Metaheuristic Optimization Design for Image Segmentation: Applications to Brain MRI Images

[1] Sara Riahi, [2] Azzeddine Riahi

[1] *Department of Mathematics and Computer Science, Chouaib Doukkali University, Faculty of Sciences , Po Box 20 , Postcode, 24000, El Jadida, Morocco.*
riahisaraphd@gmail.com

[2] *IMC Laboratory, Department of Physics , Chouaib Doukkali University , Faculty of Sciences, Po Box 20 , Postcode, 24000, El Jadida, Morocco*
riahikh@gmail.com

Abstract—*This paper presents a new approach for image segmentation based on the metaheuristic "Optimization by Ant Colonies". It is a method of classification without initial partition required or information a priori. It is based on the principle of stochastic exploration of a combined ant colony with the theory of Markov fields for modeling the field labels, and field observations. We propose to use the metaheuristic ant colonies to estimate the fields of labels and build an optimal partition of the image.*

Keywords— *Segmentation, Classification, Optimization, Ant Colony, Pareto, Metaheuristics.*

1. Introduction:

The optimization problems currently occupy a growing place in the scientific community. These problems can be combinatorial (discrete) or continuous variable, with one or more targets (single or multi-objective optimization), static or dynamic, with or without constraints. This list is not exhaustive and can be a problem, for example, both continuous and dynamic. An optimization problem is defined by a set of variables, an objective function (or cost function) and a set of constraints. The search space is the set of possible solutions of the problem. It has a dimension for each variable. For practical reasons and computing time, the search space for resolution methods is usually finished. This last limitation is not a problem, since in general the decision-maker precisely specifies the domain of definition of each variable. The objective function defines the goal to be achieved, we try to minimize or maximize it. The set of constraints is usually a set of equalities and inequalities that the variables must satisfy. These constraints limit the search space. Optimization methods look for a solution, or a set of solutions, in the search space that satisfy all constraints and minimize or maximize the objective function. Among these methods, metaheuristics are generic optimization algorithms: their goal is to allow the resolution of a wide range of different problems, without requiring any major changes in the algorithm. They form a family of algorithms to solve difficult optimization problems, for which we do not know a more efficient classical method.

The metaheuristics are generally inspired by analogies with physics (simulated annealing), with biology (evolutionary algorithms) or ethology (ant colonies, particle swarms). All kinds of extensions have been proposed for these algorithms, especially in dynamic optimization. Dynamic optimization strives to minimize or maximize an objective function that varies with time. In the previous papers [1], [2] we posed the problem of image segmentation: the extent of the search space of the solutions, the relevance of the characteristic parameters, the complexity and the diversity of the segmentation criteria (objective functions). In this paper, we reformulate the problem of image segmentation by examining it from the standpoint of optimization of technical parameters, and we present some optimization metaheuristics to solve it.

The rest of the paper is organized as follows: in part 2 we present the formulation of the segmentation problem, in part 3 we detail the multi-objective optimization, part 4 is devoted to the study of different Metaheuristics methods such as Simulated annealing , Microcanonical annealing, Genetic Algorithms, The Particle Swarm Optimization (PSO), Ant colony algorithms , the performance study and results analysis are the subject of part 5, and conclusions are given in part 6

2. Formulation of the segmentation problem

The segmentation of an image by the "region-based" approach, or by thresholding, can be reduced to an optimization problem, most often NP-difficult. Hence the need to use a metaheuristic. The segmentation of an image I using a homogeneity attribute A, is frequently defined as a partition $P = R_1, ..., R_n$ of I , such that[3]:

1. $I = \bigcup R_i, i \in [1, n]$
2. R_i is related, $\forall i \in [1, n]$
3. $A(R_i)$ True
4. $A(R_i \cup R_j) = wrong$, $\forall i \in [1, n]$ for every pair (R_i, R_j) related areas

We observe that the uniqueness of the segmentation is not guaranteed by these four conditions. The segmentation results depend not only on the information in the image, but also the method used to process that information. Generally, to reduce the problem of non uniqueness of the solution, the segmentation problem is regularized by an optimization constraint of a function F, characterizing the quality of a good segmentation. So, a fifth condition is added to the first four [4] :

5. $F(\mathrm{P}^*) = \underset{P \in P_A(I)}{\mathrm{Min}} F(P)$

Where F is a decreasing function and $P_A(I)$ is the set of possible partitions of I .It is clear that condition 5 does not entirely solve the problem of uniqueness of segmentation. There are still cases where several segmentations can have the same optimal value [5]. This explains the need to apply algorithms based on metaheuristics. The determination of a vector of optimal thresholds (a configuration) in the gray level space makes the segmentation of the images comparable to an optimization problem. Hence our approach to segmentation through techniques designed to solve this type of problem. An overview of the different image segmentation criteria was presented in our paper. As we can see, this set of criteria is not exhaustive, like the many recent papers in image processing. The conclusion we have reached is that there is no single and sufficient criterion for optimally segmenting all images [6].

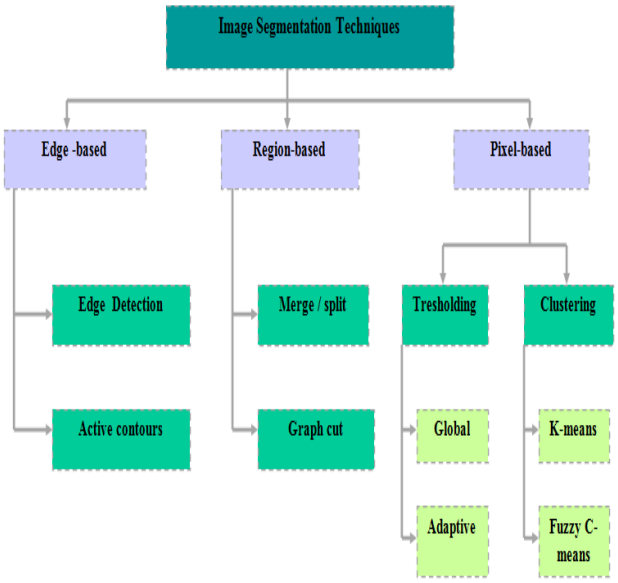

Figure 1: The different image segmentation techniques

This led us to systems schemes (algorithms) segmentation which includes several criteria. To solve image segmentation problem, we must optimize several criteria simultaneously. It is with this objective that we use the multi-objective optimization (multi-criteria). This reformulation of the problem of image segmentation into a problem of multi-objective optimization, leads us to the next section, where we will present the different approaches in this field, some of which are very little used in image processing [7].

3. The multi-objective optimization:

The multi-objective optimization is born of the need for industry to meet several conflicting requirements simultaneously. The basics of this optimization were laid by Pareto and Edgeworth in the 19th century. His theories find their first applications in economics and in recent years in the engineering sciences [8].Multi-objective problem solving approaches can be divided into three classes: approaches based on the transformation of the problem into a single objective problem (simple objective), non-Pareto approaches and Pareto approaches. These are described in the following paragraphs.

3.1 Transformation into a single objective problem:

This approach, described as a naive approach to multiobjective optimization (MO) [9], is simply to transform a multi-objective problem into a single objective problem, of which there are many methods

of resolution[10].Among the methods that use this approach, we can cite aggregation methods [11], $\varepsilon-$ constraint methods, goal-based programming methods and min-max[9].

3.1.1Aggregation method:

This method is one of the first to generate solutions to a multi-objective problem (Pareto optimal solutions).It consists in combining, linearly, several criteria f_i of the problem into a single criterion F :

$$F(\vec{x}) = \sum_{i=1}^{N} a_i f_i(\vec{x}) \qquad (1)$$

Where the parameters $a_i \geq 0$ represent the weights assigned to the criteria, $\vec{x}$ is the vector of parameters of the objective function, $\sum_{i=1}^{N} a_i = 1$ and N designate the number of criteria[12].

By varying the parameters, different supported solutions are produced. The same solution can also be produced using various parameters. The advantages of this method are its simplicity of implementation and efficiency, in that it produces a single solution, and does not require the intervention of an outside operator (a decision maker). However, to achieve this goal, two key issues need to be addressed: determining the values of the parameters associated with each criterion, and the interaction between the different criteria. Regarding the first problem, a widely used approach is to solve the problem with several values of the parameters a_i. Blind strategies have been proposed in the literature to generate, randomly, the values of the parameters [13].

Figure 2 illustrates the operation of the linear aggregation method in a bi-objective case. The calculation of the weighting parameters amounts to finding a hyperplane in the objective space (a line in the bi-objective case) with a fixed orientation. The Pareto optimal solution is the point where the hyper-plane has a common tangent with the space of feasible solutions (point x in Figure 2 (a)).Figure 2 (b) illustrates the limitations of this method. For example, in the case where the problem has a non-convex Pareto boundary, the solutions y and z can be found. While other solutions between points y and z never will be [14].

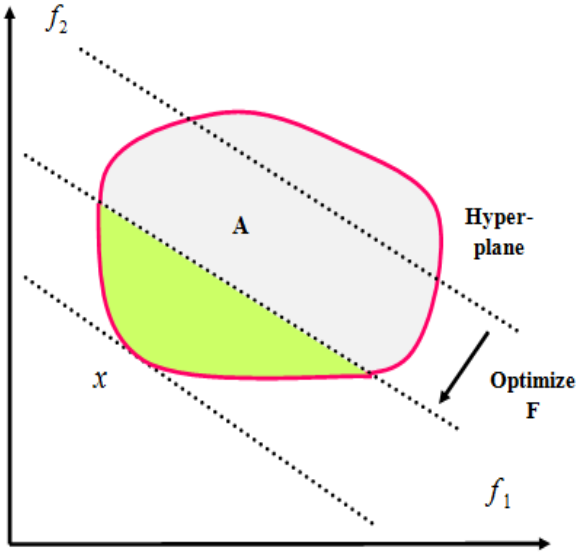

Figure 2(a) **:** Illustration of the aggregation method, in a bi-objective case. A convex Pareto boundary, A: feasible domain, x : Pareto solution.

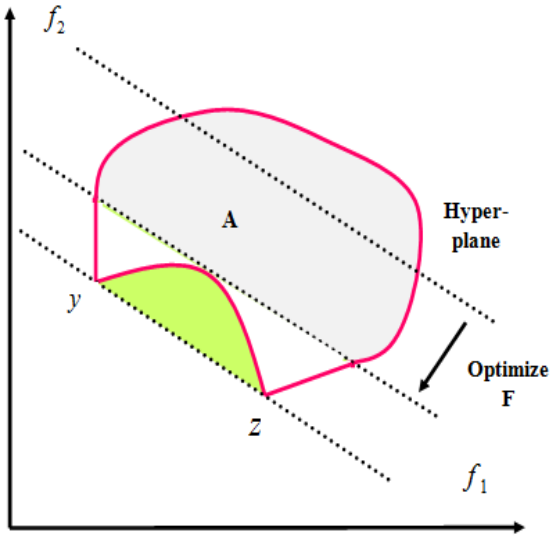

Figure 2(b)**:** Illustration of the aggregation method, in a bi-objective case. A non-convex Pareto boundary. A: feasible domain, x : Pareto solution.

3.1.2 Other processing methods:

The method, called $\varepsilon-$constraints, processing a multi-objective problem in single-objective problem is based on the optimization of an objective function subject to constraints f_j on other functions $(j \neq i)$. For example, in the case of minimizing the multiobjective problem can be written as [15]:

$$\underset{x\in A,\, f_j(x)\le \varepsilon_j}{Minimize}\ f_k(x) \qquad (2)$$
$$j=1,\ldots n;\ j\neq k$$

This makes it possible to transform a multi-objective problem into a single objective under constraints and to solve it with the mono-objective optimization methods. A priori knowledge of the appropriate intervals for the values εj is necessary for all objectives[16].Another method widely used in this multi-objective problem solving approach is goal and min-max programming. In this method, the goals (values) to be attained for each objective must be defined a priori [17].These values are then used to transform the multi-objective problem into a single-objective problem. For example, the criterion to be optimized can incorporate a weighted norm, which minimizes the deviation from the goals. The problem can be formulated as follows:

$$Minimize \left(\sum_{j=1}^{n} \lambda_i \left| f_i(x) - b_j \right|^p \right)^{1/p} \qquad (3)$$
$$x\in A$$

Where $1 \le p \le \infty$, λ_i is the vector of weights, and b_j is the goal vector. The standard used is the Chebysheff metric (L_p-metric). In the case of a Euclidean norm $p = 2$. Function (3) corresponds to a min-max function, in the case where $p = \infty$. The selection of the goal vector should not be random, because a wrong vector can lead to a solution that is not Pareto optimal [18].

3.2 Non-Pareto approach:

This approach is also called non-aggregated approach. Generally, methods based on this approach treat objectives separately. Two groups of methods exist in the literature: lexicographic selection and parallel selection. In the classical approach to lexicographical selection, the selection is performed according to a defined order a priori. This order makes it possible to define the weights of the objectives. Several metaheuristics have been used for solving multi-objective problems with lexicographical selection[19].In parallel selection approach, the first work was based on a genetic algorithm.Its algorithm, called VEGA ("vector-evaluated genetic algorithm"), selects the current solutions of the Pareto front according to each objective, independently of the others (parallel selection). The analysis of this algorithm showed that its behavior is the same as an algorithm performing a linear aggregation. Several authors have worked on improving this algorithm [20].

3.3 Pareto approach:

Unlike other approaches that combine criteria or treat separately, the Pareto approaches use the concept of dominance, to select solutions by converging the population to a set of effective solutions. This idea, based on genetic algorithms, has been proposed by Goldberg (Goldberg, 1989). This approach respects the integrity of each criterion because the values of the different criteria are not compared a priori. These methods make it possible to have a set of possible solutions (Pareto solutions) to the problem, but do not allow choosing an alternative rather than another. The choice of the final solution is up to the user (decision maker) [21].In the following paragraphs, we define the notion of Pareto dominance, the Pareto border and the notion of "constrained dominance".

3.3.1 Pareto Optimum:

Vilfredo Pareto is an Italian mathematician of the nineteenth century (Pareto, 1896). He laid the foundation for solving a multi-objective economic problem: "In a multi-objective problem, there is a balance such that we can improve a criterion without damaging at least one of the other"[22]. This equilibrium is called optimum Pareto. So, a solution x is said Pareto optimal if it is not dominated by any other solution belonging to A. These solutions are called non-dominated solutions, or not less.

3.3.2 The concept of dominance:

Let x be a potential solution to the multiobjective problem, $x \in A$ dominates $x' \in A$ if:

$$\forall i,\ f_i(x) \le f_i(x') \qquad (4)$$

with at least i such that $f_i(x) < f_i(x')$

A solution x is said weakly not dominated, if there is no solution $x' \in A$ such that:

$$f_i(x) < f_i(x') \qquad (5)$$

A solution x is said to be strongly non-dominated, if there is no solution $x' \in A$ such that:

$$f_i(x') < f_i(x) \qquad (6)$$

with at least i such that $f_i(x) < f_i(x')$

Where $i = 1, ..., n$ and n is the number of goals. Figure 3 illustrates the definition of the concept of dominance. In this example, the multi-objective problem is to minimize f_1 and f_2. The solutions represented by points 1, 4 and 5 are not dominated by any other solution [23].

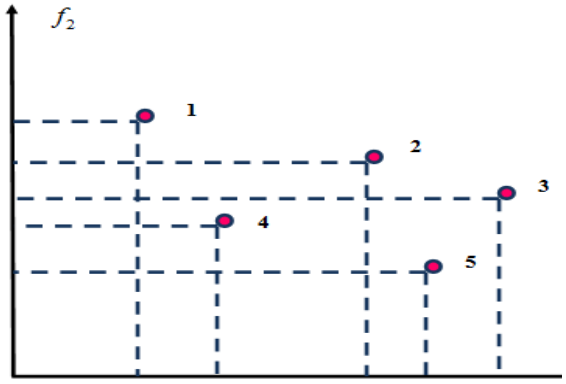

Figure 3: Illustration of the concept of dominance. f_1 and f_2 are two objective functions to minimize. Solution 4 dominates the solutions 2 and 3.

3.3.3 The Pareto border:

The border, also called the Pareto front, is the set of optimal Pareto points. Figures 4 (a) and (b) show two of a multitude of other forms of Pareto fronts [24]. In the example of Figure 3, the Pareto front is composed of points 1, 4 and 5.

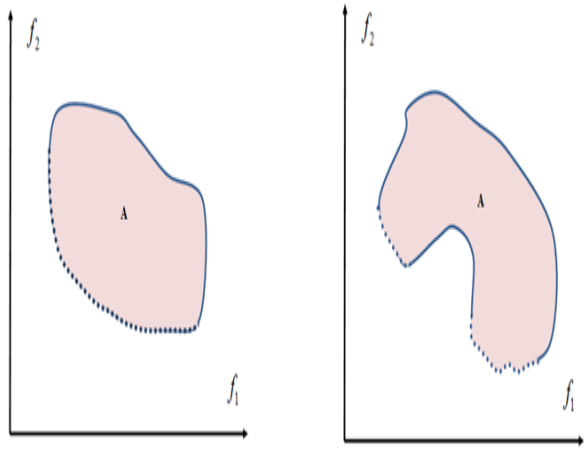

Figure 4 : Examples of Pareto fronts. A: feasible domain. (bi-objective problem).

3.3.4 The notion of dominance with constraint:

A i solution dominates with constraint j when a solution of the following occurs [21]:

- The solution i is realizable whereas the solution j is not realizable;
- Both solutions are feasible and the solution i dominates j;
- Both solutions are not feasible, but i have a number of constraint violations lower than j.

4. Metaheuristics:

In the literature, heuristic methods are divided into two classes: specific algorithms for a given problem, using domain knowledge [25], and general algorithms applicable to a wide variety of multi-objective problems: metaheuristics [26].In this part, we will focus on the second class of algorithms. To solve multiobjective problems and to determine optimal Pareto solutions, several adaptations of the metaheuristics are proposed in the literature.

4.1 Simulated annealing:

The origin of the simulated annealing comes from the analogy with the metal, wherein the method, to achieve low energy states of a solid, is to raise the temperature of the solid to high values and then allowing the cool slowly. This process is called "annealing". The basic idea of the optimization algorithm called "simulated annealing" is as follows: at decreasing temperature levels, the algorithm uses the iterative procedure of Metropolis, to reach a state of thermodynamic quasi-equilibrium. This procedure makes it possible to output local minima with a probability that increases as the temperature rises. When the algorithm reaches very low temperatures, the most probable states are in principle excellent solutions to the optimization problem [27].

4.1.1 Metropolis Algorithm:

In 1953, Metropolis proposed an iterative algorithm that achieves the thermodynamic equilibrium state of a simulated system at a temperature T. Its principle is to iterate the two following steps:

- Evaluate the energy variation associated with a random elementary transition from the current state i, of energy E_i, to a new state j, of energy $E_j : \Delta E_{ij} = E_j - E_i$;
- Accept the transition to the new state with probability P_{ij}

Where

$$\begin{cases} P_{ij}(T) = 1 & \Delta E_{ij} \le 0 \\ P_{ij}(T) = \exp\left(-\dfrac{\Delta E}{T}\right) & \text{If } \Delta E_{ij} > 0 \end{cases} \quad (7)$$

4.1.2 Simulated annealing algorithm:

The principle of simulated annealing, presented in Algorithm 1, consists of starting the algorithm with a high initial temperature and controlling the decrease of temperature (7) within the Metropolis algorithm. Several laws of decreasing temperature can be used in practice [28].

Algorithm 1: Simulated annealing algorithm

```
1 Define the objective function (f)
2 Choice of disturbance mechanisms of a configuration ΔS
3 Pull a random configuration S
4 Calculate the energy associated with this configuration E
5 Initialize the temperature (T0)
6 While Stop conditions not satisfied do
    6.1 While the thermodynamic equilibrium not reached do
        6.1.1 Pull a new configuration S'
        6.1.2 Apply the Metropolis rule
        6.1.3 If f(S') < f(S)
            f_min = f(S')
            S_opt = S'
            End if
        End while
    6.2 Decrease the temperature
  End while
7 Display the optimal solution
```

We distinguish two types of simulated annealing, according to the method of decreasing temperature:

- Homogeneous annealing: the temperature is lowered only when the thermodynamic equilibrium is reached. This algorithm assumes that the Metropolis procedure is iterated an infinite number of times and therefore has only a purely theoretical interest.
- Heterogeneous annealing: The temperature parameter is decreased after a finite number of disturbance evaluations at a given temperature.

In summary, the main control parameters are:

- The initial value of the temperature;
- The function of decreasing temperature;
- The criterion of changing temperature bearing;
- The cessation criterion

The effectiveness of the simulated annealing depends strongly on the choice of its control parameters, whose adjustment remains very empirical.

For the calculation of the flow temperature, several methods have been proposed [29]. One of the methods is based on the observation of the average variation of the function f. From an initial configuration S_0, a number of solutions $S_0^{'}$ (about 50 to 100) such that $f(S_0^{'}) > f(S_0)$ is drawn, so the average variation $\langle \Delta f \rangle$ is calculated. The initial temperature T_0 is calculated so as to initially accept a certain percentage (p) of movements degrading the function f. For an average variation $\langle \Delta f \rangle$ of the function f around the initial configuration S_0, the value of p is 50%. The value of T_0 is deduced from the following relation:

$$p = \exp\frac{-\langle \Delta f \rangle}{T_0} \quad (8)$$

The setting of the temperature decrease is very important in the simulated annealing algorithm. A sharp decrease of temperature risks trapping the algorithm in a local minimum, whereas a weak decay at the beginning of the process causes a slow convergence of the algorithm. From a theoretical point of view, the theoretical convergence of the inhomogeneous simulated annealing is ensured with a logarithmic law [30]:

$$T_k = \frac{\mu}{\log(1+k)} \quad (9)$$

Where k the number of temperature steps performed is, μ is a positive constant equal to the maximum depth of the local minima. Unfortunately, this rule induces a prohibitive calculation time. In practice, the geometric decay is often used:

$$T_{k+1} = \alpha . T_k \quad (10)$$

Where α is a constant in the interval $[0,1]$.

Regarding the change in temperature, the number of changes after which the temperature is lowered can be simply specified. The annealing has been a great success in various fields of application [31], thanks to its two main advantages: a black box performance and

ease of "adjustment" of the internal parameters. From a theoretical perspective, simulated annealing allows the optimal solution of the problem to be approached more quickly than exhaustive exploration in the research space. In practice, a good adjustment of the internal parameters of the algorithm makes it possible to accelerate the convergence towards a pseudo-optimal solution, with a predefined precision. Simulated annealing is also suitable for solving continuous optimization problems [32].The major disadvantage of this algorithm is its slowness. However, several attempts to parallelize the algorithm have been proposed in the literature, to the detriment of its theoretical convergence. It should be noted that there is fast approaches that are simple to implement and maintain the convergence properties.In multi-objective optimization simulated annealing (SA) was applied after processing of multi-objective problem into a single-objective problem [33].The two most popular methods are: the Multiple Objective Simulated Annealing (MOSA) method, where the SA was used to search for the compromise area , and the PASA method (Pareto Archived Simulated Annealing) which uses an aggregation function of the coupled objective functions, with a system for archiving non-dominated solutions.

4.2 Microcanonical annealing:

This optimization method exploits principles very similar to those of simulated annealing. The microcanonical annealing performs decreasing total energy levels by decreasing the kinetic energy between two bearings. Thus the algorithm converges by reducing the energies of a set of solutions around those that are [34].The microcanonical annealing implements Creutz algorithm (Creutz, 1983), which is based on the evaluation of a transition sequence to maximize the entropy for a constant total energy. This energy is previously set.

4.2.1 CREUTZ Algorithm:

For a total energy E_t, an iterative algorithm makes it possible to converge towards thermodynamic equilibrium by repeating a large number of times the two following steps:

- Evaluating the energy variation associated with a random elementary transition from the current state i, from potential energy E_i, to a new state j of energy $E_j : \Delta E_{ij} = E_j - E_i$
- Acceptance of this transition to the new state if $\Delta E_{ij} = E_t - E_i$. In the original article by Creutz (Creutz, 1983), the term $E_t - E_i = D$ is called "Demon" and can be interpreted as the kinetic energy of the system in the i state.

Transitions to higher potential energy states are allowed only if there is enough kinetic energy to offset the increase of potential energy, and therefore remain constant energy.

4.2.2 Microcanonical Annealing Algorithm:

The microcanonical annealing algorithm is to reduce the total energy, from a high total energy within the Creutz algorithm. Several laws of energy decay can be used, as in the case of simulated annealing. The different steps of the algorithm are summarized in Algorithm 2. Creutz's algorithm is much simpler than that of Metropolis, and requires much less computation. As there is no need of a random number generator, its implementation is simplified compared to other algorithms. Compared with simulated annealing, in the case of large problems, several studies have shown that the results obtained are very similar, with a benefit for microcanonical annealing in terms of calculation [35].On the other hand, the convergence of this algorithm is still not proved theoretically. In some cases, the microcanonical annealing may be less effective, because of the energy barriers impassable that he built during his research and that can trap in a local minimum. This algorithm can possibly be applied to solve multi-objective problems in the case where they are transformed into a single-objective problem.

Algorithm 2: microcanonical annealing algorithm

1 Initialization
1.1 Define the minimum percentage p of transitions accepted in the first level
1.2 Set initial E_t as the rate p of the transformations tested are accepted
1.3 Pull randomly an acceptable solution and calculate its energy E
1.4 Select the maximum number of transitions at each level tested of total energy
2 while the number of accepted solutions is non-zero **do**
2.1 Repeat for a given number of iterations, randomly draw a transition and calculate the associated energy variation ΔE
2.1.1 Accept the transition if $\Delta E \le E_t - E$
2.1.2 If $\Delta E \le 0$, compare the new state to the best state found since the start of the search, and memorize it if it is better
2.2 Decrease $E_t : E_t = \alpha E_t$
End while
3 Display the best state encountered during the search

4.3 Genetic Algorithms:

4.3.1 Principle:

Genetic algorithms belong to the class of evolutionary algorithms. These are metaheuristics inspired by the analogy between an optimization process and the evolution of living beings. The simulation of variation and selection mechanisms operating in natural evolutionary processes is exploited to solve artificial problems of optimization [30]. In evolutionary algorithms, the analogy with optimization is to consider potential solutions to the problem as chromosomes. These are handled by selection and mutation operators. In the case of genetic algorithms, a third operator is used, called the crossover operator. The quality of the solution corresponding to each chromosome is quantified via its own fitness. Crossing and mutation procedures are designed to permanently create new chromosomes.

Genetic algorithms remain the same size of the population of potential solutions, also known as individuals. The initial population is usually chosen randomly. However, other initialization mechanisms can be used depending on the application [32]. The number of times an individual is selected for recombination depends on its fitness relative to the rest of the population. The Algorithm 3 procedure presents the different basic steps of a genetic algorithm. In this example, $P(t)$ represents a population of candidate solutions for a given problem, at iteration t. In the following paragraphs, we describe the different procedures that make up a genetic algorithm.

In a genetic algorithm, the representation of individuals is based on the binary coding of information. The problem is that there are many ways to code the information, and the optimal choice is not easy to find .It must indeed allow an efficient search by the operators that will be applied to the chromosomes. To solve this problem, an adaptation of the representation and operators in the search space, associated with the problem addressed, is essential. In such a representation, both genotypes whose representations differ slightly should provide close solutions.

This property is not always verified in the case of a conventional binary coding. For example, the transition from the integer 7 to the integer 8, represented respectively by the chromosomes 0111 and 1000, is done in 4 mutations (this problem is known as the Hamming barrier). Various possibilities have been proposed, such as Gray encoding [27].In this coding, the transition from one number to another is done by inverting one bit of the chain. However, in practice, it is rare that these encodings are applied.

Algorithm 3: Principle of a genetic algorithm

```
1 Initialization P(t)
2 Evaluate each individual of P(t)
3 while the stopping criterion is not satisfied do
   3.1 t = t+1
   3.2 Select P(t+1) of P(t)
   3.3 Cross P(t+1)
   3.4 Mutate P(t+1)
   3.5 Evaluate P(t+1)
End while
4 Display the best condition encountered during the search
```

4.3.2 Genetic algorithms in multi-objective optimization:

Genetic algorithms have been widely used for solving multi-objective problems, with all the approaches to multi-objective optimization [33].

The GAs for the transformation approach towards the mono-objective

In the approach transforming a multi-objective problem into a single-objective problem. In the case of the $\varepsilon-$constraint approach, GAs has been used extensively to solve this class of problems [29].

The GAs for the non-Pareto approach

In the case of the non-Pareto approach, the most common GA is that proposed by Schaffer (Schaffer, 1985) known as VEGA. The particularity of this GA is to create sub-populations whose individuals are dedicated to a particular purpose.

The GAs for the Pareto approach

In this type of approach, two families of methods emerge: non-elitist methods and elitist methods. Among the non-elitist methods, we can cite the MOGA method ("Multiple Objective Genetic Algorithm"), where each individual is classified according to the number of people it dominates, NSGA method ("Not Dominated Sorting Genetic Algorithm") where the calculation of fitness is done

by separating the population into several groups, according to the degree of Pareto domination of each individual. Elitist techniques do not save the optimal Pareto individuals found during iterations. They are distinguished in two ways: the difficulty of maintaining diversity and their slow convergence towards the Pareto border. Among the many GAs adopting an elitist strategy, we quote the SPEA algorithm ("Strength Pareto Evolutionary Algorithm"), where the transition from one generation to another begins with the update backups. All non-dominated individuals are saved and dominated individuals, already present, are eliminated. After having improved this algorithm, better performances have been recorded, but at the cost of accentuating its complexity [35]. The best known and most preferred algorithm is NSGAII, a second version of NSGA.

4.4 The Particle Swarm Optimization (PSO):

The metaheuristic based on the particle swarm method ("Particle Swarm Optimization", PSO). The principle of the method is derived by analogy with collective animal behavior. The particle swarm optimization (PSO) is classified in stochastic optimization techniques to population. A brief description of the PSO algorithm is presented in the following paragraphs.

4.4.1 Principle:

The PSO method is identified with genetic algorithms. However, it has no evolution operator. In addition, a population of potential solutions is used in research. The algorithm starts with a random initialization of the particle swarm in the search space. Each particle is modeled by its position in the search space and by its velocity. At any time, all particles adjust their positions and velocities, so their paths, relative

- To their best positions.
- To the particle having the best position in the swarm.
- To their current position.

In reality, each particle is influenced, not only by its own experience, but also by that of other particles. The position and velocity of a particle in a N - dimensional search space are defined by: $P_i = (p_{i,1},...., p_{i,N})$ and $V_i = (v_{i,1},...., v_{i,N})$, respectively. Each particle is characterized by its best position $L_i = (l_{i,1},...., l_{i,N})$ at iteration t. The best position reached by the swarm is saved in the vector $G = (g_1,...., g_N)$. The velocity of each particle is updated according to the following expression [26]:

$$v_{ij}(t+1) = K\left[w.v_{ij}(t) + c1.r1.(l_{ij} - v_{ij}(t)) + c_2.r_2.(g_j - v_{ij}(t)) \right] \quad \textbf{(11)}$$

and

$$K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|} \quad \textbf{(12)}$$

where $\varphi = c_1 + c_2$ and $\varphi > 4$. The following values $j = 1,...., N$, w is a constant called inertia factor, c_1 and c_2 are constants called acceleration coefficients, r_1 and r_2 are random numbers uniformly distributed in the interval $[0,1]$. If the calculated velocity brought out a particle of the search space, its fitness is not calculated. Given the new speed, obtained from (11) and (12), the position at the iteration $t+1$ is then calculated [24]:

$$p_{ij}(t+1) = p_{ij}(t) + v_{ij}(t+1) \quad \textbf{(13)}$$

For $j = 1,...., N$

The inertia parameter w controls the influence of the old speed on the current speed, in order to allow the particles to avoid the local minima. Similarly, c_1 controls the behavior of the particle in its research on his best position and c_2 control the influence of the swarm behavior of the particle. The different steps of the algorithm are presented in Figure Algorithm 4. This method has been very successful with the optimization community. Its good performance in different applications, and the possibility of hybridization with other metaheuristics contributed to this enthusiasm. Despite its "young age" in comparison with other metaheuristics, a large number of works have been published. Most of the variants of the method, as well as the different hybridizations with other metaheuristics, are briefly presented.

Algorithm 4: Principle of PSO algorithm.

1 **Initialize** the particle population with random positions and velocities
2 **Evaluate** the objective function for each particle and calculate g
3 For each individual i, L_i is initialized to P_i
4 **Repeat** until the stop criterion
  4.1 Update particle velocities and positions
  4.2 Evaluate the objective function for each individual
  4.3 Calculate the new L_i and g
5 **Display** the best condition encountered during the search

Recent studies have shown the possibility of using PSO algorithm as a black box with the image of simulated annealing [28], which encouraged the application of the method in different fields [36].

4.4.2 The particulate Swarms Optimization in multi-objective optimization:

In order to extend the PSO to multiobjective optimization, it was necessary to modify the original formulation of the diagram of a multiobjective problem. For that we had to answer the following three questions:

- How to select particles to give preference to one that is not dominated over others
- How to retain optimal Pareto solutions compared to those already seen and encountered during the research
- How to maintain diversity in the swarm to avoid convergence to a single solution

The PSO for the approach transformation to the mono-objective

In this approach, some algorithms dedicated to the resolution of a multi-objective problem, by transforming it into a single-objective problem, have been proposed in the literature. One of the approaches, using linear targets aggregation, divides the swarm into sub-swarms of equal size. Each sub-swarm uses a vector of different parameters and evolves in his own direction. Finally, optimal Pareto solutions are chosen using the gradient method.

The PSO for the non-Pareto approach

Several algorithms have been developed. Many are inspired by GAs, like the VEPSO algorithm derived from VEGA algorithm. VEPSO uses a so-called multi-swarm approach, where each swarm is evaluated using one of the objective functions of the problem. Different Swarms communicate with each other through the exchange of their best position [37].

The PSO for the Pareto approach

In this multi-objective optimization approach that the number of published works is the most important. These approaches use selection methods of "leader" based on Pareto dominance. The basic principle of all these approaches is to consider particles as leaders, if they are not dominated in the swarm. Several leader selection schemes are possible [38].

4.5 Ant colony algorithms:

This metaheuristic is inspired by the collective deposition and tracking behavior observed in ant colonies. Ants communicate indirectly through dynamic modifications of their environment (pheromone trails) and thus construct a solution to a problem account their collective experience. The illustration of the ability of ants to find the shortest path is given through the example of the emergence of an obstacle on a path between food and nest.Figure 5 shows an illustration of the procedure followed by the ants to get food [39].

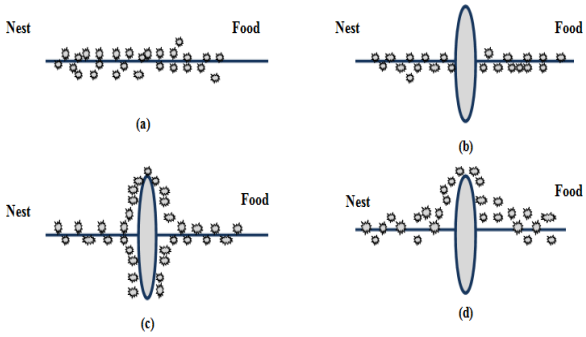

Figure 5: Illustration of the ability of ants to search for food by minimizing their course. (a) Search without obstacle,(b) Appearance of an obstacle,(c) Finding the optimal path, (d) Optimal path found.

Initially, ants move from their nest to the food (Figure 5 (a)).At the onset of the obstacle, ants alter their path to bypass (Figure 5 (b) and (c)). The colony is then divided into two groups; the first group selects the longest path, while the latter chooses the shortest. Since all ants deposit their pheromones on their path, the shortest path will eventually have the highest pheromone density. Therefore, the number of ants following this path increases.

Over time, the amount of pheromones deposited on the longest path decreases and eventually disappears;

then all the ants follow the shortest path (Figure 5 (d)). The first ant colony optimization (ACO) algorithm. The first algorithm called "Ant System" (AS), was developed specifically to solve the Traveling Salesman Problem (TSP). The adaptation of behavior and characteristics of real ants for optimization required some modifications, hence the birth of artificial ants. They have a memory, are never blind and discrete time[40].

Modeling of artificial ants to solve the traveling salesman problem is this: at each iteration t, each ant k traverses the graph and builds a complete path of n steps $(n = Card(N))$. For each ant, the journey between a city i and a city j depends on:

- The list of cities already visited. It defines the possible movements with each step, when the ant k is on the city $i : J_i^k$
- The visibility of each ant is defined by the inverse of the distance between the cities: $\eta_{ij} = \frac{1}{d_{ij}}$ This allows ants to move to the nearest cities.
- The intensity of the track corresponds to the amount of pheromone deposited on the path connecting two cities. This defines a global pseudo-memory of the system.

Displacement heuristic called the transition probability, is defined by:

$$p\left(c_i^j \left| s^p \right.\right) = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^\alpha \left[\eta_{ij}\right]^\beta}{\sum_{l \in J_i^k} \left[\tau_{il}(t)\right]^\alpha \left[\eta_{il}\right]^\beta} & \text{If } \begin{matrix} j \in J_i^k \\ j \notin J_i^k \end{matrix} \\ 0 \end{cases} \quad \textbf{(14)}$$

where α and β are two parameters controlling the importance of the intensity of the track $\tau_{ij}(t)$, and the visibility η_{ij} .The amount of pheromone $\Delta\tau_{ij}^k(t)$ left by each ant depends on the quality of the solution found, which is modeled by the following relation[41]:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \dfrac{Q}{L^k(t)} & \text{if } \begin{matrix} (i,j) \in T^k(t) \\ (i,j) \notin T^k(t) \end{matrix} \\ 0 \end{cases} \quad \textbf{(15)}$$

Where $T^k(t)$ the path made by the ant k at the iteration t, $L^k(t)$ is the length of the round and Q is a fixed parameter. For the algorithm avoids local solutions traps, tracks an update is performed:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad \textbf{(16)}$$

where $\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^k(t)$ and m is the number of ants[42]. Initially, the pheromone is initialized by a small amount $\tau_0 \geq 0$.

Figure 6 illustrates the basic principle of an ant colony optimization algorithm. In mono-objective, the metaheuristics of the ant colonies is privileged to solve combinatorial optimization problems. The algorithm is based on the same principle as that of P-ACO. It is to separately optimize each objective. Indeed, in each cycle ants exchange objective function to optimize. At the end of each cycle, the ant with the highest quality of solution updates the pheromone track, depending on the solution found.

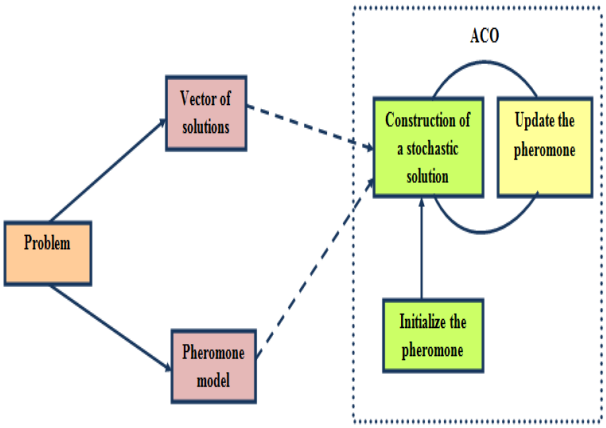

Figure 6: Basic principle of the ACO algorithm

5. Performance study and results analysis:

The image segmentation is frequently used in medical imaging method. The images from MRI (Magnetic Resonance Imaging) have indeed some stakes. Reading Them may be crucial for the diagnosis of certain neurodegenerative diseases, such as Parkinson's or Alzheimer's disease : Increasing the spatial resolution of an image (increasing the quality of the measurement, and the accuracy of the image) seems at first sight an effective tool for improving the relevance of a diagnosis. The increase in spatial resolution is characterized by the use of a more

powerful magnet, therefore more expensive. However, this resolution results in increased sensitivity of the machine, it will therefore be more sensitive to noise. The image gains in precision and paradoxically will have a form of noise that can in turn hinder the reading of the image, and thus the diagnosis. Fortunately, the correction of this noise can be done by segmentation (Figure 7).

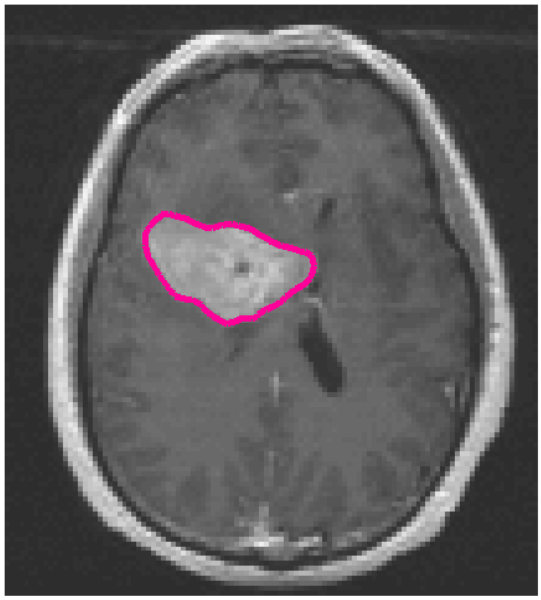

Figure 7: Location of tumor according to segmentation

The application of this method then makes it possible to take advantage of the good spatial resolution of the image, and to correct the noise that can be observed on the measurement. Each image has a noise of its own, so a good segmentation for an image is only reusable for another image. If using a segmentation seems relevant in this case, it remains sensitive to a number of parameters such as the number of classes and their compositions. The uniqueness of each solution is problematic because there is no good universal method for MRI image segmentation. The resolution of this problem can be long, and can lead to misinterpretations.

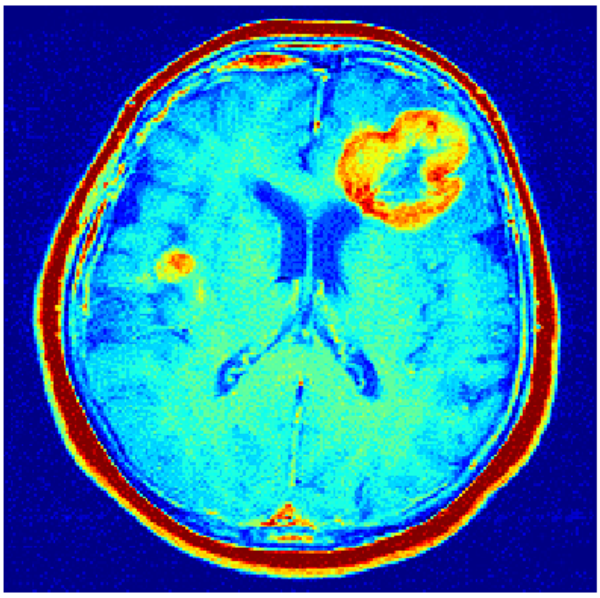

Figure 8: Original image

The step following of the image preprocessing is segmentation. We will implement the various segmentation techniques mentioned above to detect the different elements of the fabric. Subsequently we detail the results obtained from the application of these techniques on our image (Figure 8).

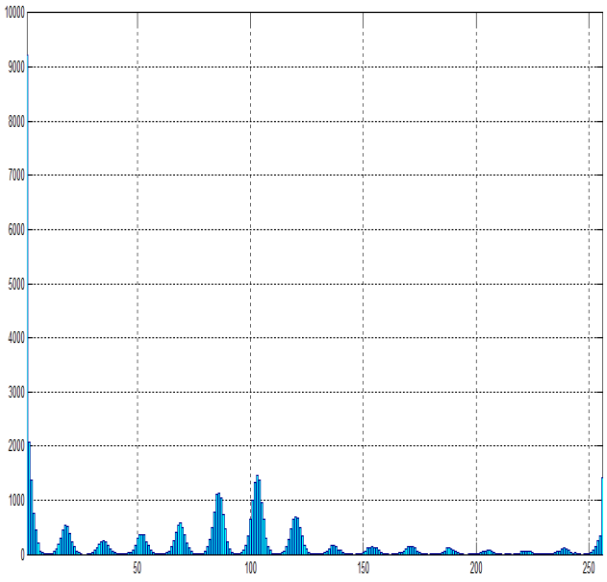

Figure 9: Histogram of original image

We opted for internal markers (local minima) to mark the objects to be segmented and external markers containing the objects to be segmented. And as we have illustrated in the figures the result is satisfactory because we detected some outlines. Contour approach segmentation is concerned with the contours of the object in the image.

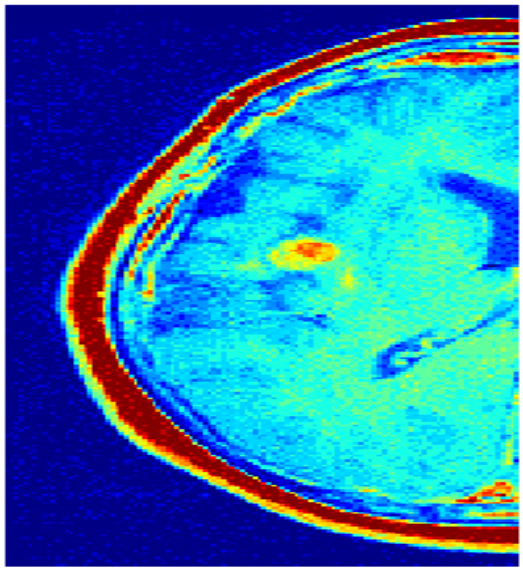

Figure 10: The left half image

Most of the algorithms associated with it are local, that is, they work at the pixel level (Figure9). Edge detection filters are applied to the image and generally give a result that is difficult to exploit.

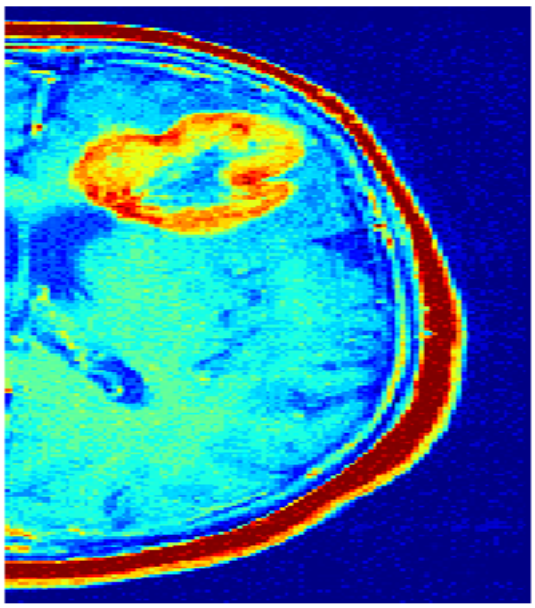

Figure 11: The right half image

The outlines extracted are most of the time fragmented and not very precise, it is then necessary to use contour reconstruction techniques by interpolation or to know a priori the shape of the sought object. Formally, this type of algorithm is close to the methods of increasing regions operating at the pixel level (Figure 10) and (Figure 11). These purely local techniques are generally too limited to handle complex images.

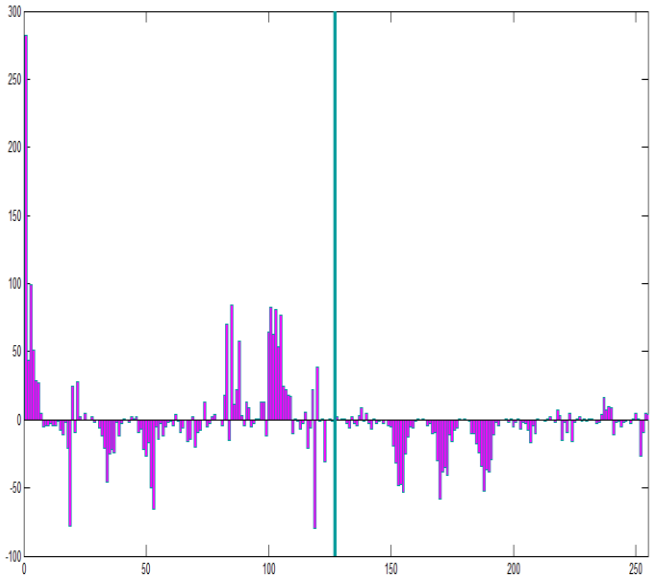

Figure 12: The difference between the histogram of left half and right half of Image

It is recalled that the gradient in a pixel of a digital image is a vector characterized by its amplitude and direction. The amplitude is directly related to the amount of local variation of the gray levels (Figure 12).

In this section we focus on the segmentation region which is a specific approach in which we seek to build surface by grouping adjacent pixels according to a criterion of homogeneity. For this, we followed two different approaches.

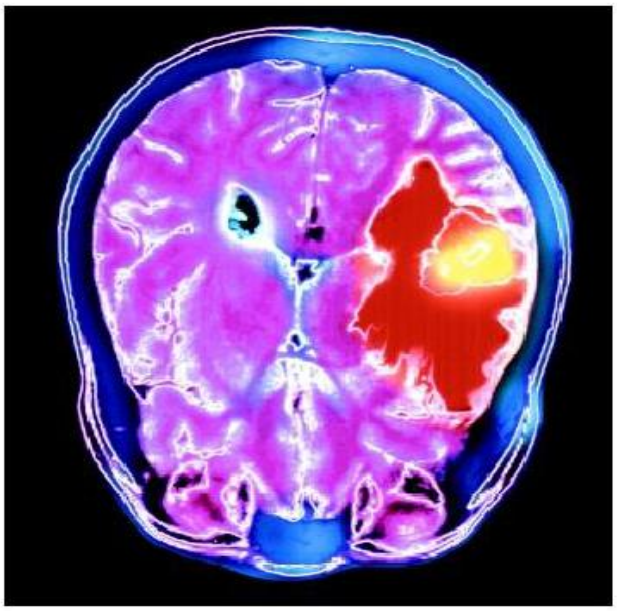

Figure 13: Original diseased color Image

The great problem of this procedure is the right choice of threshold (Figure 13). We note that the automatic thresholding does not give us bad results against the manual thresholding we offer satisfactory performance because we relied on the distribution of pixels in the image histogram to extract the threshold (Figure 15). We used a global thresholding (binarization) to partition the image into two classes (the object and background).

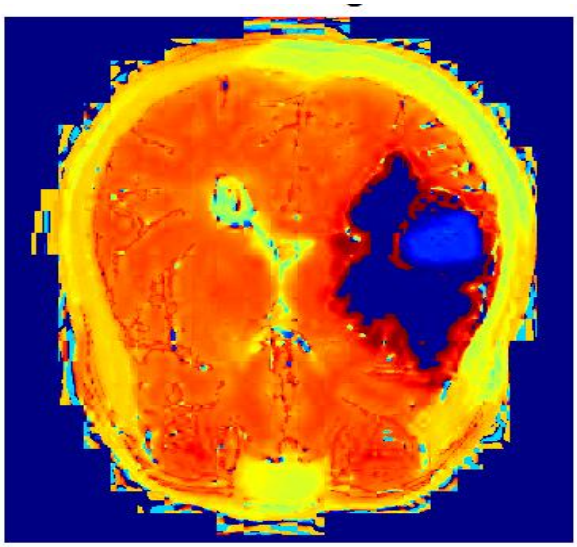

Figure 14: Hue Image

The automatic classification methods (global techniques) aim to establish a relationship between all the points of the image and all classes. The result of the segmentation is an image where each image point carries the label of a class (Figure 14). These methods differ from each other in the manner of calculating class representatives or kernels, and the limits of these classes. In our work we used three approaches:

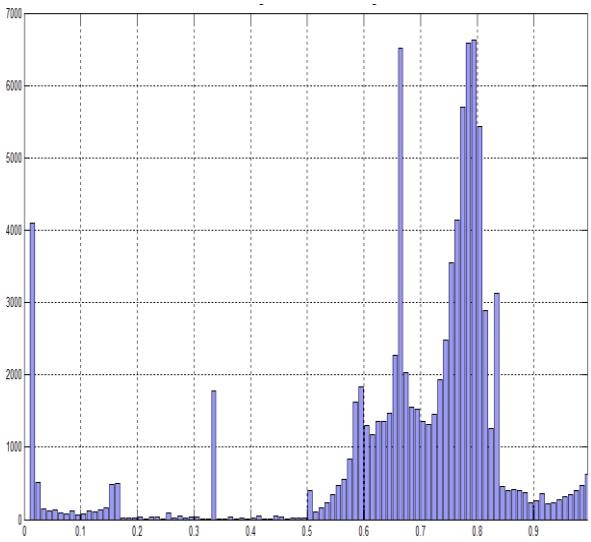

Figure 15: Histogram of hue image

K-Means, Fuzzy C-Means (FCM) and Particle Swarm Optimization (PSO).

For segmentation by K-Means, the segmentation criterion here is the color of the pixels. We choose k different color regions. Each point is assigned to the region with the closest color, and then we recalculate the color of each region until they are more distinct. We applied the algorithm on our images, we obtained the following results (Figure 16) and (Figure 17):

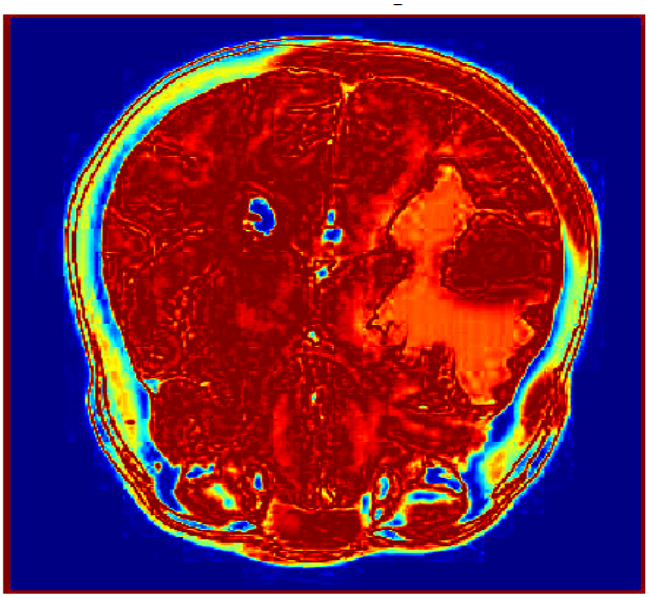

Figure 16: Value Image

K-means is an objective technique; it minimizes the value of a numerical criterion. It is an optimization technique. As is often the case in optimization, the K-means algorithm stops when it can no longer lower the value of the criterion.

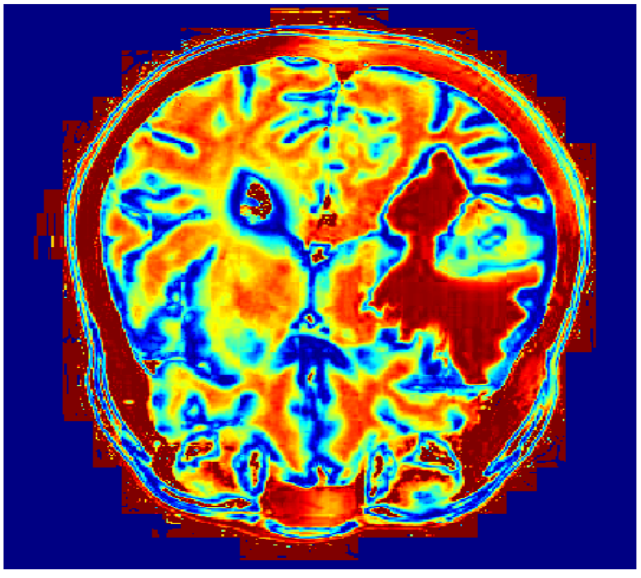

Figure 17: Saturation Image

The big disadvantage is the choice of the value of K, the number of classes. This choice can be made by simple visual examination in the case of two-dimensional data, but it is not the same for data of greater dimension. There is usually no clear indication of the appropriate number of classes, and a "wrong choice" for the value of K will then lead to a typology unrelated to reality. A notable aspect of this segmentation is the fact that two distant objects can belong to the same region. One of the difficulties of this method and the choice of the number of classes. We notice that the result of this segmentation is satisfactory and note that at each execution the class assigned to each region is not the same.

For segmentation by Fuzzy C-Means (FCM) We tested this method on our images and here is what we obtained as results: As in the case of K-Means segmentation we found difficulties in choosing the number of classes and we modified it for each image. We notice that the result of this segmentation is good.

Segmentation by Particle Swarm Optimization (PSO),The implementation of this function on our image gave us the following result:

6. Conclusion:

In this paper, we presented a formulation of image segmentation as an optimization problem in single-lens. Then we proposed a reformulation of the problem in multi-objective optimization. The basic principles of multi-objective optimization and its different approaches have been exposed. The three approaches described will be applied to image segmentation in our next work. As we pointed out in part 2, metaheuristics, in mono-objective optimization, play an important role in the field of image segmentation. Therefore, our next work will be dedicated to our contribution to segment images by metaheuristics in single lens.

References:

[1] Azzeddine Riahi ,''MRI Image Segmentation by KMeans Clustering Method and Detection of Lesions'', International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064, Volume 4 Issue 6, Page No(2484-2492),June 2015

[2] Azzeddine Riahi, Image Segmentation Techniques Based on Fuzzy C-Means and Otsu, Applied to the Brain MRI in Tumor Detection, International Journal of Computer Sciences and Engineering,Volume-03, Issue-12 E-ISSN: 2347-2693,2015

[3] Ian T. Young ,Jan J. Gerbrands ,Lucas J. van Vliet,' Fundamentals of Image Processing', Version 2.3 © 1995-2007 I.T. Young, J.J. Gerbrands and L.J. van Vliet

[4] A.-S. Capelle, O. Colot et C. Fernandez-Maloigne, 3D Segmentation of MR Brain Images into White Matter, Gray Matter and Cerebro-spinal Fluid by Means of Evidence Theory, Revue Lecture Notes of Computer Science Series, "Artificial Intelligence in Medicine", pages 112–116, M. Dojat, E. Keravnou, P. Barahona (Eds.), Springer-Verlag, 2003.

[5] Siti Noraini Sulaiman, Noreliani Awang Non, Iza Sazanita Isa, Norhazimi Hamzah ,' Segmentation of Brain MRI Image Based on Clustering Algorithm', Recent Advances in Electrical and Computer Engineering, ISBN: 978-1-61804-228-6

[6] S. Bileschi and L. Wolf. A unified system for object detection, texture recognition, and context analysis based on the standard model feature set. In BMVC, 2005.

[7] Subhagata Chattopadhyay, Dilip Kumar Pratihar, Sanjib Chandra De Sarkar'' A COMPARATIVE STUDY OF FUZZY C-MEANS ALGORITHM AND ENTROPY-BASED FUZZY CLUSTERING ALGORITHMS'', Computing and Informatics, Vol. 30, 2011, 701–720, Page No (701-720) ,2012.

[8] El Dor, A., Clerc, M., Siarry, P.: A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization. Computational Optimization and Applications 53(1), 271–295 (2012)

[9] Ahmed Saad, Ben Smith, Ghassan Hamameh, Torsten Moller, "Simultaneous Segmentation Kinetic Parameter Estimation and Uncertainty Visualization of Dynamic PET Images", Springer on Medical Image Computing and Computer Assisted Intervention' (MICCAI), vol. 4792, no. 2007, pp. 726-733, October 2007.

[10] Pitas, I.: Digital Image Processing Algorithms and Applications. John Wiley & Sons (2000)

[11]E Suganya, Archana ,Dr. S.Sountharrajan, Dr.C.Rajan, METAHEURISTIC OPTIMIZATION TECHNIQUE FOR FEATURE SELECTION TO DETECT THE ALZHEIMER DISEASE FROM MRI
,Journal of advanced research in dynamical and control systems ,ISSN : 1943 – 023X ,2017

[12] Dana Cobzas, Neil Birkbeck, Mark Schmidt,Martin Jagersand,:—3DVariational Brain Tumour Segmentation Using a High Dimensional Feature Set,‖. IEEE 11th International Conference on Computer Vision, ICCV-2007, 1-8, Brazil October 2007.

[13] Bricq.S, Collet.Ch, Armspach.J.P,:‖Unifying Framework for Multimodal brain MRI Segmentation based on Hidden Markov Chains‖, Elsevier on Medical Image Analysis,March,2008. Volume 12, Issue 6, Pages 639-652, December 2008.

[14] A. Kharrat, K. Gasmi, M. Ben Messaoud, N. Benamrane and M. Ab id, "Medical Image Classification Using an Optimal Feature Extraction Algorithm and a Supervised Classifier Technique", International Journal of Software Science and Computational Intelligence (IJSSCI), 3(2), pp. 19-33, 2011. MRI Brain Tumor Classification using Support Vector Machines and Meta-Heuristic Method (PDF Download Available).

[15] Dr.M.Karnan, N.NandhaGopal, "Hybrid Marko Random Field with with Parallel Ant Colony Optimization And Fuzzy C means For MRI Brain Image Segmentation",IEEE 2010,978-1-4244-5967-4/10.

[16] N. Lu, J. Jin, E. Michielssen, and R. L. Magin, Optimization of RF Coil Design Using Genetic HADLEY, FURSE, PARKER: RF COIL DESIGN FOR MRI USING A GENETIC ALGORITHM 285 Algorithm and Simulated Annealing Method, Nice, p. 1002, 1995.

[17] P. SANTAGO and H. D. GAGE. Quantification of MR brain images by mixture density and partial volume modeling. IEEE Trans Med Imag, 12(3):566 – 574, September 1993.

[18] S. SMITH. Fast robust automated brain extraction. Human Brain Mapping, 17:143 – 155, 2002.

[19] J. Bezdek, L. Hall, L. Clarke, "Review of MR image segmentation techniques using pattern recognition", Medical Physics, vol. 20, no. 4, pp. 1033-1048, 1993.

[20] Pham DL, Xu CY, Prince JL: A survey of current methods in medical image segmentation. Ann. Rev. Biomed. Eng. 2 (2000) 315-37 [Technical report version, JHU/ECE 99-01, Johns Hopkins University].

[21] L. P. Clarke, R. P. Velthuizen, M. A. Camacho, J. J. Heine, M. Vaidyanathan, L. O. Hall, R. W. Thatcher, M. L. Silbiger, "MRI segmentation: methods and applications", Magn. Reson. Imag., vol. 13, no. 3, pp. 343-368, 1995.

[22] R. Kikinis, F. A. Jolesz, W. E. Lorensen, H. E. Cline, P. E. Stieg, P. McL. Black, "3-D reconstruction of skull base tumors from MRI data for neurosurgical planning", Proc. Soc. Mag. Reson. Med. Conf., 1991.

[23] Z.-X. Ji, Q.-S. Sun, and D.-S. Xia, "A framework with modified fast FCM for brain MR images segmentation," Pattern Recognit., Vol. 44, no.12, pp. 999–1013, Dec. 2011
[24] Balafar MA, Ramli AR, Saripan MI, Mashohor S, Mahmud R (2010) Improved fast fuzzy C-mean and its application in medical image segmentation. J Circ Syst Comput 19: 203–214
[25] A. Karimian, S. Yazdani, and M. Askari, "Reducing the absorbed dose in analogue radiography of infant chest images by improving the image quality, using image processing techniques," Radiat. Prot Dosim., Vol. 147, no. 1–2, pp. 176–9, Jul. 2011
[26] M. Sezgin et al., "Survey over image thresholding techniques and quantitative performance evaluation", Journal of Electronic imaging, vol. 13, no. 1, pp. 146-168, 2004.
[27] Akay, B.: A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Appl. Soft Comput. 13(6), 3066–3091 (2013)
[28] Vishwakarma, B., Yerpude, A.: A meta-heuristic approach for image segmentation using firefly algorithm. Int. J. Comput. Trends Technol. (IJCTT) 11(2), 69–73 (2014)
[29] Yin, P.Y.: Multilevel minimum cross entropy threshold selection based on particle swarm optimization. Appl. Math. Comput. 184(2), 503–513 (2007)
[30] Bandyopadhyay S, Saha S, Maulik U, Deb K ,A simulated annealing based multi-objective optimization algorithm: AMOSA. IEEE Trans Evol Comput 12(3):269–283,2008
[31] Weibei Dou, Su Ruan, Daniel Bloyet, and Jean-Marc Constans. A framework of fuzzy information fusion for the segmentation of brain tumor tissues on mr images. Image and Vision Computing, 25(2), 2007.
[32] Handl, J., and Knowles, J.," On semi-supervised clustering via multiobjective optimization", 8th annual conference on Genetic and evolutionary computation (GECCO'2006), ACM Press (2006a), 1465–1472
[33]Puzicha, J., Hofmann, T. and Buhmann, J. M.," Histogram Clustering for Unsupervised Image Segmentation". Computer Vision and Pattern Recognition, Vol. 2. IEEE press, (2000), 602-608
[34] ORTIZ A, GORRIZ J M, RAMIREZ J, et al. Improving MR brain image segmentation using self-organising maps and entropy-gradient clustering . Information Sciences, 2014, 262(3):117-136
[35] riparna Saha , Sanghamitra Bandyopadhyay, MR Brain Image Segmentation Using A Multi-seed Based Automatic Clustering Technique, Fundamenta Informaticae, v.97 n.1-2, p.199-214, January 2009
[36] Luque JM, Santana-Quintero LV, Hernández-Díaz AG, Coello CAC, Caballero R (2009) g-dominance: Reference point based dominance for multiobjective metaheuristics. Eur J Oper Res 197(2):685-692.
[37] C.R. Raquel and P.C. Naval, "An effective use of crowding distance in multiobjective particle swarm optimization", in Proceedings of the 2005 Genetic and Evolutionary Computation Conference, 2005, pp. 257-264.
[38] Bong, C.W., Rajeswari, M.: Multi-objective nature-inspired clustering and classification techniques for image segmentation. Appl. Soft Comput. 11(4), 3271–3282 (2011).
[39] E. D'Agostino, F. Maes, D. Vandermeulen, and P. Suetens, "A viscous fluid model for multimodal non-rigid image registration using mutual information," in Proceedings of the Medical Image Computing and Computer Assisted Intervention (MICCAI '02), pp. 541–548, 2002.
[40] M. Battaglini, S. M. Smith, S. Brogi, and N. de Stefano, "Enhanced brain extraction improves the accuracy of brain atrophy estimation," NeuroImage, vol. 40, no. 2, pp. 583–589, 2008.
[41] C. Sagiv, N. A. Sochen, and Y. Y. Zeevi, "Integrated active contours for texture segmentation," IEEE Transactions on Image Processing, vol. 15, no. 6, pp. 1633–1646, 2006.
[42]Saranu Kavya Pooja, Koganti Koundinya, J Hemamalini, "Efficiency Measurement of Detecting Object From Video", International Journal of Applied Engineering Research, ISSN 0973-4562, Volume 10, Number 5 pp. 12165-12175,2015.