Multi-objective Evolutionary Algorithms based Operation Sequence Design for Image Segmentation

Diego P. Pinto-Roa, Julio César Mello-Román,
José Luis Vázquez Noguera,
Ramón Quintana, Fredy Roa,
Universidad Nacional de Asunción, Facultad Politécnica,
San Lorenzo, Paraguay,
{*dpinto, juliomello, jlvazquez*}*@pol.una.py*
{*fredroasam,ramonquintana*} *@gmail.com*

and

Pedro Esteban Gardel-Sotomayor
Universidad Católica Nuestra Señora de la Asunción, Facultad de Ciencias y Tecnología
Hernandarías, Paraguay
pedro.gardel@uc.edu.py

Abstract

Image segmentation transforms an image into a more understandable representation by grouping pixels with common characteristics, making it easier to identify regions of interest. There is no optimal segmentation method for all cases, which makes it challenging to select the appropriate technique for each image. We propose using a Multi-Objective Evolutionary Algorithm (MOEA) to generate sequences of operations adapted to specific applications. The evolutionary process was guided by objective functions based on Receiver Operating Characteristics (ROC) Curve analysis, using maximization of sensitivity and specificity. The experiments were performed with three types of images: cells (type A, B, C, and D), melanoma images (benign and malignant), and retinal ophthalmoscopic images. The results show that the algorithm achieves a sensitivity (TPR) and specificity (TNR) of up to 1.0 in the segmentation of images of type A and D cells, a sensitivity of 1.0 and specificity of 0.9765 in the segmentation of images of benign melanoma, a sensitivity of 0.9857 and specificity of 0.9825 in the segmentation of malignant melanoma images, and sensitivity of 0.8931 and specificity of 0.9104 in the extraction of retinal veins in ophthalmoscopic images.

Keywords: image segmentation, sequences of operations, sensitivity, specificity, MOEA.

1 Introduction

Given that no single algorithm can fulfill all image segmentation requirements, various proposed techniques and strategies address this problem, each with resulting features that may or may not be advantageous, depending on the specific application [1,2]. One feasible approach to tackle the segmentation issue through existing techniques and strategies is to combine them automatically in an appropriate sequence, allowing for desirable outcomes for specific applications. However, this is quite a complex task due to the multitude of parameters and variables inherent to each technique, the emergent properties that may arise when combining them, and the peculiarities of each possible application.

The outcomes of the segmentation must meet the demands of each application, necessitating an exhaustive analysis of each application and the characteristics of the available segmentation techniques to ascertain which methods are apt for each application. This requires a substantial investment of time and effort if done manually.

The proposed Multi-Objective Evolutionary Algorithm (MOEA) employs the simultaneous maximization of sensitivity and specificity as objective functions. In our experiments, we use images sourced from databases of melanoma images, ophthalmological images of retinal veins, and images featuring various types of cells. The result is a set of non-dominated solutions or a Pareto set, where each solution is a sequence of operations generated by the algorithm and classified based on the predefined objectives and evaluated on the image sets used for training. It is noteworthy that this work expands upon the contribution of the preliminary version [3] by presenting results in more complex case studies.

This work aims to experimentally ascertain the feasibility of generating operation sequences using MOEA for segmenting images with shared characteristics. Unlike other research [1,2,4–7], this approach yields valid results for binary, gray-scale, and color images.

An additional aim of this study is to engineer robust sequences of operations through an automated process. The intention is to ensure that the derived solutions are not only effective but also versatile, proving their applicability across a diverse array of images that exhibit shared characteristics. The overarching goal is to guarantee that the results are consistently satisfactory across these various images. To facilitate this, the training phase of our methodology employs an extensive set of image pairs, strategically chosen to represent a wide spectrum of shared traits, thereby enhancing the robustness and adaptability of the segmentation solutions.

The main contributions of this research to the state of the art can be summarized as follows:

- Development of an approach for automatically generating operation sequences for binary, grayscale, and color image segmentation, applicable across various case studies.
- Utilization of a supervised training mechanism with image pairs (original and ideal segmentation), enhancing the robustness of the generated sequences for diverse images.
- Incorporation of a broad range of operation types to enhance the expressive capabilities of the sequences, at the expense of a larger search space. The approach is scalable regarding the number of operations and flexible in the choice of operations for sequence composition.
- Adaptability of the method for scopes beyond image segmentation, by replacing ideal segmentation pairs with other types of images that can be derived from the original through operational sequences.

The paper is organized in the following manner: Section 2 reviews pertinent literature and introduces fundamental concepts. Section 4 delineates the methodology of the proposed approach. The experimental setup and the corresponding results are detailed in Section 5. Section 7 discusses the principal findings and implications of this work.

2 Related Works

Due to the increasing need for manipulating and processing digital information, specifically digital images, it has become almost obligatory to use techniques that take advantage of the computer resources available today to automate such processing.

In recent years, several techniques and approaches have been proposed that use of computer resources and the evolutionary techniques that have been widely disseminated through them.

Zhou *et al.* [8] propose a mechanism that classifies images of segmentation of dermoscopic lesions based on specimens, where segmented images are selected as specimens and associated with them a histogram that is used to define the probabilities that the pixels of an image are classified as part of the lesion or not.

Schmid-Saugeona *et al.* [9] present a computer-assisted diagnostic system for skin lesions, with solutions for detecting the lesion and quantifying the degree of symmetry in the shape of the lesions.

Silveira *et al.* [10] perform a comparison between dermoscopic image segmentation techniques based on a threshold, edge detection, and region classification is presented.

Beuren *et al.* [1] apply a melanoma segmentation approach based on morphological operations on color images. A threshold technique is applied to segment regions of interest in melanoma. Binary morphology techniques are used to filter out the region of interest. The approach has been tested on two databases of benign and malignant melanomas, and the results have been compared with the corresponding ideal segmentation.

Roberts and Claridge [2] present an approach that evolves a gray-scale medical image segmentation procedure using genetic programming and training mechanisms from ideal images and segmentations.

Quintana *et al.* in [7] apply Genetic Programming to evolve sequences of morphological operations for binary images that given an input image, will convert it to an output image containing a specific characteristic or region of the input image.

Hong *et al.* [11] propose an approach based on genetic algorithms that optimizes a set of standard filters by determining their types, order, and associated noise factors.

In [5], an automatic image enhancement method for grayscale images, based on an evolutionary optimization process, is introduced.

Wang and Tan propose in [6] to apply genetic programming to generate grayscale image enhancement procedures for subsequent segmentation. The procedures are generated from an input image and its corresponding ideal segmentation. The enhancement procedures generated by the genetic programming are applied to the grayscale images in order to enhance image characteristics that facilitate their subsequent segmentation. In this work seven morphological operations and five logical operations are used. The study adopts a mono-objective approach, focusing on optimizing a single criterion, and conducts tests on a set of infrared finger vein images, selected for their unique imaging challenges.

In [12], Lezcano et al. propose an evolutionary multi-objective approach for image processing. This method generates a sequence of mathematical morphology operations applied to binary images. From an original image, it produces a target image that highlights specific features or regions of the original image.

Although the previously mentioned techniques have proven effective in specific contexts, they present limitations that prevent their universal application to the image segmentation problem. For example, approaches based on genetic programming and evolutionary algorithms are often optimized for particular image types or criteria, reducing their generalizability and effectiveness in applications with different characteristics. Furthermore, many of these works focus on mono-criterion objectives, such as grayscale image enhancement or threshold-based segmentation, without simultaneously considering sensitivity and specificity as objective functions, which is crucial for precise segmentation. On the other hand, methodologies that use regular morphological elements and standard filters do not offer the flexibility to adapt to the complexities of different image types and conditions. This study, therefore, focuses on developing a Multi-Objective Evolutionary Algorithm (MOEA) that integrates a variety of operations representative of conventional segmentation techniques and simultaneously maximizes sensitivity and specificity. This approach is necessary to address the diversity of features in accurate data more effectively, providing a robust and adaptable solution to multiple segmentation scenarios.

3 Background

3.1 Image Segmentation

When an image is automatically analyzed or interpreted, a mechanism must be in place to uniquely identify the pixels that correspond to a region or feature of interest. The process of identifying these pixels is known as segmentation [13]. Segmentation is the process of grouping pixels with common attributes.

Image segmentation is one of the first steps in most image processing tasks, its objective is to obtain a more meaningful or simplified representation of the image, and its result significantly impacts subsequent steps.

3.2 Pixel Classification

3.2.1 Image

An image is defined as a bi-dimensional function $I(x, y)$, where x and y are spatial (Cartesian) coordinates, and the *amplitude* of I in any coordinate pair (x, y) is called the intensity of the image at that point [14]. When the values of x, y and the amplitude values of I are all discrete finite values, we can say that we are dealing with a digital image. If the pixels are arranged in a grid or matrix of size $M \times N$, the domain of the function is given by:

$$D = \{(x, y) : x = 1, 2, \ldots, M,\ y = 1, 2, \ldots, N\} \tag{1}$$

The resulting segmented images will be presented as binary images so we define the range for the binary images as

$$R = \{0, 1\}. \tag{2}$$

3.2.2 Pixel Classification

This section considers an image PI, and its corresponding segmentation I_m, both of $M \times N$ dimensions.

True Positives: A true positive occurs when a pixel in the segmented region of the proposed image coincides in value with the pixel in the same position in the segmented region of the ideal image. From this, we can define the function TP that us allows to quantify the true positives that occur between the proposed and ideal image:

$$\mathrm{TP}(PI,\ I_m) = \sum_{x=1}^{M}\sum_{y=1}^{N} \mathrm{IsTP}(PI,\ I_m,\ x, y) \tag{3}$$

$$\mathrm{IsTP}\left(PI,\ I_m, x, y\right) = \begin{cases} 1 & \text{If } PI(x,y) = I_m(x,y) = 0 \\ 0 & \text{Otherwise} \end{cases} \tag{4}$$

True Negatives: Analogous to the true positives, a true negative occurs when a pixel corresponding to the background of the segmentation in the proposed image coincides in value with the pixel located in the same position as the background of the segmentation in the ideal image. We define the function TN that quantifies the true negatives between the proposed image and the ideal image:

$$\mathrm{TN}(PI,\ I_m) = \sum_{x=1}^{M}\sum_{y=1}^{N} \mathrm{IsTN}(PI,\ I_m,\ x, y) \tag{5}$$

$$\mathrm{IsTN}\left(PI,\ I_m, x, y\right) = \begin{cases} 1 & \text{If } PI(x,y) = I_m(x,y) = 1 \\ 0 & \text{Otherwise} \end{cases} \tag{6}$$

False Positives: False positives are indicators of error, which occur when a pixel in the proposed image has been classified as part of the segmented region. However, the pixel located in the same position as the ideal image corresponds to the background of the segmentation, differing in value. The function FP quantifies the false positives between the ideal and proposed images.

$$\mathrm{FP}(PI,\ I_m) = \sum_{x=1}^{M}\sum_{y=1}^{N} \mathrm{IsFP}(PI,\ I_m,\ x, y) \tag{7}$$

$$\mathrm{IsFP}\left(PI,\ I_m, x, y\right) = \begin{cases} 1 & \text{If } PI(x,y) = 0 \wedge I_m(x,y) = 1 \\ 0 & \text{Otherwise} \end{cases} \tag{8}$$

False Negatives: False negatives are indicators of error, which occur when a pixel in the proposed image has been classified as part of the background of the segmented region. However, the pixel at the same position in the ideal image corresponds to the segmented region, and there is a difference in value between them.

$$\mathrm{FN}(PI,\ I_m) = \sum_{x=1}^{M}\sum_{y=1}^{N} \mathrm{IsFN}(PI,\ I_m,\ x, y) \tag{9}$$

$$\mathrm{IsFN}\left(PI,\ I_m, x, y\right) = \begin{cases} 1 & \text{If } PI(x,y) = 1 \wedge I_m(x,y) = 0 \\ 0 & \text{Otherwise} \end{cases} \tag{10}$$

To determine the quality of the segmentation, each proposed segmentation called PI is compared with an ideal segmentation called II. That is a comparison between two images. To assess this comparison quantitatively, we use two metrics: the $Sensitivity$ and the $Specificity.$

3.2.3 Sensitivity

In this paper, sensitivity is the probability of correctly classifying a pixel whose actual state is defined as positive concerning the condition being studied by the test, also referred to as a fraction of true positives (TPR). In the case of images, a *pixel* is defined as positive when it is part of the segmented region. Therefore the sensitivity is the probability of classifying a pixel as part of the segmented region.

$$\mathrm{TPR}\left(PI, I_m\right) = \frac{\mathrm{TP}\left(PI, I_m\right)}{\mathrm{TP}\left(PI, I_m\right) + \mathrm{FN}\left(PI, I_m\right)} \tag{11}$$

3.2.4 Specificity

In this paper, specificity is the probability of correctly classifying a pixel whose actual state is defined as negative. It is equal to the result of subtracting the fraction of false positives from one. For the case of the pixels in an image, a pixel is defined as negative when it belongs to the background of the image (i.e., is outside the segmented region) and establishing specificity as the probability of classifying a pixel as part of the background of the segmentation, i.e., as a pixel that does not correspond to the segmented region.

$$\mathrm{TNR}\left(PI, I_m\right) = \frac{\mathrm{TN}\left(PI, I_m\right)}{\mathrm{TN}\left(PI, I_m\right) + \mathrm{FP}\left(PI, I_m\right)} \tag{12}$$

3.2.5 Use of Sensitivity and Specificity

The analysis of sensitivity and specificity (ROC analysis) [15, 16] of a diagnostic test is a powerful analysis tool for classification problems, providing specialists with more detailed information about the validity of a solution. In addition, using sensitivity and specificity as targets in the assessment task, rather than the accuracy rate, provides a mathematically equivalent method because these two factors are proportionally included in the overall accuracy calculation. Therefore, a careful and balanced consideration of both sensitivity and specificity is essential for test validity. Relying solely on overall accuracy, despite its simplicity, is not sufficient because its dependence on prevalence can lead to misleading conclusions.

Recent studies in areas such as screening, classification, and ROC analysis have used multi-target algorithms in order to optimize the sensitivity and specificity of a given classifier [17], [18], [19].

3.3 Specimen, Input and Operations Sequences

3.3.1 Specimen

In this work the set E is formed by elements Ie called specimen, which is composed by two images: the original image I_o and the target image I_m. The original image is the image to be processed, and the target image represents the ideal segmentation of the reference image.

$$Ie = (I_o, I_m) \tag{13}$$

3.3.2 Input

Given k specimens that form a k-tuple representing the E input to the evolutionary process. This input will be used during the training process.

$$E = \{Ie_1, Ie_2, \ldots, Ie_{k-1}, Ie_k\} \tag{14}$$

Figure 1 graphically illustrates the concepts of 'specimen' and 'input'.

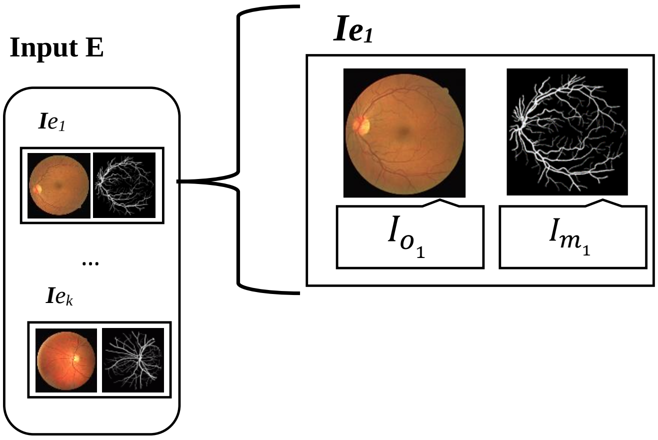


Figure 1: Specimens and Input. A 'specimen' is defined as a pair of images $Ie_i = (I_{o_i}, I_{m_i})$, as illustrated in the figure. Here, I_{o_i} represents the original image, and I_{m_i} is its corresponding ideal segmentation.

3.3.3 Operations Sequence

There is a P set of operations from the field of digital image processing. P contains the components that will make up each sequence of operations that will transform each image I_o into the image I_m. The operations that make up this set can be seen in Tables 1 and 2. In addition, for some types of operations, specifically those of a morphological type, there are two sets of regular and irregular structuring elements. All the characteristics mentioned above favour the expressive capacities of the sequences of operations (solutions). In addition, there are two sets of regular and irregular structuring elements for some types of operations, specifically those of a morphological type. All the characteristics mentioned above favor the expressive capacities of the sequences of operations (solutions).

This sequence of operations, which we call S, is an ordered tuple of n operations defined as:

$$S = (s_1, s_2, \ldots, s_{n-1}, s_n) \tag{15}$$

where $s_j \in P$.

We can also see S as a function that receives as input an image I_o, to which a process applies successively, and in order, the operations that make up S producing as a result an output image that we call PI.

$$PI \Leftarrow S(I_o) \tag{16}$$

$$\begin{gathered} \text{where } I^{j+1} \leftarrow s_j(I^j) \\ \text{with } \; j = 1, 2, ..., n-1, n \\ \text{and } \; I^1 = I_o, I^n = PI \end{gathered}$$

3.4 Multi-objective optimization

In the real world, many problems require the simultaneous optimization of several performance measures or objectives. While it is possible to optimize these objectives separately in some cases, more often than not, they are in conflict. For example, optimizing one objective to its absolute best often results in sub-optimal performance, or even unacceptable outcomes, in the others. In situations with conflicting objectives, a compromise is necessary. This implies that the solutions might be sub-optimal for each individual objective, but collectively acceptable for all objectives.

As the number of conflicting objectives grows, the task of finding an acceptable compromise solution becomes significantly more complex.

A general multi-target optimization problem is defined as a set of n variables representing the decision variables of the problem, a set of k target functions, and a set of q constraints [20]. The objective functions and constraints are functions of the decision variables. The multi-objective optimization problem can then be defined as:

Optimize

$$y = f(x) = [f_1(x), f_2(x), \ldots, f_p(x)] \tag{17}$$

subject to:

$$e(x) = [e_1(x), e_2(x), \ldots, e_q(x)] \geq 0 \tag{18}$$

where:

$$\begin{aligned} x &= (x_1, x_2, \ldots, x_n) \in X \subseteq \mathbb{R}^n \\ y &= (y_1, y_2, \ldots, y_p) \in Y \subseteq \mathbb{R}^p \end{aligned}$$

x is a *n-dimensional decision vector*, X denotes the decision space, y is a *p-dimensional objective vector*, e Y denotes the objective space. Optimizing, depending on the problem, can mean either minimizing or maximizing. The set of constraints $e(x) \geq 0$ determines the set of feasible solutions X_f and its corresponding set of feasible target vectors Y_f.

Given two decision vectors, u, $v \in X$. It is said that u dominates v (denoted as $u \succ v$) if it is better or equal to v in each of the objectives and strictly better in at least one objective. This is,

We say that two vectors u and v are not comparable, if and only if $u \not\succ v \wedge v \not\succ u \wedge u \neq v$.

In this scenario, u and v are vectors representing different solutions, and neither can be considered superior to the other across all objectives. While u and v are not strictly identical, they are equally viable since neither dominates the other in a multi-objective context. Consequently, u and v are said to be not comparable, denoted as $u \sim v$. The notation $u \rhd v$ is used to indicate that u either dominates or is not comparable to v.

Pareto optimality can be defined in terms of a decision vector $x \in X_f$, it is said that x is non-dominated with respect to a set $Q \subseteq X_f$ if and only if: $\forall v \in Q : (x \rhd v)$.

In case x is non-dominated with respect to the whole set X_f, and only in that case, it is said that x is a Pareto 'optimal' solution. From this concept we can define the optimal Pareto set P^* to be the set of non-dominated decision vectors that belong to X_f, that is: $P* = \{x \in X_f \; : \forall v \in X_f : (x \rhd v)$.

The corresponding set of objective vectors $P_{true} = F(P^*)$ conforms the Optimal Pareto Front.

3.5 Evolutionary Algorithms

Evolutionary algorithms are search and optimization strategies inspired by biological evolution and natural selection processes. The fundamental idea of evolutionary algorithms is to maintain a population of individuals, where each individual represents an alternative solution to the problem. The individuals in this population undergo an evaluation process that determines their degree of quality as a solution to the problem.

For this paper, the NSGA-II evolutionary algorithm has been chosen (see [21]), and any other MOEA can be selected for future testing.

The NSGA-II algorithm, as presented in [21], is an advanced version of the original NSGA (Non-dominated Sorting Genetic Algorithm) algorithm [22]. NSGA-II enhances the process with elitism, which involves selecting the most promising individuals from both parent and offspring populations for the next generation. Additionally, it introduces a fast, non-dominance based sorting mechanism, which efficiently categorizes solutions based on their level of dominance.

The principal innovation of NSGA-II lies in its front-based ordering mechanism. In this approach, the first front consists of all non-dominated solutions within the population. Subsequent fronts are determined iteratively; the second front, for instance, is formed by identifying non-dominated individuals from the remaining population after removing those in the first front. This classification process continues until each solution in the population is assigned to a front. A rank is assigned to each solution, indicating its level of non-dominance, with lower ranks signifying more optimal individuals. To preserve population diversity, NSGA-II implements the Crowding Distance calculation. This metric evaluates how closely solutions are grouped in the solution space, aiding in the selection of diverse individuals. The selection mechanism in this study is a binary tournament, where the individual with the lower rank is preferred. In cases where individuals have the same rank, the one with the higher Crowding Distance, indicating a less crowded region, is selected.

4 Solution Proposal

4.1 Problem Formulation

"Given a set of operations and a set of image pairs, where each pair is composed of a reference image and its corresponding ideal segmentation, which we call the target image, an optimal sequence of operations must be found, which receives as input each reference image of the set of images, and produces its corresponding target image."

The problem can be expressed as a multi-objective MOP optimization problem where given an input set $E = \{Ie_i\}_{i=1}^{k}$ we seek to identify sequences of S operations that simultaneously maximize the following target functions:

$$F_1 = \frac{1}{k}\sum_{i=1}^{k} \mathrm{TPR}\left(PI_i, I_{m_i}\right) \tag{19}$$

$$F_2 = \frac{1}{k}\sum_{i=1}^{k} \mathrm{TNR}\left(PI_i, I_{m_i}\right) \tag{20}$$

where $PI_i = S(Io_i)$ and subject to the following constraints:

1. (R_1) Required color image: It is given when an operation requires a color image, and receives as input a gray-scale or binary image.

2. (R_2) Gray-scale image required: This is when an operation requires a gray-scale image, and receives a different type of image as input.

3. (R_3) Required binary image: It is given when an operation requires a binary image, and receives as input any other type of image.

The primary constraints in this context are inherent at the level of the operations composing the sequences; that is, they are embedded within each operation's requirements, particularly regarding the type of input parameters required. Considering this, and given the sequential nature of these operations, the output of one operation typically serves as the input for the next. However, if an operation within a sequence receives an invalid input parameter, the sequence is abruptly interrupted. Such sequences, unable to be executed due to these invalid inputs, are deemed invalid and consequently excluded from the set of feasible solutions to the problem.

4.2 Training Mechanism Based on MOEAs

Given an input E composed of k specimens, an approach based on evolutionary algorithms is proposed to automatically generate sequences of S operations, which allow the segmentation of images, and where, by applying each S sequence to the $Ie = (I_o, I_m)$ specimens of E, it allows to transform each I_o image of it into a segmentation similar to the corresponding ideal I_m segmentation. The degree of similarity between the proposed segmentation (PI) and the ideal segmentation I_m is quantified by the metrics of *sensitivity* and $specificity$.

The proposed algorithm is called Multi-Objective Evolutionary Segmenting Sequence Generator (*GEMSS*), and is based on NSGA-II [21].

At the end of the execution, GEMSS generates, as a result, a set of non-dominated solutions, trade-off solutions considering the target functions, over the training set.

4.3 Set of Operations

As mentioned above, this work aims to generate sequences of operations that achieve an adequate segmentation for a set of input images. For this purpose, defining the operations that will make up the sequences is necessary. GEMSS considers a set of 40 primary operations classified into two big groups according to the number of parameters they receive as input, 32 binary operations and 8 unitary operations. Some of them can have variants according to specific implementations or configurations. In addition, for the morphological operations, we use two sets of structuring elements of four different sizes; the first set is composed of regular structuring elements, and the second of irregular structuring elements that were generated randomly, which gives a total of 128 structuring elements. Combining these characteristics favors the expressive capacities of the sequences of operations generated by the algorithm.

4.3.1 Unitary Operations

The unitary operations receive only one image as an input parameter. This input image can be either the image resulting from applying the previous operation (in the sequence) or the contents of *storage* (an auxiliary memory used during the execution of the operation sequences). This set is composed of a total of 32 operations, including morphological operations, filters, and some special operations. Certain operations have variants or implementations that can be applied, depending on the evolutionary process's course. The morphological operations use two sets of structuring elements (regular and irregular), each composed of four subsets of 16 elements each, according to their size (3x3, 5x5, 7x7, 9x9), which gives a total of 128 structuring elements. These characteristics favor the expressive capacities of the sequences of operations. Table 1 shows the unitary operations used for this work.

4.3.2 Binary operations

The binary operations receive two images as parameters. These images come from *storage* and the result of applying a previous operation in the sequence. There are seven binary operations, of which 3 are logical operations on images (there is a logical operation NOT which is unary), and the remaining 4 are arithmetic operations at a pixel level. The specific operations used in this paper are listed in Table 2.

4.3.3 No Operation

The absence of operations is represented by a *No Operation* operator, which receives an image as input and returns the image without modifications as a result. This operator allows that, although the size of the chromosome is fixed during the process of evolution, the number of operations, which are effective within the chromosome, can vary. In addition, maximizing the number of *No Operation* type genes within a chromosome could look for sequences that minimize the number of operations.

4.4 Solution Structure

4.4.1 Chromosome

A chromosome defines a sequence of operations that corresponds to a candidate solution to the problem. Each chromosome comprises a sequence of n genes where each gene encodes an operation.

$$S = \{s_1, s_2, ..., s_{n-1}, s_n\}$$

Table 1: Unitary Operations GEMSS

Unitary Operations	
1	Morphological dilatation
2	Morphological Erosion
3	Morphological reconstruction
4	Morphological closing
5	Morphological Opening
6	Morphological Top Hat
7	Morphological Black Hat
8	ISH filter
9	HSI filter
10	Grey-scale conversion by one of the following methods: Intensity, Gleam, Luminance, Luma, Lightness, Value, Luster, Min Decomposition
11	Threshold with one of the following methods: Renyi Entropy, Huang, Intermodes, Isodata, Max Entropy, Otsu, Minimum Mean
12	Binary opening
13	Outline
14	Skeletonization
15	Fill Holes
16	Morphological gradient
17	GetChannel (Obtaining color image channel)
18	CLAHE [23]
19	Normalize Stack
20	Normalize Color Stack
21	Filter on the image with one of the following methods: Min, Max, Variance, Mean
22	Remove Connected (Removal of connected items)
23	Sharpen
24	Smooth
25	Find Edges
26	SubtrackBackground
27	Color Threshold
28	Histogram stretch contrast improvement
29	Despeckle
30	Median filter
31	Outliers filter
32	Mean Filter

Table 2: Binary Operations GEMSS

Binary Operations	
1	Difference: between the input image and the storage content
2	Addition: between the input image and the storage content
3	Multiplication: between the input image and the storage content
4	Division: between the input image and the storage content
5	Logical AND: between the input image and the storage content
6	Logical OR: between the input image and the storage content
7	Logical XOR: between the input image and the storage content
8	Logical NOT: between the input and bit

The algorithm initializes the chromosomes randomly. The number of genes r on each chromosome is always fixed. However, the number of effective operations on each chromosome is variable and depends on the type of each gene.

The chromosomes receive an image as input, apply the first operation of the sequence to it, and then apply the rest of the operations to the result of the previous operation generating a new image which we call *proposal*.

4.4.2 Gene

A gene represents an operation. It is a vector of binary variables, this is:

$$s_i = \{s_{i_0}, s_{i_1}, s_{i_2}, ..., s_{i_{r-1}}, s_{i_r}\}$$

with $r = 19$ for our implementation.

In addition, an auxiliary memory called *storage* is used to store intermediate results during the application of genes from a chromosome to an input image. This memory allows for storing partial and parallel results that may be required by any of the operations indicated by the genes, mainly when it comes to binary operations. The structural meaning of the binary variables that make up the gene is described below.

Input image: The variable s_{i_0} defines the image that will be used as input for the operation. When $s_{i_0} = 0$, the Storage content replaces the input, but when $s_{i_0} = 1$, no changes are made to the input.

Operation / No Operation: The variable s_{i_1} defines whether it is an effective operation or a gene of type *No Operation*. When $s_{i_1} = 1$, the rest of the variables of the gene determine the operation to be carried out, whereas when $s_{i_1} = 0$, it indicates that the gene is of type *No Operation*, which means that this operation does not make any change on the input image.

Unitary or Binary Operations: The variable s_{i_2} determines the type of operation according to the number of images used as operands. If the value $s_{i_2} = 0$, the operation is of unitary type, and if the value is $s_{i_2} = 1$, it is of binary type. This variable determines the structural meaning of the rest of the variables that make up the gene since the remaining variables have different meanings for unitary and binary operations.

a) Unitary Operations: As previously mentioned, when the variable $s_{i_2} = 0$, the structure of the gene takes on the meaning of a unitary operation. In order to reference the unitary operations, five binary variables are used, which gives a total of 32 possible operations. The variables s_{i_3} to s_{i_7} are used to reference these operations, and the s_{i_8} is destined to increase the number of operations that can be referenced in the future. In contrast, s_{i_9} to $s_{i_{18}}$ are used to establish characteristics, parameters, and configurations of each one of these operations.

b) Binary Operations: When the variable $s_{i_2} = 1$, the structure of the gene takes on the meaning of a binary operation. In order to reference the binary operations, three variables are used, which gives a total of 8 possible operations. The variables s_{i_3}, s_{i_4} and s_{i_5} are used to reference these operations. The variable s_{i_6} is reserved for referencing more operations in future implementations, while s_{i_7} is used to establish parameters of some of these operations.

Output image storage in auxiliary memory: The variable $s_{i_{19}}$ determines whether the operation result will be stored in the auxiliary memory *storage*. If the value is one, the image will be stored in *storage*; otherwise, the operation result will not be stored in the auxiliary memory.

Basic steps for applying a gene to an image: When applying an operation (gene), either to an input image or to the image resulting from the previous operation, the following procedure is followed:

1. It is checked whether the entry should be replaced with the Storage contents according to the value of s_{i_0}.

2. A check is made on s_{i_2}, to determine whether the gene in question represents an effective operation or whether it is a *No Operation* type gene. If the latter is the case, the gene has no effect on the image, and the image is passed on as input to the next gene in the sequence (or returned as a final result in the case of the last gene). If it is an effective operation, the next step is performed.

3. If it is an effective operation, s_{i_3} must be checked to discriminate the structural meaning of the variables that make up the gene; determining whether it is a unitary or binary operation.

4. The operation specified by the values of the gene variables is applied to the entry with the parameters specific to each operation.

5. It is checked whether the result of the operation should be stored in the auxiliary memory *storage*.

6. Once the operation indicated by the gene is applied, the result is returned and used as input for the next gene in the chromosome, or as output, in the case of the last gene in the sequence.

4.5 Solution Evaluation

The evolutionary mechanism requires that the quality of each solution coded on the chromosome be determined according to the objective functions. The values of the targets are calculated from metrics that compare the proposed image and the target image (which corresponds to an ideal segmentation). The proposed image is generated by applying a solution (sequence of operations), to the target image, from a pair (target image, target image) of the input. To calculate the value of a target function on a chromosome, the arithmetic mean of the values obtained by comparing all the proposed images PI (generated by applying the chromosome on I_o) with their respective target images I_m using sensitivity or specificity (depending on whether they are F_1 or F_2) is calculated. The process of evaluating each of the S sequences of operations, based on the target functions (sensitivity (11) and specificity (12) consists of the following key steps.

1. Apply the sequence S to each of the target images I_{o_i}, of each specimen that composes the input, generating an image PI_i, complying with the expression 16. This process requires the previous decoding of each binary gene to transform them into real operations on images.

2. The PI_i image generated by S is quantitatively compared with the ideal segmentation of the I_{m_i} specimen, using sensitivity (TPR) and specificity (TNR).

3. Finally, the arithmetic mean of the values of sensitivity (TPR) and specificity (TNR) calculated in the previous step is calculated, thus making it possible to quantitatively estimate the quality of S on the input, which represents the two values of the target functions of the S solution.

These steps of the evaluation are summarized in the objective functions expressed in (19) and (20). The evaluation process is graphically illustrated in Figure 2.

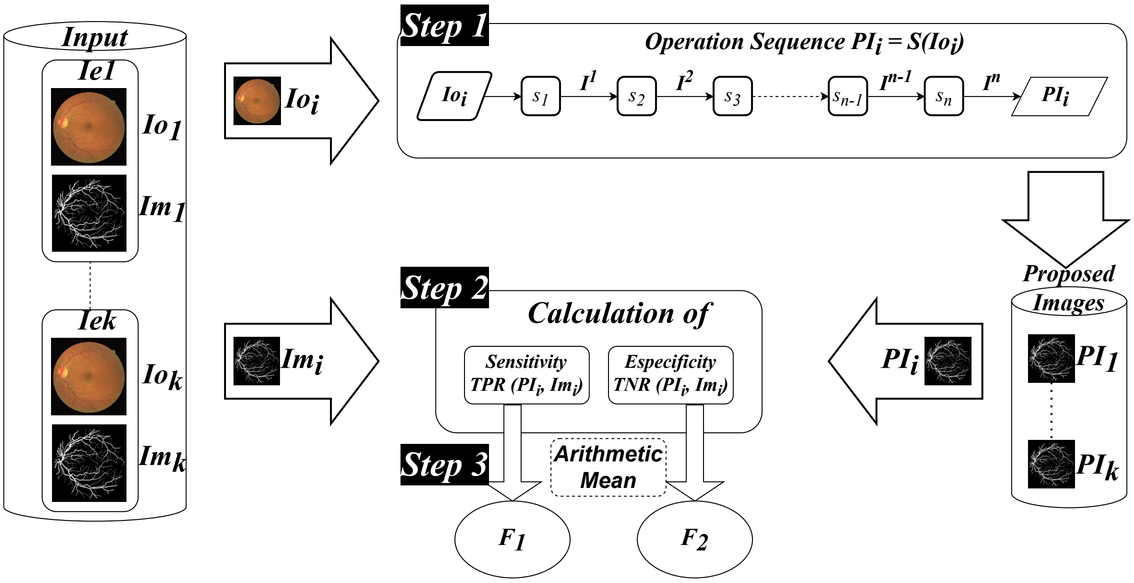


Figure 2: Operations Sequence Evaluation Scheme S Applied to Input Specimens; (1) The operation sequence S is applied to the target images Io_i of each specimen in the input, generating corresponding proposed images PI_i; (2) These proposed images are then compared with the target images Im_i of the input specimens using sensitivity TPR and specificity TNR metrics; (3) The arithmetic mean of all sensitivity values is calculated, determining the function F_1 of S for the input. Similarly, the function F_2 is determined.

This evaluation scheme is fully scalable in relation to the following factors:

- Input size: The scheme accommodates variable input sizes (k), adaptable to the complexity or specific requirements of the algorithm's application scope.

- Chromosome length: In this implementation, the solution chromosome length is set to $n = 20$ genes (operations). This value is adjustable, depending on the problem's complexity or the imaging time to be analyzed.
- Number of metrics and objectives: The scheme supports varying the number of metrics and objective functions, provided they facilitate image comparison based on specific criteria. Additional metrics for comparing other quantitative image aspects can be integrated, enhancing suitability for specific application areas. For instance, to prioritize shorter solutions that optimize existing objectives, an objective function measuring the count of effective operations (excluding *NoOperation*) in the chromosome can be added.
- Operations set: The operations set is fully customizable. Operations can be replaced, extended, or reduced to suit specific application areas, enhance the expressive capacity of solutions, or improve the algorithm's convergence time.

It is important to note that increasing the size of each of these factors enhances the expressive capabilities of the solutions generated by the algorithm, in contrast increases the size of the space for finding solutions which would imply an increase in the time of convergence of the algorithm.

4.6 Execution parameters and stop criteria

For the implementation of this work, several Q of independent evaluations have been established as a stop criterion. Once the stop criterion is reached, the algorithm returns a Pareto set for each run, with the chromosomes representing the best solutions obtained until that moment, based on the objectives measured on the input. Among the performance parameters that can be modified are the crossover rate, the mutation rate, the number of evaluations, and the size and composition of the input. These factors contribute to the fact that the proposed approach can be adapted to different scopes.

5 Experiments

To measure the effectiveness of this evolutionary approach, we conducted a quantitative evaluation based on three test cases: (a) extraction of cell types according to their shape, (b) segmentation of melanoma images, and (c) extraction of retinal veins. Thirty independent runs of the three cases mentioned above were performed, and an un-dominated ordering of the results was obtained for each case.

In our experiments, when a solution dominates all others, the Pareto Front comprises that single solution, and the choice of a candidate solution is straightforward. In cases where the Pareto front includes multiple non-dominated solutions, the usual case in multi-objective optimization is a specific "candidate solution" arbitrarily selected. Then, we use this "candidate solution" to illustrate the proposed approach application. The main intention is to present the objective function results of a solution and visually demonstrate the method's performance in terms of its capability in image segmentation.

5.1 Extraction of cell types according to their shape

The first test case is considered the most basic. It has a set of binary images where cells of various types are observed; these cells are differentiated at first sight by their shapes.

In this experiment, a total of five images are available: the original image containing a combination of four different types of cells (Figure 3), and four target images, each containing a single cell type; these target images represent different ideal segmentation of the original image (Figure 4). For simplicity, we name the cell types as A, B, C, and D.

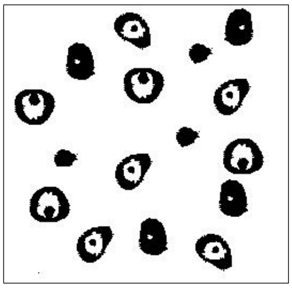


Figure 3: Four types of cells combined in one image

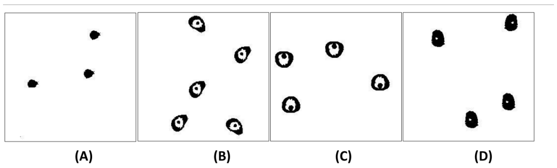


Figure 4: Target images, the four ideal segmentations are observed for Type A, B, C, and D cells.

The experiment is designed to determine whether GEMSS can generate sequences of operations that segment the images and extract a given cell type from them.

5.1.1 Execution Parameters

For this test case, 120 independent executions were performed, 30 executions for each of the cell types. For each of the GEMSS executions in this experiment, the following execution parameters were considered:

- Stop criteria: $Q = 100,000$ evaluations, equivalent to 4,000 generations.
- Chromosome length: Set to a maximum of n=15 genes, meaning each solution can have up to 15 effective operations.
- Solution population size: Fixed at 25 for the iterations of the evolutionary algorithm.
- Crossover rate: 0.95.
- Mutation rate: 0.1.
- Input size: $k = 1$.

5.1.2 Type A cells: Experimental Results

Figure 5 shows the Pareto front of the set of solutions formed by the union and non-dominated ordering of the Pareto sets of each of the 30 independent executions, where each solution is classified according to the target function values on the input used in the experiment. As can be observed, a solution (candidate solution) was found that dominates the rest of the solutions generated during the experiment. Note that, the candidate solution is highlighted.

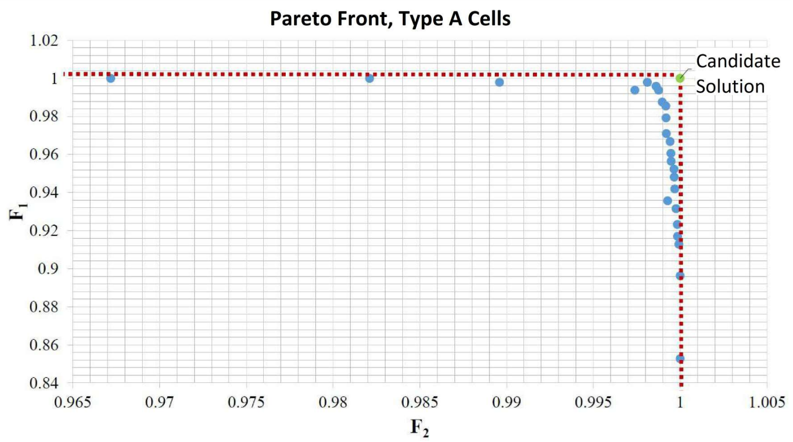


Figure 5: Solutions generated by GEMSS for the cell shape experiment, in its execution for the extraction of type A cells.

A solution has been selected from this Pareto front, which we call a candidate solution (see Figure 5). The candidate solution simultaneously obtains the highest value of the target functions. Table 3 shows a quantitative evaluation of this candidate solution through the target functions, image comparison metrics, and error metrics.

For the target functions, the best values are closest to one, while for the error metrics, the best values are close to zero.

Table 3: Quantitative Evaluation of Solution Candidate: Type A Cells

Solution Candidate Cell Type A		
Performance Metrics		Value
Objective Functions	F_1	1
	F_2	1
Error Metrics	FPR	0
	FNR	0

Figure 6 visually compares the target image, its corresponding ideal segmentation with type A cells, and the segmentation produced by the candidate solution.

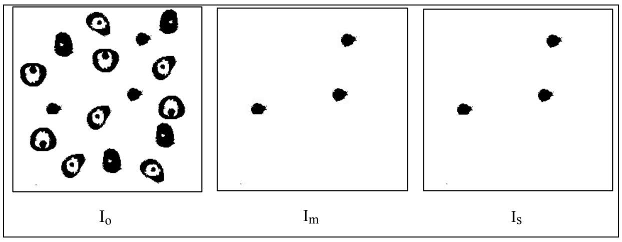


Figure 6: Comparison of target image, ideal segmentation, and segmented image generated by the candidate solution for Type A Cells. The figure illustrates the target image, its ideal segmentation highlighting type A cells, and the segmentation result as produced by the evaluated candidate solution.

As mentioned above, each solution is a sequence of operations automatically generated by a training mechanism based on evolutionary algorithms.

5.1.3 Type B Cells: Experimental Results

In each of the 30 runs with type B cells, GEMSS produced a Pareto set. These sets underwent non-dominated ordering to form the Pareto front, as illustrated in Figure 7. In this case, it was also possible to identify a dominant candidate solution, which outperformed the other solutions generated during the experiment.

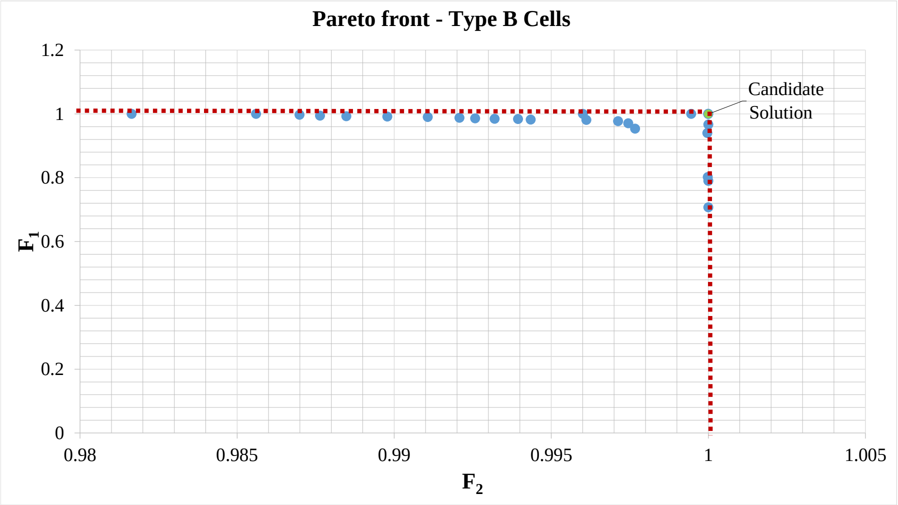


Figure 7: GEMSS-Generated Solutions for Type B Cell Shape Experiment. This figure classifies solutions based on their target function values. The depicted candidate solution, which dominates all others generated in the experiment, is the sole member of the Pareto front.

From this set, a candidate solution is selected for quantitative evaluation in terms of target functions and error metrics, as detailed in Table 4.

Table 4: Quantitative Evaluation of the Candidate Solution for Type B Cells.

Solution Candidate Type B Cells		
Evaluation Metric		Value
Objective Functions	F_1	1.0
	F_2	0.999456479540337
Error Metrics	FPR	0.000543520459663
	FNR	0.0

Figure 8 visually compares the segmentation results produced by the candidate solution with the target image containing Type B cells.

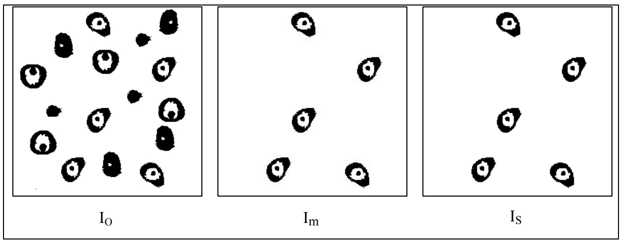


Figure 8: Comparison of Target Image, Ideal Segmentation, and Segmented Image for Type B Cells. This figure illustrates the target image alongside its ideal segmentation and the segmentation result obtained from the candidate solution, specifically for Type B cells, showcasing the effectiveness of the proposed segmentation approach.

5.1.4 Type C Cells: Experimental Results

The Pareto Front, derived from the non-dominated ordering of the solution set from 30 GEMSS runs for Type C cells, is depicted in the graph shown in Figure 9.

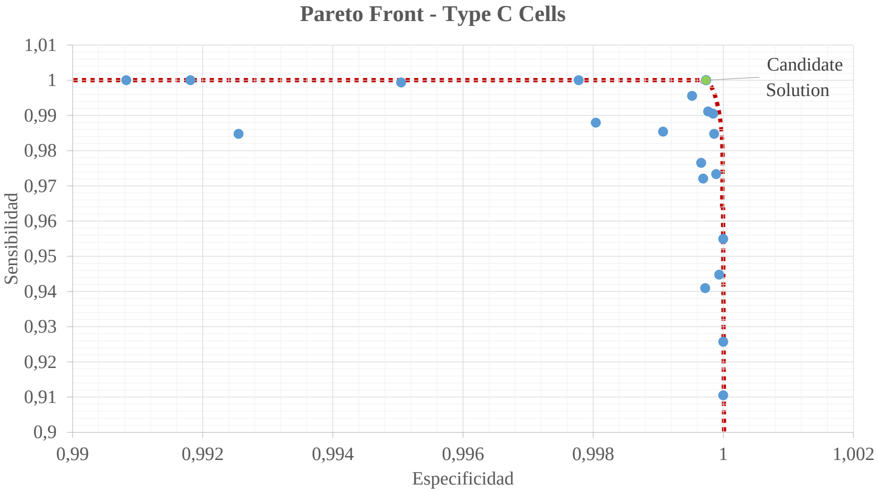


Figure 9: GEMSS-Generated Solutions for Type C Cell Shape Experiment. This figure classifies the solutions based on their target function values, executed for the extraction of Type C cells. It also highlights the solution selected as the candidate solution.

The quantitative evaluation of the candidate solution is presented in Table 5.

Table 5: Quantitative Evaluation of Solution Candidate: Type C Cells

Solution Candidate Type C Cells		
Performance Metrics		Value
Objective Functions	F_1	0.9371428571428572
	F_2	0.9998920486089692
Error Metrics	FPR	0,00010795139103078
	FNR	0.06285714285714286

Applying the candidate solution to the target image of the experiment results in a segmented image. Figure 10 visually compares this segmented image with the ideal segmentation.

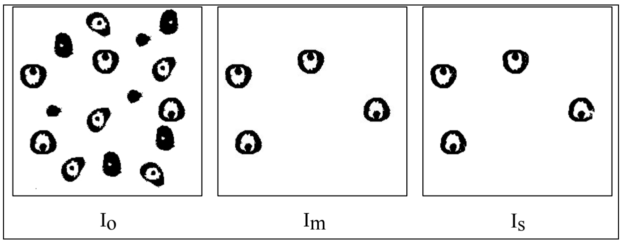


Figure 10: Comparison of Target Image, Ideal Segmentation, and Segmented Image for Type C Cells. This figure illustrates the target image alongside its ideal segmentation and the segmentation result obtained from the candidate solution, specifically for Type C cells, showcasing the effectiveness of the proposed segmentation approach.

5.1.5 Type D Cells: Experimental Results

The final experiment concerning cellular forms focuses on Type D cells. Figure 11 displays the Pareto front, which is derived from the combined solution sets of 30 GEMSS runs, followed by their non-dominated sorting. Notably, one solution (the candidate solution) is identified as dominating the rest of the solutions generated during the experiment.

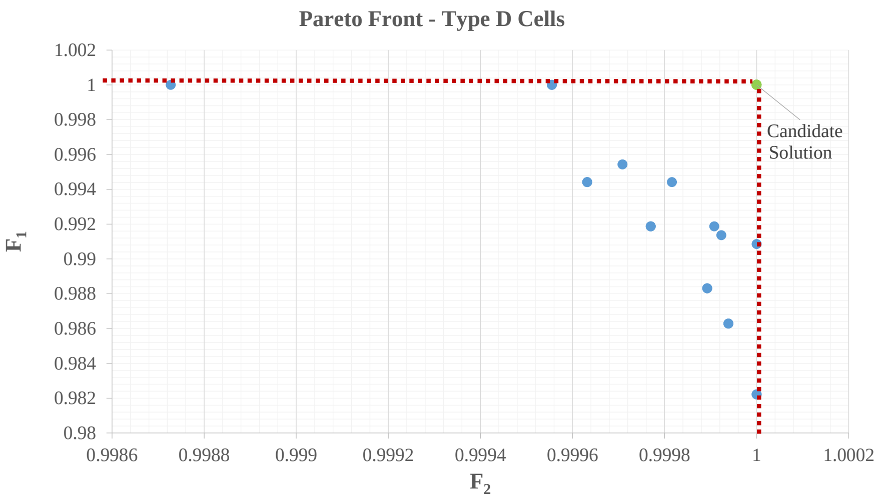


Figure 11: GEMSS-Generated Solutions for the D-Type Cell Shape Experiment. This figure categorizes the solutions based on their target function values, executed for the extraction of D-type cells. It highlights the candidate solution, which not only dominates the rest of the solutions generated during the experiment but is also the sole solution belonging to the Pareto front.

The quantitative analysis of the candidate solution for this experiment is presented in Table 6.

Table 6: Quantitative Evaluation of Solution Candidate: Type D cells.

Solution Candidate Type D Cells		
Performance Metrics		Value
Objective Functions	F_1	1.0
	F_2	1.0
Error Metrics	FPR	0.0
	FNR	0.0

A visual representation of the candidate solution's application is shown in Figure 12.

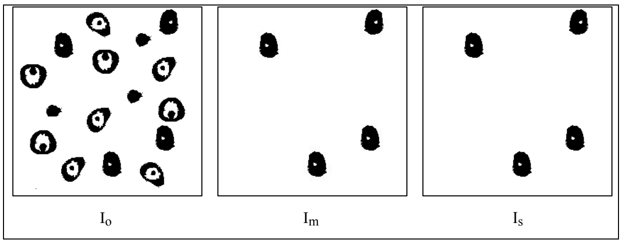


Figure 12: Comparison of Target Image, Ideal Segmentation, and Segmented Image for Type D Cells. This figure illustrates the target image alongside its ideal segmentation and the segmentation result obtained from the candidate solution, specifically for Type D cells, showcasing the effectiveness of the proposed segmentation approach.

5.2 Melanoma Image Segmentation

Two databases of melanoma images, benign and malignant, downloaded from [24], were considered for this test case. These databases contain color photographs of different types of lesions and their corresponding ideal binary segmentations, manually produced by experts in the field of medical imaging segmentation of this type. For this experiment, 60 independent runs were performed, 30 for benign melanoma images and another 30 for malignant melanoma images; for both cases, an entry of size $k = 10$ was considered.

Figure 13 presents examples of image pairs, each consisting of a target image and its corresponding ideal segmentation, as used in the specimens of the input.

As a result of each of the runs, GEMSS generates a Pareto set with solutions that are applicable to all specimens in the input.

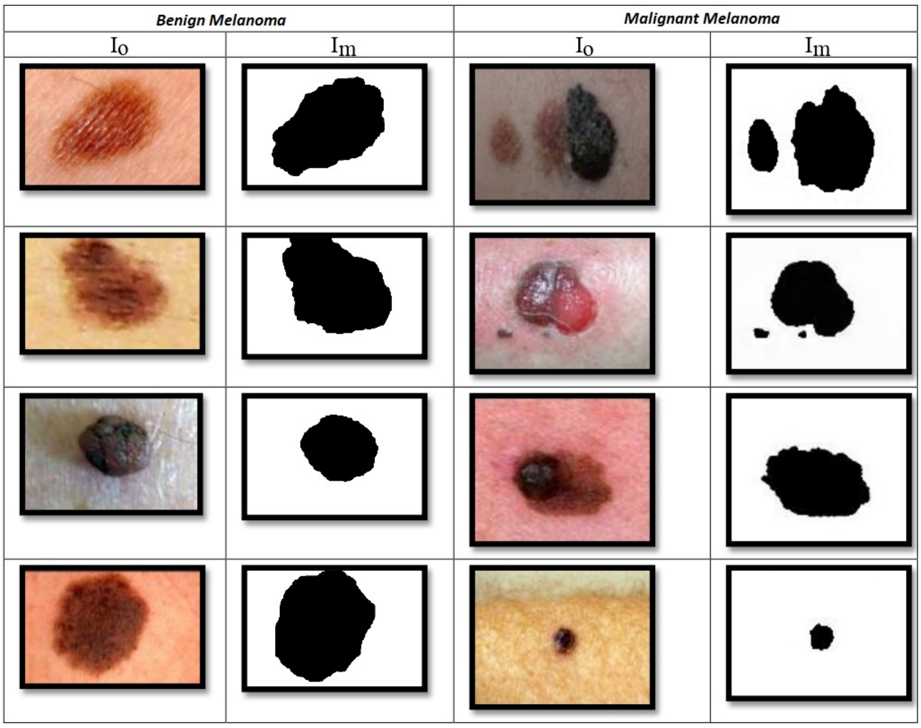


Figure 13: Examples of Malignant and Benign Melanoma Images Used in the Experiment. The figure illustrates representative samples of both malignant and benign melanomas, showcasing the visual differences and characteristics pertinent to this study.

5.2.1 Input Composition

Unlike the first test case, GEMSS receives a specimen entry, where the I_o are the original images, and the I_m are the ideal segmentations obtained from the databases in [24].

5.2.2 Execution Parameters

For this experiment, two runs are executed: one focusing on the benign melanoma step and another on the malignant melanoma case. Both runs employ the same set of parameters:

- Stop criteria: $Q = 100,000$ evaluations, equivalent to 4,000 generations.
- Chromosome length: $n = 15$.
- Solution population size: 25.
- Crossover rate: 0.95.
- Mutation rate: 0.1.
- Input size: $k = 10$.

5.2.3 Benign Melanoma: Experimental Results

Unlike the cell shape experiment, in this test case, GEMSS processes an input M comprising several specimens. This input serves as a foundation for evaluating solution quality by comparing the target function values generated across the input.

At the conclusion of each run, GEMSS produces a Pareto set of solutions. After aggregating the solutions from all 30 runs, a non-dominated sorting is applied to them. This process results in a selected set of solutions, with their Pareto front depicted in Figure 14. Each solution in the figure is positioned according to the values of its objective functions.

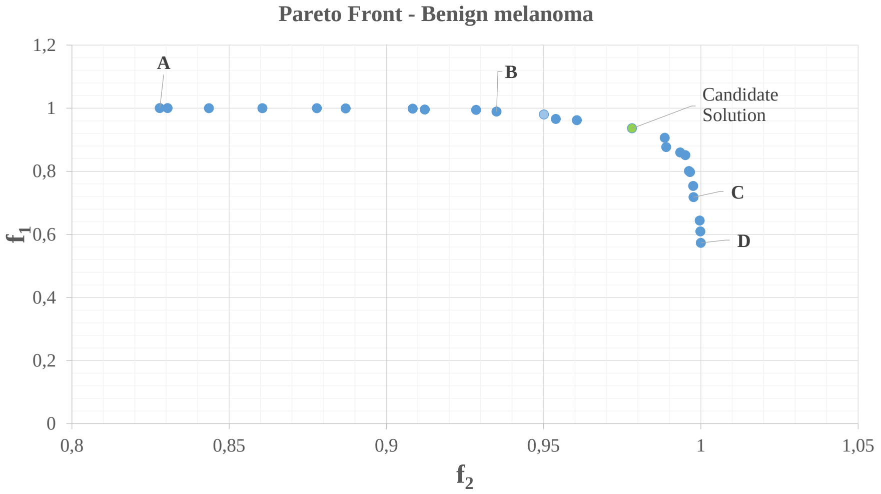


Figure 14: Pareto Front of Solutions Generated by GEMSS for Benign Melanoma Image Segmentation. This figure categorizes the solutions based on their target function values, executed for benign melanoma images. It highlights the solution identified as the candidate solution, illustrating its position relative to others in terms of optimization objectives.

Table 7 displays the segmentations generated by applying solutions corresponding to points A, B, C, and D on the Pareto front from Figure 14. This table not only illustrates the target function values for these solutions but also includes their Sensitivity (TPR) and Specificity (TNR) metrics, calculated by comparing each generated image with its ideal segmentation. The variation in visual characteristics of the segmentations demonstrates the impact of using solutions with different target function values.

Table 7: Visual Comparison of GEMSS Solutions for Benign Melanoma. This table showcases a side-by-side comparison of the segmentation results obtained from various GEMSS solutions, highlighting the differences in their approach to benign melanoma segmentation.

Sol.	A	B	C	D
F1	0.99988823	0.99446071	0.71775582	0.57298902
F2	0.84360887	0.92857977	0.99769531	0.99999999
TPR	1.00000000	1.00000000	0.95819336	0.77454274
TNR	0.95468205	0.98180189	0.99873080	1.00000000

Based on the target function values depicted in Figure 14, a candidate solution has been selected for detailed analysis.

Table 8 provides a quantitative comparison between the segmentation created by this candidate solution on the target image e_{obj} of a randomly selected specimen and its ideal segmentation.

Table 9 offers a comprehensive quantitative assessment of the candidate solution's quality across the entire input. It details target function values and error metrics for the solution, with values derived from the arithmetic mean of the candidate solution's performance on each specimen in the input. Additionally, the table includes the standard deviation of the error metrics, calculated during the evaluation process.

For the Figure 15, one of the specimens of the entry has been taken at random and the candidate solution has been applied to the target image in order to visually compare it with its corresponding target image (ideal segmentation).

Table 8: Quantitative comparison of the segmentation generated by the candidate solution and the ideal segmentation of one of the images in the entry Benign Melanoma.

Benign Melanoma		
Candidate Solution Vs Ideal Segmentation		
Comparison Metrics	Sensitivity	1.00000000
	Specificity	0.97646378
Error Metrics	FPR	0.02353622
	FNR	0.00000000

Table 9: Quantitative evaluation of candidate solution on entry. Benign Melanoma.

Benign Melanoma			
Solution Candidate Vs Input			
		Value	**Standard Deviation**
Objective Functions	F1	0.96714772	0.04894739
	F2	0.90567026	0.12686165
Error Metrics	FPR	0.09432974	0.12686165
	FNR	0.03285228	0.04894739

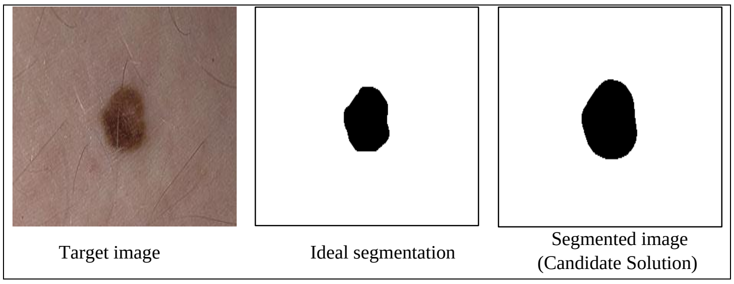
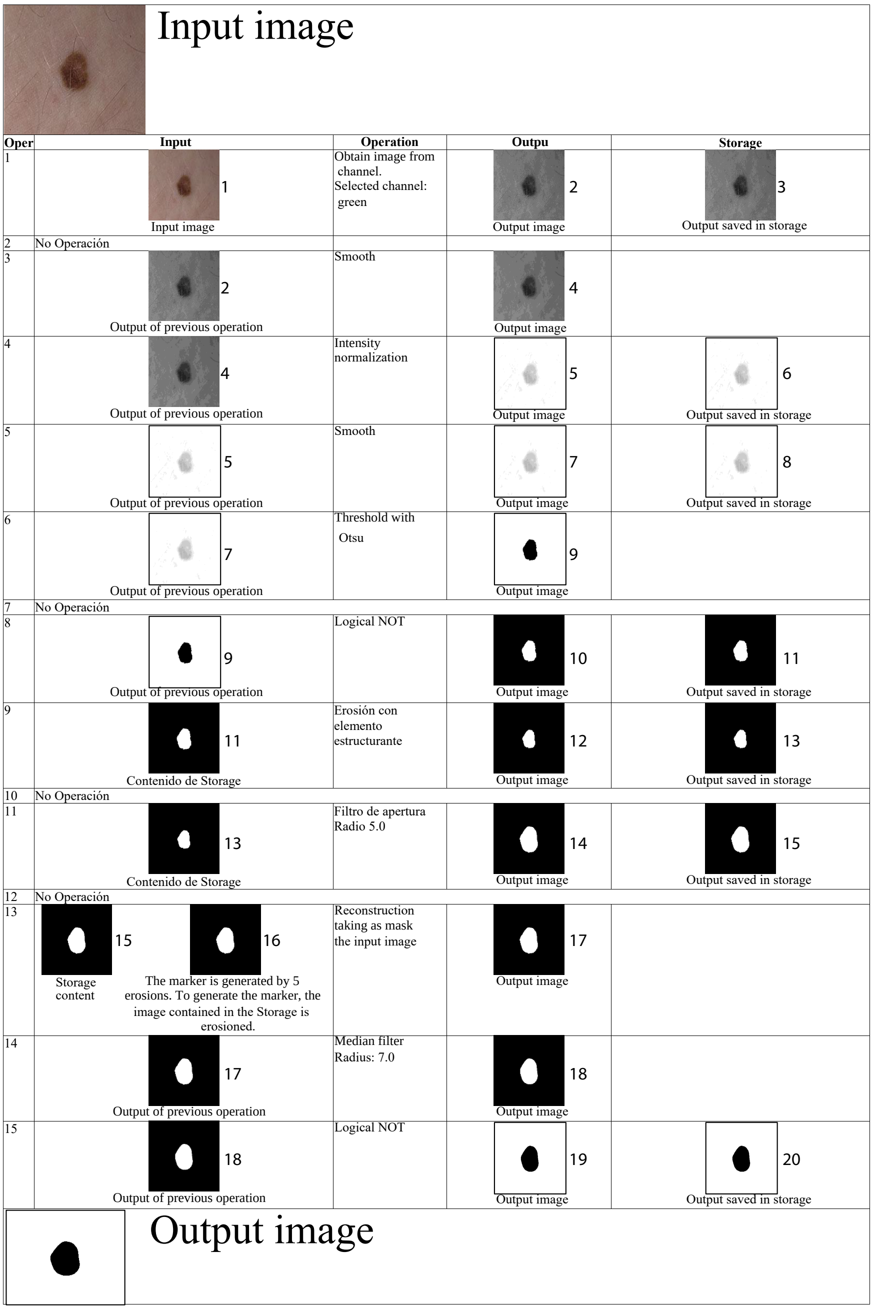


Figure 15: Target image, ideal segmentation, and segmented image generated from the candidate solution for benign melanoma image segmentation.

Taking the same specimen used in the Figure 15 the sequence of operations represented by the candidate solution is presented in detail, in the Figure 16.



Input image

Oper	Input	Operation	Outpu	Storage
1	1 Input image	Obtain image from channel. Selected channel: green	2 Output image	3 Output saved in storage
2	No Operación			
3	2 Output of previous operation	Smooth	4 Output image	
4	4 Output of previous operation	Intensity normalization	5 Output image	6 Output saved in storage
5	5 Output of previous operation	Smooth	7 Output image	8 Output saved in storage
6	7 Output of previous operation	Threshold with Otsu	9 Output image	
7	No Operación			
8	9 Output of previous operation	Logical NOT	10 Output image	11 Output saved in storage
9	11 Contenido de Storage	Erosión con elemento estructurante	12 Output image	13 Output saved in storage
10	No Operación			
11	13 Contenido de Storage	Filtro de apertura Radio 5.0	14 Output image	15 Output saved in storage
12	No Operación			
13	15 Storage content 16 The marker is generated by 5 erosions. To generate the marker, the image contained in the Storage is erosioned.	Reconstruction taking as mask the input image	17 Output image	
14	17 Output of previous operation	Median filter Radius: 7.0	18 Output image	
15	18 Output of previous operation	Logical NOT	19 Output image	20 Output saved in storage

Output image

Figure 16: Sequence of operations of the candidate solution applied to one of the images of the entry in benign melanoma images.

5.2.4 Malignant melanoma: Experimental Results

After each execution, GEMSS generates the Pareto set of solutions. All the solutions generated during the 30 runs were taken, and a non-dominated order was carried out on them, selecting a set of solutions whose front is represented in Figure 17.

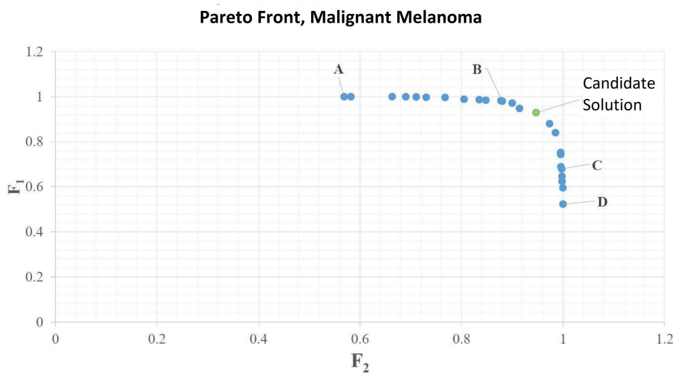


Figure 17: Pareto front of solutions generated by GEMSS for malignant melanoma, in its execution for malignant melanoma images. The solutions are classified according to their target function values. The candidate solution is highlighted.

Table 10 presents the segmentations generated when applying the solutions corresponding to points A, B, C, and D of the Pareto front of Figure 17. It also presents the values of the target functions of the solutions, in addition to the Sensitivity (TPR) and Specificity (TNR) values that were obtained by comparing each of the images generated with their corresponding ideal segmentation. This Table shows the variation in visual characteristics of the generated segmentations as solutions with different target function values are taken.

Table 10: Visual comparison of GEMSS solutions. Malignant melanoma

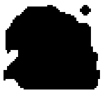
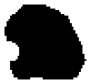
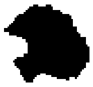
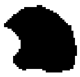
				
Sol	A	B	C	D
F1	0.99999999	0.98192199	0.68856149	0.52289952
F2	0.56886118	0.87777147	0.99577101	0.99999999
TPR	0.99999999	0.97729337	0.95277021	0.83832879
TNR	0.65794066	0.95636998	0.98254799	0.99999999

Table 11 shows a quantitative comparison between the segmentation generated by the candidate solution on the target image I_o of a randomly selected specimen of the entry and its corresponding ideal segmentation.

A quantitative evaluation of the candidate solution, from Figure 17, based on the complete input, is shown in Table 12. It shows the values of the target functions, image comparison metrics, and error metrics commonly applied in digital image processing.

Table 11: Quantitative comparison of the segmentation generated by the candidate solution and the ideal segmentation of one input image

Malignant Melanoma		
Candidate Solution vs Ideal Segmentation		
Comparison Metrics	Sensitivity	0.98573693
	Specificity	0.98249784
Error Metrics	FPR	0.01750216
	FNR	0.01426307

We take as a candidate solution the one indicated in Figure 17, and taking at random a specimen of the input, we apply it on the target image to later compare it visually with the target image (ideal segmentation), in Figure 18.

Table 12: Quantitative evaluation of candidate solution on the input. Malignant melanoma

Malignant melanoma			
Solution Candidate vs Input			
		Value	**Standard Deviation**
Objective Functions	F1	0.92897303	0.06561326
	F2	0.94699778	0.04334569
Error Metrics	FPR	0.05300222	0.04334569
	FNR	0.07102697	0.06561326

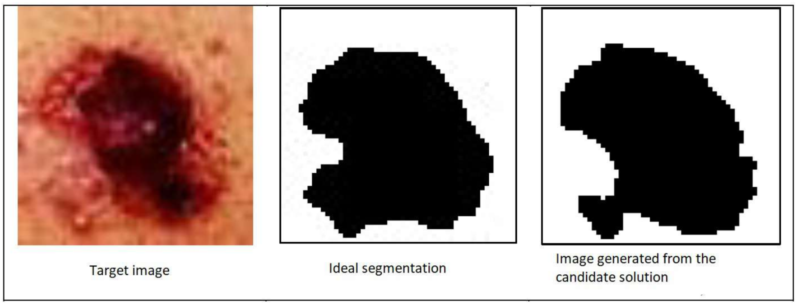


Figure 18: Target, ideal and segmented images generated from the candidate solution for malignant melanoma.

5.3 Retinal Vein Extraction

In this experiment, GEMSS receives an input composed of color ophthalmoscopic images of human retinas and their corresponding binary ideal segmentations where the pattern of veins extracted from these images is observed. The images used for this experiment were downloaded from [25]. This experiment aims to show the flexibility of GEMSS for different types of applications since the characteristics of this type of image differ significantly from the images treated in the test cases of 5.1 and 5.2 sections. Thirty independent runs of this experiment were performed.

Examples of the image types to be analyzed in this test case are shown in Figure 19.

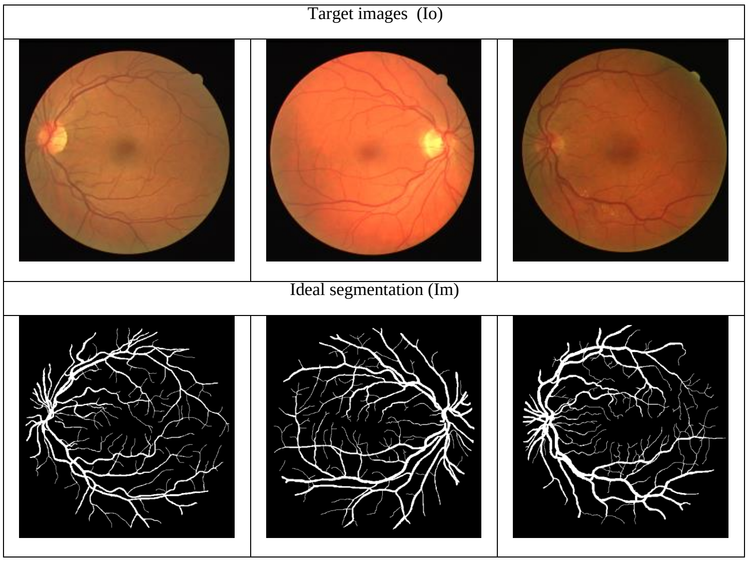


Figure 19: Examples of target images and corresponding ideal segmentations used in experimental inputs. This figure illustrates a selection of target images alongside their ideal segmentations, highlighting the types of data used to construct the inputs for the experiment.

5.3.1 *Execution parameters*

For each of the GEMSS runs in this experiment, the following execution parameters have been considered:

- Stop criteria: $Q = 100,000$ evaluations, equivalent to 4,000 generations.
- Chromosome length: $n = 15$.
- Solution set size: 25.
- Crossover rate: 0.95.
- Mutation rate: 0.1.
- Input size: $k = 5$.

5.3.2 *Experimental Results*

In the retinal sales extraction testing cases, the input has also been composed of multiple specimens. GEMSS generated a Pareto set for each of the 30 executions of the algorithm. Later, a non-dominant ordering was performed on the union of these sets, and the result is represented in the Pareto front of Figure 20, where all the solutions found and their position about the values of the target functions are shown. One of the solutions is also highlighted, which is taken as a candidate solution and will be the object of a more detailed analysis.

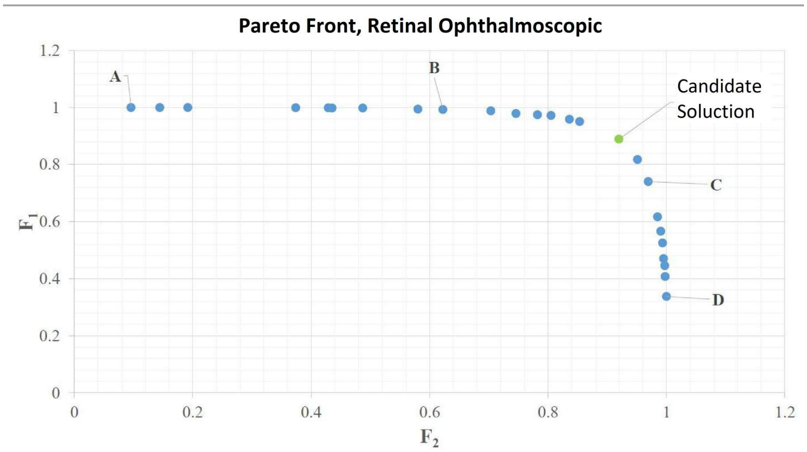


Figure 20: Pareto front of solutions generated by GEMSS for the retinal ophthalmoscopic image segmentation experiment. The solutions are classified according to their target function values. The candidate solution is highlighted.

Table 13 presents the segmentations generated when applying the solutions corresponding to points A, B, C, and D of the Pareto front of Figure 20. It also presents the values of the target functions of the solutions; in addition to the values of Sensitivity (TPR) and Specificity (TNR) that were obtained by comparing each of the images generated with their corresponding ideal segmentation. This Table shows the variation in visual characteristics of the generated segmentations as solutions with different target function values are taken.

Table 14 shows a quantitative comparison between the segmentation generated by the candidate solution on the target image I_o of a randomly selected specimen of the entry and its corresponding ideal segmentation.

The candidate solution was taken, and its quality was evaluated from the input specimens to generate the Table 15. This table shows the values of the target functions, the metrics commonly used for image comparison, and the degrees of error between them.

Figure 21 compares the result of applying the candidate solution to the target image of a specimen selected randomly from the input and the specimen's corresponding target image (ideal segmentation).

Table 13: Visual comparison of GEMSS solutions. Retinal Veins

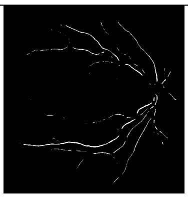
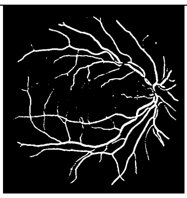
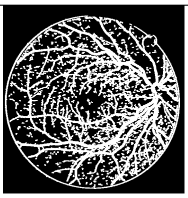
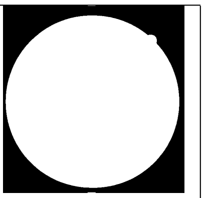
				
Sol.	A	B	C	D
F1	1.00000000	0.99295543	0.74043079	0.33772022
F2	0.09583510	0.62234229	0.96934800	1.00000000
FPR	1.00000000	0.99243340	0.76566161	0.34098997
FNR	0.11547795	0.71379106	0.97780408	1.00000000

Table 14: Quantitative comparison of the segmentation generated by the candidate solution and the ideal segmentation of one of the images in the input. Retinal Veins

Retinal Veins		
Candidate Solution Vs Ideal Segmentation		
Comparison Metrics	Sensitivity	0.89305288
	Specificity	0.91035778
Error Metrics	FPR	0.08964222
	FNR	0.10694712

Table 15: Quantitative evaluation of candidate solution on input. Retinal Veins

Retinal Veins			
Solution Candidate Vs Input			
		Value	**Standard Deviation**
Objective Functions	F1	0.88904096	0.01602606
	F2	0.91987295	0.02196550
Error Metrics	FPR	0.08012705	0.02196550
	FNR	0.11095904	0.01602606

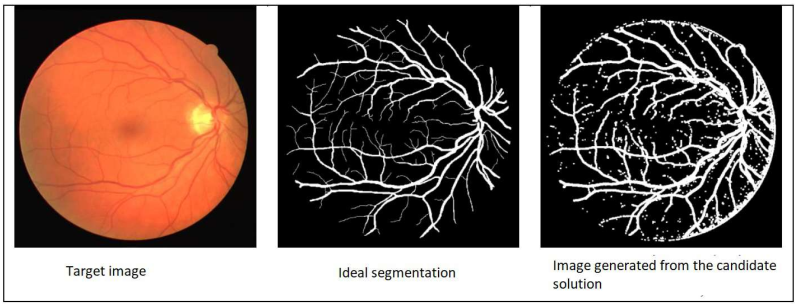


Figure 21: Objective, ideal segmentation and segmented images generated from the candidate solution, retinal vein extraction.

6 Analysis of Results

In our experiments, we showed the effectiveness of using multi-objective evolutionary algorithms, such as NSGA-II, for image segmentation in different contexts. The results indicate that this approach achieves high segmentation accuracy, essential in medical applications where accurate feature identification is required.

Tests were performed using three types of data sets to validate the versatility of the method:

1. **Binary Images of Type A, B, C, and D Cells:** The approach correctly identified a cellular form in this data set. Maximizing sensitivity and specificity helped minimize false positive and false negative errors. The results show that the algorithm achieves a sensitivity (TPR) and specificity (TNR) of up to 1.0 in the segmentation of images of type A and D cells.

2. **Images of Benign and Malignant Melanoma:** In this scenario, segmentation accuracy is crucial to differentiate between benign and malignant areas of the skin. Sequences of operations could be generated that achieved detailed and accurate segmentation, essential for accurate clinical diagnosis. The results showed that the approach could adapt to variations in the color and texture of the images, providing consistent and reliable segmentation. Specifically, the algorithm achieved a sensitivity of 1.0 and specificity of 0.9765 in the segmentation of images of benign melanoma, and a sensitivity of 0.9857 and specificity of 0.9825 in the segmentation of malignant melanoma images.

3. **Ophthalmoscopic Retinal Images:** This data set presented a more complex challenge due to subtle color variations and delicate patterns in retinal images. Despite these challenges, the multi-objective evolutionary approach showed its ability to handle these complexities and accurately segment areas of interest. The results indicate a sensitivity of 0.8931 and specificity of 0.9104 in the extraction of retinal veins in ophthalmoscopic images.

Some works in the literature seek to optimize the sequence of operations to obtain an output image from an input image [6, 7]. However, Lezcano et al. [4] are the only ones to use a multi-objective optimization approach for this purpose. In their work, evolutionary algorithms are used to simultaneously optimize the similarity of the resulting image and the computational cost of the operations.

Despite sharing the idea of using multi-objective optimization, our approach differs from that of Lezcano et al. [4] in key aspects:

- **Type of Images Processed:** Lezcano et al. [4] were exclusively focused on binary image segmentation. In contrast, our approach includes grayscale and color image processing, which extends its applicability to a wider variety of image segmentation problems.

- **Objective Functions:** While the approach of Lezcano et al. [4] seeks to minimize the number of operations and computational cost, our method focuses on maximizing segmentation accuracy, evaluated through metrics such as sensitivity and specificity. This difference reflects distinct objectives: Lezcano's approach [4] is more oriented towards computational efficiency, whereas our proposal prioritizes segmentation quality.

Due to the differences in objective functions, a direct comparison of the results of the two approaches is not appropriate. Each method is optimized for different criteria and applications. The choice of objective functions in our work aligns with the need for high accuracy in critical applications, such as medical diagnostics, whereas the Lezcano approach [4] might be more suitable for scenarios where computational efficiency is a priority. Additionally, direct comparisons with single-objective works are also not suitable, as our multi-objective optimization considers multiple criteria simultaneously, providing a more comprehensive assessment of performance.

7 Conclusions

This work presents a novel approach to address the pressing issue of image segmentation using Multi-Objective Evolutionary Algorithms (MOEAs). This approach was developed in response to the growing need to segment images with high precision, especially in critical applications where it is essential to identify features accurately.

Maximization of sensitivity and specificity were used as objective functions to guide the optimization process. These criteria ensure high segmentation accuracy, which is crucial in scenarios where errors can have significant consequences, such as detecting anomalies or patterns in various types of images. The approach was validated on three types of images: i) cell shapes in binary images, ii) images of benign and malignant melanoma, and iii) retinal images obtained by ophthalmoscopy. The results confirmed the method's ability to generate sequences of operations that achieve accurate and effective segmentation.

This work not only confirmed the feasibility of using a multi-objective evolutionary approach for digital image processing but also highlighted the method's versatility when applied to different types of images, whether binary, grayscale, or color. The ability to adapt the approach to multiple contexts expands its potential for use in a wide range of image-processing applications beyond just medical fields.

In future work, we propose diversifying the application of GEMSS to other fields of application to measure and compare the degree of robustness of the solutions generated by GEMSS based on images that share common characteristics. Other Multiobjective Evolutionary Optimization Algorithms as SPEA II [26] or MOPSO [27] can be implemented to compare the solutions generated by the NSGA-II. The proposed Multiobjective approach can be extended to other image processing problems different from segmentation, where given a input image and a target image, it is necessary to generate a sequence of operations that transforms the input image into an image similar or comparable to the target image.

Acknowledgment

This research was funded by CONACYT, Paraguay, grant number PINV18-846 and the Facultad Politécnica, Universidad Nacional de Asunción.

References

[1] A. T. Beuren, R. J. Pinheiro, and J. Facon, "Color approach of melanoma lesion segmentation," in *2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*. IEEE, 2012, pp. 284–287.

[2] M. E. Roberts and E. Claridge, "An artificially evolved vision system for segmenting skin lesion images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2003, pp. 655–662.

[3] D. P. Pinto-Roa, J. C. Mello-Román, J. L. V. Noguera, R. Quintana, F. Roa, and P. E. Gardel-Sotomayor, "Operation sequence design for image segmentation based on multi-objective evolutionary algorithms," in *2023 XLIX Latin American Computer Conference (CLEI)*, 2023, pp. 1–10.

[4] C. Lezcano, J. L. V. Noguera, D. P. Pinto-Roa, M. García-Torres, C. Gaona, and P. E. Gardel-Sotomayor, "A multi-objective approach for designing optimized operation sequence on binary image processing," *Heliyon*, vol. 6, no. 4, 2020.

[5] C. Munteanu and A. Rosa, "Gray-scale image enhancement as an automatic process driven by evolution," *IEEE transactions on systems, man, and cybernetics, part B (cybernetics)*, vol. 34, no. 2, pp. 1292–1298, 2004.

[6] J. Wang and Y. Tan, "Morphological image enhancement procedure design by using genetic programming," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 1435–1442.

[7] M. I. Quintana, R. Poli, and E. Claridge, "Morphological algorithm design for binary images using genetic programming," *Genetic Programming and Evolvable Machines*, vol. 7, no. 1, pp. 81–102, 2006.

[8] H. Zhou, J. M. Rehg, and M. Chen, "Exemplar-based segmentation of pigmented skin lesions from dermoscopy images," in *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro.* IEEE, 2010, pp. 225–228.

[9] P. Schmid-Saugeona, J. Guillodb, and J.-P. Thirana, "Towards a computer-aided diagnosis system for pigmented skin lesions," *Computerized Medical Imaging and Graphics*, vol. 27, no. 1, pp. 65–78, 2003.

[10] M. Silveira, J. C. Nascimento, J. S. Marques, A. R. Marçal, T. Mendonça, S. Yamauchi, J. Maeda, and J. Rozeira, "Comparison of segmentation methods for melanoma diagnosis in dermoscopy images," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 1, pp. 35–45, 2009.

[11] J.-H. Hong, S.-B. Cho, and U.-K. Cho, "A novel evolutionary approach to image enhancement filter design: method and applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1446–1457, 2009.

[12] C. Lezcano and J. Kanazawa, "Generación de soluciones de morfología matemática utilizando algoritmo evolutivo multiobjetivo," B. Eng. thesis, Facultad Politecnica, Universidad Nacional de Asunción, San Lorenzo, Paraguay, 2015.

[13] N. Efford, *Digital image processing: a practical introduction using java (with CD-ROM).* Addison-Wesley Longman Publishing Co., Inc., 2000.

[14] R. C. Gonzalez and E. Richard, *Woods, digital image processing.* Prentice Hall Press, 2002.

[15] J. García-Nieto, E. Alba, L. Jourdan, and E. Talbi, "Sensitivity and specificity based multiobjective approach for feature selection: Application to cancer diagnosis," *Information Processing Letters*, vol. 109, no. 16, pp. 887–896, 2009.

[16] C. E. Metz, "Basic principles of ROC analysis," in *Seminars in nuclear medicine*, vol. 8, no. 4. Elsevier, 1978, pp. 283–298.

[17] M. A. Kupinski and M. A. Anastasio, "Multiobjective genetic optimization of diagnostic classifiers with implications for generating receiver operating characteristic curves," *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 675–685, 1999.

[18] R. M. Everson and J. E. Fieldsend, "Multi-class roc analysis from a multi-objective optimisation perspective," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 918–927, 2006.

[19] T. Suttorp and C. Igel, "Multi-objective optimization of support vector machines," in *Multi-objective machine learning.* Springer, 2006, pp. 199–220.

[20] C. A. C. Coello, A. H. Aguirre, and E. Zitzler, *Evolutionary multi-criterion optimization.* Springer, 2005.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[22] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation*, vol. 2, no. 3, pp. 221–248, 1994.

[23] A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 38, pp. 35–44, 2004.

[24] (2011) Testez-vous en images. oncopeau. [Online]. Available: http://info-melanome.net/pub/en_savoir_plus/testez_vous_en_images/

[25] (2011) Digital retinal images for vessel extraction. [Online]. Available: http://www.isi.uu.nl/Research/Databases/DRIVE/

[26] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK report*, vol. 103, 2001.

[27] J. Moore, "Application of particle swarm to multiobjective optimization," *Technical report*, 1999.