

DYNAMIC MINIMUM SPANNING CIRCLE COMPUTATION USING ARTIFICIAL IMMUNE COMPUTING PARADIGM

SRIMANTA PAL

Electronics and Communication Sciences Unit, Indian Statistical Institute
203 B T Road, Kolkata 700 108, India
srimanta@isical.ac.in

ABSTRACT. This paper presents a method for the computation of dynamic minimum spanning circle (MSC) from a given set of points using negative selection method of artificial immune computing paradigm. This proposed algorithm is formulated for a two-dimensional data set but this can be extended to n -dimension. The analysis of this algorithm and an application in digital mammogram are also presented in this paper.

AMS (MOS) Subject Classification. 52B55, 68U05, 65D16, 97R40, 68T27.

1. INTRODUCTION

This paper suggests a method for computing the *minimum spanning circle* (MSC) or the *smallest enclosing circle* for a given set of points. The problem is to compute the radius and the center of the circle having minimum area that encloses n given points in a plane or space.

In computational geometry, (Berg et al., 1997; Edelsbrunner, 1987; Preparata & Shamos, 1985; Shamos, 1977) the MSC problem was first raised by Sylvester (1857). In subsequent years many solutions of solving MSC (Elzinga & Hearn, 1972; Francis & White, 1974; Hearn & Vijay, 1982; Megiddo, 1983; Melville, 1985; Nair & Chandrasekaran, 1971; Preparata & Shamos, 1985; Shamos & Hoey, 1975; Shamos, 1977; Skyum, 1991; Smallwood, 1965; Toussaint & Bhattacharya, 1981) have been suggested. For n number of input points, the (worst-case) time complexities for the above solutions ranges from $O(n^3)$ to $O(n \log n)$. Megiddo (1983) formulated this problem as a linear programming problem which could be solved in $O(n)$ time. A randomized algorithm is available for computing the MSC that takes expected $O(n)$ time (Welzl, 1991). Algorithm for the computation of MSC from a given set of points using neurocomputing paradigm is solved by Datta (Datta, 1996; Datta & Parui, 2002). Other than this, some of the algorithms suffer from implementation complexities, i.e., they are not simple from the point of view of programming implementation.

The MSC finds its use in pattern recognition, image analysis, statistical estimation, etc. It has applications in transmission and transportation problems. For example, in radio/TV broadcasting the serving transmitter needs to be located centrally with respect to the receivers to minimize transmission power for a given quality of services. In other words, we need to find where the facility should be located so as to minimize the maximum distance from the facility to any user. In case of transportation, the optimal location for a distributor will be minimum of the maximum distances that a customer would have to travel from his place to the distributor to make the transaction more time and cost efficient. In the above examples the set of individuals, to be served, is considered as a set of points and the center of the MSC is the optimal position of service center.

Another important application of MSC is to mark the region of interest (ROI) for the detection of breast cancer in the digital mammogram. Also this MSC can be used for the detection of outliers as discrete false positive point in the mammogram. So an optimal region of interest can be marked by MSC.

In this paper we have adopted the theory of *artificial immunecomputing* to find an optimized solution for a set of planer points. In recent years interests are increasing in solving problems using *artificial immunecomputing techniques*. The article is organized as follows: Section 2 briefly describes the method of *negative selection* as a novel computational technique of *artificial immune systems* (AIS). Section 3 contains the model used to compute *minimum spanning circle* (MSC) or the *smallest enclosing circle*. Analysis of this algorithm is discussed in Section 4. Results with an application of MSC in digital mammogram is discussed in Section 5. A conclusion has been incorporated in Section 6.

2. NEGATIVE SELECTION METHOD

With the advances in biology; molecular computing, genetics, neurocomputing, evolutionary computing, etc. are growing rapidly. On the other hand, there is a rapid increase of comprehension of the behaves of immune system (IS). The knowledge about the IS functioning has disclosed several of its main operative mechanism, *negative selection* is one of them (Castro & Timmis, 2002).

The main task of the immune system is to perform the living being in the search for malfunctioning cells from their own bodies (e.g., cancer and tumor cells), and foreign disease causing elements (e.g., viruses and bacteria). Every element that can be recognized by the immune system is called an *antigen* (Ag). The cells that originally belong to our body and are harmless to its functioning are termed *self* (or *self antigens*), while the disease causing elements are named *nonself* (or *nonself antigens*). The immune system, thus, has to be capable of distinguishing between

what is *self* from what is *nonself*, a process called *self/nonself* discrimination, and performed basically through pattern recognition events.

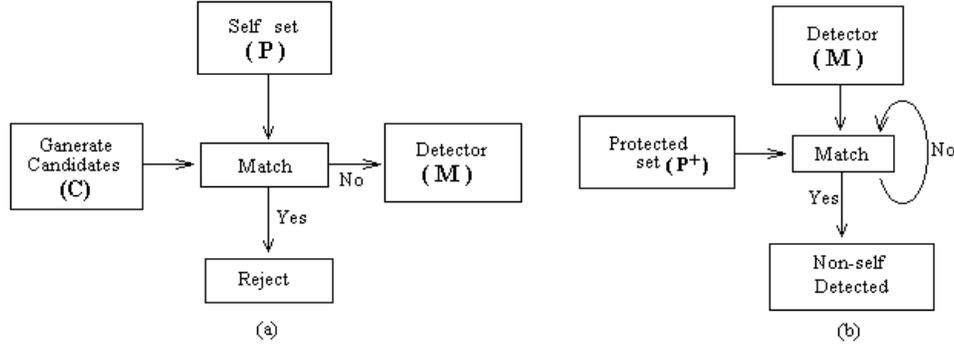


FIGURE 1. Schematic diagram for the functioning of *negative selection* algorithm: (a) Production of detectors, and (b) Monitoring for the presence of undesired (*nonself*) pattern.

In perspective of *artificial immune systems* we first determine the set of pattern to be protected and name it the *self-set* (\mathbf{P}). Based upon the *negative selection* algorithm, generate a set of *detector* (\mathbf{M}) that will be responsible to identify all the elements that do not belong to the *self-set*, i.e., the *nonself* elements. This negative nature, that, we allowed to mature only those detectors that can identify elements not belonging to *self-set* justifies the name *negative selection*. Now the negative selection method is described in Algorithm 1 as shown in Figure 1(a).

Algorithm 1: Negative selection method

- Step 1.** [Generation] Generate random candidate elements (\mathbf{C}) which are sorted afterwards to form detectors.
- Step 2.** [Matching] Compare (match) the elements in \mathbf{C} with the elements in \mathbf{P} . If an element of \mathbf{P} is recognized by an element of \mathbf{C} (match occurred), then discard this element of \mathbf{C} ; else store this element of \mathbf{C} in the detector set \mathbf{M} .
- Step 3.** [Termination] Stop.

The newly formed detector set (\mathbf{M}) is used in monitoring the system for the presence of *nonself* patterns (Figure 1(b)). This set to be monitored (\mathbf{P}^+) might be composed of the set (\mathbf{P}) plus other new patterns ($\mathbf{P}^+ \subseteq \mathbf{P}$), or it can be completely novel set ($\mathbf{P}^+ = \mathbf{P}$).

3. THE MODEL

3.1. Philosophy of the Model. As discussed earlier the concept of *minimum spanning circle* over a set of planer points is nothing but to choose a point location (i.e., center of the circle) on the plane such that the maximum distance of the points from the center is minimized.

The idea of computing MSC using immune system results from the interesting fact that, when a part of our body is affected by foreign diseases immune system tries to fight against it and assured that least area is damaged due to the infection, which in $2D$ case may be viewed as a smallest enclosing circle. This explains the reason that boils on our skin generally takes a regular circular shape.

3.2. Formulation. Minimum spanning circle for a set of points is the smallest enclosing circle which contains all the points.

Let $P = \{X_1, X_2, \dots, X_n\}$ be a set of n given points where coordinate of $X_i = (x_i, y_i)$ for $i = 1, 2, \dots, n$ in the plane. The problem is to find the center and radius of the smallest circle such that no point of P falls outside the circle.

According to the theory of *artificial immune system* (AIS), the input set P is called *self-set* which is to be protected. Based upon the *Negative Selection* algorithm, generate a random candidate circle element C , where $C = (X_c, r_c)$, $X_c = (x_c, y_c)$ the center, and $r_c =$ radius of the circle.

Now depending on the logic for matching we segregate the elements of *self-set* into two sets namely *Accepted set* (A) and *Rejected set* (R). The set A comprises of those points which are identified by the candidate element C , i.e, lies inside the circle C and set R consists of those points which are un-identified by the candidate element C , i.e, lies outside the circle C , as described in Eqn. (3.1).

$$(3.1) \quad X_i \in \begin{cases} A, & \text{if } r_c \leq \|X_i - X_c\| \\ R, & \text{otherwise} \end{cases}$$

for $i = 1, 2, \dots, n$.

After determining sets A and R , we extract information from them which will be used in the maturation of the candidate C . Let d_m^A and d_m^R be the maximum Euclidean distances between the center of C from A and R respectively as given in Eqns. (3.2) and (3.3).

$$(3.2) \quad d_m^A = \max \{0, \|X_m - X_c\|\} \quad \text{where } \|X_m - X_c\| = \max_j \|X_j - X_c\|, \quad X_j \in A$$

$$(3.3) \quad d_m^R = \max \{0, \|X_{m'} - X_c\|\} \quad \text{where } \|X_{m'} - X_c\| = \max_k \|X_k - X_c\|, \quad X_k \in R$$

The randomly selected initial value of the candidate circle be $C(0)$. At iteration t the maturation which takes place in $C(t) = (X_c(t), r_c(t))$ is given in Eqns. (3.4) and (3.5).

$$(3.4) \quad r_c(t+1) = (1 - \alpha) r_c(t) + \alpha(\xi d_m^A + \eta d_m^R)$$

$$(3.5) \quad X_c(t+1) = X_c(t) + \beta (X_m - X_c)$$

where $0 \leq \alpha, \beta \leq 1$ and

$$(3.6) \quad \xi = \begin{cases} 1, & \text{if } |A| > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$(3.7) \quad \eta = \begin{cases} 1, & \text{if } |R| > 0 \\ 0, & \text{otherwise} \end{cases}$$

The idea is as follows: The *Minimum Spanning Circle* (MSC) will contain all the points in P , so we have changed the radius with a tuning parameter α with iteration in Eqn. (3.4). If some points lie outside then $d_m^A < d_m^R$, so the radius $r_c(t)$ will increase with iteration until all the points come within the circle (even for a single point lying outside $d_m^A < d_m^R$) then the *Rejected set* (R) will become empty ($d_m^R = 0$) and as t goes to infinity $r_c(t)$ will then tend to d_m^A . Since we are trying to minimize the radius, we also move the center of circle C towards the point X_m , which is the farthest point of set A from X_c . After each iteration the farthest point of the set A from X_c is recalculated. It is clear that during this process, some new point $X_{m'}$ ($m' \neq m$) becomes the farthest point from the center X_c and, as soon as it happens, X_c starts moving toward $X_{m'}$ instead of moving toward X_m (using Eqn. (3.4) and (3.5)).

Algorithm 2: Dynamic MSC

- Step 1.** [Initialization] Initialize iteration number $t = 0$.
Generate a random candidate circle $C(t) = (X_c(t), r_c(t))$.
- Step 2.** [Grouping] Split the input set (*self-set*) P into two sets A and R using *match logic* according to Eqn. (3.1).
- Step 3.** [Computation] Calculate d_m^A and d_m^R using Eqns. (3.2) and (3.3)
- Step 4.** [Modification] Modify the candidate circle C defined in Eqns. (3.4) and (3.5).
- Step 5.** If necessary add a new input point into P or delete an existing input point from P .
- Step 6.** [Looping] Set $t = t + 1$ and repeat from Step 2 to Step 5 until the parameters of candidate circle C does not change.
- Step 7.** [Termination] Stop.

4. ANALYSIS

In this section we prove the convergence of the algorithm. Rearranging Eqn. (3.5) we get

$$(4.1) \quad X_c(t+1) = (1 - \beta)X_c(t) + \beta X_m$$

Substituting $V(t)$ as a center of the random detector circle instead of $X_c(t)$, $X(t)$ as a member of *self-set* in place of X_m (where $X(t) \in P$) and making β dynamic, i.e.,

replace β by $\beta(t)$ we get

$$(4.2) \quad V(t+1) = (1 - \beta(t))V(t) + \beta(t) X(t)$$

Using Eqn. (4.2) we get by putting $t = 0, 1, \dots, t$ as:

$$\begin{aligned} V(1) &= (1 - \beta(0))V(0) + \beta(0)X(0) \\ V(2) &= (1 - \beta(1))V(1) + \beta(1)X(1) \\ &= (1 - \beta(1))(1 - \beta(0))V(0) + \beta(0)(1 - \beta(1))X(0) + \beta(1)X(1) \\ &= \prod_{i=0}^1 (1 - \beta(i))V(0) + \sum_{k=0}^{1-1} \beta(k) \left(\prod_{i=k+1}^1 (1 - \beta(i)) \right) X(k) + \beta(1)X(1) \\ &\quad \vdots \end{aligned}$$

Similarly we can write,

$$\begin{aligned} V(t+1) &= (1 - \beta(t))(1 - \beta(t-1)) \cdots (1 - \beta(0))V(0) \\ &\quad + \beta(0)(1 - \beta(1))(1 - \beta(2)) \cdots (1 - \beta(t))X(0) \\ &\quad + \beta(1)(1 - \beta(2))(1 - \beta(3)) \cdots (1 - \beta(t))X(1) \\ &\quad + \cdots + \beta(t-1)(1 - \beta(t))X(t-1) + \beta(t)X(t) \end{aligned}$$

$$(4.3) \quad V(t+1) = \prod_{i=0}^t (1 - \beta(i))V(0) + \sum_{k=0}^{t-1} \left(\beta(k) \left(\prod_{i=k+1}^t (1 - \beta(k)) \right) X(k) \right) + \beta(t)X(t)$$

Before analyzing Eqn. (4.3), we shall study the properties of the maturity factor $\beta(t)$ of detector circle.

Property 1: If $\beta(t)$ is the maturity factor of detector circle at time t then $\prod_{t=0}^{\infty} (1 - \beta(t)) = 0$.

Proof. Since $0 < \beta(t) < 1$ for all t , $\sum_{t=0}^{\infty} \beta(t) \rightarrow \infty$ if and only if $\prod_{t=0}^{\infty} (1 - \beta(t)) = 0$. □

Result 1: If $G(t) = \sum_{k=0}^t \beta(k) \left(\prod_{i=k+1}^t (1 - \beta(k)) \right) + \beta(t)$ then $G(t) = 1$ as $t \rightarrow \infty$ for $0 < \beta(k) < 1$, $0 \leq k \leq t$.

Proof. We can write

$$\prod_{i=0}^t (1 - \beta(i)) = (1 - \beta(0)) \prod_{i=1}^t (1 - \beta(i))$$

i.e.,

$$\prod_{i=0}^t (1 - \beta(i)) = \prod_{i=1}^t (1 - \beta(i)) - \beta(0) \prod_{i=1}^t (1 - \beta(i))$$

Therefore

$$\beta(0) \prod_{i=1}^t (1 - \beta(i)) = \prod_{i=1}^t (1 - \beta(i)) - \prod_{i=0}^t (1 - \beta(i))$$

Hence we can write the following equations

$$\begin{aligned} \beta(0) \prod_{i=1}^t (1 - \beta(i)) &= \prod_{i=1}^t (1 - \beta(i)) - \prod_{i=0}^t (1 - \beta(i)) \\ \beta(1) \prod_{i=2}^t (1 - \beta(i)) &= \prod_{i=2}^t (1 - \beta(i)) - \prod_{i=1}^t (1 - \beta(i)) \\ \beta(2) \prod_{i=3}^t (1 - \beta(i)) &= \prod_{i=3}^t (1 - \beta(i)) - \prod_{i=2}^t (1 - \beta(i)) \\ &\vdots = \vdots \\ \beta(t-1) \prod_{i=t}^t (1 - \beta(i)) &= \prod_{i=t}^t (1 - \beta(i)) - \prod_{i=t-1}^t (1 - \beta(i)) \\ \beta(t) &= 1 - (1 - \beta(t)) \end{aligned}$$

Adding both the sides of the above equations we get

$$(4.4) \quad \sum_{k=0}^{t-1} \beta(k) \left(\prod_{i=k+1}^t (1 - \beta(k)) \right) + \beta(t) = 1 - \prod_{i=0}^t (1 - \beta(i))$$

Now by Property 1 $\prod_{i=0}^t (1 - \beta(i)) = 0$ as $t \rightarrow \infty$. Hence $G(t) = 1$ as $t \rightarrow \infty$.

Equation (4.3) can be rewritten as

$$\begin{aligned} V(t+1) &= \prod_{i=0}^t (1 - \beta(i)) (V(0) - X_q) \\ &\quad + \sum_{k=0}^{t-1} \left(\beta(k) \left(\prod_{i=k+1}^t (1 - \beta(k)) \right) (X(k) - X_q) \right) \\ &\quad + \beta(t) (X(t) - X_q) \\ &\quad + \left(\prod_{i=0}^t (1 - \beta(i)) + \sum_{k=0}^{t-1} \left(\beta(k) \left(\prod_{i=k+1}^t (1 - \beta(k)) \right) \right) + \beta(t) \right) X_q \end{aligned}$$

or,

$$\begin{aligned} V(t+1) - X_q &= \prod_{i=0}^t (1 - \beta(i)) (V(0) - X_q) \\ &\quad + \sum_{k=0}^{t-1} \left(\beta(k) \left(\prod_{i=k+1}^t (1 - \beta(k)) \right) (X(k) - X_q) \right) \\ &\quad + \beta(t) (X(t) - X_q) \quad [\text{using Result 1}] \end{aligned}$$

$$\begin{aligned}
(4.5) \quad \|V(t+1) - X_q\| &\leq \prod_{i=0}^t (1 - \beta(i)) \|V(0) - X_q\| \\
&\quad + \sum_{k=0}^{t-1} \left(\beta(k) \left(\prod_{i=k+1}^t (1 - \beta(k)) \right) \|X(k) - X_q\| \right) \\
&\quad + \beta(t) \|X(t) - X_q\|
\end{aligned}$$

□

The radius updation formula defined in Eqn. (3.4) defines the following three types of system.

Type 1: ($\xi = 0$ and $\eta = 1$). In this type the center of the circle does not move but the radius increases with the distance of the furthest point in A from the center of the current circle. The formula in Eqn. (3.4) can be written as

$$(4.6) \quad r_c(t+1) = (1 - \alpha)^t r_c(0) + \alpha \sum_{i=0}^{t-1} (1 - \alpha)^i d_m^R(i)$$

Type 2: ($\xi = 1$ and $\eta = 1$). In this type the center of the circle is moving towards the furthest point of set A and also the radius is changing according to the previous values of $r_c(t)$, $d_m^A(t)$, and $d_m^R(t)$. The formula in Eqn. (3.4) can be written as

$$(4.7) \quad r_c(t+1) = (1 - \alpha)^t r_c(0) + \alpha \sum_{i=0}^{t-1} (1 - \alpha)^i (\xi d_m^A(i) + \eta d_m^R(i))$$

Type 3: ($\xi = 1$ and $\eta = 0$). In this type both the center and the radius of the circle changes with the set A. Here the center of the circle is moving towards the furthest point of A and it is changing (reducing) the radius of the circle. The formula in Eqn. (3.4) can be written as

$$(4.8) \quad r_c(t+1) = (1 - \alpha)^t r_c(0) + \alpha \sum_{i=0}^{t-1} (1 - \alpha)^i \xi d_m^A(i)$$

Lemma 1: If the point set is $P = \{X_1\}$ then the center of the MSC is X_1 and radius is 0.

Proof. By using Eqn. (4.5) we get

$$\begin{aligned}
(4.9) \quad \|V(t+1) - X_1\| &\leq \prod_{i=0}^t (1 - \beta(i)) \|V(0) - X_1\| \\
&\quad + \sum_{k=0}^{t-1} \left(\beta(k) \left(\prod_{i=k+1}^t (1 - \beta(k)) \right) \|X(k) - X_1\| \right) \\
&\quad + \beta(t) \|X(t) - X_1\|
\end{aligned}$$

In this case the term $\|V(0) - X_1\|$ in $\prod_{i=0}^t (1 - \beta(i)) \|V(0) - X_1\|$ is independent of i . Hence by Property 1, $\prod_{i=0}^t (1 - \beta(i)) \|V(0) - X_1\| = 0$ as $t \rightarrow \infty$. Again both

$\|X(k) - X_1\|$ and $\|X(t) - X_1\|$ are 0. Since $X(k) \rightarrow X_1$ and $X(t) \rightarrow X_1$. Therefore $\|V(t+1) - X_1\| \leq 0$ for large t . Hence $V(t+1) \rightarrow X_1$ as $t \rightarrow \infty$, i.e., the center of the MSC converges to X_1 .

By using Eqn. (4.6) of Type 1 we observe

$$(4.10) \quad r_c(t+1) = (1-\alpha)^t r_c(0) + \alpha \sum_{i=0}^{t-1} (1-\alpha)^i d_m^R(i)$$

In this case, there exists a $t = T_0$ such that $r_c(T_0) \geq d_m^R(T_0)$ where $d_m^R(T_0) = \|X_c(0) - X_1\|$. Also $r_c(t) < r_c(t+1)$ for all $t < T_0$.

If $t > T_0$ then it follows Eqn. (4.8) of Type 3. Here $r_c(0) = r_c(T_0)$ and $d_m^A(T_0) = \|X_c(T_0) - X_1\|$ but $X_c(T_0) = X_c(0)$. So we can write

$$(4.11) \quad r_c(t+1) = (1-\alpha)^t r_c(T_0) + \alpha \sum_{i=T_0}^{t-1} (1-\alpha)^i d_m^A(i)$$

Also we have seen $X_c(t) \rightarrow X_1$ as $t \rightarrow \infty$, i.e., $d_m^A(t) = \|X_c(t) - X_1\| \rightarrow 0$ as $t \rightarrow \infty$ when $t > T_0$. This also says $r_c(t) \rightarrow 0$ as $t \rightarrow \infty$. \square

Lemma 2: If the point set, $P = \{X_1, X_2\}$ then the center of the MSC is (i) laying on the line joining X_1 and X_2 , and (ii) $X_c \rightarrow \frac{X_1+X_2}{2}$ as $t \rightarrow \infty$.

Proof. By using Eqn. (4.5) we get

$$(4.12) \quad \begin{aligned} & \|V(t+1) - X_1\| + \|V(t+1) - X_2\| \\ & \leq \prod_{i=0}^t (1-\beta(i)) (\|V(0) - X_1\| + \|V(0) - X_2\|) \\ & \quad + \sum_{k=0}^{t-1} \left(\beta(k) \left(\prod_{i=k+1}^t (1-\beta(k)) \right) (\|X(k) - X_1\| + \|X(k) - X_2\|) \right) \\ & \quad + \beta(t) (\|X(t) - X_1\| + \|X(t) - X_2\|) \end{aligned}$$

Here $X(k)$ and $X(t)$ is either X_1 or X_2 .

$$(4.13) \quad \begin{aligned} & \|V(t+1) - X_1\| + \|V(t+1) - X_2\| \\ & \leq \prod_{i=0}^t (1-\beta(i)) (\|V(0) - X_1\| + \|V(0) - X_2\|) \\ & \quad + \sum_{k=0}^{t-1} \beta(k) \left(\prod_{i=k+1}^t (1-\beta(k)) + \beta(t) \right) (\|X_1 - X_2\|) \\ & = \prod_{i=0}^t (1-\beta(i)) (\|V(0) - X_1\| + \|V(0) - X_2\|) \\ & \quad + \left(1 - \prod_{i=0}^t (1-\beta(i)) \right) (\|X_1 - X_2\|) \end{aligned}$$

As $t \rightarrow \infty$ then

$$\begin{aligned}
(4.14) \quad & \|V(t+1) - X_1\| + \|V(t+1) - X_2\| \\
& \leq \|X_1 - X_2\| + \prod_{i=0}^t (1 - \beta(i)) (\|V(0) - X_1\| + \|V(0) - X_2\| - \|X_1 - X_2\|) \\
& \quad \|V(t+1) - X_1\| + \|V(t+1) - X_2\| \\
& = \|X_1 - X_2\|
\end{aligned}$$

as $t \rightarrow \infty$ [by Property 1]

This is only possible when the point $V(t+1)$ lies on the line joining X_1 and X_2 .

If we introduce the term $G(t)$ then we can write

$$\begin{aligned}
& \|V(t+1) - X_2\| + \|V(t+1) - X_1\| \\
& \leq \prod_{i=0}^t (1 - \beta(i)) \{ \|V(0) - X_2\| + \|V(0) - X_1\| \} \\
& \quad + \left(\sum_{k=0}^{t-1} \beta(k) \prod_{i=k+1}^t (1 - \beta(k)) + \beta(t) \right) \{ \|X_1 - X_2\| \}.
\end{aligned}$$

or, $\|V(t+1) - X_2\| + \|V(t+1) - X_1\| = (1 - G(t)) (\|V(0) - X_1\| + \|V(0) - X_2\|) + G(t) \|X_1 - X_2\|$.

This indicates as $t \rightarrow \infty$, $V(t+1)$ will be on the line joining X_1 and X_2 (using Result 1, i.e., $G(t) = 1$ as $t \rightarrow \infty$).

Now we prove that $V(t+1)$ tends to the midpoint of X_1 and X_2 , i.e., $V(t+1) \rightarrow \frac{X_1 + X_2}{2}$ as $t \rightarrow \infty$.

Using Eqn. (4.2) we get $\|V(t+1) - V(t)\| = \beta(t) \|X(t) - V(t)\|$. Again $\beta(t) \rightarrow 0$ as $t \rightarrow \infty$. So for any positive number δ there exists some t_0 for which $\|V(t_0+1) - V(t_0)\| < \delta$.

Let H_1 and H_2 be two half plains divided by the perpendicular bisector L_{12} of the line joining X_1 and X_2 , where $X_1 \in H_1$ and $X_2 \in H_2$. Without loss of generality we assume $V(t_0)$ belongs to H_1 , it will move towards X_2 till it reaches H_2 at some time $t_1 > t_0$ and evidently the perpendicular distance from $V(t_1)$ to L_{12} will be less than $\|V(t_1) - V(t_1 - 1)\|$. Once $V(t_1)$ reaches H_2 , $V(t)$ starts moving towards X_1 and for some $t_2 > t_1$, $V(t_2)$ will fall on H_1 .

Hence, $V(t)$ will approach to $(X_1 + X_2)/2$ as $t \rightarrow \infty$. \square

Lemma 3: If the point set is $P = \{X_1, X_2, X_3\}$ then the center of the MSC is lying on the intersection of three farthest point Voronoi polygons.

Proof. The farthest point Voronoi diagram (FPVD) of X_1, X_2 , and X_3 will results two distinct casees. In case of obtuse angled triangle (Figure 2(a)) the vertex of

the FPVD of X_1, X_2 and X_3 will lie outside the triangle and in case of acute angled triangle (Figure 2(b)) the vertex will be inside the triangle. In Figure 2 Q_1, Q_2 and Q_3 represents the farthest point Voronoi polygons for X_1, X_2 , and X_3 respectively. \square

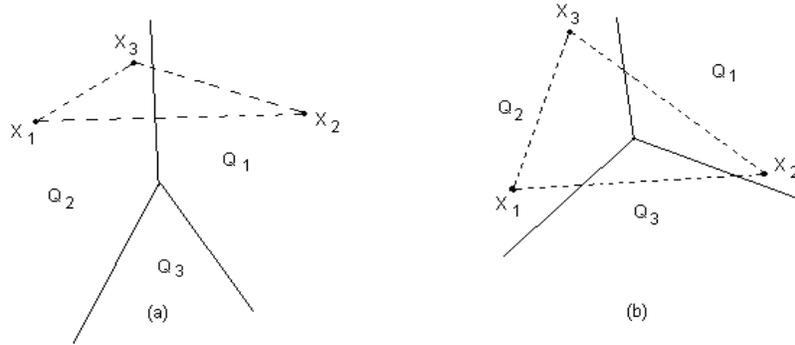


FIGURE 2. The position of the center of minimum spanning circle in case of three points. (a) Center lies on the middle of X_1 and X_2 and (b) Center lies on the intersection of three farthest point Voronoi polygons.

Case 1. In case of obtuse angled triangle (Figure 2(a)) we will always find a t_0 , for which $V(t)$ will lie outside Q_3 for all $t > t_0$. From Lemma 2, it is then clear that $V(t)$ will tend to $(X_1 + X_2)/2$ as $t \rightarrow \infty$.

Case 2. In case of acute angled triangle (Figure 2(b)), the three perpendicular bisectors L_{ij} of the line joining X_i and X_j ($1 \leq i, j \leq 3, i \neq j$) determines the vertex of the FPVD of X_1, X_2 and X_3 . Arguments in the proof of Lemma 2 justifies the fact that $V(t)$ will be arbitrarily close to each of the three L_{ij} 's. Thus $V(t)$ converges to the intersection point of the three bisectors. Here, the FPV vertex is the center of minimum spanning circle.

4.1. Generalization. Initially the system is one of the three state as in Types 1 to 3. Suppose the system is in Type 1 state then after certain number of iterations the system changes more likely to Type 2 or may be Type 3 state. If the system is in Type 2 state then again after few iterations it changes to Type 3 state. The state transition are shown in Figure 3. This property can be proven by Lemma 4.

Lemma 4: The system with n points, $P = \{X_1, X_2, \dots, X_n\}$, $|P| > 1$ is in Type 1 state, then there exists a $t = T_0$ such that the system transformed to Type 2 or 3 state.

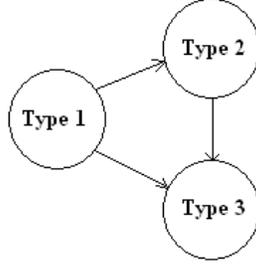


FIGURE 3. The state transition diagram of a system – Type 1 system: $\xi = 0$, $\eta = 1$, Type 2 system: $\xi = 1$, $\eta = 1$ and Type 3 system: $\xi = 1$, $\eta = 0$.

Proof. Using Eqn. (4.6) we can write

$$\begin{aligned}
 r_c(T_0 + 1) &= (1 - \alpha)^{T_0} r_c(0) + \alpha\eta \sum_{i=0}^{T_0-1} (1 - \alpha)^i d_m^R(i) \\
 &= (1 - \alpha)^{T_0} r_c(0) + \alpha\eta \frac{1 - (1 - \alpha)^{T_0}}{1 - (1 - \alpha)} d_m^R \\
 &= (1 - \alpha)^{T_0} r_c(0) + (1 - (1 - \alpha)^{T_0}) d_m^R \\
 &= (1 - \alpha)^{T_0} (r_c(0) - d_m^R) + d_m^R \\
 &= d_m^R \quad [\text{For large } T_0 \text{ and using Lemma 1}]
 \end{aligned}$$

For $t < T_0$, $d_m^R(i)$ is fixed say d_m^R because the system does not change the center and points in P are outside the circle but in every iteration the radius of the circle is changing by using Eqn. (4.6). When the radius $\geq d_m^R$ few points of P either inside the circle or on the circumference. In this situation the $|A| > 0$ therefore $\xi = 1$ and $\eta = 1$ or 0 . So the system moves to state as defined in Type 2 or Type 3. In this situation the center of the circle starts changing by using Eqn. (4.3). \square

Lemma 5: The system with n points, $P = \{X_1, X_2, \dots, X_n\}$, $|P| > 1$ is in Type 2 state, then there exists a $t = T_1$ such that the system transformed to Type 3 state.

Proof. Using Eqn. (4.7) we can write

$$(4.15) \quad r_c(t + 1) = (1 - \alpha)^{t-T_0} r_c(T_0) + \alpha \sum_{i=T_0}^{t-1} (1 - \alpha)^i (\xi d_m^A(i) + \eta d_m^R(i)) \quad \text{for } t > T_0$$

In this situation the center of the circle is moving towards the furthest point of A , so its radius changes either in increasing or decreasing order. The target is to $|R| \rightarrow 0$ and $|A| \rightarrow |P|$ as t increases to T_1 , that is, set A will contain all the points in P . At that situation $\eta = 0$ and ξ remains 1. Hence the system becomes in a state defined by Type 3. \square

Lemma 6: The system with n points, $P = \{X_1, X_2, \dots, X_n\}$, $|P| > 1$ is in Type 3 state, then there exists a $t = T_2$ such that the system remains in Type 3 state but the center and radius of the circle remains unchange.

Proof. Using Eqn. (4.7) we can write

$$(4.16) \quad r_c(t+1) = (1-\alpha)^{t-T_1} r_c(T_1) + \alpha \sum_{i=T_1}^{t-1} (1-\alpha)^i \xi d_m^A(i) \quad \text{for } t > T_1$$

Also the center is updated by the formula in Eqn. (3.4)

$$(4.17) \quad X_c(t+1) = X_c(t) + \beta (X_m - X_c).$$

Here $\xi = 1$ for $t > T_2$ where T_2 large integer. $\|X_c(t+1) - X_c(t)\| < \epsilon$ and $\|r_c(t+1) - r_c(t)\| < \delta$ where ϵ and δ are a small quantities and

$$r_c(t+1) = (1-\alpha)^{t-T_1} r_c(T_1) + \alpha \sum_{i=T_1}^{t-1} (1-\alpha)^i \xi d_m^A(i) \quad \text{for } t > T_2 > T_1.$$

As $t \rightarrow \infty$ $(1-\alpha)^{t-T_1} \rightarrow 0$, $d_m^A(i)$ is fixed to d_m^A (say) then $r_c \rightarrow d_m^A$ since $\xi \alpha \sum_{i=0}^{\infty} (1-\alpha)^i d_m^A = \alpha \frac{d_m^A}{1-(1-\alpha)} = d_m^A$. Hence the center and the radius of the circle remains unchanged. \square

Theorem 1: The center and radius of the MSC, i.e., $(X_c(t), r_c(t))$ obtain from $P = \{X_1, X_2, \dots, X_n\}$, $n = |P| > 0$ using Algorithm dynamic MSC remains unchanged as $t \rightarrow \infty$.

Proof. Easily follows from Lemma 1 to Lemma 6. \square

5. RESULT

The proposed model is tested on several $2D$ point set. Figure 4 shows the test result where the MSC passes through 3 points. This figure has 4 panels: panel (a) of Figure 4 shows the case when all the points lyes outside the randomly generated circle, i.e., set A is empty where as R set contains all the points (using Eqn. (3.1)) and $d_m^A = 0$, $d_m^R \neq 0$ (using Eqn. (3.2)). Panel (b) of Figure 4 shows the case where none of the sets A and R are empty and $d_m^A < d_m^R$ (using Eqn. (3.2)), while panel (c) of Figure 4 indicates the circle contains all the points, i.e., set R is empty and $d_m^A \neq 0$, $d_m^R = 0$. Final result is shown in Panel (d) of Figure 4 obtained after 490 iteration.

One application of MSC is shown in breast cancer detection using digital mammogram. A digital mammogram (see Figure 5(a)) is an low-dose x-ray image of soft breast tissues. It is used to detect breast cancer by any computational techniques like computational intelligence tools and techniques (Pal et al., 2008). The objective of the detection technique is to identify the location of the presence of microcalcification (if any) from the given digital mammogram (Figure 5(a)) and then further

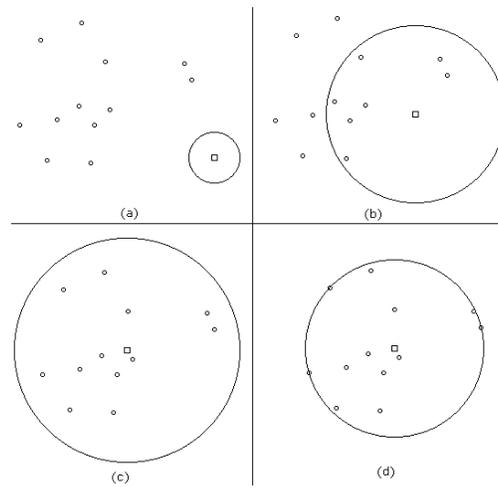


FIGURE 4. The intermediate and final results on a point set ($\alpha = \beta = 0.01$), where center of the circle is represented by a rectangle: (a) Initial position of randomly generated circle with respect to 12 point data (small circle), (b) Position of circle after 44th iteration, (c) Circle enclosing all points after 126th iteration, and (d) Final position of circle after 490th iteration.

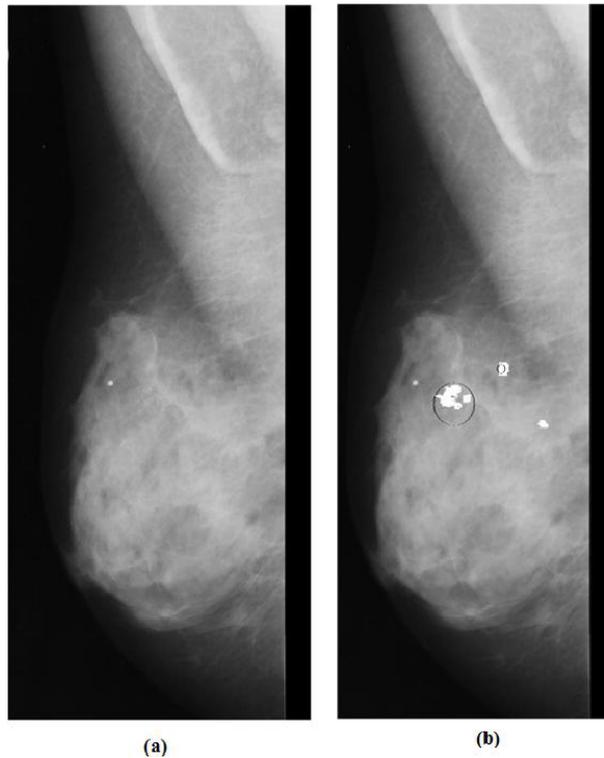


FIGURE 5. Digital mammogram and its region of interest (ROI) marked by minimum spanning circle: (a) original mammogram, and (b) marked output mammogram

investigate that the microcalcification is benign or malignant. The location of microcalcification is known as region of interest (ROI). Figure 5(a) is an original image of digital mammogram. A breast cancer detection technique detects positive calcified pixels and marked by a circle as shown in Figure 5(b). This circle is obtained by MSC algorithm. This algorithm marks the affected region of the breast on the digital mammogram by the minimum spanning circle. This circle indicates a region of interest (ROI) and also gives an estimate about the area affected by the cancer. The exact affected area will be estimated by the expert after further study. This marking technique using MSC is also applicable for 3-D breast image. In this case it will be the minimum spanning sphere.

6. CONCLUSION

The performance of the algorithm in terms of time complexity and approximation depends on the value of α and β ($0 < \alpha, \beta < 1$). By using Eqn. (3.4) α and β control the value of allowable mutation (change in the candidate circle) and maturity of the circle towards the MSC. Higher accuracy can be obtained by decreasing α and β slowly when the equilibrium condition is reached, but in the beginning α and β must be appreciably large for fast convergence of algorithm.

The result of the algorithm does not depend on the initial choice of circle and the algorithm is completely dynamic in nature, i.e., data set can be modified in any time as required. This algorithm can easily be extended in 3 or more dimensions. This method of MSC construction works on the dynamic data set.

REFERENCES

- [1] M. Berg, M. Kreveld, M. Overmars, & O. Schwarzkopf, *Computational geometry algorithms and applications*. Springer-Verlag, 1997.
- [2] L. N. de Castro and J. Timmis. *Artificial Immune System: A New Computational intelligence Approach*. Springer, 2002.
- [3] A. Datta, Computing minimum spanning circle by self-organization. *Neurocomputing*, 13:75–83, 1996.
- [4] A. Datta and S. K. Parui, A neural network model for minimum spanning circle: its convergence, architecture design and applications. *International Journal of Pattern Recognition and Artificial Intelligence*, 16:797–815, 2002.
- [5] H. Edelsbrunner, *Algorithms in combinatorial geometry*. Springer-Verlag, 1987.
- [6] J. Elzinga and D. E. Hearn, Geometrical solutions for some minimax location problems. *Transportation Sci.*, vol. 6, pp. 379–394, 1972.
- [7] R. L. Francis and J. A. White, *Facility Layout and Location*. Prentice-Hall, 1974.
- [8] D. W. Hearn and J. Vijay, Efficient algorithms for the (weighted) minimum circle problem. *Operation Research*, 30:777–795, 1982.
- [9] N. Megiddo, Linear time algorithms for linear programming in R^3 and related problems. *SIAM J. of Computing*, 12:759–776, 1983.

- [10] R. C. Melville, An implementation study of two algorithms for the minimum spanning circle problem. In *Computational Geometry*, ed. G. T. Toussaint, North-Holland, 1985, pp. 267–294.
- [11] K. P. K. Nair and R. Chandrasekaran, Optimal location of a single service center of certain types. *Naval Res. Logist. Quart.*, 18:503–510, 1971.
- [12] N. R. Pal, B. Bhowmick, S. K. Patel, S. Pal and J. Das, A multi-stage neural network aided system for detection of microcalcifications in digitized mammograms. *Neurocomputing*, 71:2625–2634, 2008.
- [13] F.P. Preparata and M.I. Shamos, *Computational Geometry: An Introduction*. New York: Springer-Verlag, 1985.
- [14] M. Shamos and D. Hoey, Closest-point problems. In *Proc. 16th Annual IEEE Symposium on Foundations of Computer Science*, Los Angeles, pp. 151–162, 1975.
- [15] M. Shamos. *Problems in Computational Geometry*. Carnegie-Mellon Univ., 1977.
- [16] R. Skyum, A simple algorithm for smallest enclosing circle. *Infor. Proc. Letters*, 37:121–125, 1991.
- [17] R. D. Smallwood, Minimax detection station placement. *Operation Research*, 13:636–646, 1965.
- [18] J. J. Sylvester, A question in the geometry of situation. *Quart. J. Math.*, 1:79, 1857.
- [19] G. Toussaint and B. Bhattacharya, “On geometric algorithms that use the farthest-point Voronoi diagram”, *Tech. Rept. SOCS-81.3*, McGill Univ. School of Computer Sci., 1981.
- [20] E. Welzl, Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science*, ed. H. Maurer, *Lecture Notes in Computer Science*, vol. 555, pp. 359–370, 1991.