Convolutional kernel and neural network for Pareto front

Liu Ruilin, Zhang Tao

School of Information and Mathematics, Yangtze University

**Abstract**

In this paper,

**1 Introduction**

In real world, most of the optimization problems are multi-objective optimization problems (MOOPs) actually. Multi-objective optimization involves optimizing a number of objectives simultaneously. The problem becomes challenging when the objectives are of conflict to each other. In solving such problems, with or without the presence of constraints, these problems give rise to a set of trade-off optimal solutions, popularly known as Pareto-optimal solutions[1]. There are mainly two approaches to obtain a finite set of Pareto-optimal solutions, scalarization and the methods based on Pareto optimal. Now researchers mainly involve in the methods based on Pareto optimal because of the ability that can depict the solution of MOPs more generally. In addition, recently some researchers proposed some algorithms based on machine learning, these methods are not yet mature. Current state-of-the-art approaches for MOPs are represented by the evolutionary multi-objective optimization algorithms (EMOAs)[2].In these algorithms, the Pareto dominance of arbitrary two candidate solutions is determined by computing and comparing their objective vectors[3].

However, it is a time consuming procedure that gaining the Pareto dominance relationship of the objective vectors. Actually, it is a sorting procedure of vectors. In order to overcome the curse of computation cost in this procedure, some researchers proposed a method of predicting Pareto dominance using machine learning, a supervised classifier[3]. It is a exploratory work, but can’t change the curse because of that the classifier depends on specific problem and even depends on some specific data. Actually, this method using the statistical learning or machine learning to solve the vectors sorting problem.

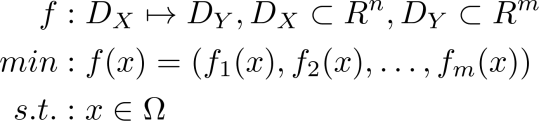
Inspired by the image processing method, convolutional neural network, we transform the objective vectors images into binary image. According to some simply mathematical conclusions, the procedure to sort the vectors is equal to recognize the specific contour of the binary image of vectors. There are many operators to obtain the contour in image processing such as Robert, Sobel and Prewitt ect.. However, all of them can not recognize this specific contour. So we proposed a class of operators and use it to build convolutional neural networks to obtain the non-dominant vectors. This method provides an efficient for relieving the curse of computation cost in solving complicated MOPs and even for vectors sorting when it needed.

**2 Problem setting**

In this part, some basic definition and conclusion will be listed in the mathematical form briefly. Base on these we will introduce the algorithm and proof it next.

2.1 Basic definition

**Definition 1** (MOP) An MOP is formulated as below:



Where  is the decision variable space of n dimensions, and  is the objective function space of m dimensions.  is a decision variable of n dimensional vector.  is the feasible set and typically specified as set of constraints on the decision variables. Vector value function C:/Users/71903/AppData/Local/Temp/qt_temp.x10796qt_temp is the objective function.

**Definition 2** (Global Minimum) For  the valueis called a global minimum if and only if

C:/Users/71903/AppData/Local/Temp/qt_temp.K10796qt_temp.

Then  is called global minimum solution.

**Definition 3** (Dominate) It is called  dominate (or  dominated by ) and denoted as , if

C:/Users/71903/AppData/Local/Temp/qt_temp.p10796qt_temp .

**Definition 4** (Pareto optimal solution) A specified  is termed as a Pareto optimal solution if there is no exist an element in the set  to dominate the .

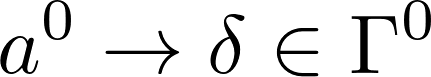
**Definition 5** (PS and PF) A set consists of all of the Pareto optimal solution is called Pareto set (PS) of the MOP. A set consists of images in the objective function mapping from PS by objective function  is called Pareto front (PF).

**Definition 6** (binary image of a vector and a set)  is a n dimensional vector. is n dimensional image of point (re: delete), and assume that the ith axis is . Assume that the digital image resolution in axis  is . Denote set . Then n-D tensor  is the binary image of the vector ,if

is a pixel of image , and  is the index in ith dimension. Such that

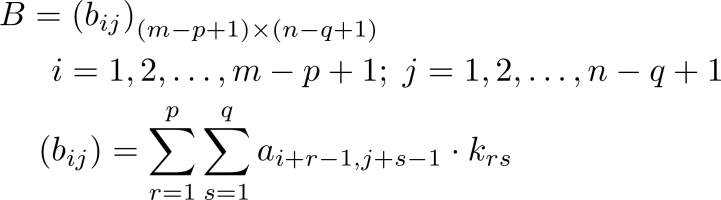
 if only if point  in the position of pixel  (re: ), and for others.

And pixel  in binary image  is the corresponding pixel of vector , it is denoted as:

.

If a set ,  is the binary image of the vector , then is the binary image of set A.

**Definition 7** (1 stride 2 dimensional convolution operator) Assume that , are 2 dimensional matrices.



Denote as:

C:/Users/71903/AppData/Local/Temp/qt_temp.z16380qt_temp

Matrix  is the result of convolution of A. Matrix  is termed convolutional kernel.

A conclusions we used that is well known in this field is that the PF is a part of the contour of the image region in objective function space which the ends are the minimum points of  and (for two dimensional objective function space for example).

2.2 Problem researched

In this paper, we concern on the MOPs with 2 dimensional objective functions. The problem we researched is that how to obtain the non-dominant vectors form some specific vectors. The input of the problem is a binary image of a set of vectors, the output is the image of PF or non-dominant vectors set.

**3 Algorithm**

In this chapter, we will proposed two algorithm base on the same principle. The special convolution kernel will be presented and the method to use it will be explained. And we proof the all results of that.

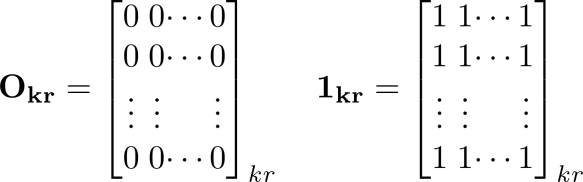
3.1 The input of algorithm

Just like mentioned above, the input of the problem is a binary image of a set of vectors. According to Definition 7, it can be found that the dimension of the result will be less than the original matrix. In order to get a result as same dimension as the input, the actually input of convolution is the result after padding operator of the original input matrix. The padding operator refers that adding zeros around the matrix such that the convolutional result has a same dimension as the original. The input of our algorithm is the matrix padded, it will not be noticed especially later.

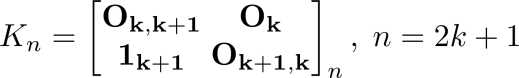
3.2 Procedure of algorithm

First, we give the PF recognition convolution kernel.

**Definition 8** (PF recognition convolution kernel) Two special  matrices are denoted as

.

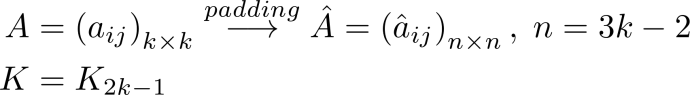
Then matrix



is a n dimensional PF recognition convolution kernel (n-d kernel).

Using this n-d kernel we construct the algorithm, the procedure of algorithm I is shown as:

1. Initialization and input:

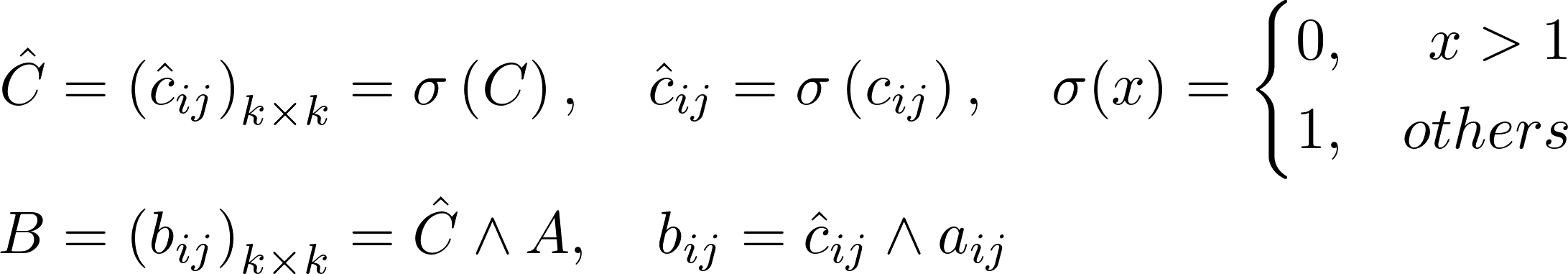


Where matrix  is the given binary image, matrix  is the padding result of matrix , matrix  is a PF recognition convolution kernel.

1. Convolution:

C:/Users/71903/AppData/Local/Temp/qt_temp.W16380qt_temp

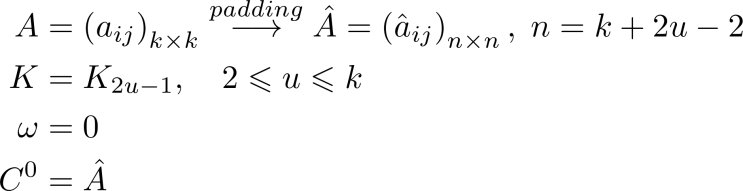
1. Activation:



Matrix  is the output of the algorithm.

This algorithm just uses one step of convolution operator, it it clearly that the convolution kernel  is related to the size of input and increases with the increasing of input matrix size. So we propose another algorithm also base on the same principle. Algorithm II is shown as:

1. Initialization and input:



The symbols is same as the algorithm I, just the dimension of the convolution kernel  is changed. Matrix  is equal to the matrix .

1. Convolution:

C:/Users/71903/AppData/Local/Temp/qt_temp.d16380qt_temp

If input of convolution operator need to padding, then padding it firstly, this will not be mentioned in the algorithm.

1. Loop:

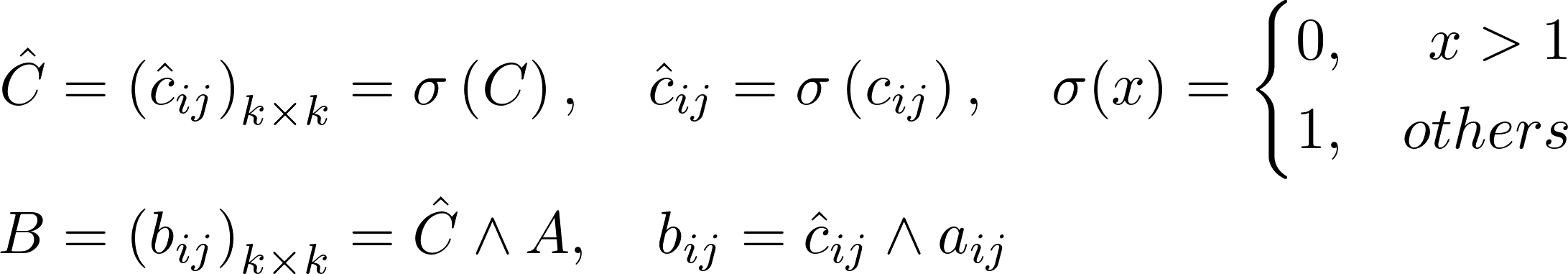
Set:

C:/Users/71903/AppData/Local/Temp/qt_temp.F12668qt_temp

and then go to step(2) until:

C:/Users/71903/AppData/Local/Temp/qt_temp.T14180qt_temp.

1. Activation:



This step is also same as the algorithm I.

The algorithm I is a particular case of algorithm II when . The algorithm II has a constant convolution kernel for different size of input matrix, but more convolution operators are needed for the same input than algorithm I. There are a trade-off between the convolution kernel size and the count of convolution operators, what is funny is that this is also a MOP.

3.3 Proof of algorithms

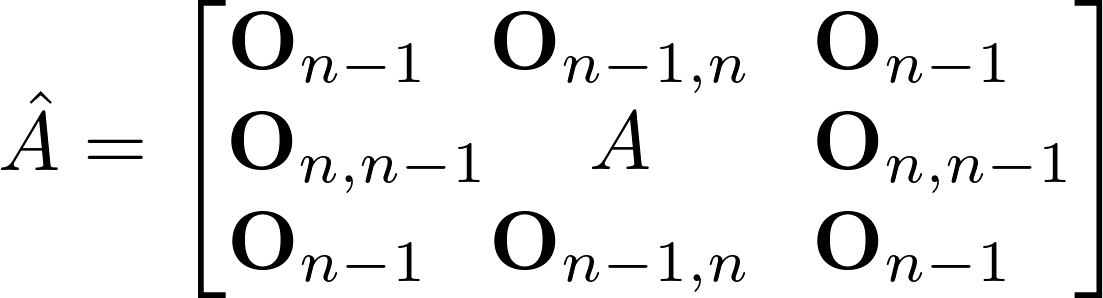
In this section, two algorithm mentioned above will be proved in mathematical form. In fact, as mentioned above, the algorithm I just is a particular case of algorithm II. But we also give the proof of the algorithm I in order to make the proof next more clear.

**Lemma 1** Matrix  is a binary image of vectors set , vector  and , then

C:/Users/71903/AppData/Local/Temp/qt_temp.L14180qt_temp

**Proof** This lemma is obviously according to the definition of Pareto dominance.

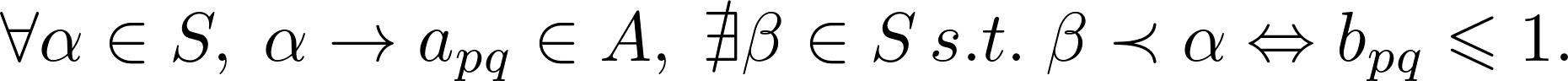
**Lemma 2** Matrix  is a binary image of vectors set , matrix ,  is a PF recognition convolution kernel, matrix



is the padding operator result of matrix , matrix

C:/Users/71903/AppData/Local/Temp/qt_temp.g16380qt_temp.

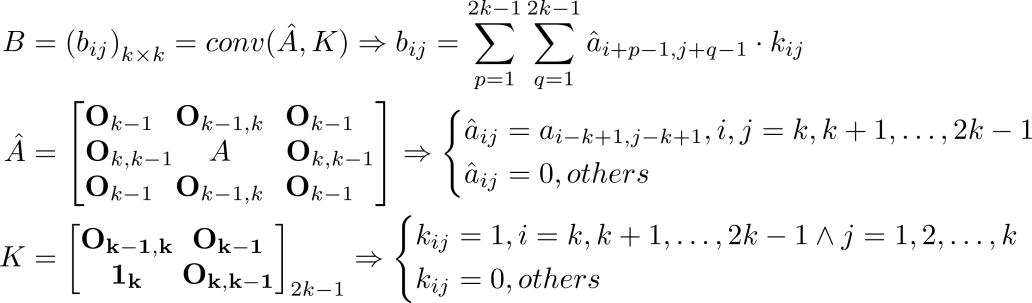
Then



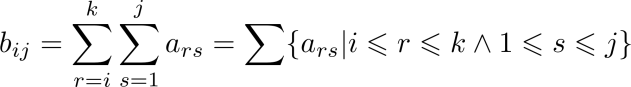
Of course set  is the PF of the set .

**Proof**

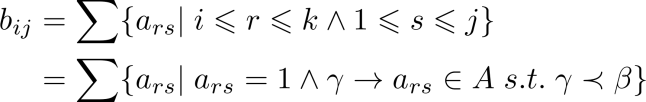
According to:



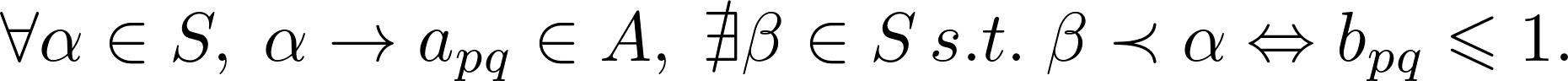
then

.

Assume that  and , this is equal to  according to the definition of binary image obviously. According to Lemma 1, then

.

So that means . Then

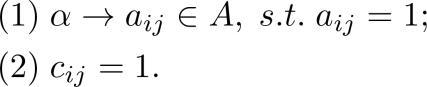


**Theorem 1** As shown in the procedure of the algorithm I, output binary image matrix  is the PF of the input binary image matrix .

**Proof**

Assume that matrix  is the binary image of vectors set , the set  is consist of all of the non-dominant vectors of set  and  is the PF of set . The binary image of set  is the PF of binary image . Assume that  is the binary image of set . Then what we have to proof is that .

According to step (1), (2) and Lemma 2, If the vector  is non-dominant vector, it should be equal to these two conditions obviously:



Then after step (3), it can be inferred that:

1. , then , indicates that , so ;
2. , then , then , so the vector  is a non-dominant vector of set , in other words, , so .

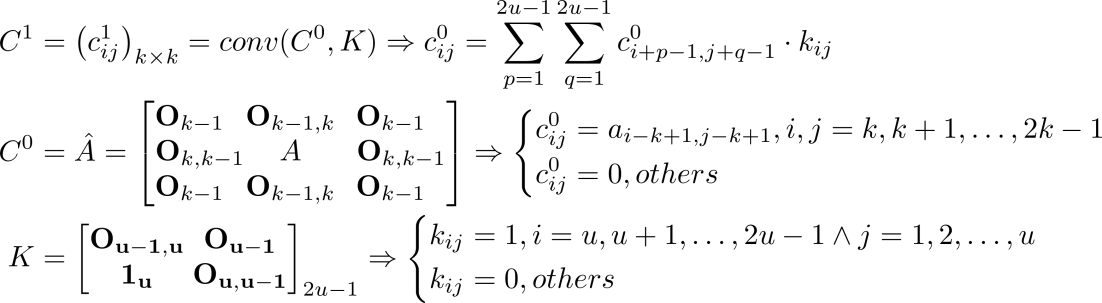
Then .

**Theorem 2** As shown in the procedure of the algorithm II, output binary image matrix  is the PF of the input binary image matrix .

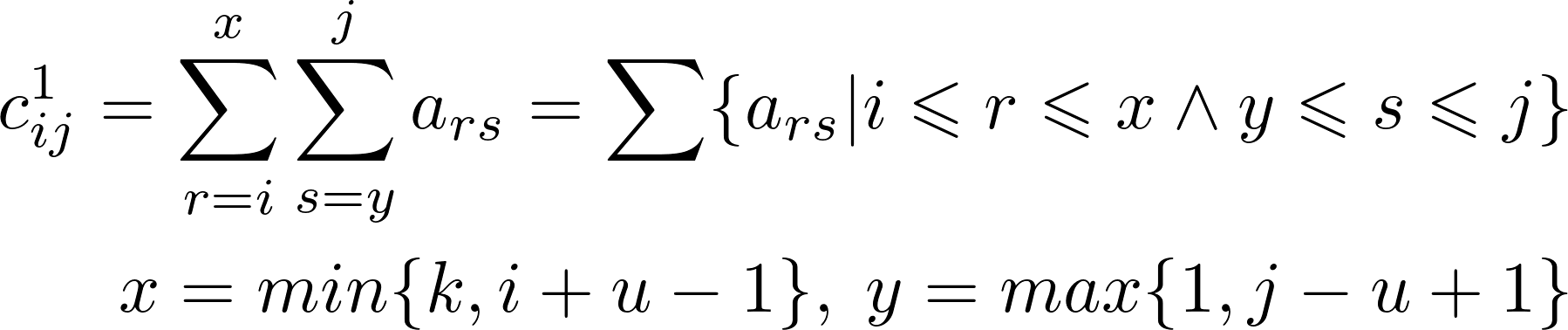
**Proof**

As mentioned before, algorithm I is just a special case of algorithm II when . We will use the proof of algorithm I to prove algorithm II. The key point of the proof is that how to prove the matrix  after step (3) has been done is same significance for indicating the dominant relationship as the matrix  in algorithm I after step (2).

Initialize as the step (1), then after step (2), according to the proof of algorithm I:



then



Obviously , it indicates that  can reflect if there is a pixel dominates the point  around it and the distance no more than .

Then step (3) is a loop condition. When  and the step (2) has been done t-th times, according to the analysis above, it indicates that  can reflect if there is a pixel dominates the point  around it and the distance no more than . Consider the extreme situation that , then the farthest point which can dominate the point  probably is the point . So when the condition  is satisfied, the loop will halt.

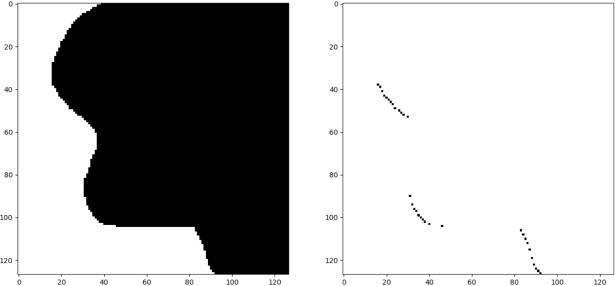
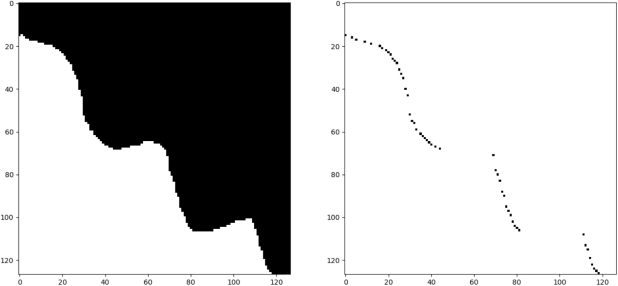
Now assume that step (3) has been done, according to the analysis of convolution step, it can be inferred that the matrix  is same significance for indicating the dominant relationship as the matrix  in algorithm I after step (2).

And the step (4) as same as the algorithm I, so it is same as the proof of Theorem 1 that the output matrix  is the PF of the input matrix .

**4 Simulation**

In this chapter, we construct two convolutional neural networks using algorithm I and algorithm II separately. Then we test the networks via some examples.

The architectures of networks is simple. The network I is based on algorithm I and there is only one layer of convolutional layer. The network II is based on algorithm II and it is a deep convolutional neural network with the depth which varies depends on the size of input and kernel. The results of networks is shown as Figure 1 and Figure 2, in each figure, the left is the input and the right is the output.



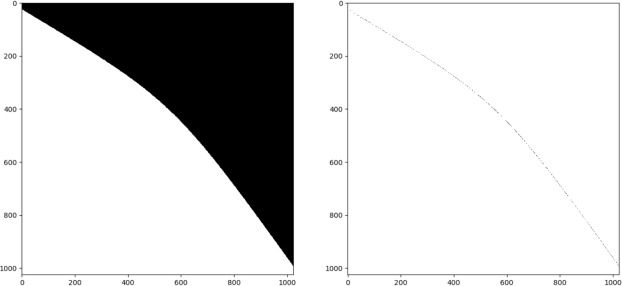
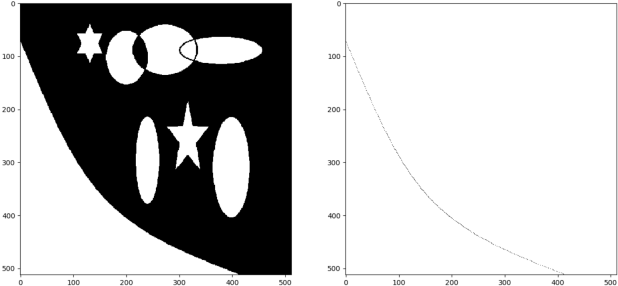
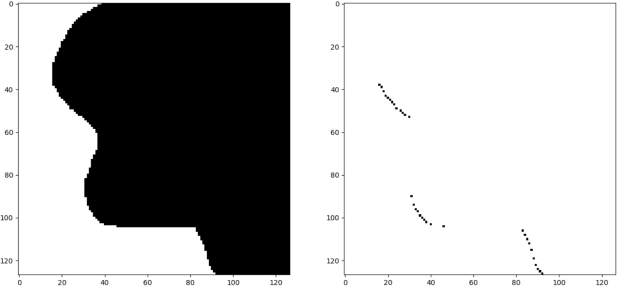
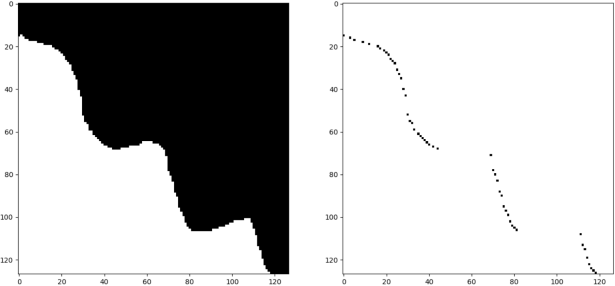


Figure 1 The PF recognition results by network I



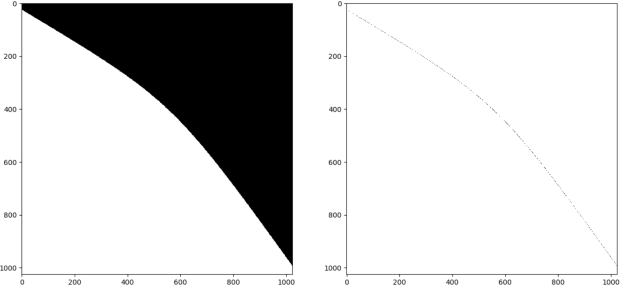
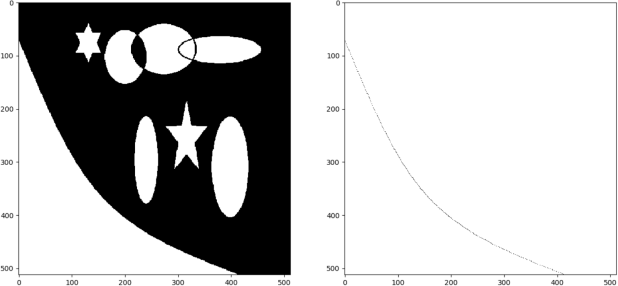


Figure 2 The PF recognition results by network II

All networks can obtain the PF correctly, but there is obviously difference between them. Due to the deep neural network architecture, network II have a small size convolution kernel, so it computes more faster.

**5 Conclusion**

In this paper, we proposed two algorithm based on the same principle to recognize the PF of a input binary image. The key point of the algorithms is the PF recognition convolution kernel we proposed firstly and the proofs of the algorithms have been given. We construct convolutional neural networks to test the algorithm and it indicates the algorithms work well.

Algorithm II is more advanced, it is faster and more general. The deep convolutional neural network based on algorithm II will be a new approach for obtaining the non-dominant vectors when solving MOPs or other situations when vectors sorting needed. It will provide an efficient method for relieving the curse of computation cost in solving complicated MOPs and even for vectors sorting when it needed.