i^{(t)}) = \frac{100 \times (E_m(\boldsymbol{x}_i^{(t)}) - E_m^{min})}{E_m^{max} - E_m^{min}} \tag{12}$$

$$g_2(\boldsymbol{x}_i^{(t)}) = \frac{100 \times (E_c(\boldsymbol{x}_i^{(t)}) - E_c^{min})}{E_c^{max} - E_c^{min}} \tag{13}$$

where E_m^{max}, E_m^{min}, E_c^{max}, and E_c^{min} are maximum and minimum values for E_m and E_c, respectively, obtained experimentally using various test images.

The objective values are calculated once for each individual in the offspring population. However, we keep as many fitness values as defined search directions. A combined objective value is calculated for each $\boldsymbol{\omega}^k$ $(k = 1, 2, \cdots, N)$ by

$$g^k(\boldsymbol{x}_i^{(t)}) = \sum_{j=1}^{M} \omega_j^k g_j(\boldsymbol{x}_i^{(t)}) = \omega_1^k g_1(\boldsymbol{x}_i^{(t)}) + \omega_2^k g_2(\boldsymbol{x}_i^{(t)}) \tag{14}$$

and the individuals' fitness in the k-th search direction is assigned by

$$f^k(\boldsymbol{x}_i^{(t)}) = g^k(\boldsymbol{x}_W^{(t)}) - g^k(\boldsymbol{x}_i^{(t)}) \tag{15}$$

where $g^k(\boldsymbol{x}_W^{(t)})$ is the combined objective value of the worst individual in the k-th search direction at the t-th generation.

For each search direction $\boldsymbol{\omega}^k$, CM creates a corresponding λ_{CM}^k number of offspring. Similarly, SRM creates λ_{SRM}^k offspring (see detailed information about CM and SRM implementation for halftoning problem in **4.3**). Thus, the total offspring number for each search direction is

$$\lambda^k = \lambda_{CM}^k + \lambda_{SRM}^k. \tag{16}$$

The offspring created for all N search directions coexist within one single offspring population. Hence the overall offspring number is

$$\lambda = \sum_{k=1}^{N} \lambda^k. \tag{17}$$

SRM's mutation rates are adapted based on a normalized mutants survival ratio. The normalized mutant survival ratio used in [9, 10] is extended to

$$\gamma = \frac{\sum_{k=1}^{N} \mu_{SRM}^k}{\sum_{k=1}^{N} \lambda_{SRM}^k} \cdot \frac{\lambda}{\sum_{k=1}^{N} \mu^k} \tag{18}$$

where μ^k is the number of individuals in the parent population of the k-th search direction $P^k(t)$, μ_{SRM}^k is the number of individuals created by SRM present in $P^k(t)$ after extinctive selection, λ_{SRM}^k is the offspring number created by SRM and λ is the overall offspring number as indicated in **Eq. (17)**.

We chose (μ, λ) Proportional Selection[17] to implement the extinctive selection mechanism. Since we want to search simultaneously in various directions, selection to choose the parent individuals that will reproduce either with CM or SRM is accordingly applied for each one of the predetermined search directions. Thus, selection probabilities for each search direction $\boldsymbol{\omega}^k$ are computed by

$$P_s^k(\boldsymbol{x}_i^{(t)}) = \begin{cases} f^k(\boldsymbol{x}_i^{(t)}) \;/\; \sum_{j=1}^{\mu^k} f^k(\boldsymbol{x}_j^{(t)}) & (1 \le i \le \mu^k \le \lambda^k) \\ 0 & (\mu^k < i \le \lambda) \end{cases} \tag{19}$$

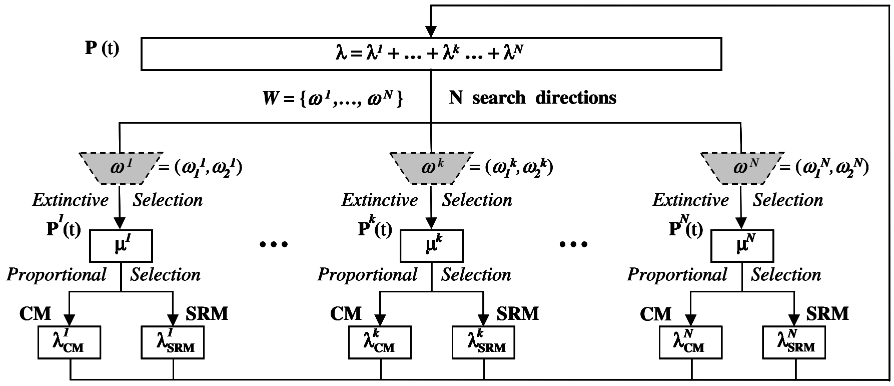


Fig. 1. Block diagram of the extended multiobjective GA-SRM

where $\boldsymbol{x}_i^{(t)}$ is an individual at generation t which has the i-th highest fitness value in the k-th search direction $f^k(\boldsymbol{x}_i^{(t)})$, μ^k is the number of parents and λ^k is the number of offspring in the k-th search direction, and λ is the overall number of offspring.

Note that for each search direction only $\lambda^k < \lambda$ individuals are created. However, the parent population μ^k is chosen among the overall λ offspring population. In this way information sharing is encourage among individuals created for neighboring search directions provided that the neighbors' fitness are competitive with the locals'. **Fig. 1** presents the block diagram of the extended multiobjective GA-SRM.

Once the offspring has been evaluated, a set of non-dominated solutions is sought for each search direction, i.e. for the k-th search direction non-domination is checked only among the offspring created for that search direction. Two secondary populations keep the non-dominated solutions. $P_{cur}(t)$ keeps the non-dominated solution obtained from the offspring population at generation t and P_{nds} keeps the set of the non-dominated solutions found through the generations. P_{nds} is updated at each generation with $P_{cur}(t)$. In the halftoning problem an image is divided into blocks and the GA is applied to each image block. Hence, the GA would generate a set of non-dominated solutions for each image block. Since we are interested in generating simultaneously various Pareto optimal "whole" images, a decision making process is integrated to chose only one solution for each search direction in each image block. Thus, among the various non-dominated solutions found for a given search direction, we chose the one that minimizes the combined error E_m and E_c in that particular direction.

4.3 CM and SRM for Halftoning Problem

In the halftoning problem an individual is represented as a $n \times n$ two-dimensional structure. In this work we use the same two-dimensional operators, CM (Crossover and Mutation) and SRM-ADB (Self Reproduction with Mutation - Adaptive Dynamic Block), presented in [9, 10] to create offspring.

CM first crosses over two previously selected parents interchanging either their rows or columns, similar to [6,7], and then it applies standard mutation inverting bits with a small mutation probability per bit, $p_m^{(CM)}$, analogous to canonical GAs. Thus, mutation in CM is of a quantitative nature after which the number of 0s and 1s may change. It may be worth trying more specialized approaches to implementing crossover, however this point will not be discussed in this work.

SRM, on the other hand, first creates an exact copy of a previously selected individual from the parent population and then applies mutation only to the bits inside a mutation block. SRM is provided with an Adaptive Dynamic-Block (ADB) mutation schedule similar to Adaptive Dynamic-Segment mutation (ADS)[12,14]. With ADB mutation is directed only to a block (square region) of the chromosome and the mutation block area $\ell \times \ell$ is dynamically adjusted to $\ell/2 \times \ell/2$ every time the normalized mutants survival ratio γ by **Eq. (18)** falls under a threshold τ. The block's side length ℓ varies from n to 2, $[n, 2]$. The offset position of the mutation block is chosen at random for each chromosome. The adaptive mechanism in SRM is designed to control the required exploration-exploitation balance during the search process.

The effect of ADB's mutation on the distribution of 0s and 1s within an individual could be of a qualitative or quantitative nature. It has been verified in [9,10] that for the halftoning problem ADB with qualitative mutation shows superior performance than ADB with quantitative mutation (i.e. bit flipping mutation). Since qualitative mutation do not change the number of 0s and 1s within an individual it has an impact only on the spatial resolution error E_c, while quantitative mutation has an impact on both E_m and E_c in **Eq. (3)** and **(14)**. Thus, qualitative mutation is less disruptive and can take better advantage of the high correlation among contiguous pixels in an image[25] contributing to a more effective search. Therefore, in this work we use ADB with qualitative mutation, which is implemented as a bit swapping process. Note that there is no need to set a mutation probability in qualitative mutation since all pairs of bits within the mutation block are simply swapped.

5 Experimental Results and Discussion

We observe and compare the performance of four kinds of GAs generating halftone images: (i) a simple GA that uses CM and proportional selection, similar to [6,7], (denoted as cGA) (ii) an extended cGA using the same multiobjective technique described in **4.2** (denoted as a moGA), (iii) a GA with SRM that uses CM, SRM and (μ, λ) proportional selection[9,10] (denoted as GA-SRM), and (iv) the extended multiobjective GA-SRM (denoted as moGA-SRM).

The GAs are applied to SIDBA's benchmark images in our simulation. The size of the original image is 256×256 pixels with $R = 256$ gray levels. An image is divided into 256 non-overlapping blocks, each one of size $n \times n = 16 \times 16$ pixels. For each block, the algorithms were set with different seeds for the random initial population.

We define 11 search directions, $N = 11$, setting $\boldsymbol{W} = \{\boldsymbol{\omega}^1, \boldsymbol{\omega}^2, \cdots, \boldsymbol{\omega}^{11}\} = \{(0.0, 1.0), (0.1, 0.9), \cdots, (1.0, 0.0)\}$. With $\boldsymbol{\omega}^1 = (0.0, 1.0)$ the search focuses exclusively in E_c's space and with $\boldsymbol{\omega}^{11} = (1.0, 0.0)$ in E_m's; whereas with $\boldsymbol{\omega}^k$, $2 \leq k \leq 10$, the search focuses in the combined space of E_c and E_m. moGA and moGA-SRM generate simultaneously one image for each direction in a single run. On the other hand, to generate the 11 images with either cGA or GA-SRM an equal number of separate runs are carried out, each one using a different $\boldsymbol{\omega}^k$ as weighting parameter. Unless stated otherwise, the GAs are set with the parameters detailed in **Table 1**[1)] and the experimental image used is "Lenna". The values set for crossover and mutation probabilities in cGA are the same used in [6, 7]. The image quality attained by the cGA with a 200 parent population and the same $T = 4 \times 10^4$ evaluations used in [6, 7] are taken as a reference for comparison in our study. The number of generations performed for each algorithm is calculated as T/λ.

Table 1. Genetic algorithms parameters

Parameter	*cGA*	*moGA*	*GA-SRM*	*moGA-SRM*
Selection	Proport.	(μ, λ) Proport.	(μ, λ) Proport.	(μ, λ) Proport.
Mating	$(\boldsymbol{x}_i, \boldsymbol{x}_j),\ i \neq j$	$(\boldsymbol{x}_i, \boldsymbol{x}_j),\ i \neq j$	$(\boldsymbol{x}_i, \boldsymbol{x}_j),\ i \neq j$	$(\boldsymbol{x}_i, \boldsymbol{x}_j),\ i \neq j$
p_c	0.6	0.6	1.0	1.0
$p_m^{(CM)}$	0.001	0.001	0.001	0.001
$\mu^k : \lambda^k$	-	1 : 1	1 : 2	1 : 2
$\lambda_{CM}^k : \lambda_{SRM}^k$	-	-	1 : 1	1 : 1
τ	-	-	0.40	0.40

Table 2 shows the average in all image blocks of the non-normalized combined errors $e^k(\boldsymbol{x}) = \omega_1^k E_m(\boldsymbol{x}) + \omega_2^k E_c(\boldsymbol{x})$ by cGA(200) after T evaluations for each search direction $\boldsymbol{\omega}^k$, $1 \leq k \leq 11$, under column $\boldsymbol{W}$. For the other algorithms under $\boldsymbol{W}$ we present the fraction of T at which the algorithm reach similar image quality (for cGA(200) these values are all 1.00 and are shown right below the combined error). Column $T^{\mathbf{W}}$ indicates the overall evaluations needed to generate the 11 images. Since the cGA generates one image at a time, it needs $11T$[2)] evaluations to generate all 11 images. The first moGA row show results by the multiobjective simple GA with a $\mu^k = 18$ parents and a $\lambda^k = 18$, $\lambda = 198$ offspring configuration. moGA simultaneously generates the 11 images and needs approximately $2.43T$[3)] to guarantee that all images would have at least the same quality as cGA(200). moGA's second row show results by moGA with a $\mu^k = 4$ parents and a $\lambda^k = 4$, $\lambda = 44$ offspring configuration. In this case population size reduction in moGA accelerates a little bit more the overall convergence and still produces better images than cGA(200). It should be noticed that population

1) GA-SRM search only in one direction at a time and the population related parameters μ^k, λ^k, λ_{CM}^k, and λ_{SRM}^k should be read without the index k

2) The entire number of evaluations required by the single objective GAs to generate all 11 images are given by the sum of the evaluations expended in each direction

3) In the case of multiple objective GAs, due to the concurrent search, the maximum number of the evaluations among all search directions determines the overall number of evaluations needed to generate all 11 images

reductions in cGA accelerates convergence but it is affected by a lost of diversity and the final image quality is inferior than cGA(200)'s[6, 7]. moGA benefits from the information sharing induced by selection (see explanation below for **Fig. 2**) and can tolerate population reductions. Compared with cGA, the results by moGA represents an enormous reduction in processing time and illustrates the benefits that can be achieved by including multiobjective techniques within GAs.

Table 2. Evaluations to generate high quality images (Lenna)

Algorithm	$\boldsymbol{W} = \{\boldsymbol{\omega}^1, \boldsymbol{\omega}^2, \cdots, \boldsymbol{\omega}^{11}\}$											$T^{\mathbf{W}}$
	$\boldsymbol{\omega}^1$	$\boldsymbol{\omega}^2$	$\boldsymbol{\omega}^3$	$\boldsymbol{\omega}^4$	$\boldsymbol{\omega}^5$	$\boldsymbol{\omega}^6$	$\boldsymbol{\omega}^7$	$\boldsymbol{\omega}^8$	$\boldsymbol{\omega}^9$	$\boldsymbol{\omega}^{10}$	$\boldsymbol{\omega}^{11}$	
combined error	121.0	111.4	100.6	89.5	78.2	66.9	55.5	44.2	32.8	21.5	10.1	-
cGA(200)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	$11T^{2)}$
moGA(18, 198)	1.43	2.43	1.65	1.27	1.21	1.00	0.86	0.76	0.70	0.65	0.72	$2.43T^{3)}$
moGA(4, 44)	1.12	2.30	1.44	1.36	1.20	1.02	0.85	0.79	0.73	0.66	0.79	$2.30T^{3)}$
GA-SRM(2, 4)	0.40	0.23	0.15	0.13	0.12	0.11	0.10	0, 09	0.09	0.08	0.08	$1.58T^{2)}$
moGA-SRM(9, 198)	1.12	1.07	0.58	0.44	0.30	0.27	0.24	0.23	0.22	0.21	0.21	$1.12T^{3)}$
moGA-SRM(2, 44)	1.56	1.03	0.50	0.30	0.20	0.16	0.15	0.13	0.12	0.12	0.12	$1.56T^{3)}$
*moGA-SRM**(2, 44)	0.96	0.92	0.40	0.31	0.22	0.17	0.15	0.14	0.13	0.13	0.13	$0.96T^{3)}$

Row GA-SRM(2,4) presents results by GA-SRM with a 2 parents and 4 offspring configuration. GA-SRM even with a very scaled down population configuration considerably reduces processing time to generate high quality images for all combinations of weighting parameters. GA-SRM, for this particular image, would need approximately $1.58T^{2)}$ to generate all 11 images. Note that GA-SRM sequentially generating the 11 images is faster than moGA.

The first moGA-SRM row show results by the multiobjective proposed GA-SRM with a $\mu^k = 9$ parents and a $\lambda^k = 18$, $\lambda = 198$ offspring configuration. Compared with moGA we can see that the inclusion of SRM notoriously increases the multiobjective algorithm's performance needing no more than $1.12T^{3)}$ to generate the 11 images, which is faster than GA-SRM. Results by a scaled down population configuration is shown in row moGA-SRM(2,44) that represents a $\mu^k = 2$ parents and a $\lambda^k = 4$, $\lambda = 44$ offspring configuration. The population size reduction in moGA-SRM accelerates convergence in all but one search direction (see under $\boldsymbol{\omega}^1$) and the overall evaluation time is similar to GA-SRM. From GA-SRM and moGA-SRM results we see that parallel mutation SRM can greatly improve the performance of single objective as well as multiobjective genetic algorithms in the halftoning problem.

We observe that moGA(2,44), which uses CM but not SRM, only for $\boldsymbol{\omega}^1$ produces faster convergence than moGA-SRM ($e^1 = 0.0E_m + 1.0E_c$). It seems that CM alone is particularly useful for searching in E_c's search space. However, when the search involves both E_m's and E_c's spaces the interaction of CM and SRM produce better results. We conduct an experiment in which we favor CM's offspring over SRM's only in the $\boldsymbol{\omega}^1$ direction. In row moGA-SRM*(2,44) we show results using a configuration that creates offspring in $\boldsymbol{\omega}^1$ direction only with CM, i.e. $\lambda^1_{CM} = 4$, $\lambda^1_{SRM} = 0$ and $\lambda^k_{CM} = 2$, $\lambda^k_{SRM} = 2$ for $2 \leq k \leq 11$. This has the effect of accelerating convergence in $\boldsymbol{\omega}^1$ search direction and therefore reducing the overall evaluation time to $0.96T$.

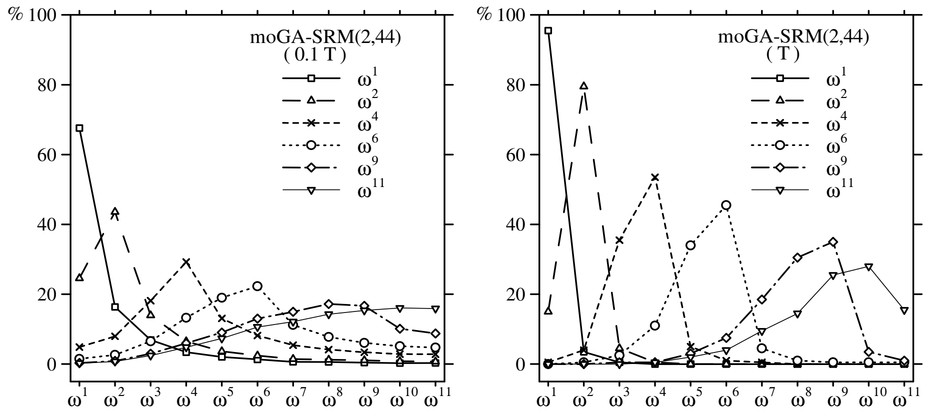
E_m and E_c represent fitness landscapes with different degree of difficulty for the GAs. E_m's landscape is smoother than E_c's and the GAs are expected to converge faster in E_m's direction. This is corroborated by the results obtained by the GAs. In **Table 2** we can see that for $\boldsymbol{\omega}^k$ with $k \geq 6$, E_m's directions, the algorithms need less time to converge. It should be specially noticed that moGA-SRM for those directions finds high quality images in less than $0.2T$. This behavior and the results by the last experiment mentioned above suggest that it may be worth trying dynamic configurations so that more resources could be assigned to those directions that require more time to converge accelerating the overall time needed to generate images simultaneously.

Table 3. Actual percentage of evaluations expended in each search direction

	$\boldsymbol{W} = \{\boldsymbol{\omega}^1, \boldsymbol{\omega}^2, \cdots, \boldsymbol{\omega}^{11}\}$										
Algorithm	$\boldsymbol{\omega}^1$	$\boldsymbol{\omega}^2$	$\boldsymbol{\omega}^3$	$\boldsymbol{\omega}^4$	$\boldsymbol{\omega}^5$	$\boldsymbol{\omega}^6$	$\boldsymbol{\omega}^7$	$\boldsymbol{\omega}^8$	$\boldsymbol{\omega}^9$	$\boldsymbol{\omega}^{10}$	$\boldsymbol{\omega}^{11}$
$cGA(200)$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$moGA(18, 198)$	13.0	22.1	15.0	11.5	11.0	9.1	7.8	6.9	6.4	5.9	6.5
$moGA(4, 44)$	10.2	20.9	13.1	12.4	10.9	9.3	7.7	7.2	6.6	6.0	7.2
$GA\text{-}SRM(2, 4)$	40.0	23.0	15.0	13.0	12.0	11.0	10.0	9.0	9.0	8.0	8.0
$moGA - SRM(9, 198)$	10.2	9.7	5.3	4.0	2.7	2.5	2.2	2.1	2.0	1.9	1.9
$moGA - SRM(2, 44)$	14.2	9.4	4.5	2.7	1.8	1.5	1.4	1.2	1.1	1.1	1.1
$moGA - SRM^*(2, 44)$	8.7	8.4	3.6	2.8	2.0	1.5	1.4	1.3	1.2	1.2	1.2

In **Table 2** moGA's and moGA-SRM's rows show the evaluations expended by the algorithm in all search directions. The actual percentage of the evaluations expended in each search direction is shown in **Table 3**. From this table it can be seen that with the multiobjective algorithms there is a substantial reduction of the actual number evaluations for each search direction. These reductions are explained by the information sharing induced by the selection process. As mentioned in **4.2** and indicated by **Eq. (19)**, the individuals with higher fitness in a specific direction are selected as parents. Thus, the individuals chosen to be parents for the k-th search direction at generation t may have been created for neighboring directions at generation t-1. To verify this point we also observe the composition of the parent population for each search direction. **Fig. 2** shows the average distribution for some of the $\boldsymbol{\omega}^k$ directions after $0.1T$ and T evaluations, respectively. For example, in **Fig. 2(a)**, the parent population of $\boldsymbol{\omega}^4$ is in average composed by 18% of individuals coming from $\boldsymbol{\omega}^3$, 30% from $\boldsymbol{\omega}^4$ itself, and 13% from $\boldsymbol{\omega}^5$. From these figures we can see that each search direction benefits from individuals that initially were meant for other neighboring directions. This information sharing pushes forward the search reducing convergence times. Looking at **Fig. 2(a)** and **Fig. 2(b)** we can see that the information sharing is higher during the initial stages of the search.

Fig. 3 illustrates typical transitions of the non-normalized combined error $e(\boldsymbol{x})$ over the number of evaluations for some of the search directions by the GAs. The plots are cut after T evaluations. From these figures it can be visually



(a) after $0.1T$ (b) after T

Fig. 2. moGA-SRM's average parent population distribution

appreciated the higher convergence velocity and higher convergence reliability (lower errors) by the algorithms that include SRM, GA-SRM and moGA-SRM. In general, moGA is faster than the cGA, but their final image quality tends to be the same. Also, it should be noticed that results by moGA and moGA-SRM are achieved simultaneously in one run (thus, T for these algorithms indicates the evaluations expended in all search directions).

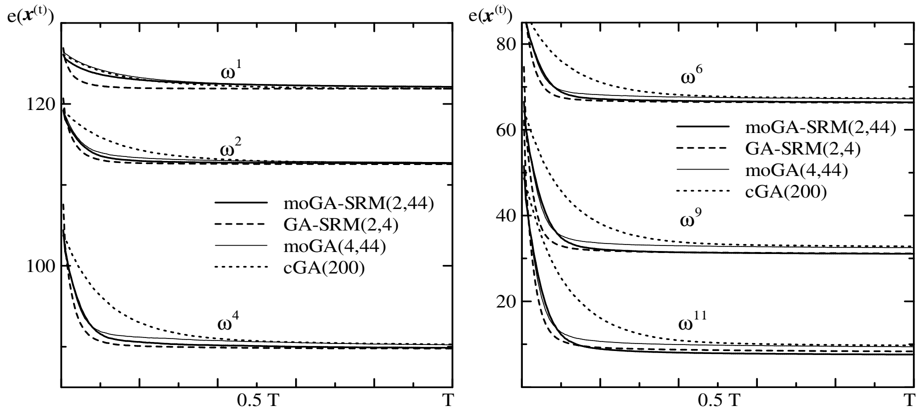


Fig. 3. Error transition for various $\boldsymbol{\omega}^k$

Fig. 4 show the original image "Lenna" and the images generated by two conventional halftoning techniques: ordered dithering (screen) and error diffusion[8]. **Fig. 5** show some of the simultaneously generated images by moGA-SRM. From these figures we can see that moGA-SRM generates more pleasant images to the human observer than traditional techniques. Another point to be remarked is that traditional halftoning techniques can generate only one image. On the other hand, among the images generated by moGA-SRM there is a gradual difference according to spatial and gray level resolution, which makes the

(a) original image (b) ordered dithering (screen) (c) error diffusion

Fig. 4. Lenna's original and generated images by two conventional techniques

Fig. 5. Lenna's simultaneously generated images by moGA-SRM*(2,44) after $0.96T$

GA based halftoning technique more flexible to users' requirements as well as more robust to constraints imposed by displaying and printing devices.

With regards to processing time, running software implementations of the algorithms in a Pentium III processor (600 MHz), to generate one image conventional techniques need only few seconds while GA-SRM (also implemented in software) needs about 8 minutes. Note that GA based techniques in this study process one block at a time always starting with random initial populations. Due to the high correlation among neighbor blocks of an image, reductions on processing time are expected by using previously generated image blocks in the initial populations of the subsequent blocks. However it is clear that, from a processing time standpoint, in order to apply GA based halftoning techniques on-line they must be improved further to reduce as much as possible the number of evaluations needed to generate higher quality images. Also, the GA's final implementation for industrial application must be in hardware.

Finally, we should also say that similar results were obtained for other SIBDA's benchmark images.

6 Conclusions

In this work we have extended an improved GA (GA-SRM) to a multiobjective optimization GA (moGA-SRM) for the image halftoning problem aiming to simultaneously generate halftone images with various combinations of gray level and spatial resolution.

GA-SRM is based on an empirical model of GA that puts parallel genetic operators in a cooperative-competitive stand with each other. To extend GA-SRM we follow a cooperative population search with aggregation selection preserving the fundamental features of the cooperative-competitive model. We compare the performance of four genetic algorithms generating halftone images: (i) a single objective simple GA (cGA), (ii) a single objective GA-SRM, (iii) a multiobjective simple GA (moGA), (iv) the proposed multiobjective GA-SRM (moGA-SRM).

From our experimental results we observe that multiobjective techniques benefit from information sharing and can greatly reduce processing time to generate simultaneously high quality images. To generate 11 images moGA requires only about 21% of the evaluations used by cGA. The cooperative-competitive model for parallel operators helps to increase the performance of single and multi objective GAs in this problem reducing even further processing time. GA-SRM requires about 15% and moGA-SRM about 9% of the evaluations used by cGA.

As future works, important issues to be explored related to the halftoning problem are (i) the effect of the definition of the weights set on the algorithm's stability and convergence, (ii) dynamic and parallel hierarchical configurations for moGA-SRM in order to accelerate the overall time needed to generate images simultaneously. Also, we are planning to continue studying moGA-SRM's behavior in a wider range of problems that include more than two objectives[18] and use it in other real world applications.

References

1. C. Fonseca and P. Fleming, "An overview of Evolutionary Algorithms in Multiobjective Optimization", *Evolutionary Computation*, 3(1):1-16, 1995.
2. J. Horn, "Multicriterion Decision Making", *Handbook of Evolutionary Computation*, Oxford University Press, Volume 1, pp.F1.9:1-F1.9:15, 1997.
3. C. Coello, "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques", *Knowledge and Information Systems*, 1(3):269-308, 1999.
4. D. Van Veldhuizen and G. Lamont, "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art", *Evolutionary Computation*, 8(2):125-147, 2000.

5. K. S. Tang, K. F. Man, S. Kwong and Q. He, "Genetic Algorithms and Their Applications", *IEEE Signal Processing Magazine*, vol.13, no. 6, pp.22-37, Jun. 1996.
6. N. Kobayashi and H. Saito, "Halftoning Technique Using Genetic Algorithm", *Proc. IEEE ICASSP'94*, vol.5, pp.105-108, Apr. 1994.
7. N. Kobayashi and H. Saito, "Halftone Algorithm Using Genetic Algorithm", *IEICE Trans.*, vol.J78-D-II, no. 10, pp.1450-1459, Oct. 1995 (in Japanese).
8. R. Ulichney, *Digital Halftoning*, MIT Press, Cambridge, 1987.
9. H. Aguirre, K. Tanaka and T. Sugimura, "Accelerated Image Halftoning Technique Using Improved Genetic Algorithm with Tiny Populations", *Proc. IEEE SMC'99*, vol.IV, pp.905-910, Oct. 1999.
10. H. Aguirre, K. Tanaka and T. Sugimura, "Accelerated Halftoning Technique Using Improved Genetic Algorithm", *IEICE Trans.*, vol.E83-A, no. 8, pp.1566-1574, Aug. 2000.
11. H. Aguirre, K. Tanaka and T. Sugimura, "Cooperative Crossover and Mutation Operators in Genetic Algorithms ", *Proc. GECCO'99*, p.772, Jul. 1999.
12. H. Aguirre, K. Tanaka and T. Sugimura, "Cooperative Model for Genetic Operators to Improve GAs", *Proc. IEEE ICIIS'99*, pp.98-106, Nov. 1999.
13. H. Aguirre, K. Tanaka, T. Sugimura and S. Oshita, "Cooperative-Competitive Model for Genetic Operators: Contributions of Extinctive Selection and Parallel Genetic Operators", *Proc. Late Braking Papers GECCO 2000*, pp.6-14, Jul. 2000.
14. H. Aguirre, K. Tanaka and T. Sugimura, "Empirical Model with Cooperative-Competitive Genetic Operators to Improve GAs: Performance Investigation with 0/1 Multiple Knapsack Problems", IPSJ Journal, vol.41, no.10, pp.2837-2851, Oct. 2000.
15. J. H. Holland, *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press, 1975.
16. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, 1989.
17. T. Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford Univ. Press, 1996.
18. H. Ishibashi, H. Aguirre, K. Tanaka and T. Sugimura, "Multi-objective Optimization with Improved Genetic Algorithm", *Proc. IEEE SMC2000*, pp.3852-3857, Oct. 2000.
19. J. Schaffer and J. Grefenstette, "Multiobjective Optimization with Vector Evaluated Genetic Algorithms", *Proc. 1st Int. Conf. on Genetic Algorithms*, pp.93-100, Jul. 1985.
20. M. Fourman, "Compaction of Symbolic Layout Using Genetic Algorithms", *Proc. 1st Int. Conf. on Genetic Algorithms*, pp.141-153, Jul. 1985.
21. F. Kurwase, "A Variant of Evolution Strategies for Vector Optimization", *Lecture Notes in Computer Science*, vol.496, pp.193-197, Springer-Verlag, 1991.
22. T. Murata and H. Ishibuchi, "MOGA:multi-objective genetic algorithms", *Proc. 2nd IEEE Int. Conf. on Evolutionary Computation*, pp.289-294, Nov. 1995.
23. J. Horn, N. Nafpliotis and D. E. Goldberg, "A Niched Pareto Genetic Algorithm for Multiobjective Optimization", *Proc. 1st Int. IEEE Conf. on Evolutionary Computation*, IEEE Service Center, vol.1, pp.82-87, 1994.
24. N. Srinivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms", *Evolutionary Computation*, 2(3):221-248, 1994.
25. R. C. Gonzalez and R. E. Wood, *Digital Image Processing*, Addison-Wesley, 1992.