



ERASMUS MUNDUS

MATHMODS PROGRAMME

**MULTIOBJECTIVE
OPTIMIZATION**

Author:
Chainarong KESAMOON

Supervisors:
Pierre KORNPORST
Emilien TLAPALE

July 23, 2009

1 Introduction

Multiobjective optimization can be found in several fields; in product and process design, finance, aircraft design, the oil and gas industry, automobile design which there are many conflicting criteria. For example in product and process design, we would like to maximize the profit while we have to keep customers royalty. We may minimize the product cost to increase the profit but it also affect the product's performance which may course the customers disappoint. Now we can see that those criteria are conflicting. While we are increasing quality of one criterion, it may course the others be unacceptable. So we would like to find the optimal solution to compromise those criteria. Multiobjective optimization can give answers for these problems. This makes it very useful in various fields.

For neuroscience, in the NeuroMathComp project team at INRIA Sophia Antipolis Mediteranee, we are researching to discover the principles that govern the functioning of neurons and assemblies thereof. There are several regions in brain working in different objects but they work together to control the body. The situations in neuroscience seem similar to the other fields mentioned above. There are many criteria to be compromised. While researchers in other fields have studied and been making use of multiobjective optimization so far, the application of multiobjective optimization in neuroscience is not well-studied yet. Hence my training work in the NeuroMathComp project team is to study multiobjective optimization and try to apply to some problems in neuroscience.

2 Multiobjective Optimization

This section is mainly about the idea of multiobjective optimization. We will study the meaning of multiobjective optimization, basic definitions and theorems in multiobjective optimization in finite dimensional spaces. Then we will study some methods used in multiobjective optimization. Some definitions and theorems are mentioned in general view to project on the nature of multiobjective optimization, the precise definitions and theorems can be found in [1].

Indeed, we are going to solve the multiobjective optimization of the form

$$\text{minimize } \{f_1(x), \dots, f_k(x)\}$$

$$\text{subject to } x \in S = \{x \in \mathbb{R}^n | g(x) = (g_1(x), \dots, g_m(x))^T \leq 0\}$$

We call $S \subset \mathbb{R}^n$ a feasible region and $Z = f(S) \subset \mathbb{R}^k$ a feasible objective region.

2.1 Multiobjective Optimization

Multiobjective optimization problem consists of vector of objective functions $f(x) = (f_1(x), \dots, f_k(x))^T$ where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, k \geq 2$. $x = (x_1, \dots, x_n)^T \in S$ is called decision vector and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m$ are called constraint functions forming S . Each multiobjective optimization problem has properties depending on objective functions and constraint functions. If the objective functions and constraint functions are linear, then the multiobjective optimization problem is called linear. If at least one of the objective or the constraint functions is nonlinear, the problem is called a nonlinear multiobjective optimization problem. Moreover, multiobjective optimization problem can be characterized by the convexity and differentiability of the objective functions and the constraint functions. Multiobjective optimization problem is convex (quasiconvex, pseudoconvex) if all the objective functions and the feasible region are convex (quasiconvex, pseudoconvex). The pseudoconvexity is defined only where the considered function is differentiable. We also handle nondifferentiable multiobjective optimization problems. In our case a nondifferentiable function is a function which is locally Lipschitzian.

We will concentrate on cases where non-linear functions are included. If we always deal with strong conditioning problems such as strictly convex differentiable problem, it is easy to solve. But in practice, we cannot choose the problem to solve. That is why we need to know weaker conditions such as pseudoconvexity, quasiconvexity and nondifferentiability. The precise definitions of these conditions can be seen in [1].

2.2 Pareto Optimality

In a single objective optimization problem, there is one objective function. When we solve the problem we obtain only one objective value but possibly many solutions in decision space. Therefore we are interested in the decision space. But in the case of multiobjective optimization problem, when the objective functions are conflict, we will obtain several feasible objective values which are vectors in $Z \subset \mathbb{R}^k$. In this case we cannot choose the best solution directly since Z is only partially ordered. Normally, we are more interested in the objective space because of the lower dimension.

The objectives in multiobjective optimization problem cannot be ordered, so we need a tool to extract the proper objectives from the whole feasible objective region. The Pareto optimality is that tool. The Pareto optimal set consists of objective vectors which none of the components can be improved without deterioration to at least one of other components.

We define the locally Pareto optimal objective which is Pareto optimal in some neighbourhood. This is useful in practice.

2.3 Decision Maker

There are many (possible infinite) solutions in a Pareto optimal set. Those solutions are equivalent in mathematical meaning, we can select any one of them to be the final solution. But in real life, one who is responsible to make the final decision so called "decision maker" is not a mathematician. He/she may have hidden reasons to select his/her preference. The decision maker has right to decide whatever he/she thinks it is good enough. The mathematician gives him/her only the information to decide. We usually assume that decision maker is only interested in Pareto optimal points and the rest can be excluded. This means that the decision maker will move from one solution to another in the Pareto optimal set.

2.4 Ranges of the Pareto Optimal Set

Since we are dealing with the Pareto optimal set, we should know how big it is. We can estimate the range of the Pareto optimal set is between the ideal objective vector and the nadir objective vector. The ideal objective vector is an objective vector minimizing each of the objective functions. The nadir objective vector is the upper bound of the Pareto optimal set which can be estimated from a payoff table.

2.5 Value Function

We assume that the decision maker makes decisions based on a function of some kind. This function is called a value function. This function $U : \mathbb{R}^k \rightarrow \mathbb{R}$ represents the preference of the decision maker among the objective vectors. The value function is dependent on the decision maker only and it is difficult to specify the value function explicitly. Generally, it is assumed to be strongly decreasing and it is more important in the development of solution method than in practice.

2.6 Efficiency and Weak Pareto Optimality

Other than the Pareto optimality, we define other optimalities which are more general. Those are weak Pareto optimality and efficiency. The weak Pareto optimal set consists of objective vectors such that cannot be improved all the components in the same time. It is possible to improve some components. While the efficiency consists of all the vectors in feasible objectives set which there does not exist another vector dominating it with respect to a cone D . The ordering cone D is used to induce ordering in Z . The Pareto optimality is a special case of efficiency where $D = \mathbb{R}_+^k$. Hence we can easily see that the weak Pareto optimality and the efficiency are more general than the Pareto optimality.

2.7 Trade-Offs and Marginal Rate of Substitution

We can move from one solution to another solution in multiobjective optimization. We can compare changes in two objective functions f_i, f_j using the trade-off. Trade-off is the ratio of changes between those two functions denoted by $\Lambda_{i,j} = \Lambda_{i,j}(x^1, x^2) = \frac{f_i(x^1) - f_i(x^2)}{f_j(x^1) - f_j(x^2)}$. There are two types of trade-offs which are total trade-off and partial trade-off. The partial trade-off happens when there is only one pair of indexes i, j such that the function values at x^1 and x^2 are different. Then we can define the trade-off rate as $\lambda_{i,j} = \lambda_{i,j}(x^*, d^*) = \lim_{\alpha \rightarrow 0^+} \lambda_{i,j}(x^* + \alpha d^*, x^*)$.

The marginal rate of substitution is defined on an indifference curve. $m_{i,j}(x^*)$ represents the preference of the decision maker at a decision vector $x^* \in S$. It is the amount of decrement in the value of the objective function f_i that compensates the decision maker for the one-unit increment in the value of the objective function f_j , while the values of all the other objectives remain unaltered. It is actually the negative slope of the tangent to the indifference curve at a certain point. Hence the final solution of a

multiobjective optimization problem is a Pareto optimal solution where the indifference curve is tangent to the Pareto optimal set.

If the Pareto optimal set is smooth, the trade-off rate of a point is in fact the slope of the tangent of the Pareto optimal set at that point. We can determine when a Pareto optimal solution is a final solution. That is the tangent of the indifference curve and the Pareto optimal set coincide at it. That means $-m_{i,j} = \lambda_{i,j}$ for all $i, j = 1, \dots, k, i \neq j$. In this case we say that f_i is a reference function since the trade-off and the marginal rate of substitution are generated with respect to it.

2.8 Proper Pareto Optimality

Proper Pareto Optimality is another type of Pareto optimality. The idea is that unbounded trade-offs between objectives are not allowed. ϵ -Proper Pareto optimal is another type of Pareto optimal. It is stronger than Pareto optimal since the larger convex cone \mathbb{R}_ϵ^k is used instead of \mathbb{R}_+^n . The trade-offs are bounded by ϵ and $\frac{1}{\epsilon}$.

2.9 Methods

In multiobjective optimization, generating Pareto optimal solutions play a very important role. But whenever we have discovered the Pareto optimal solutions, our task has not finished yet. Since the final solution is decided by decision maker. So the representation of the obtained Pareto optimal solutions to the decision maker is also important because he/she is not mathematician. We will classify methods to obtain the final solution into four classes according to the interaction between the the analyst and the decision maker.

1. **no preference methods** where the opinions of the decision maker are not taken into consideration
2. **a posteriori methods** where the decision maker select the most preferred among the alternative after the Pareto optimal set has been generated and presented to the decision maker
3. **a priori methods** where the decision maker must specify his/her preferences, hopes and opinions before the solution process.
4. **interactive methods** where the final solution comes from co-operation , interaction of the decision maker during the solution process. This is the most developed of the four classes of methods presented here.

Normally, the ideas of solving multiobjective optimization are based on scalarization which tries to convert the problem into a single or a family of single objective optimization problems with a real-valued objective function, termed the scalarizing function, depending possibly on some parameters.

I will present here three methods in multiobjective optimization problems which are posteriori methods since the others we need to know the value functions which is difficult to define or we need to select our preferences during the procedure which may cost the decision maker's time. The object of the following methods is to generate the Pareto optimal solutions to be decided later by the decision maker.

2.9.1 Weighting Method

The idea is to associate each objective function with a weighting coefficient and minimize the weighted sum of the objectives. In this way, the multiple objective functions are transformed into a single objective function. We also suppose that the weighting coefficients w_i are real numbers such that $w_i \geq 0$ for all $i = 1, \dots, k$. It is usually supposed that the weights are normalized, that is $\sum_{i=1}^k w_i = 1$. To be more exact, the multiobjective optimization problem is modified into the following problem, to be called a weighting problem.

$$\begin{aligned} & \text{minimize } \sum_{i=1}^k w_i f_i(x) & (1) \\ & \text{subject to } x \in S \end{aligned}$$

where $w_i \geq 0$ for all $i = 1, \dots, k$ and $\sum_{i=1}^k w_i = 1$.

Then the solution of weighting problem (1) is weakly Pareto optimal and if $w_i > 0$ for all $i = 1, \dots, k$ then the solution of (1) is Pareto optimal.

2.9.2 ϵ -Constraint Method

One of the objective functions is selected to be optimized and all the other objective functions are converted into constraints by setting an upper bound to each of them. The problem will be called an ϵ -constraint problem.

$$\begin{aligned} & \text{minimize } f_l(x) & (2) \\ & \text{subject to } f_j(x) \leq \epsilon_j \text{ for all } j = 1, \dots, k, j \neq l, \\ & x \in S \end{aligned}$$

Then the solution of the ϵ -Constraint problem is weakly Pareto optimal. If a point $x^* \in S$ is a unique solution of ϵ -constraint problem (2) for some l with

$\epsilon_j = f_j(x^*)$ for $j = 1, \dots, k, j \neq l$, then x^* is Pareto optimal. The following theorem is very useful in this method.

Theorem 2.1 (Karus-Kuhn-Tucker necessary optimality condition)

Let the objective and the constraint functions be continuously differentiable at x^ which is regular point of the constraint problem of the ϵ -constraint problem. A necessary condition for x^* to be a solution of the ϵ -constraint problem is that there exist vector $0 \leq \lambda \in \mathbb{R}^{k-1}$ and $0 \leq \mu \in \mathbb{R}$ such that*

1. $\nabla f_l(x^*) + \sum_{j=1, j \neq l}^n \lambda_j \nabla(f_j(x^*) - \epsilon_j) + \sum_{i=1}^m \mu \nabla g(x^*) = 0$
2. $\lambda_j(f_j(x^*) - \epsilon_j) = 0$ for all $j \neq l$, $\mu_i g_i(x^*) = 0$ for all $i = 1, \dots, m$

2.9.3 Hybrid method

The idea is to combine the weighting method and the ϵ -constraint method. The hybrid problem to be solved is

$$\text{minimize } \sum_{i=1}^k w_i f_i(x) \tag{3}$$

$$\begin{aligned} \text{subject to } & f_j(x) \leq \epsilon_j \text{ for all } j = 1, \dots, k, \\ & x \in S, \end{aligned}$$

where $w_i > 0$ for all $i = 1, \dots, k$.

Then the solution of hybrid problem (3) is Pareto optimal for any upper bound vector $\epsilon \in \mathbb{R}^k$. On the other hand, if x^* is Pareto optimal, then it is a solution of problem (3) for $\epsilon = f(x^*)$.

3 Case study

In our work, we would like to apply the methods in multiobjective optimization into neuroscience problems. For this case study, we chose the Ambrosio-Tortorelli segmentation functional to be minimized by ϵ -constraint method

3.1 Image segmentation

In computer vision, segmentation refers to the process of partitioning a digital image into multiple segments (sets of pixels) (also known as superpixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s).

3.2 The Mumford-Shah functional

The Mumford-Shah functional for image segmentation was introduced by Mumford and Shah in 1989. It is a famous approach for image segmentation but there is a difficulty since we have to calculate gradient with respect to curve segmenting set.

$$E_{MS}(u, B) = \alpha \int \int_{R \setminus B} \|\nabla u\|^2 dx dy + \beta \int \int_R (u - g)^2 dx dy + |B|$$

where

R is connected, bounded, open set of \mathbb{R}^2

B is a curve segmenting R

$|B|$ is the length of B

g is the feature intensity

u is the smoothed image $\subset \mathbb{R}^2 \setminus B$

α, β are the weights.

The minimizer u of this functional is a smooth approximation of g in each sub-domain segmented by B .

3.3 The Ambrosio-Tortorelli functional

Because of the difficulty of Mumford-Shah functional, Ambrosio and Tortorelli replaced B by a continuous variable v and obtained the Ambrosio-Tortorelli edge-strength function as

$$E_{AT}(u, v) = \int \int_R \left\{ \alpha(1-v)^2 \|\nabla u\|^2 + \beta(u-g)^2 + \frac{\rho}{2} \|\nabla v\|^2 + \frac{v^2}{2\rho} \right\} dx dy$$

where

R is connected, bounded, open set of \mathbb{R}^2

g is the feature intensity

u is the smoothed image

v is a continuous variable

α, β, ρ are the weights.

We are going to solve this variational optimization problem by ϵ -constraint method. First of all, let us review the standard approach which based of the gradient descent equations;

$$\frac{\partial u}{\partial t} = -2\nabla v \cdot \nabla u + (1-v)\nabla^2 u - \frac{\beta(u-g)}{\alpha(1-v)} \quad (4)$$

$$\frac{\partial v}{\partial t} = \nabla^2 v - \frac{v}{\rho^2} + \frac{2\alpha}{\rho}(1-v)\|\nabla u\|^2 \quad (5)$$

$$\frac{\partial u}{\partial n} \Big|_{\partial R} = 0, \quad \frac{\partial v}{\partial n} \Big|_{\partial R} = 0 \quad (6)$$

After we obtain the gradient descent equation, we implement into computer using finite difference with diffusion equation by center scheme [2].

3.4 Alternative Approach

To solve the optimization problem of the Ambrosio-Tortorelli functional by ϵ -constraint method we split the Ambrosio-Tortorelli functional into two functionals.

$$E_1(u, v) = \int \int_R \alpha(1-v)^2 \|\nabla u\|^2 dx dy \quad (7)$$

$$E_2(u, v) = \int \int_R \beta(u-g)^2 + \frac{\rho}{2} \|\nabla v\|^2 + \frac{v^2}{2\rho} dx dy \quad (8)$$

$$(9)$$

Then solve

$$\begin{aligned} & \text{minimize } \{E_1(u, v), E_2(u, v)\} \\ & \text{subject to } (u, v) \in (W^{1,2}(R))^2 \end{aligned}$$

The ϵ -constraint problem corresponding to this multiobjective optimization problem is

$$\begin{aligned} & \text{minimize } E_1(u, v) \\ & \text{subject to } E_2(u, v) \leq \epsilon \end{aligned}$$

This problem is variational minimization problem with inequality constraint which is different from what we have learned before. Since we have studied only multiobjective optimization in finite dimensional spaces, but this problem lies in infinite dimensional spaces. To overcome this difficulty, we have to spend some more time on multiobjective optimization in infinite dimensional spaces. Nevertheless, I try to solve this problem roughly with some basic procedures. Therefore, please note that this step has not been verified yet. From the ϵ -constraint problem, the Lagrange function is

$$L(u, v, \lambda) = E_1(u, v) + \lambda(E_2(u, v) - \epsilon)$$

Keep u and v fixed respectively, the Euler equations are

$$\text{div}(E_1(u, v) + \lambda E_2(u, v))_{\nabla u} = (E_1(u, v) + \lambda E_2(u, v))_u \quad (10)$$

$$\text{div}(E_1(u, v) + \lambda E_2(u, v))_{\nabla v} = (E_1(u, v) + \lambda E_2(u, v))_v \quad (11)$$

We now obtain the gradient descent equations correspond (10) and (11)

$$\frac{\partial u}{\partial t} = \text{div}(\alpha(1-v)^2 \nabla u) - \lambda \beta(u-g) \quad (12)$$

$$\frac{\partial v}{\partial t} = \nabla^2 v + \frac{2\alpha}{\lambda \rho} (1-v) \|\nabla u\|^2 - \frac{v}{\rho^2} \quad (13)$$

$$\frac{\partial u}{\partial n} \Big|_{\partial R} = 0, \quad \frac{\partial v}{\partial n} \Big|_{\partial R} = 0 \quad (14)$$

Then impose the condition on λ from the Karush-Kuhn-Tucker condition

$$\lambda(E_2(u, v) - \epsilon) = 0 \text{ and } \lambda \geq 0.$$

That is

$$\int \int_R \lambda \beta (u-g)^2 + \frac{\lambda \rho}{2} \|\nabla v\|^2 + \frac{\lambda v^2}{2\rho} dx dy - \lambda \epsilon = 0 \text{ and } \lambda \geq 0. \quad (15)$$

To find evolution of λ corresponds the evolution of u and v . Multiply (12) by $(u - g)$ and integrate over R . If steady state has been reached, $\frac{\partial u}{\partial t} = 0$. So we have

$$0 = \int \int_R (u - g) \operatorname{div}(\alpha(1 - v)^2 \nabla u) - \lambda \beta (u - g)^2 dx dy.$$

That is

$$\int \int_R \lambda \beta (u - g)^2 dx dy = \int \int_R (u - g) \operatorname{div}(\alpha(1 - v)^2 \nabla u) dx dy. \quad (16)$$

Replace (16) in (15), we finally obtain

$$\lambda = \frac{\int \int_R (u - g) \operatorname{div}(\alpha(1 - v)^2 \nabla u) dx dy}{\epsilon - \int \int_{R^2} \frac{\rho}{2} \|\nabla v\|^2 + \frac{v^2}{2\rho} dx dy} \quad (17)$$

From now on, we are able to implement (12),(13) and (17) into computer by the association of finite difference(see [2]). By varying ϵ and keep other parameters fixed, we obtain the results.

4 Results

4.1 The Experiments

We do experiments on two images, one is an image with simple detail and another is more complicating. The parameters are fixed except for ϵ as the following.

α	β	ρ	ϵ	Time step for u	Time step for v
0.01	0.001	1.0	0.05	0.1	0.005
0.01	0.001	1.0	0.10	0.1	0.005
0.01	0.001	1.0	0.15	0.1	0.005
0.01	0.001	1.0	0.20	0.1	0.005

The numbers of iterations for image (1) and image (2) are 10 and 300 respectively.

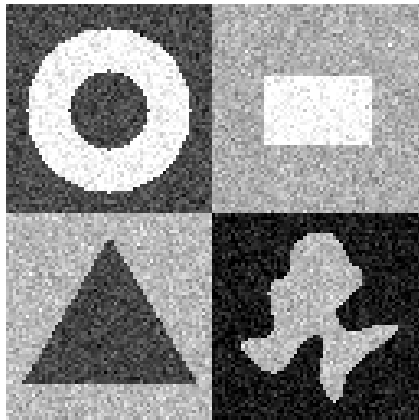


Figure 1: Sample 1



Figure 2: Sample 2

4.2 Results

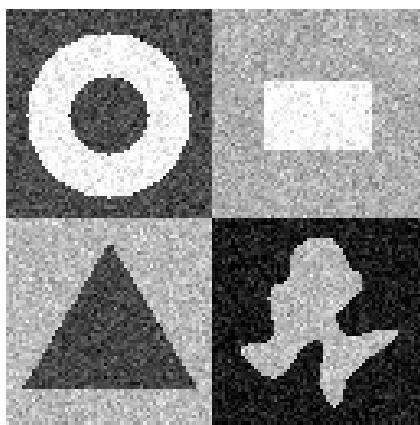


Figure 3: Original image

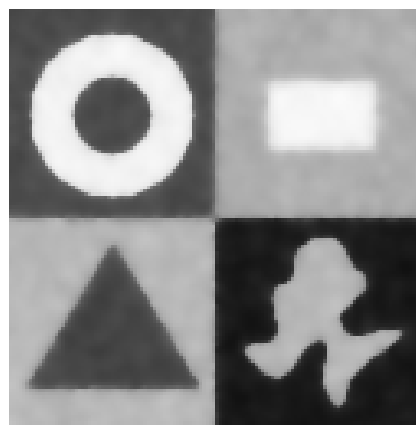


Figure 4: Standard approach

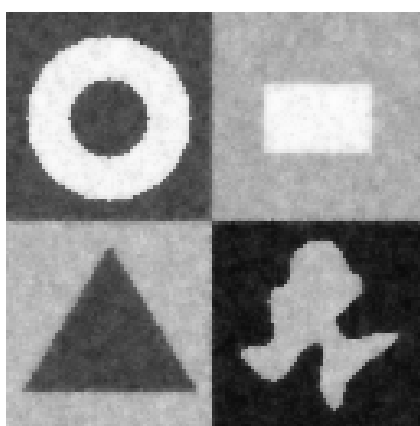


Figure 5: Alternative $\epsilon = 0.05$

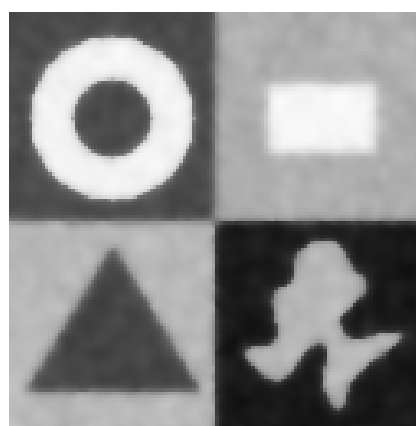


Figure 6: Alternative $\epsilon = 0.10$

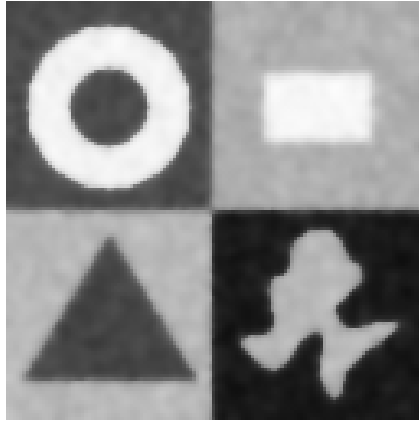


Figure 7: Alternative $\epsilon = 0.15$

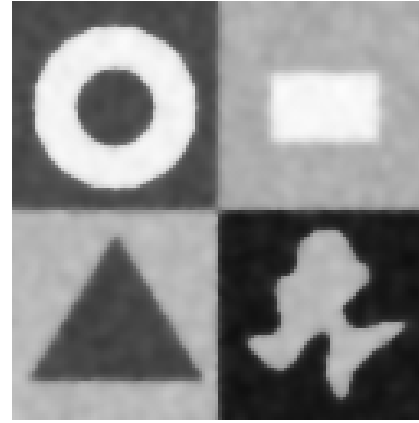


Figure 8: Alternative $\epsilon = 0.20$



Figure 9: Original image

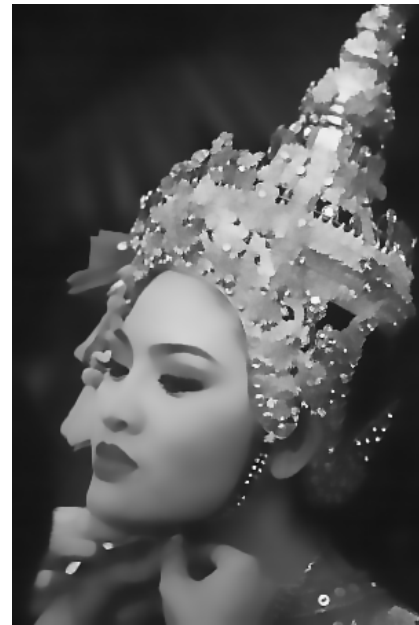


Figure 10: Standard approach



Figure 11: Alternative $\epsilon = 0.05$



Figure 12: Alternative $\epsilon = 0.10$

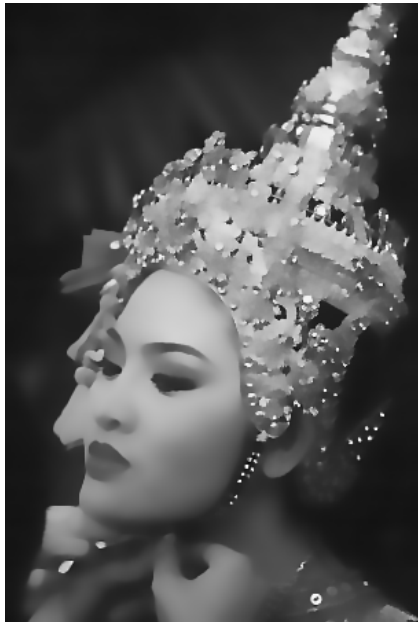


Figure 13: Alternative $\epsilon = 0.15$



Figure 14: Alternative $\epsilon = 0.20$

Then try to compare the results from the standard approach and the alternative approach. Since the alternative approach differs from the standard approach by varying λ (see (15) and (16)) which is a coefficient of $E_2(u, v)$. We may compare the results from two approaches by putting average value of λ in alternative approach as coefficient of $E_2(u, v)$ terms in standard approach.

For the alternative approach, we calculate the average value of λ for each ϵ

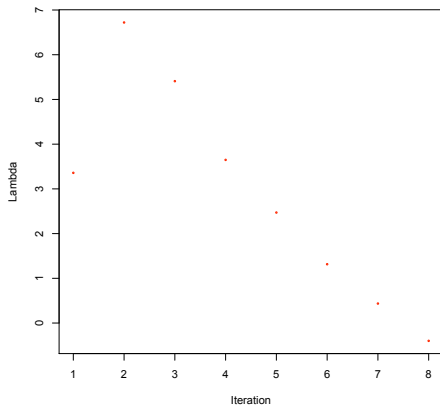


Figure 15: Evolution of λ for image (1) where $\epsilon = 0.10$

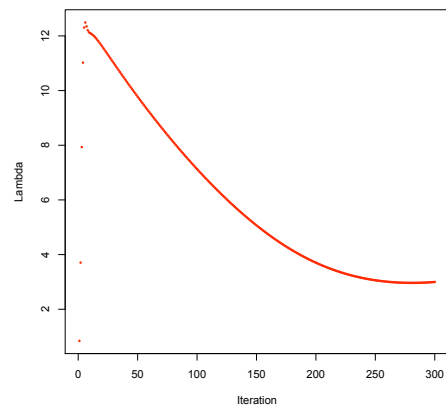


Figure 16: Evolution of λ for image (2) where $\epsilon = 0.10$

and obtain the following.

ϵ	Average λ	
	Image (1)	Image (2)
0.05	19.1875	13.7491
0.10	2.3358	5.9559
0.15	0.0358	3.0393
0.20	0.0232	1.5955

We then apply those average values of λ into standard approach.

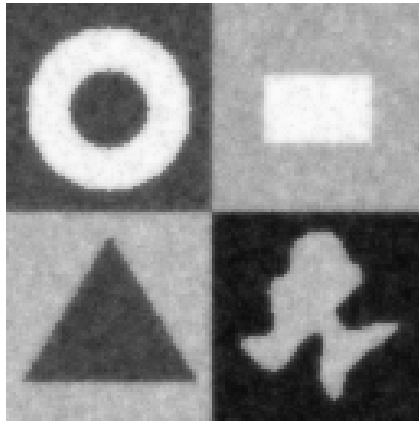


Figure 17: Standard $\lambda = 19.1875$

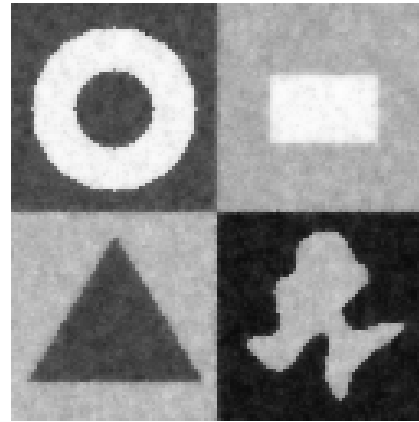


Figure 18: Alternative $\epsilon = 0.20$

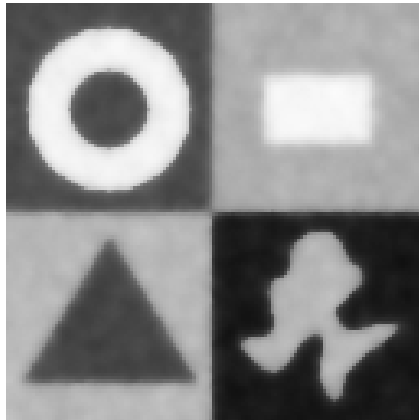


Figure 19: Standard $\lambda = 2.3358$

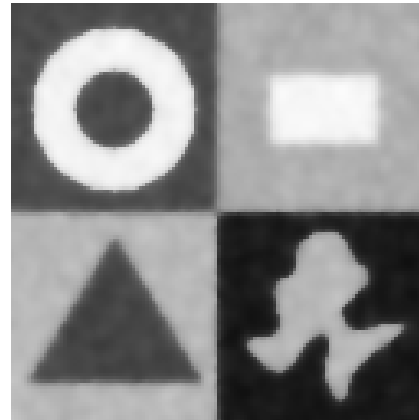


Figure 20: Alternative $\epsilon = 0.20$



Figure 21: Standard $\lambda = 0.0358$

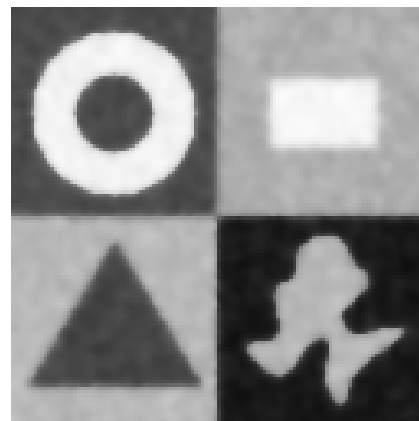


Figure 22: Alternative $\epsilon = 0.20$



Figure 23: Standard $\lambda = 0.0232$

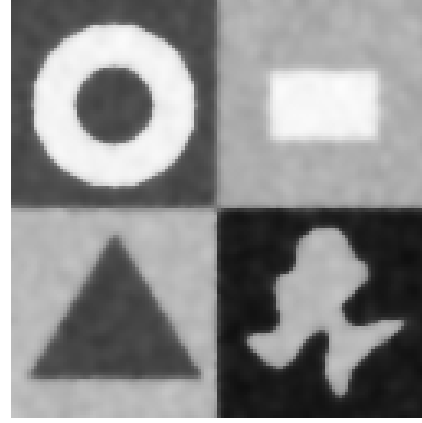


Figure 24: Alternative $\epsilon = 0.20$



Figure 25: Standard $\lambda = 19.1875$



Figure 26: Alternative $\epsilon = 0.20$



Figure 27: Standard $\lambda = 2.3358$



Figure 28: Alternative $\epsilon = 0.20$



Figure 29: Standard $\lambda = 0.0358$

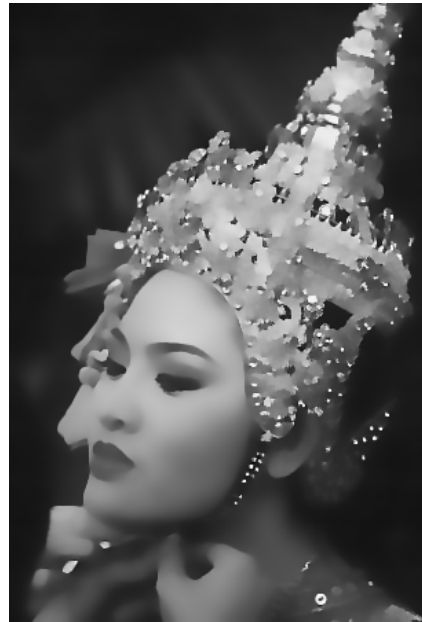


Figure 30: Alternative $\epsilon = 0.20$



Figure 31: Standard $\lambda = 0.0232$



Figure 32: Alternative $\epsilon = 0.20$

We also compare the running time between both approaches.

Approach	Running time (sec.)	
	Image (1)	Image (2)
Standard	0.11	25.78
Alternative $\epsilon = 0.05$	0.14	33.10
Alternative $\epsilon = 0.10$	0.11	33.07
Alternative $\epsilon = 0.15$	0.09	33.19
Alternative $\epsilon = 0.20$	0.09	31.05

5 Discussion and Future Work

5.1 Discussion on the results

- The results from both approaches are not significantly different. Both approaches converge to some acceptable solutions with very little differences on running time.
- We don't have a measurement to decide which solution is better than the others. But we know that the Ambrosio-Tortorelli functional is not strictly convex, the solutions we obtain from the standard approach are local solutions. That means by the standard approach, we obtain one solution which may not be the best solution. But by the association of multiobjective optimization, the solutions from the ϵ -constraint method are weak Pareto optimal. And we obtain various solutions depended on the choices of ϵ . This means that we have more candidates for the final solution even now we don't have a method to select the final solution.
- We have some reformations from continuous domains to vector domains which have not been checked the availability well. The results are calculated approximately.

5.2 Future work

- Try to apply other methods and problems.
- We have to study multiobjective optimization in infinite dimensional domains. Where most neuroscience problems locate in.
- The evolutionary algorithm for multiobjective optimization is also promising to be apply in this field [6],[7].

References

- [1] K. M. Miettinen, *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Dordrecht, 1999.
- [2] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing : Partial Differential Equations and the Calculus of Variations*. Springer-Verlag, New York, 2002.
- [3] I. B. Russak, *Calculus of Variations : MA 4311 Lecture Notes*. California, 1996.
- [4] J. Shah, *A common Framework for Curve Evolution, Segmentation and Anisotropic Diffusion*. IEEE Conf. on Computer Vision and Pattern Recognition, 1996.
- [5] L. I. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*. Physica D 60, pp.259-268, North-Holland, 1992.
- [6] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization : Methods and Applications*. Swiss Federal Institute of Technology Zurich, 1999.
- [7] B. Bhanu, *Adaptive Image Segmentation Using a Genetic Algorithm*. IEEE Transaction on Systems, Man, and Cybernetics, Vol.25, No.12, pp.1543-1567, December, 1995.