Chapter 5

A Local Network Neighbourhood Artificial Immune System with Application to Unsupervised Data Clustering

The co-operation and co-stimulation or suppression between lymphocytes to respond and adapt to invading antigens can result in the formation of lymphocyte network structures in the natural immune system, according to the network theory of immunology. An antigen stimulated lymphocyte not only secretes antibodies but also proliferates by generating mutated clones to adapt to the antigen structure. The proliferation of a lymphocyte stimulates the immediate neighbouring lymphocytes, which in turn might also proliferate to adapt to the antigen structure and stimulate neighbouring lymphocytes. Thus, a network of lymphocytes *learns* the structure of an antigen by co-stimulating each other. The network topology of co-stimulated lymphocytes inspired the modelling of the local network neighbourhood artificial immune system (LNNAIS). The different parts of the LNNAIS algorithm are discussed in sections 5.1 to 5.4. The differences and similarities between existing network based AIS models and the proposed LNNAIS are discussed in section 5.5.

5.1 The Algorithm

The proposed LNNAIS algorithm is given in pseudo code in algorithm 5.1 and consists of seven high level steps to respond to an antigen/training pattern. Figure 5.1 shows a flow chart for the steps in the LNNAIS algorithm. These steps are:

1. Initialise the ALC population

2. Present an antigen to each ALC in the population and return the ALC with the highest calculated binding *affinity* with the antigen.

3. The returned highest affinity ALC reacts to the antigen pattern by initialising the antigen pattern as an antigen mutated clone and binds to the clone.

4. If the highest affinity ALC *activates*, the activated ALC spawns a mutated clone.

5. The spawned clone then binds to those antigen mutated clones of the activated ALC with which the spawned clone has a higher binding affinity than the activated ALC.

6. The mutated clone or activated ALC then *co-stimulates* ALCs which is within the *local neighbourhood* of the activated ALC.

7. Co-stimulation of neighbouring ALCs can result in co-suppression and/or the non-proliferation of other ALCs in the population.

The first step initialises the ALC population. The second and third step simulate the *affinity maturation* of a lymphocyte in the natural immune system. The second step models the *clonal selection* of the natural immune system. The antigen pattern selects the ALC with which the antigen has the highest binding affinity for cloning. The third step models the *proliferation* of a lymphocyte in the natural immune system. When a lymphocyte reaches a certain level of proliferation (clone size), the lymphocyte activates and spawns a mutated clone (*somatic hyper mutation* in the fourth step). The fifth and sixth steps simulate the network theory of co-stimulation and/or suppression, and the final step the non-proliferation of other lymphocyte clones due to the proliferation of neighbouring lymphocytes. The above high level steps are grouped into four phases, namely *initialise*, *react*, *adapt* and *suppress*. Each of these phases are explained next.

5.2 Initialising an ALC and the ALC population

The ALC population, $\mathcal{B}$, in LNNAIS is initialised as an empty set. The ALC population expands to a maximum size, $\mathcal{B}_{max}$, over time. The patterns in data set, $\mathcal{A}$, that needs to be partitioned are seen as antigen patterns and are randomly presented to the ALC population. The ALCs and antigen mutated clones in LNNAIS are encoded with the same structure as the antigen patterns in $\mathcal{A}$. If patterns in the data set are real-valued (or binary) vectors then the ALCs and antigen mutated clones are also real-valued (or binary) vectors. ALCs with antigen mutated clones are used in LNNAIS to adapt to the antigen patterns to form network structures and eventually cluster

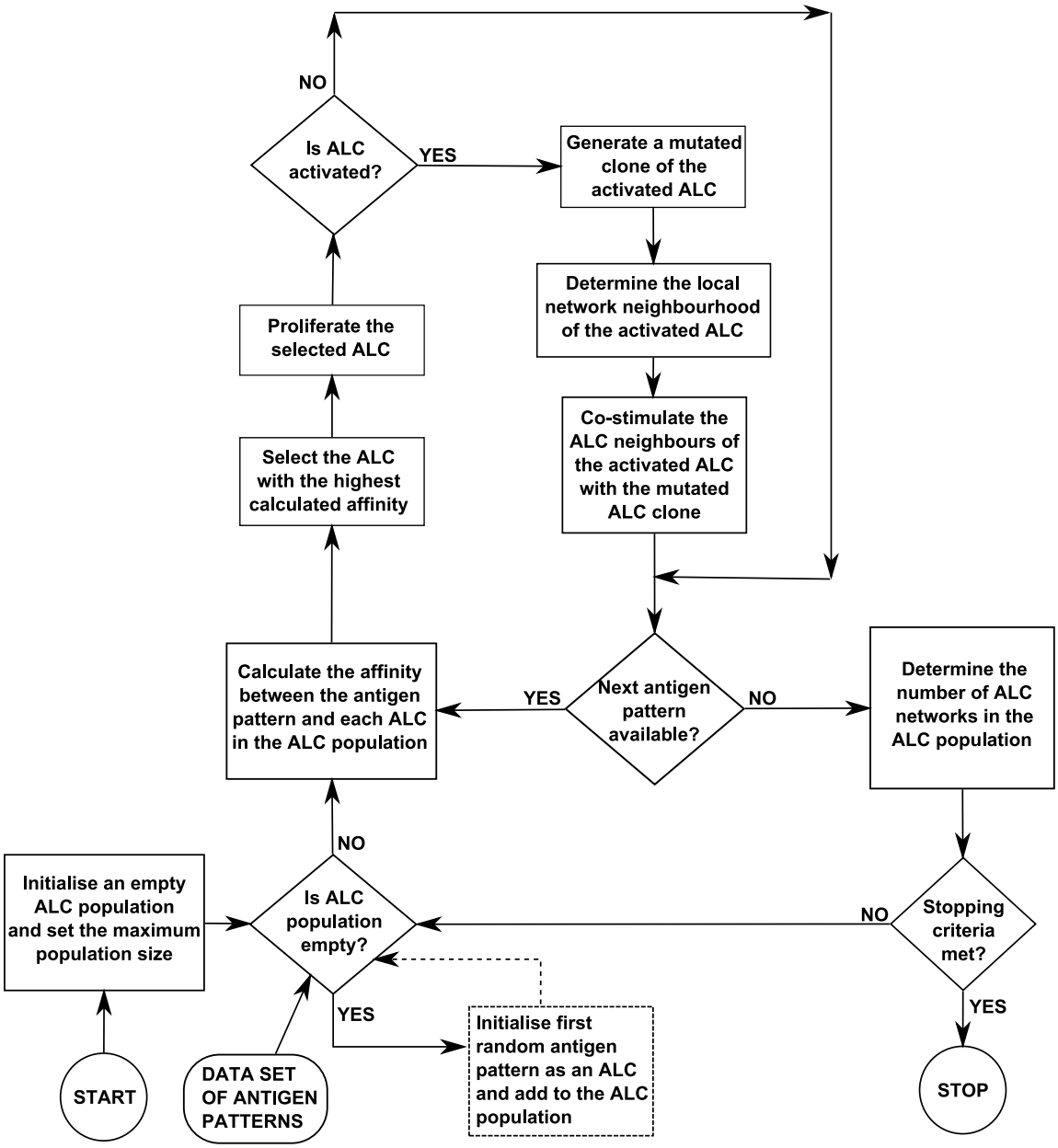


Figure 5.1 Flow chart of LNNAIS algorithm

Algorithm 5.1: High Level LNNAIS Algorithm

Set the maximum size of the ALC population as $\mathcal{B}_{max}$;
Initialise an empty set of ALCs as population $\mathcal{B}$;
for *each antigen,* $\mathbf{a}_j \in \mathcal{A}$*, at index position* j *in* $\mathcal{A}$ **do**
  if $|\mathcal{B}| = 0$ *(empty population of ALCs)* **then**
    Initialise a new ALC, $\mathbf{b}$, with the same structure as pattern $\mathbf{a}_j$;
    $\mathcal{B} = \mathcal{B} \cup \mathbf{b}$;
  end
  Calculate the antigen affinity between $\mathbf{a}_j$ and each $\mathbf{b}_i \in \mathcal{B}$ using equation (2.3);
  Select $\mathbf{b}_h \in \mathcal{B}$, at index h, as the ALC with highest calculated antigen affinity;
  Proliferate $\mathbf{b}_h$ as discussed in section 5.3.2;
  if $\mathbf{b}_h$ *is activated* ($|C_h| > \varepsilon_{clone}$) **then**
    Generate a mutated clone, $\mathbf{b}_h^{'}$, using equation (5.4);
    Secrete an antibody, $\mathbf{b}^*$, as discussed in section 5.3.4;
    Determine the local network neighbourhood of $\mathbf{b}_h$ using equation (5.5);
    Co-stimulate the local network neighbourhood of $\mathbf{b}_h$ with $\mathbf{b}^*$, as discussed in section 5.4.3;
  end
end

the data set. The initialisation of antigen mutated clones and the insertion of initialised ALCs into $\mathcal{B}$ are discussed next.

5.3 Reacting to an Antigen

The high level steps of the *react* phase are basically the steps responsible for calculating the affinity levels between the ALCs in population $\mathcal{B}$ and an antigen, selecting the ALC with the highest affinity and proliferating the selected ALC. The sections to follow explain and define each of these aspects.

5.3.1 Calculating the Affinity

The affinity between an antigen pattern, $\mathbf{a}$, and an ALC, $\mathbf{b}$, is known as the antigen affinity and is calculated as the Euclidean distance between $\mathbf{b}$ and $\mathbf{a}$. Euclidean distance is defined in equation (2.3) and is also used to measure the network affinity between two ALCs. The affinity determines the binding strength between an ALC and an antigen pattern or neighbouring ALC. Therefore, a lower Euclidean distance implies a higher affinity (stronger binding) between an

ALC and an antigen pattern or neighbouring ALC, and vice versa.

5.3.2 Proliferating the Clonal Selected ALC

The ALC with the highest binding affinity with an antigen pattern is selected as $\mathbf{b}_h$, where h is the index position of the selected ALC in $\mathcal{B}$. The antigen pattern $\mathbf{a}$ is then initialised as an antigen mutated clone $\mathbf{a}^{'}$. The antigen mutated clone $\mathbf{a}^{'}$ is grouped with $\mathbf{b}_h$ by inserting $\mathbf{a}^{'}$ at the first index position of the clonal set $\mathcal{C}_h$. Each ALC, $\mathbf{b}_i$, at index position i in $\mathcal{B}$, contains a set of antigen mutated clones, $\mathcal{C}_i$. Inserting an antigen mutated clone into $\mathcal{C}_i$ increases the clonal level of $\mathbf{b}_i$. Whenever the clonal level, $|\mathcal{C}|$, of an ALC exceeds the clonal level threshold, ε_{clone}, the ALC activates and generates a mutated ALC clone. When an antigen mutated clone is inserted at the first index of $\mathcal{C}$ and $|\mathcal{C}| > \varepsilon_{clone}$, the antigen mutated clone at the last index position $|\mathcal{C}|$, is removed from $\mathcal{C}$. This gives more current antigen mutated clones a higher probability to survive and influence the generation of the mutated ALC clone. The sections to follow discuss different definitions used to generate a mutated ALC clone.

5.3.3 Normalising the Affinity of an Antigen Mutated Clone

The normalised affinity between an antigen mutated clone, $\mathbf{a}^{'} \in \mathcal{C}_i$, and an ALC $\mathbf{b}_i$, is defined as

$$\sigma^* \left(\mathbf{b}_i, \mathbf{a}^{'}, \mathcal{C}_i \right) = 1.0 - \frac{\sigma \left(\mathbf{b}_i, \mathbf{a}^{'} \right)}{\sigma_{max} + 1.0} \tag{5.1}$$

where

$$\sigma_{max} = max_{c=1,\ldots,|\mathcal{C}_i|} \left\{ \sigma \left(\mathbf{b}_i, \mathbf{a}_c^{'} \right) \right\} \tag{5.2}$$

and $\mathbf{a}_c^{'}$ is an antigen mutated clone at index position c in clonal set $\mathcal{C}_i$ of ALC $\mathbf{b}_i$. In the above definition, σ^* calculates the normalised affinity between an antigen mutated clone, $\mathbf{a}_c^{'} \in \mathcal{C}_i$, and an ALC, $\mathbf{b}_i$, with respect to the lowest affinity (highest Euclidean distance) in the set of antigen mutated clones, $\mathcal{C}_i$. A lower affinity between an antigen mutated clone and an ALC will result in a lower normalised affinity and vice versa. Thus the higher an ALC's affinity towards an antigen mutated clone, the more the ALC's clone will be mutated towards the antigen mutated clone, as explained in the next section.

5.3.4 Generating a Mutated Clone of an Activated ALC

The vector difference between two vectors $\mathbf{q}$ and $\mathbf{r}$ is defined as:

$$\theta(\mathbf{r},\mathbf{q}) = \mathbf{q} - \mathbf{r} \tag{5.3}$$

The above function, θ, returns a vector with the same number of attributes (components) as $\mathbf{q}$. These attributes are calculated by subtracting each attribute in $\mathbf{r}$ from the corresponding attribute in $\mathbf{q}$. The set of antigen mutated clones, $\mathcal{C}_i$, which is contained by an ALC $\mathbf{b}_i$ determines the mutated clone which will be generated when an ALC is activated. The mutated clone, $\mathbf{b}_i^{'}$, is calculated using

$$\mathbf{b}_i^{'} = \mathbf{b}_i + \frac{\sum_{c=1}^{|\mathcal{C}_i|} \sigma^*\left(\mathbf{b}_i, \mathbf{a}_c^{'}, \mathcal{C}_i\right) \theta\left(\mathbf{b}_i, \mathbf{a}_c^{'}\right)}{\sum_{c=1}^{|\mathcal{C}_i|} \sigma^*\left(\mathbf{b}_i, \mathbf{a}_c^{'}, \mathcal{C}_i\right)} \tag{5.4}$$

In the above definition, $\mathbf{b}_i$ is mutated by adding a calculated average vector (second term in equation (5.4)) to $\mathbf{b}_i$. The numerator of the fraction in the second term contains the product of the normalised affinity between $\mathbf{b}_i$ and an antigen mutated clone, and the vector difference between $\mathbf{b}_i$ and the applicable antigen mutated clone. The normalised affinity between an ALC and an antigen mutated clone was discussed in section 5.3.3. The influence of the vector difference between $\mathbf{b}_i$ and an antigen mutated clone is therefore weighted by the normalised affinity. The numerator is thus calculated as the sum of weighted vector differences for all the antigen mutated clones contained by $\mathbf{b}_i$. Antigen mutated clones in $\mathcal{C}_i$ with a higher binding affinity with ALC $\mathbf{b}_i$ have a higher influence on the mutation of the clone in comparison with antigen mutated clones with a lower binding affinity. The result is that the ALC clone is mutated more towards higher affinity antigen mutated clones in $\mathcal{C}_i$. The calculated sum of weighted vector differences (numerator) is then divided by the sum of the normalised affinities to obtain an average vector for mutating $\mathbf{b}_i$.

5.3.5 Secreting an Antibody for Co-stimulation

The antigen mutated clones in $\mathcal{C}_i$ with which $\mathbf{b}_i^{'}$ has a higher affinity than the parent ALC $\mathbf{b}_i$, is added to the clonal set of $\mathbf{b}_i^{'}$ (bind to $\mathbf{b}_i^{'}$). If more than half of the number of antigen mutated clones in $\mathcal{C}_i$ bind to $\mathbf{b}_i^{'}$, the parent ALC $\mathbf{b}_i$ is added as an antigen mutated clone to the clonal set of $\mathbf{b}_i^{'}$. The parent ALC is then replaced by $\mathbf{b}_i^{'}$ in $\mathcal{B}$ and secreted as a co-stimulating antibody to neighbouring ALCs. If less than half of the number of antigen mutated clones in $\mathcal{C}_i$ bind to $\mathbf{b}_i^{'}$, the parent ALC $\mathbf{b}_i$ is suppressed by removing all of the antigen mutated clones in $\mathcal{C}_i$. This

prevents frequently activated ALCs from dominating the population. The mutated ALC clone, $\mathbf{b}_i'$, is then inserted into $\mathcal{C}_i$; not only to co-stimulate the parent ALC, but also to preserve the memory of the antigen structure. The mutated ALC clone is secreted as a co-stimulating antibody to neighbouring ALCs. The following section discusses the co-stimulation of neighbouring ALCs within a local network neighbourhood.

5.4 Adapting the ALCs in a Local Network Neighbourhood

The co-stimulating antibody which is secreted during the activation of a proliferated ALC is presented to the immediate ALC neighbour(s) in the local network neighbourhood of the activated ALC. The neighbouring ALCs within a local network neighbourhood adapt to the antibody as it would react to an antigen (as explained in section 5.3). The following sections discuss the manner in which a local network neighbourhood of an activated ALC is determined.

5.4.1 Determining the Local Network Neighbourhood of an Activated ALC

An ALC's neighbourhood, $\mathcal{N}$, is determined by a network neighbourhood window of size, ρ, and the highest average network affinity between the potential neighbouring ALCs. The neighbourhood, $\mathcal{N}_{i,\rho}$, of an ALC, $\mathbf{b}_i \in \mathcal{B}$, is defined as

$$\mathcal{N}_{i,\rho} = \left\{ \forall \mathbf{b}_j \in \mathcal{B} : \min_{j=i-(\rho-1),\ldots,i} \{\mu(j, j+(\rho-1))\} \right\} \tag{5.5}$$

where

$$\rho \leq |\mathcal{B}| \tag{5.6}$$

$$\mathcal{N}_{i,\rho} \subseteq \mathcal{B} \tag{5.7}$$

$$\mathbf{b}_i \in \mathcal{N}_{i,\rho} \tag{5.8}$$

and μ calculates the average network affinity between ALCs in the population from index position i to $i+(\rho-1)$; μ is defined in section 5.4.2. The above definition is a network window of size ρ which starts at position $i-(\rho-1)$, sliding over the ALC population in search of the highest average network affinity (minimum average distance). Figure 5.2 illustrates a local network neighbourhood where $\rho = 5$ and the network with the highest average network affinity starts at index position $h-2$.

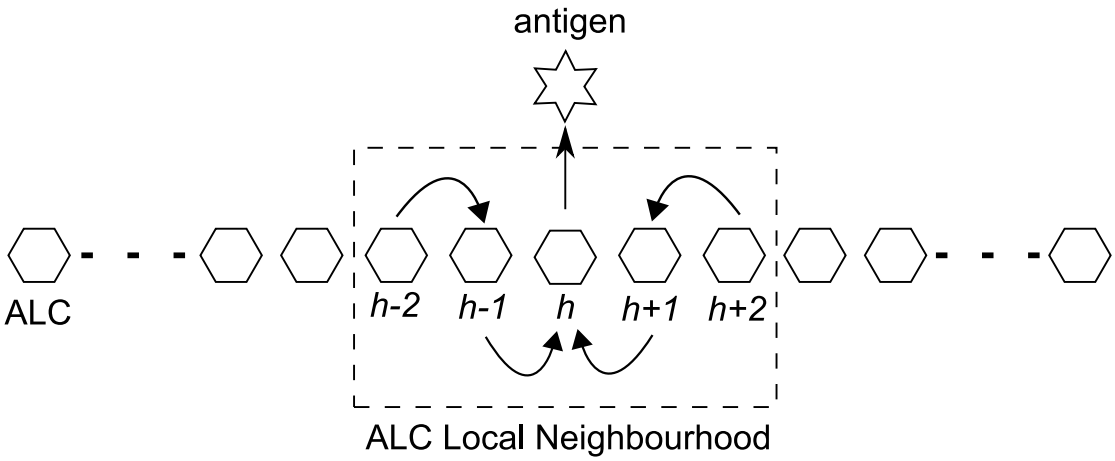


Figure 5.2 Adapting an ALC Network Neighbourhood

5.4.2 Average Network Affinity in a Local Network Neighbourhood

The average network affinity level of a network of ALCs starting at index position x to y, is defined as

$$\mu(x,y) = \frac{\sum_{i=x}^{y-1} \sigma(\mathbf{b}_i, \mathbf{b}_{i+1})}{y-x} \tag{5.9}$$

where σ is the Euclidean distance (as defined in equation (2.3)).

5.4.3 Co-stimulating the Local Network Neighbourhood

The neighbouring ALCs within a local network neighbourhood, $\mathcal{N}_{i,\rho}$, adapt to the secreted antibody of its predecessor in the neighbourhood. Figure 5.2 illustrates a local network neighbourhood with $\rho = 5$ adapting to an antigen. In this figure, ALC $\mathbf{b}_h$ is selected by the antigen for cloning and proliferation (as explained in section 5.3.2). As a result of proliferating $\mathbf{b}_h$, the ALC became active ($|\mathcal{C}_h| > \varepsilon_{clone}$) and secreted an antibody for co-stimulation of the immediate neighbours of $\mathbf{b}_h$. The immediate neighbours of $\mathbf{b}_h$ at indices $h-1$ and $h+1$ react to the secreted antibody by adding the clonal set of the antibody to $\mathcal{C}_{h-1}$ and $\mathcal{C}_{h+1}$, respectively. If either or both of the neighbouring ALCs, $\mathbf{b}_{h-1}$ and $\mathbf{b}_{h+1}$ becomes activated, either or both will secrete antibodies (as explained in section 5.3.4), which will co-stimulate their immediate ALC neighbours at indices $h-2$ and $h+2$, respectively. If a neighbouring ALC is not activated by the co-stimulation of a predecessor's antibody, the antibody is inserted into the local network at the index of the neighbouring ALC, increasing the population size through *clonal expansion* (discussed in section 5.4.4). The neighbouring ALCs with the highest network affinity in the

population, which are not within the local network neighbourhood, are merged to stabilise the population size. Merging of ALCs simulate the non-proliferation of other ALC clones in the population (discussed in section 5.4.5). The process of co-stimulation continues until the ALCs on the boundary of the local network neighbourhood are co-stimulated or until a neighbouring ALC is not activated by the co-stimulation of a predecessor's antibody. Algorithm 5.2 lists the pseudo code for adapting the ALCs in a local network neighbourhood.

5.4.4 Clonal Expansion of a Local Network Neighbourhood

A local network neighbourhood is clonally expanded whenever a neighbouring ALC, $\mathbf{b}_i$, is not activated by the co-stimulation of a predecessor's secreted antibody. The secreted antibody, $\mathbf{b}^*$, is inserted at position i^* which is defined as

$$i^*(\mathbf{b}^*, \mathbf{b}_i) = \begin{cases} i & \text{if } \frac{\sigma(\mathbf{b}^*, \mathbf{b}_{i-1}) + \sigma(\mathbf{b}^*, \mathbf{b}_i)}{2} < \frac{\sigma(\mathbf{b}^*, \mathbf{b}_i) + \sigma(\mathbf{b}^*, \mathbf{b}_{i+1})}{2} \\ i+1 & \text{otherwise} \end{cases} \tag{5.10}$$

The secreted antibody is inserted at the index position where the average network affinity is the highest between the secreted antibody and its potential neighbouring ALCs.

5.4.5 Non-proliferation of the ALC Population

The maximum ALC population size, $\mathcal{B}_{max}$, is exceeded whenever clonal expansion occurs in a local network neighbourhood. Therefore, the non-proliferation and suppression of other ALCs in the population keeps the size of the ALC population stable. Non-proliferation (suppression) is simulated by merging two ALCs in the population which are not within the clonally expanded local network neighbourhood, and which have the highest network affinity in the population.

5.5 Similarities and Differences with Other Network based AIS Models

This section discusses some of the differences and similarities between the proposed algorithm and existing network based AIS models.

Algorithm 5.2: Adapting the Neighbourhood, $\mathcal{N}_{h,\rho}$, to an Activated ALC, $\mathbf{b}_h$

Let $\mathbf{b}^*$ be the secreted antibody of the activated ALC $\mathbf{b}_h$;
$l = h-1; r = h+1$;
Let $\mathbf{b}_l^* = \mathbf{b}^*$ and $\mathbf{b}_r^* = \mathbf{b}^*$ be the secreted antibodies for co-stimulation of neighbouring ALCs $\mathbf{b}_l$ and $\mathbf{b}_r$, respectively;
Activated=true;
Costimulated=false;
for $\mathbf{b}_l \in \mathcal{N}_{h,\rho}$ *and Activated* **do**
  Add antigen mutated clones of $\mathbf{b}_l^*$ to clonal set $\mathcal{C}_l$ of neighbouring ALC $\mathbf{b}_l$;
  if $\mathbf{b}_l$ *is activated (i.e.* $|\mathcal{C}_l| > \varepsilon_{clone}$*)* **then**
    Generate a mutated clone, $\mathbf{b}_l^{'}$, using equation (5.4);
    Secrete an antibody $\mathbf{b}_l^*$ from $\mathbf{b}_l$, as discussed in section 5.3.4;
    $l = l-1$;
    Costimulated=true;
  end
  else
    Activated=false;
    Insert $\mathbf{b}_l^*$ into $\mathcal{N}_{h,\rho}$ at position $i^*\left(\mathbf{b}_l^*, \mathbf{b}_l\right)$ (as defined in equation (5.10));
    Merge two ALCs in the population with the highest network affinity, as discussed in section 5.4.5;
  end
end
Activated=true;
for $\mathbf{b}_r \in \mathcal{N}_{h,\rho}$ *and Activated* **do**
  Add antigen mutated clones of $\mathbf{b}_r^*$ to clonal set $\mathcal{C}_r$ of neighbouring ALC $\mathbf{b}_r$;
  if $\mathbf{b}_r$ *is activated (i.e.* $|\mathcal{C}_r| > \varepsilon_{clone}$*)* **then**
    Generate a mutated clone, $\mathbf{b}_r^{'}$, using equation (5.4);
    Secrete an antibody $\mathbf{b}_r^*$ from $\mathbf{b}_r$, as discussed in section 5.3.4;
    $r = r+1$;
    Costimulated=true;
  end
  else
    Activated=false;
    Insert $\mathbf{b}_r^*$ into $\mathcal{N}_{h,\rho}$ at position $i^*\left(\mathbf{b}_r^*, \mathbf{b}_r\right)$ (as defined in equation (5.10));
    Merge two ALCs in the population with the highest network affinity, as discussed in section 5.4.5;
  end
end
if *not Costimulated and* $|\mathcal{B}| < \mathcal{B}_{max}$ **then**
  Insert $\mathbf{b}^*$ into $\mathcal{N}_{h,\rho}$ at position $i^*\left(\mathbf{b}^*, \mathbf{b}_h\right)$ (as defined in equation (5.10));
end

5.5.1 Training Data

Although the proposed LNNAIS model can be trained on normalised data, the normalisation of training data is not a prerequisite for LNNAIS. Similar to other network based AIS models, LNNAIS sees all training patterns as antigen patterns.

5.5.2 Population of ALCs

The population of ALCs can be initialised with a number of randomly initialised ALCs or a number of randomly selected training patterns as ALCs, i.e. a cross section of the training data is used to initialise the ALCs. The initial population of ALCs in LNNAIS is an empty set. The first randomly selected training pattern is initialised as an ALC and added to the population of ALCs. This concept is known as *dendritic injection* in the natural immune system. The population of ALCs are grown and pruned in LNNAIS. The *growth* of the population of ALCs in LNNAIS is based on the process of *affinity maturation*. When an activated ALC of a local network neighbourhood does not adapt to the presented antigen pattern, the clonal level of the ALC is penalised and a mutated clone of the ALC is inserted into the local network of ALCs.

5.5.3 ALC Presentation

An ALC in LNNAIS is presented by a continuous-valued array with the same dimension as the antigen patterns in the training set, as is the case for other network based AIS models.

5.5.4 Affinity Measurement

The affinity between an antigen pattern and an ALC is measured using the Euclidean distance as defined in section 5.3.1. The affinity between two ALCs, referred to as network affinity, is also measured using the Euclidean distance. Some of the existing network based AIS models also measure antigen and network affinity using Euclidean distance. The difference between LNNAIS and the existing network based AIS models is that LNNAIS has no threshold to determine whether two ALCs are linked to form a network. LNNAIS introduces a new concept of an ALC network neighbourhood size, as defined in section 5.4.1 and proposed by Graaff and Engelbrecht [64].

5.5.5 Learning the Antigen Structure

Another similarity between existing network based AIS models and the proposed LNNAIS is that some ALCs are cloned and mutated to adapt to antigen patterns. LNNAIS also models the process of *affinity maturation* to introduce new ALCs into the population as discussed in section 5.4.3. LNNAIS also models the non-proliferation of ALCs, as discussed in section 5.4.3. The difference between LNNAIS and existing network based AIS models is that *expansion* of the ALC population is done on a per local network neighbourhood bases. LNNAIS models the *idiotopic network theory* of ALCs. This means that the insertion of new ALCs into a population will be done within a local network neighbourhood (as discussed in section 5.4.3). Non-proliferation on the other hand is only done on ALCs which do not form part of the *activated* local network neighbourhood. This means that only ALCs outside a network neighbourhood will be non-proliferated in the ALC population (as discussed in section 5.4.3). This approach penalises the population of ALCs by *non-proliferating* the population but also reinforces the network neighbourhood by *clonal expansion*.

5.5.6 Determining the Number of Clusters

The number of ALC networks formed in existing network based AIS models represent potential clusters in the data set. In most of the existing network based AIS models the number of ALC networks in a population is determined by a network affinity threshold or a hybrid approach is taken by clustering the ALC population into sub-nets (as discussed in section 4.6). The thresholding technique uses a proximity matrix of network affinities between the ALCs in the population. The ALCs with a network affinity below the threshold value are allowed to be linked and form networks. Therefore the specified value of the network affinity threshold determines the number of ALC networks and it can be a formidable task to specify the correct network affinity threshold to obtain the correct or required number of clusters. A potential drawback of the hybrid approach is that the clusters (sub-nets) might contain ALCs which do not have a *good* or generic representation of the data. Both of these techniques are also computationally expensive.

The proposed LNNAIS model has the advantage that an ALC can only link to its immediate neighbours to form an ALC network. This is due to the network topology and an index based neighbourhood technique. Therefore, there is no need for a network affinity threshold and/or a proximity matrix of network affinities to determine the number of ALC networks in LNNAIS. It is also not necessary to follow a hybrid approach of clustering the ALC population. Determining

the number of clusters in LNNAIS is thus computationally less expensive and is explained next.

In order to obtain a specified number of clusters, K, the network affinities between neighbouring ALCs in the population need to be calculated. The boundaries of each cluster are then determined by pruning the network links between the K lowest calculated network affinities. Figure 5.3 illustrates this technique where $K = 3$. The edges between ALCs have an associated network affinity. The K edges that forms the boundaries between the ALCs (dotted lines) have the lowest network affinity in the ALC population, i.e. highest Euclidean distance. The centroid of each of the formed ALC networks (illustrated as clouds) is calculated using equation (2.18).

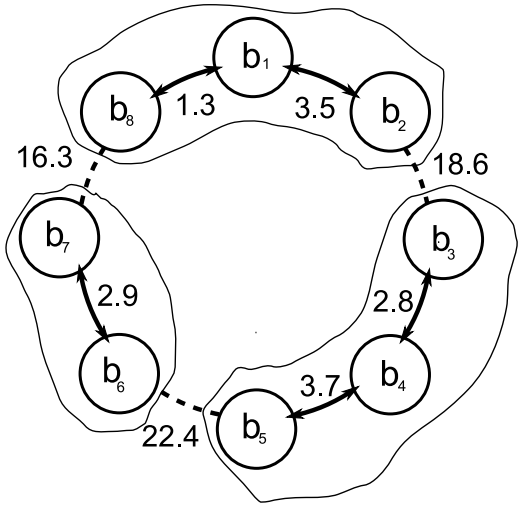


Figure 5.3 Determining the Number of Clusters in LNNAIS

5.5.7 The Number of Parameters

Focusing on existing network based AIS models which are used in the experimental work of this chapter, there is also a significant difference in the number of parameters that need to be specified for each of the models. The DWB model has a total of 12 parameters, SMAIN has a total of seven parameters and Opt-aiNet a total of six parameters. The proposed LNNAIS model has only three parameters which are the maximum population size, $\mathcal{B}_{max}$, the neighbouring radius, ρ, and the activation level for ALC cloning, ε_{clone}.

5.6 Time Complexity of LNNAIS

The time complexity of LNNAIS is based on the complexity of partitioning $\mathcal{A}$, sorting the network affinities between the ALCs in the ALC population and pruning K boundaries between the ALCs in the ALC population of size $\mathcal{B}_{max}$ to obtain K ALC networks (clusters). The time complexity of partitioning $\mathcal{A}$ is based on presenting $\mathcal{A}$ to ALC population $\mathcal{B}$ and adapting the ALC population. Assume that t_1 is the number of iterations taken by LNNAIS to converge. The worst case of time complexity for LNNAIS to partition $\mathcal{A}$ is when there is always an activated ALC in $\mathcal{B}$ and when the network neighbourhood size of the activated ALC is the entire ALC population ($\mathcal{N} = \mathcal{B}$). Then the time complexity of partitioning $\mathcal{A}$ is $O\left(t_1 |\mathcal{A}| (\mathcal{B}_{max})^2 N \chi_1\right)$ where $|\mathcal{A}|$ is the size of the data set that needs to be partitioned and N is the number of dimensions of $\mathcal{A}$. The χ_1 parameter is the time complexity for an activated ALC to generate an antibody for co-stimulation of neighbouring ALCs. The t_1, $\mathcal{B}_{max}$, N and χ_1 parameters are fixed in advance and usually $\mathcal{B}_{max} << |\mathcal{A}|$ and $\left|\mathcal{N}\right| << \mathcal{B}_{max}$. If $t_1 \mathcal{B}_{max} N \chi_1 << |\mathcal{A}|$ then the time complexity of partitioning $\mathcal{A}$ is $O(|\mathcal{A}|)$. If however, $\mathcal{B}_{max} \approx |\mathcal{A}|$ and $\left|\mathcal{N}\right| \approx \mathcal{B}_{max}$ then the time complexity of partitioning $\mathcal{A}$ is $O\left(|\mathcal{A}|^2\right)$. The maximum number of boundaries in an ALC population of size $\mathcal{B}_{max}$ is $\mathcal{B}_{max}$. The time complexity of sorting the $\mathcal{B}_{max}$ network affinities depends on the sorting algorithm used. Assume the time complexity of the sorting algorithm is some constant, χ_2. The worst case of time complexity for LNNAIS to determine K ALC networks is when $K = \mathcal{B}_{max}$, giving a time complexity of $O(\mathcal{B}_{max})$.

5.7 Experimental Results and Analysis

This section discusses and compares the clustering results obtained by K-means, CPSO, SMAIN, DWB, Opt-aiNet and LNNAIS. Furthermore, a sensitivity analysis of LNNAIS is done on the different data sets.

5.7.1 Data clustering problems

Table 5.1 lists the selection of data sets used to benchmark the clustering performance and quality of the proposed LNNAIS model against the clustering quality of existing clustering methods like K-means clustering and CPSO (as discussed in sections 2.3.2 and 2.7.1, respectively) and network based AIS models for data clustering like SMAIN, DWB-AIS and Opt-aiNet (as discussed in section 4.6). The characteristics of each data set are also listed in the table. These are the number of patterns in the dataset ($|P|$), the number of features per pattern in the data set (N

Table 5.1 List of Eleven Benchmarking Data Sets for Clustering

Category	Data set name	$\|P\|$	N	σ_{max}	K	Overlap?
Group 1	Iris	150	4	7.7	3	Y
	Two-spiral	190	2	3.045	12	Y
	Hepta	212	3	13.383	7	N
Group 2	Engytime	4096	2	14.806	2	Y
	Chainlink	1000	3	4.383	6	Y
	Target	770	2	8.627	5	Y (outliers)
Group 3	Ionosphere	351	34	11.358	2	Y
	Glass	214	9	16.449	6	Y
Group 4	Image Segmentation	2310	19	1775.117	7	Y
	Spambase	4601	57	18758.75	2	Y
	Letter Recognition	20000	16	60	26	Y

- number of dimensions), the maximum distance between the patterns in the data set (σ_{max}), the number of clusters selected for partitioning the data set (K) and whether there are any overlapping patterns in the data set. The two-spiral, hepta, engytime, chainlink and target data sets are part of a fundamental clustering problems suite [95]. The other data sets were collected from the UCI Machine Learning repository [6].

The data sets in table 5.1 can be categorised into four groups:

- Group 1 (small number of features / small number of patterns): The data sets within this group have a small number of features and a small number of patterns. The iris data set, two-spiral problem and hepta data set form part of this group. All of these data sets have less than 500 patterns and less than five features per pattern.

- Group 2 (small number of features / large number of patterns): The data sets within this group also have a small number of features but a larger number of patterns in comparison to the data sets in group 1. The engytime data set, chainlink data set and the target data set (to a lesser extent) form part of this group. All of these data sets have more than 500 patterns and less than five features per pattern.

- Group 3 (large number of features / small number of patterns): This group contains data sets with a larger number of features in comparison to groups 1 and 2, but a small number of patterns. The ionosphere data set and the glass data set form part of this group and both have less than 500 patterns, with each pattern having more than eight features.

- Group 4 (large number of features / large number of patterns): The last group contains data sets with a larger number of features (compared to groups 1 and 2) and a larger number of patterns (compared to groups 1 and 3). The image segmentation data set, spambase data set and letter recognition data set form part of this group. All of these data sets have more than 500 patterns and more than eight features.

Taken as a whole, the data sets listed in table 5.1 represent a good distribution of data clustering problems with the number of patterns in the range $[150, 20000]$ and the number of features in the range $[2, 57]$. All the data sets have overlapping patterns except the hepta data set. The target data set also contains outlier patterns.

5.7.2 Experimental setup and methodology

All experimental results in this chapter are averages taken over 50 runs, unless stated otherwise. The stopping criteria for all algorithms was set to 1000 iterations ($t_{max} = 1000$). Populations/Swarms in the respective algorithms were initialised by randomly selecting patterns from the data set. The patterns in a data set were randomly presented to each model. None of the data sets were normalised for training. All algorithms were implemented using the Java 6 framework which interfaced to a MySQL 5 database for collection of data sets and exporting of results. Algorithms were executed on a 24 core Sun Grid Engine. Tables 5.2 to 5.6 summarise the parameter values used by the respective algorithms for each data set. All parameter values for the respective algorithms were found empirically to deliver the best performance for clustering the applicable data set. The Q_{ratio} validity index (defined in equation (2.49)), intra error distance (J_{intra} as defined in equation (2.17)) and inter error distance (J_{inter} as defined in equation (2.16)) are used as performance measures to determine the clustering quality of the different models. These clustering performance measures were discussed in sections 2.3.2 and 2.4, respectively.

The following sections investigate whether there is a difference between the clustering quality, Q_{ratio}, of two models for a specific data set or not. The hypothesis is defined as

- *Null* hypothesis, H_0: There is no difference in Q_{ratio}.
- *Alternative* hypothesis, H_1: There is a difference in Q_{ratio}.

The above hypothesis was tested with a non-parametric Mann-Whitney U hypothesis test (0.95 confidence interval, i.e. $\alpha = 0.05$) between the clustering quality of LNNAIS and the clustering

Table 5.2 CPSO Parameter Values

Data set	K	$\|S\|$	d	w	c_1	c_2	δ
Iris	3	6	3	0.82	1.33	1.218	0.301
Two-spiral	12	9	4	0.558	0.656	1.94	0.326
Hepta	7	44	4	0.697	1.696	0.963	0.62
Engytime	2	63	11	0.641	0.719	0.156	0.359
Chainlink	6	11	5	0.234	0.656	1.969	0.266
Target	5	23	2	0.789	0.422	1.658	0.258
Ionosphere	2	45	8	0.683	1.518	1.207	0.961
Glass	6	13	6	0.914	1.344	1.246	0.115
Image Segmentation	7	10	5	0.77	0.875	1.545	0.312
Spambase	2	42	18	0.812	0.125	1.152	0.938
Letter Recognition	26	49	3	0.836	0.828	1.641	0.055

Table 5.3 SMAIN Parameter Values

Data set	K	$\mathcal{B}_{init}$	R_γ	R_Λ	NAT	R_k	R_{max}	R_{init}
Iris	3	0.25	0.836	3	1.115	0.422	238	37
Two-spiral	12	0.182	0.516	91	0.039	0.656	975	92
Hepta	7	0.191	0.938	38	0.259	0.375	900	88
Engytime	2	0.019	0.672	35	2.322	0.469	725	36
Chainlink	6	0.2	0.859	23	0.038	0.094	425	91
Target	5	0.049	0.824	22	0.077	0.852	819	31
Ionosphere	2	0.157	0.637	34	0.099	0.727	319	68
Glass	6	0.157	0.637	34	0.015	0.727	319	68
Image Segmentation	7	0.29	0.926	2	24.618	0.898	281	76
Spambase	2	0.123	0.805	33	6.571	0.359	388	43
Letter Recognition	26	0.051	0.93	24	8.595	0.109	988	68

Table 5.4 DWB Parameter Values

Data set	K	$\mathcal{B}_{max}$	ϕ_{init}	m_{min}	A	a_{min}	a_{max}	k_{clone}	ς	τ	τ_α	τ_β	$k_{compress}$
Iris	3	39	0.362	0.087	37	49	49	1.688	0.24	14	5	10	6
Two-spiral	12	46	0.668	0.025	55	6	6	2.438	0.913	12	13	13	5
Hepta	7	47	0.959	0.959	78	35	54	1.625	0.592	1	6	15	2
Engytime	2	39	0.485	0.209	24	24	86	3.188	0.852	6	6	1	4
Chainlink	6	40	0.592	0.102	41	72	91	2.125	0.714	7	11	2	2
Target	5	47	0.554	0.982	36	62	62	3.844	0.89	13	12	11	3
Ionosphere	2	17	0.561	0.929	44	7	68	1.25	0.929	5	7	9	3
Glass	6	46	0.845	0.018	13	11	11	4.031	0.569	2	5	9	7
Image Segmentation	7	47	0.201	0.538	16	2	2	2.906	0.477	12	14	7	1
Spambase	2	46	0.27	0.546	71	9	9	4.562	0.025	3	8	4	4
Letter Recognition	26	45	0.148	0.423	9	27	46	3.062	0.148	8	10	13	3

Table 5.5 Opt-aiNET Parameter Values

Data set	K	$\mathcal{B}_{init}$	η	$\varepsilon_{network}$	$\varepsilon_{fitness}$	φ	$\frac{1}{\zeta}$
Iris	3	44	10	0.186	1.317	0.131	0.356
Two-spiral	12	14	1	0.324	0.902	0.219	0.169
Hepta	7	39	1	0.297	1.54	0.491	0.459
Engytime	2	29	2	0.037	0.412	0.403	0.322
Chainlink	6	7	22	0.178	0.723	0.306	0.283
Target	5	12	3	0.362	1.109	0.338	0.412
Ionosphere	2	28	3	0.477	1.97	0.294	0.144
Glass	6	35	1	0.155	0.961	0.456	0.431
Image Segmentation	7	14	5	0.021	1.184	0.316	0.134
Spambase	2	6	5	0.32	1.985	0.409	0.191
Letter Recognition	26	45	2	0.416	1.258	0.444	0.394

Table 5.6 LNNAIS Parameter Values

Data set	K	$\mathcal{B}_{max}$	ρ	ε_{clone}
Iris	3	14	3	8
Two-spiral	12	39	3	6
Hepta	7	29	3	6
Engytime	2	10	3	22
Chainlink	6	24	3	8
Target	5	28	3	6
Ionosphere	2	10	3	17
Glass	6	24	3	8
Image Segmentation	7	20	2	27
Spambase	2	10	5	22
Letter Recognition	26	104	3	10

quality of each of the other models. The result is statistical significant if the calculated probability (p-value is the probability of H_0 being true) is less than α. The results for each data set group are discussed next.

5.7.3 Testing for statistical significance - data group 1

Table 5.7 summarises the results obtained for data group 1 using the applicable parameter values in tables 5.2-5.6 for each of the data sets. The corresponding statistical hypothesis tests between LNNAIS and the remaining models for each of the data sets in group 1 are summarised in table 5.8 (based on the clustering quality, Q_{ratio}). The Mann-Whitney U statistical hypothesis test accepts H_0 that the means are the same at a 0.05 level of significance between LNNAIS and Opt-aiNet and between LNNAIS and CPSO for data set hepta. The remainder of the Mann-Whitney U statistical hypothesis tests showed a significant difference in performance between LNNAIS and the other clustering algorithms. LNNAIS tends to deliver clusters of a higher quality when compared to K-means, CPSO, DWB and Opt-aiNet for data sets iris and hepta. Although SMAIN tends to deliver clusters of a higher quality when compared to LNNAIS for all data sets in group 1, LNNAIS delivers more compact clusters for the iris data set. Also, K-means tends to deliver clusters of a higher quality for data set two-spiral (refer to table 5.7). SMAIN tends to find clusters in the data sets of group 1 with a higher quality, followed by LNNAIS.

5.7.4 Testing for statistical significance - data group 2

The results obtained for data group 2 with the applicable parameter values in tables 5.2-5.6 are summarised in table 5.9. The Mann-Whitney U statistical hypothesis test accepts H_0 that the mean clustering quality, Q_{ratio}, are the same between LNNAIS and DWB for data set chainlink; and rejects H_0 for all other cases (as summarised in table 5.10). Referring to table 5.9, LNNAIS tends to deliver clusters of a higher quality when compared to CPSO, DWB and Opt-aiNet for all data sets in group 2. K-means tends to deliver clusters of a higher quality when compared to LNNAIS for data sets chainlink and target, but of lower quality for data set engytime. SMAIN also tends to deliver clusters of a higher quality for all data sets in group 2, followed by LNNAIS.

5.7.5 Testing for statistical significance - data group 3

The results of the Mann-Whitney U statistical hypothesis test accepts H_0 that the mean clustering quality, Q_{ratio}, are the same between LNNAIS and DWB, and LNNAIS and CPSO for data set

Table 5.7 Descriptive Statistics: Data Group 1

Data set	Algorithm	J_{intra}	J_{inter}	Q_{ratio}
Iris	K-means	0.689 (± 0.073)	3.269 (± 0.201)	0.509 (± 0.268)
	CPSO	0.725 (± 0.089)	2.964 (± 0.201)	0.658 (± 0.354)
	SMAIN	0.766 (± 0.041)	3.705 (± 0.207)	0.295 (± 0.021)
	DWB	0.753 (± 0.152)	3.103 (± 0.282)	0.547 (± 0.304)
	Opt-aiNet	0.887 (± 0.021)	2.977 (± 0.095)	0.882 (± 0.168)
	LNNAIS	0.738 (± 0.054)	3.546 (± 0.309)	0.333 (± 0.048)
Two-spiral	K-means	0.212 (± 0.005)	1.014 (± 0.021)	0.521 (± 0.102)
	CPSO	0.251 (± 0.025)	0.829 (± 0.079)	1.648 (± 0.978)
	SMAIN	0.213 (± 0.004)	1.096 (± 0.013)	0.433 (± 0.015)
	DWB	0.241 (± 0.010)	0.988 (± 0.065)	1.094 (± 0.501)
	Opt-aiNet	0.279 (± 0.027)	0.813 (± 0.105)	2.740 (± 3.020)
	LNNAIS	0.233 (± 0.009)	1.030 (± 0.041)	0.847 (± 0.296)
Hepta	K-means	0.976 (± 0.232)	4.041 (± 0.147)	0.999 (± 0.465)
	CPSO	0.893 (± 0.355)	3.930 (± 0.344)	1.095 (± 1.748)
	SMAIN	0.641 (± 0.001)	4.147 (± 0.005)	0.219 (± 0.001)
	DWB	1.187 (± 0.260)	3.990 (± 0.238)	1.254 (± 0.618)
	Opt-aiNet	1.179 (± 0.462)	3.681 (± 0.499)	1.643 (± 1.353)
	LNNAIS	0.748 (± 0.102)	4.140 (± 0.099)	0.345 (± 0.206)

Table 5.8 Statistical Hypothesis Testing between LNNAIS and Other Models based on Q_{ratio}: Data Group 1 ($\alpha = 0.05$; with continuity correction; unpaired; non-directional)

Data set	Algorithm	z	p	Outcome
Iris	K-means	4.539	< 0.001	Reject H_0
	CPSO	5.958	< 0.001	Reject H_0
	DWB	5.115	< 0.001	Reject H_0
	SMAIN	3.726	< 0.001	Reject H_0
	Opt-aiNet	6.646	< 0.001	Reject H_0
Two-spiral	K-means	5.773	< 0.001	Reject H_0
	CPSO	4.361	< 0.001	Reject H_0
	DWB	2.21	0.027	Reject H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	6.246	< 0.001	Reject H_0
Hepta	K-means	3.726	< 0.001	Reject H_0
	CPSO	1.331	0.183	Accept H_0
	DWB	5.892	< 0.001	Reject H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	1.804	0.071	Accept H_0

ionosphere, and rejects H_0 for all other cases (as summarised in table 5.11). LNNAIS tends to deliver clusters of a higher quality for all data sets in group 3 when compared to K-means, CPSO and DWB (refer to table 5.12). However, SMAIN and Opt-aiNet tend to deliver clusters of a higher quality for data set ionosphere when compared to cluster quality of LNNAIS. SMAIN also tend to deliver clusters of a higher quality for the data sets in group 3, followed by LNNAIS. LNNAIS does however deliver more compact clusters than SMAIN for the glass data set.

5.7.6 Testing for statistical significance - data group 4

Table 5.13 summarises the results obtained for data group 4. The corresponding statistical hypothesis tests between LNNAIS and the remaining models for each of the data sets in group 4 are summarised in table 5.14. The Mann-Whitney U statistical hypothesis test accepts H_0 that the means are the same between LNNAIS and K-means for data set image segmentation, and between LNNAIS and Opt-aiNet for data set letter recognition. The Mann-Whitney U statistical hypothesis test rejects H_0 for all other cases (as summarised in table 5.14). In most cases LNNAIS tends to deliver clusters of a higher quality except for data set image segmentation and letter recognition where SMAIN tends to deliver clusters of a higher quality (refer to table 5.13).

Table 5.9 Descriptive Statistics: Data Group 2

Data set	Algorithm	J_{intra}	J_{inter}	Q_{ratio}
Engytime	K-means	1.431 (± 0.000)	2.998 (± 0.000)	0.477 (± 0.000)
	CPSO	1.435 (± 0.001)	2.935 (± 0.012)	0.489 (± 0.002)
	SMAIN	2.097 (± 0.103)	5.975 (± 0.670)	0.355 (± 0.039)
	DWB	1.599 (± 0.120)	3.057 (± 0.526)	0.540 (± 0.115)
	Opt-aiNet	1.435 (± 0.001)	2.932 (± 0.025)	0.490 (± 0.004)
	LNNAIS	1.944 (± 0.281)	4.557 (± 1.043)	0.438 (± 0.069)
Chainlink	K-means	0.488 (± 0.006)	1.550 (± 0.049)	0.517 (± 0.031)
	CPSO	0.592 (± 0.053)	1.412 (± 0.150)	1.092 (± 0.667)
	SMAIN	0.487 (± 0.007)	1.643 (± 0.039)	0.471 (± 0.023)
	DWB	0.538 (± 0.025)	1.506 (± 0.074)	0.751 (± 0.320)
	Opt-aiNet	0.646 (± 0.059)	1.363 (± 0.185)	1.352 (± 0.554)
	LNNAIS	0.535 (± 0.021)	1.493 (± 0.116)	0.640 (± 0.118)
Target	K-means	0.544 (± 0.030)	2.393 (± 0.244)	0.337 (± 0.032)
	CPSO	0.749 (± 0.077)	2.340 (± 0.556)	1.133 (± 0.578)
	SMAIN	1.008 (± 0.000)	5.794 (± 0.000)	0.238 (± 0.001)
	DWB	0.649 (± 0.059)	2.058 (± 0.319)	0.752 (± 0.285)
	Opt-aiNet	0.792 (± 0.050)	2.706 (± 0.494)	1.750 (± 1.491)
	LNNAIS	0.804 (± 0.124)	2.985 (± 0.525)	0.559 (± 0.155)

Table 5.10 Statistical Hypothesis Testing between LNNAIS and Other Models based on Q_{ratio}: Data Group 2 ($\alpha = 0.05$; with continuity correction; unpaired; non-directional)

Data set	Algorithm	z	p	Outcome
Engytime	K-means	3.097	0.002	Reject H_0
	CPSO	3.4	< 0.001	Reject H_0
	DWB	3.888	< 0.001	Reject H_0
	SMAIN	4.931	< 0.001	Reject H_0
	Opt-aiNet	3.4	< 0.001	Reject H_0
Chainlink	K-means	4.886	< 0.001	Reject H_0
	CPSO	3.748	< 0.001	Reject H_0
	DWB	0.85	0.395	Accept H_0
	SMAIN	6.32	< 0.001	Reject H_0
	Opt-aiNet	5.759	< 0.001	Reject H_0
Target	K-means	6.513	< 0.001	Reject H_0
	CPSO	4.517	< 0.001	Reject H_0
	DWB	2.964	0.003	Reject H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	4.657	< 0.001	Reject H_0

Table 5.11 Statistical Hypothesis Testing between LNNAIS and Other Models based on Q_{ratio}: Data Group 3 ($\alpha = 0.05$; with continuity correction; unpaired; non-directional)

Data set	Algorithm	z	p	Outcome
Ionosphere	K-means	2.24	0.025	Reject H_0
	CPSO	1.833	0.067	Accept H_0
	DWB	1.582	0.114	Accept H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	3.837	< 0.001	Reject H_0
Glass	K-means	4.664	< 0.001	Reject H_0
	CPSO	6.513	< 0.001	Reject H_0
	DWB	6.291	< 0.001	Reject H_0
	SMAIN	4.916	< 0.001	Reject H_0
	Opt-aiNet	6.646	< 0.001	Reject H_0

Table 5.12 Descriptive Statistics: Data Group 3

Data set	Algorithm	J_{intra}	J_{inter}	Q_{ratio}
Ionosphere	K-means	2.302 (± 0.125)	3.192 (± 0.486)	0.728 (± 0.045)
	CPSO	2.806 (± 0.221)	4.197 (± 1.306)	0.778 (± 0.387)
	SMAIN	2.767 (± 0.000)	6.047 (± 0.000)	0.458 (± 0.000)
	DWB	2.632 (± 0.168)	3.488 (± 0.888)	0.799 (± 0.195)
	Opt-aiNet	2.781 (± 0.068)	4.623 (± 1.086)	0.662 (± 0.275)
	LNNAIS	2.807 (± 0.207)	3.962 (± 0.576)	0.725 (± 0.127)
Glass	K-means	1.035 (± 0.038)	4.557 (± 0.464)	0.901 (± 0.309)
	CPSO	1.581 (± 0.120)	3.017 (± 1.121)	1.685 (± 0.674)
	SMAIN	1.709 (± 0.003)	7.663 (± 0.038)	0.381 (± 0.007)
	DWB	1.198 (± 0.089)	3.716 (± 0.899)	1.458 (± 0.471)
	Opt-aiNet	1.446 (± 0.170)	3.256 (± 1.179)	2.188 (± 0.701)
	LNNAIS	1.358 (± 0.149)	5.367 (± 0.423)	0.541 (± 0.113)

Table 5.13 Descriptive Statistics: Data Group 4

Data set	Algorithm	J_{intra}	J_{inter}	Q_{ratio}
Image Segmentation	K-means	65.274 (± 0.523)	356.964 (± 32.396)	0.694 (± 0.033)
	CPSO	77.522 (± 7.161)	177.950 (± 24.600)	1.493 (± 0.598)
	SMAIN	126.990 (± 0.283)	787.028 (± 1.906)	0.400 (± 0.001)
	DWB	71.657 (± 3.074)	245.495 (± 133.903)	1.060 (± 0.301)
	Opt-aiNet	74.457 (± 6.321)	174.931 (± 28.219)	1.621 (± 0.990)
	LNNAIS	87.984 (± 9.635)	597.456 (± 116.260)	0.989 (± 1.015)
Spambase	K-means	216.058 (± 0.000)	2003.263 (± 0.000)	0.108 (± 0.000)
	CPSO	301.660 (± 19.617)	136.613 (± 30.941)	2.301 (± 0.452)
	SMAIN	239.369 (± 27.139)	1599.789 (± 831.832)	0.194 (± 0.096)
	DWB	185.926 (± 22.246)	1216.169 (± 1509.055)	0.236 (± 0.120)
	Opt-aiNet	247.833 (± 18.812)	71.578 (± 37.181)	5.586 (± 6.421)
	LNNAIS	432.734 (± 221.003)	6720.659 (± 2691.210)	0.074 (± 0.046)
Letter Recognition	K-means	5.383 (± 0.012)	11.121 (± 0.157)	1.090 (± 0.043)
	CPSO	6.571 (± 0.121)	11.028 (± 0.764)	1.480 (± 0.225)
	SMAIN	7.297 (± 0.238)	17.299 (± 0.455)	0.751 (± 0.029)
	DWB	6.562 (± 0.108)	12.268 (± 0.704)	1.758 (± 0.662)
	Opt-aiNet	6.419 (± 0.108)	11.778 (± 0.630)	1.367 (± 0.179)
	LNNAIS	6.072 (± 0.080)	12.601 (± 0.331)	1.351 (± 0.202)

Table 5.14 Statistical Hypothesis Testing between LNNAIS and Other Models based on Q_{ratio}: Data Group 4 ($\alpha = 0.05$; with continuity correction; unpaired; non-directional)

Data set	Algorithm	z	p	Outcome
Image Segmentation	K-means	1.922	0.055	Accept H_0
	CPSO	5.093	< 0.001	Reject H_0
	DWB	3.6	< 0.001	Reject H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	5.064	< 0.001	Reject H_0
Spambase	K-means	3.984	< 0.001	Reject H_0
	CPSO	6.646	< 0.001	Reject H_0
	DWB	5.603	< 0.001	Reject H_0
	SMAIN	5.5	< 0.001	Reject H_0
	Opt-aiNet	6.646	< 0.001	Reject H_0
Letter Recognition	K-means	5.404	< 0.001	Reject H_0
	CPSO	2.144	0.032	Reject H_0
	DWB	3.053	0.002	Reject H_0
	SMAIN	6.646	< 0.001	Reject H_0
	Opt-aiNet	0.288	0.773	Accept H_0

Also, K-means tends to deliver clusters of a higher quality for data set letter recognition.

The experimental results show that, in general, LNNAIS delivers clusters of similar or higher quality than classical data clustering models like K-means and CPSO, and network based AIS models like DWB and Opt-aiNet. Overall, SMAIN tends to deliver clusters of a higher quality for all data sets, followed by LNNAIS. Although SMAIN tends to deliver clusters of a higher quality than LNNAIS, a cursory assessment indicates that SMAIN tends to utilise a larger ALC population than LNNAIS. This might indicate an overfit of the data which results in superior clustering quality of SMAIN. A disadvantage of SMAIN when compared to LNNAIS is that SMAIN follows a hybrid approach to determine the number of ALC networks (clusters) and is therefore computationally more expensive than LNNAIS. Furthermore, LNNAIS has less user specified parameters. The next section compares and discusses the ALC population sizes of SMAIN, DWB and LNNAIS to elaborate on the cursory assessment of overfitting the data. This is then followed by a sensitivity analysis of the LNNAIS parameters on the clustering quality of the model.

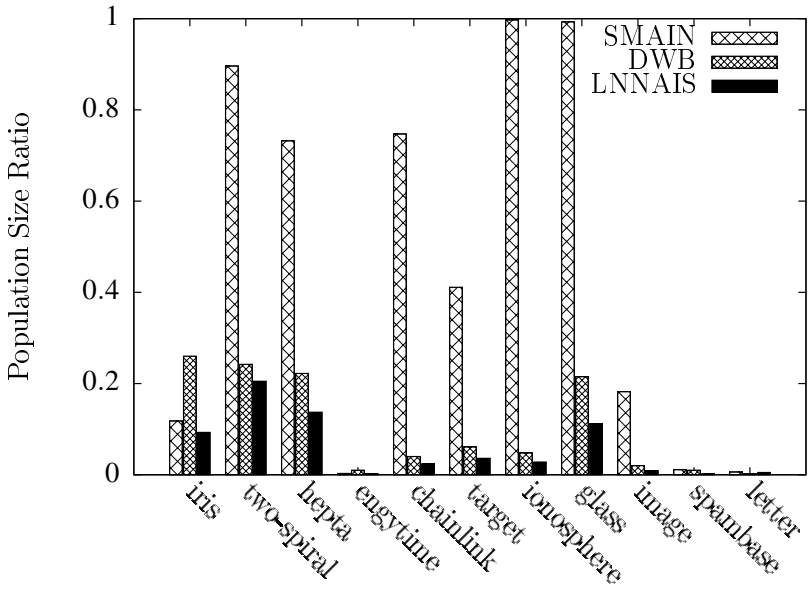


Figure 5.4 ALC Population Size Ratios of SMAIN, DWB and LNNAIS

5.7.7 ALC Population Size - Overfitting the Data

This section investigates the ALC population sizes between SMAIN, DWB and LNNAIS to indicate potential overfit of the data. Overfitting of the data could result in superior clustering quality of a specific model when compared to other models which utilise a smaller ALC population size. Figure 5.4 illustrates a histogram of the ALC population size of SMAIN, DWB and LNNAIS to cluster the data sets. The size of the ALC population is expressed as a ratio of the applicable data set size. Therefore, an ALC population size ratio closer to 1.0 indicates a higher level of overfit of the applicable data set. The figure illustrates that LNNAIS has a population size ratio of less than 0.2 for all of the data sets. On the contrary, SMAIN has a population size ratio of more than 0.4 for six of the data sets (two-spiral, hepta, chainlink, target, ionosphere and glass). For data sets glass and ionosphere, the ALC population size of SMAIN is almost equal to the size of the data sets (ratio close to 1.0). In general, SMAIN utilises a larger ALC population to cluster the data than DWB and LNNAIS. This not only explains the superior clustering quality of SMAIN in the previous section but also a drawback of SMAIN that tends to overfit the data. Compared to SMAIN in view of these findings, LNNAIS delivers clusters of high quality without overfitting the data.

5.7.8 Influence of LNNAIS parameters

This section investigates the influence of the LNNAIS parameters on the clustering quality of the model with reference to Q_{ratio}, J_{intra}, J_{inter} and the number of obtained clusters K. These parameters are the maximum population size, $\mathcal{B}_{max}$, the neighbourhood size, ρ, and the clonal level threshold, ε_{clone}. Compared to the network based AIS models which are used in this chapter, LNNAIS has significantly less parameters. The clustering results of a representative data set were selected from each of the defined data groups for the discussion. All of the other clustering results of the remaining data sets within the same data group, followed similar trends unless stated otherwise. The identified data sets include two-spiral from group 1, chainlink from group 2, glass from group 3 and image segmentation from group 4. The LNNAIS model has been executed with population sizes of 10 to 50 ALCs, clonal level threshold values of 6 to 27 and neighbourhood sizes which are calculated as a ratio of the population size. Neighbourhood size ratios from 0.05 to 0.9 were used to calculate the neighbourhood size ρ using $\rho = \rho_r \times \mathcal{B}_{max}$ (ρ_r is the neighbourhood size ratio). In cases where a parameter was kept constant, the parameter was set to the value as listed in table 5.6 for each of the applicable data sets.

Population Size: Figures 5.5 to 5.8 show the effect of different ALC population sizes, $\mathcal{B}_{max}$, at different neighbourhood size ratios, ρ_r, and a constant clonal level threshold, ε_{clone}. These figures show that for small neighbourhood sizes an increase in the ALC population size has a less significant influence on the clustering quality, Q_{ratio}, when compared to larger neighbourhood sizes. The cluster compactness and separation do however tend to decrease at low neighbourhood sizes with an increase in the ALC population size (increasing J_{intra} and decreasing J_{inter}). Furthermore, figures 5.5 to 5.8 also show that no significant improvement is achieved for all the different neighbourhood sizes in the number of obtained clusters for ALC population sizes larger than a specific optimal value (which is problem dependant). This can also be observed in figures 5.13 to 5.16. Figures 5.13 to 5.16 show that an increase in the ALC population size increases the cluster compactness and separation (decreasing J_{intra} and increasing J_{inter}) for different clonal level threshold values with a low constant neighbourhood size. Therefore, an increase in the ALC population size increases diversity which obtains the required number of clusters and improves the clustering quality.

Neighbourhood Size: Figures 5.9 to 5.12 show the effect of different neighbourhood size ratios, ρ_r, at different clonal level threshold values, ε_{clone}, and a constant ALC population size, $\mathcal{B}_{max}$. An increase in the neighbourhood size decreases the cluster compactness and separation

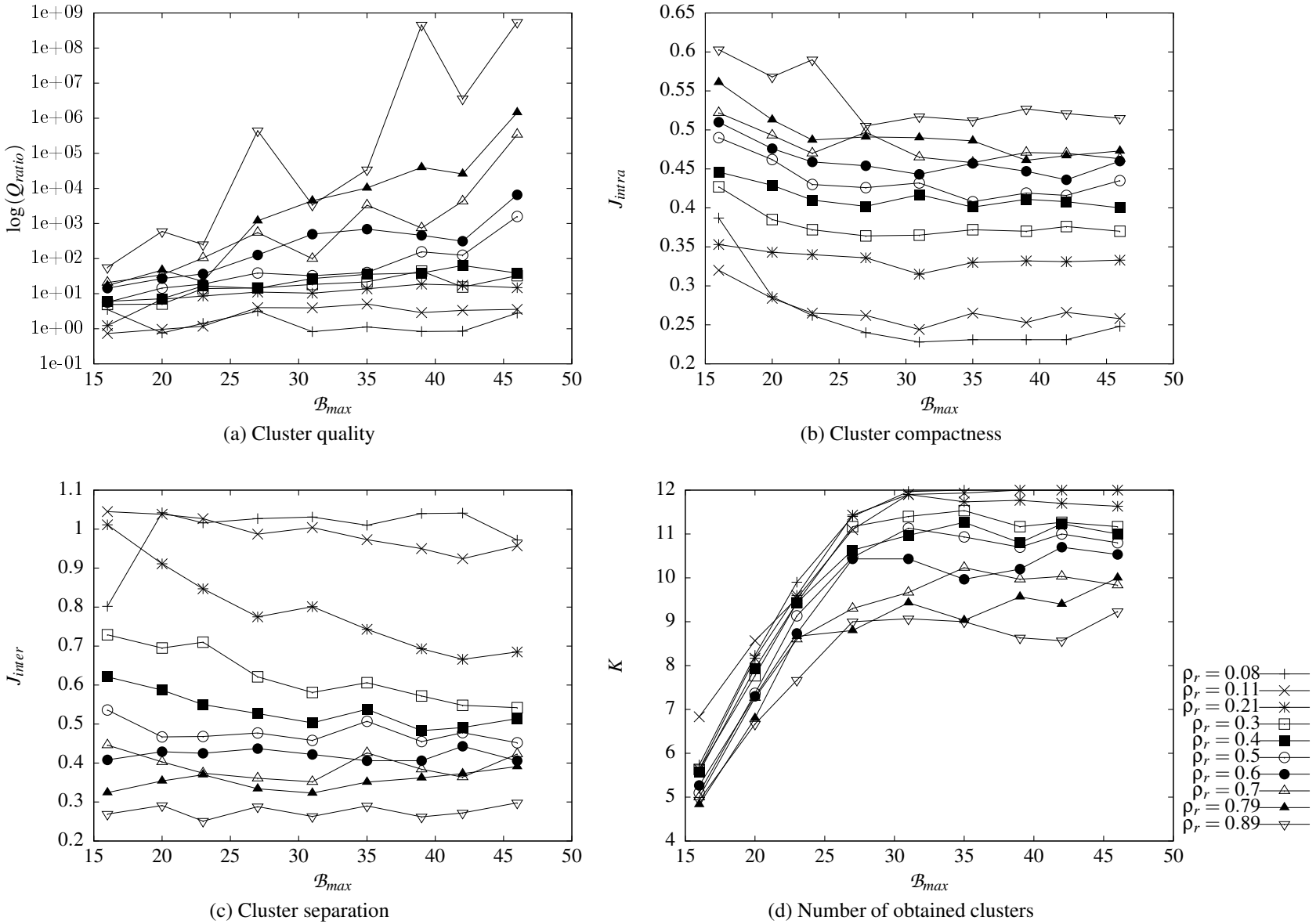
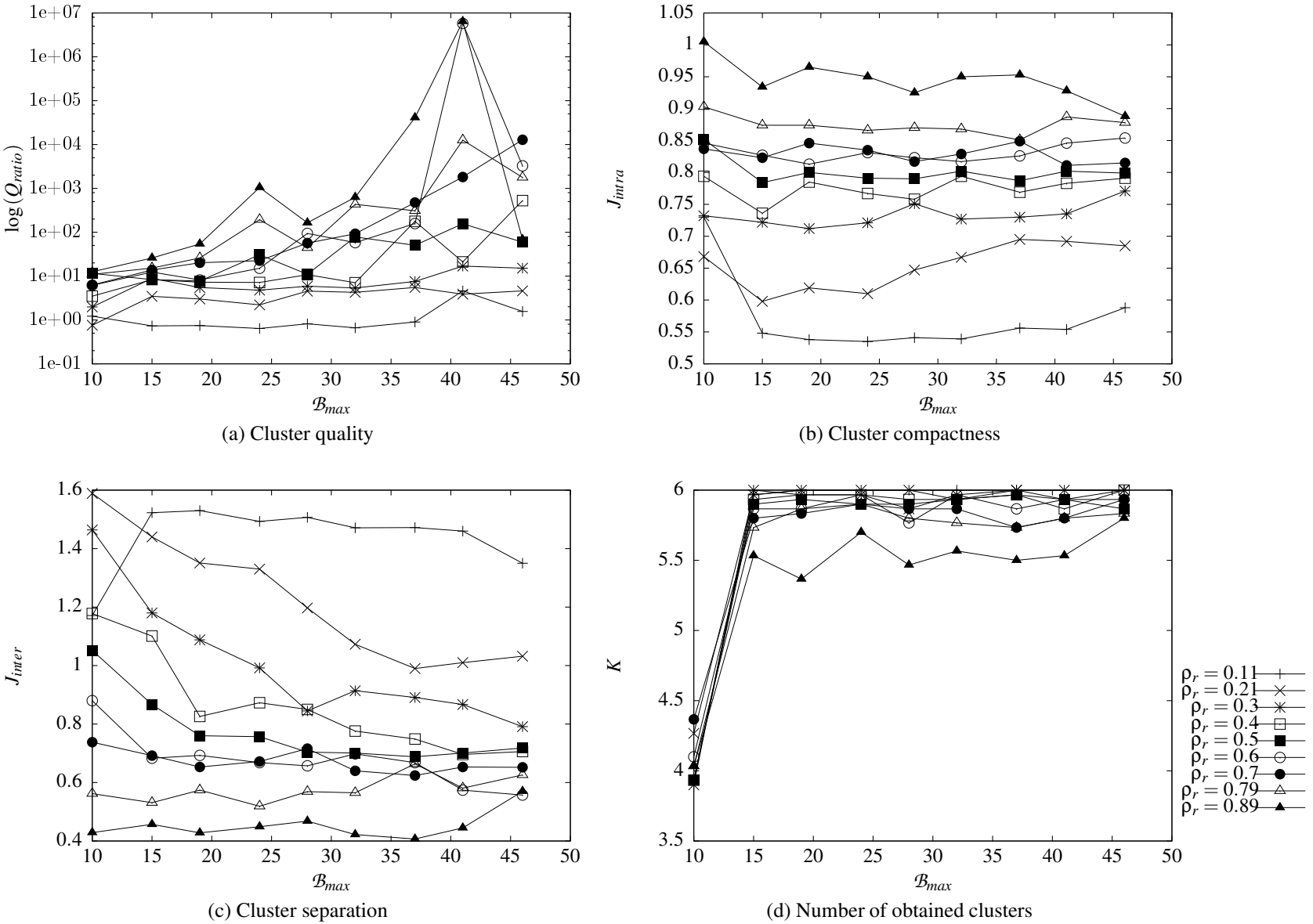


Figure 5.5 Two-spiral data set ($\varepsilon_{clone} = 6$): Effect of the ALC population size with a constant clonal level threshold



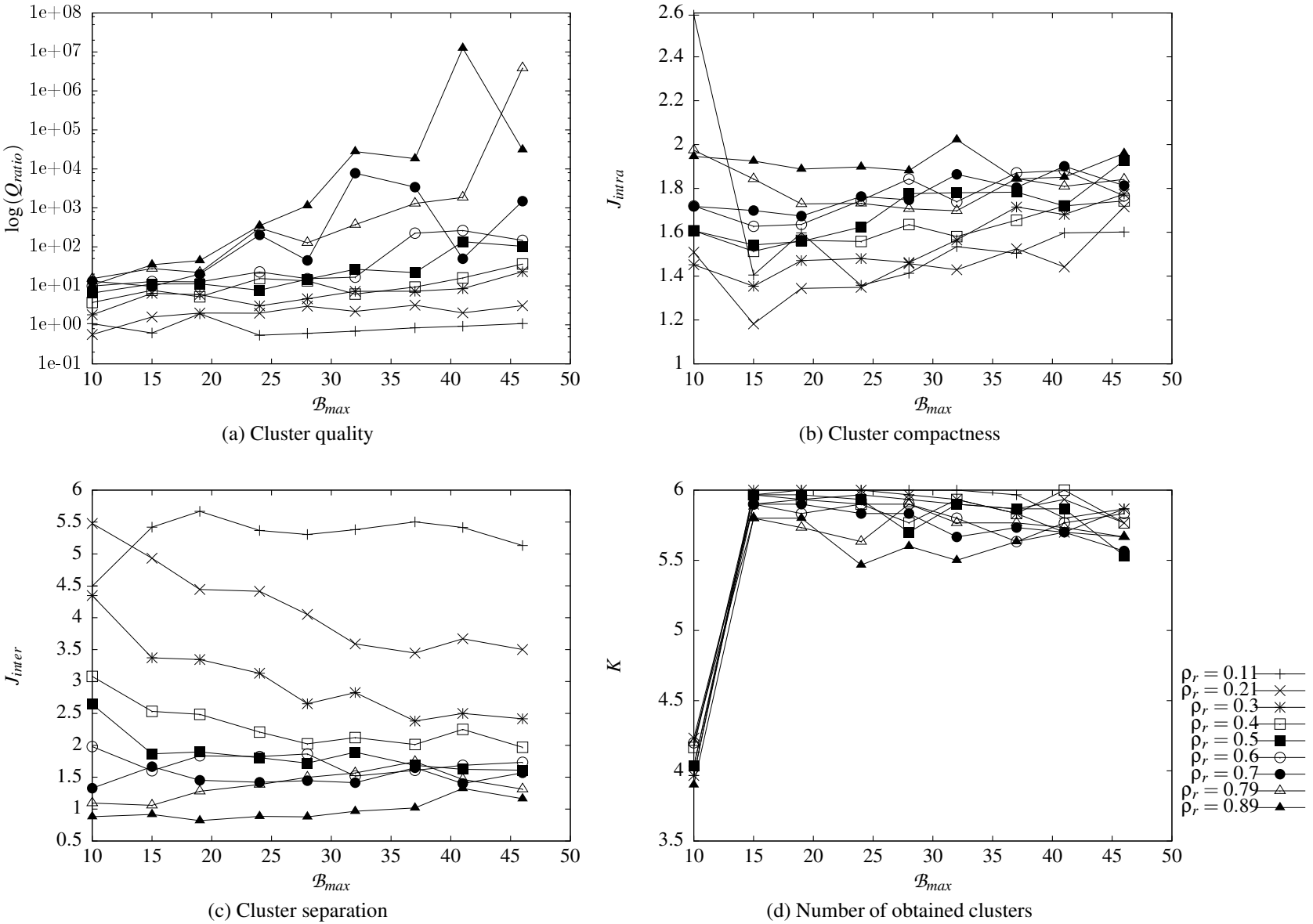
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.6 Chainlink data set ($\varepsilon_{clone} = 8$): Effect of the ALC population size with a constant clonal level threshold



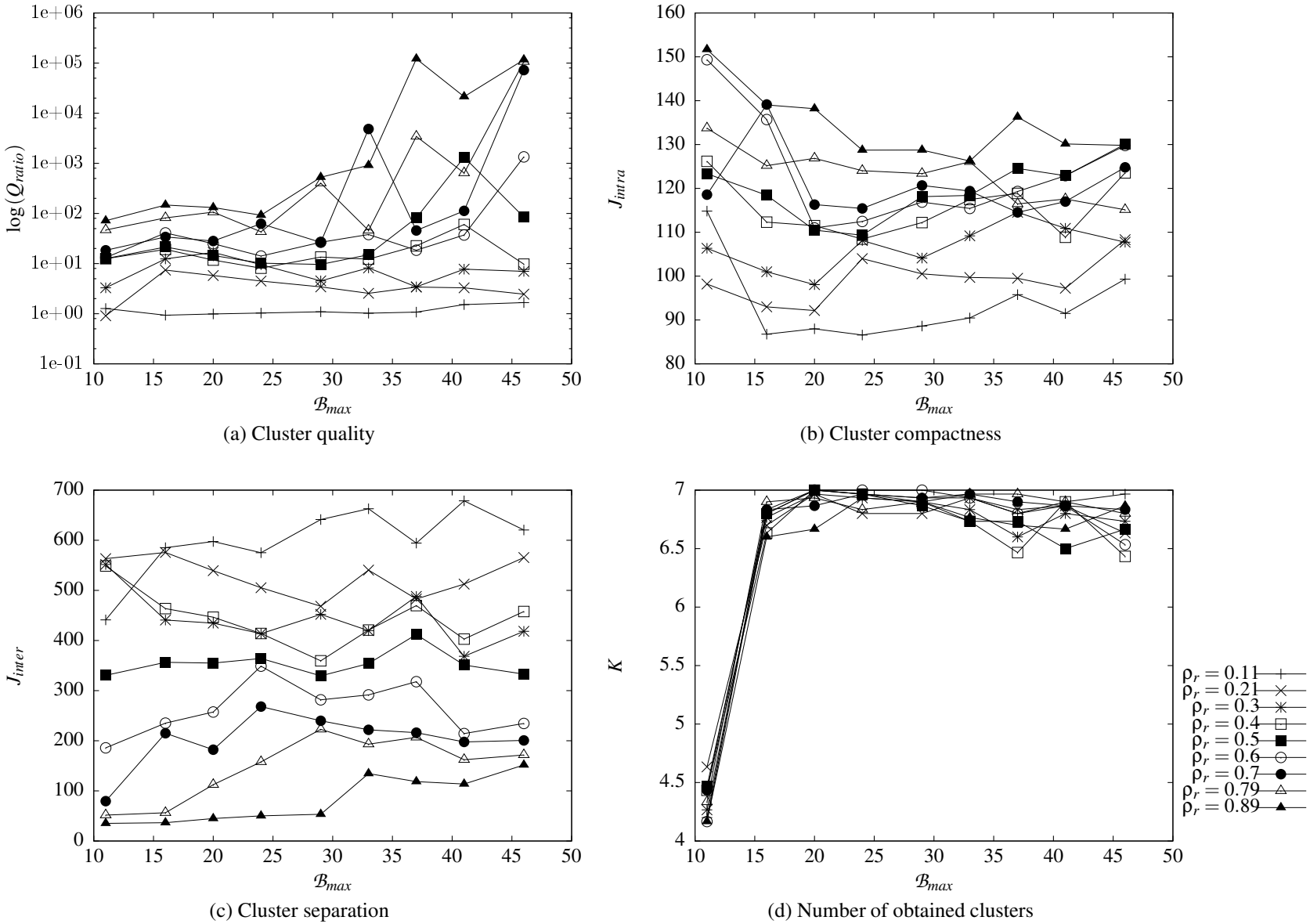
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.7 Glass data set ($\varepsilon_{clone} = 8$): Effect of the ALC population size with a constant clonal level threshold



(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.8 Image Segmentation data set ($\varepsilon_{clone} = 27$): Effect of the ALC population size with a constant clonal level threshold

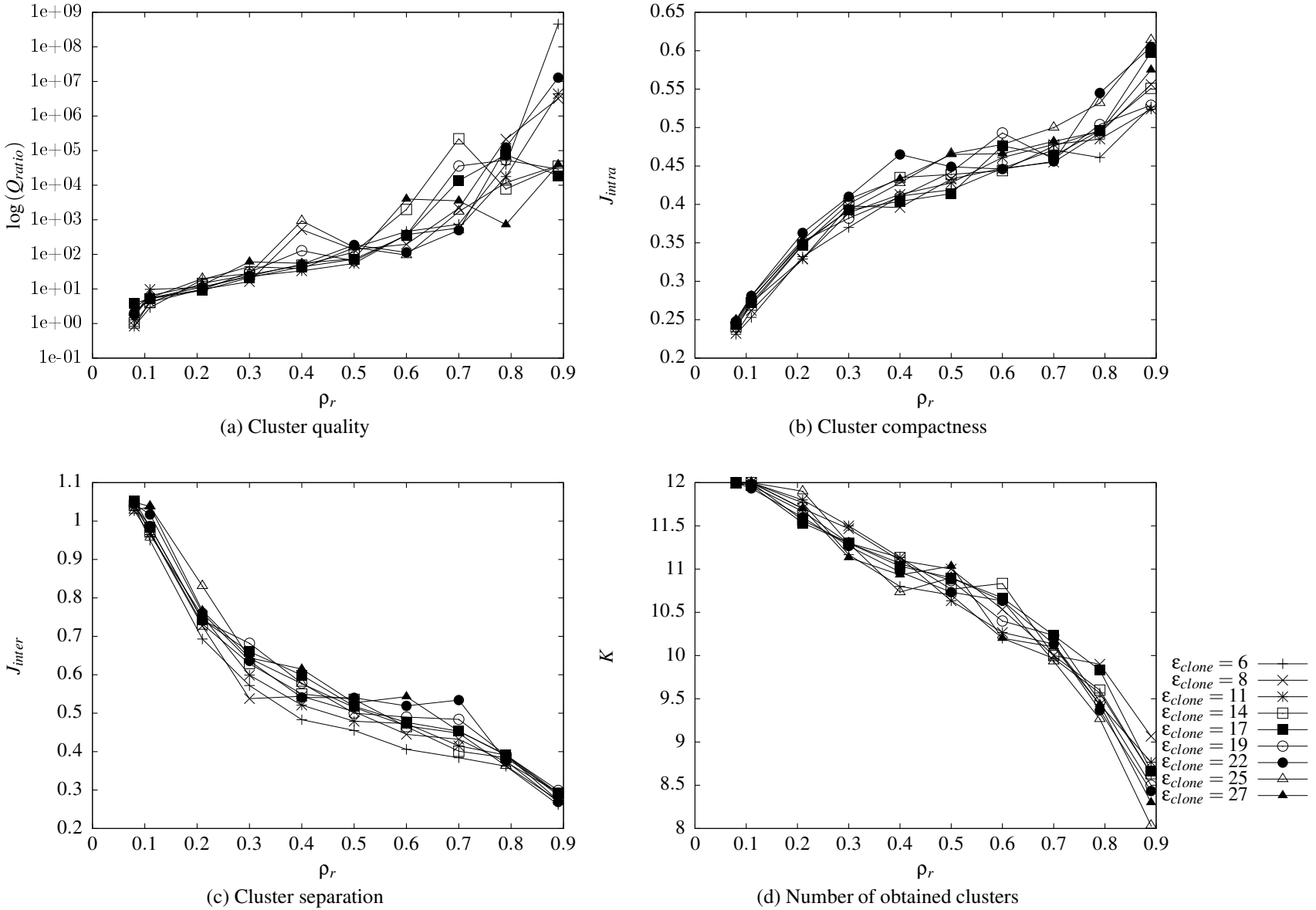
for all of the different clonal level threshold values, resulting in clusters of a lower quality (increasing Q_{ratio} and J_{intra} with a decreasing J_{inter}). This effect is also shown in figures 5.5 to 5.8 where an increase in the neighbourhood size ratio decreases the cluster compactness (increases J_{intra}) and decreases the cluster separation (decreases J_{inter}) for all values of $\mathcal{B}_{max}$. From these observations it can be concluded that small values of ρ_r deliver more compact and more separated clusters (lower J_{intra}, higher J_{inter}) and therefore clusters of higher quality (lower Q_{ratio}) when compared to higher values of ρ_r. From the above mentioned figures, lower neighbourhood sizes also tend to obtain the required number of clusters.

Clonal Level Threshold: Figures 5.13 to 5.16 show the effect of different clonal level threshold values, ε_{clone}, at different ALC population sizes, $\mathcal{B}_{max}$, and a constant neighbourhood size, ρ. An increase in the clonal level threshold has no significant improvement in the number of obtained clusters at different ALC population sizes (as illustrated in figures 5.13 to 5.16) and also not at different neighbourhood sizes (as illustrated in figures 5.9 to 5.12). Furthermore, the different clonal level threshold values follow similar trends with reference to the quality, compactness and separation of the clusters when the neighbourhood size increases (as illustrated in figures 5.9 to 5.12 and explained in the previous paragraph). In the case of the chainlink and image segmentation data sets, increasing the clonal level threshold also results in less compact clusters at different ALC population sizes (as illustrated in figures 5.14 and 5.16), whereas there is no significant change in the compactness of the clusters for the two-spiral and glass data sets (as illustrated in figures 5.13 and 5.15). Therefore, the clonal level threshold influences the cluster compactness and is problem specific.

In summary, the clustering performance of LNNAIS is sensitive to the values of the ALC population size and neighbourhood size. The ALC population size is problem specific and in general low neighbourhood size values deliver clusters of higher quality. The clustering performance of LNNAIS is generally insensitive to the value of the clonal level threshold.

5.8 Conclusion

A new network based AIS model (LNNAIS) was proposed for data clustering. LNNAIS utilises a different network topology, which is an index based ALC neighbourhood topology to determine the network connectivity between ALCs. The clustering performance of the LNNAIS model was compared against classical clustering algorithms (K-means clustering and CPSO) and existing



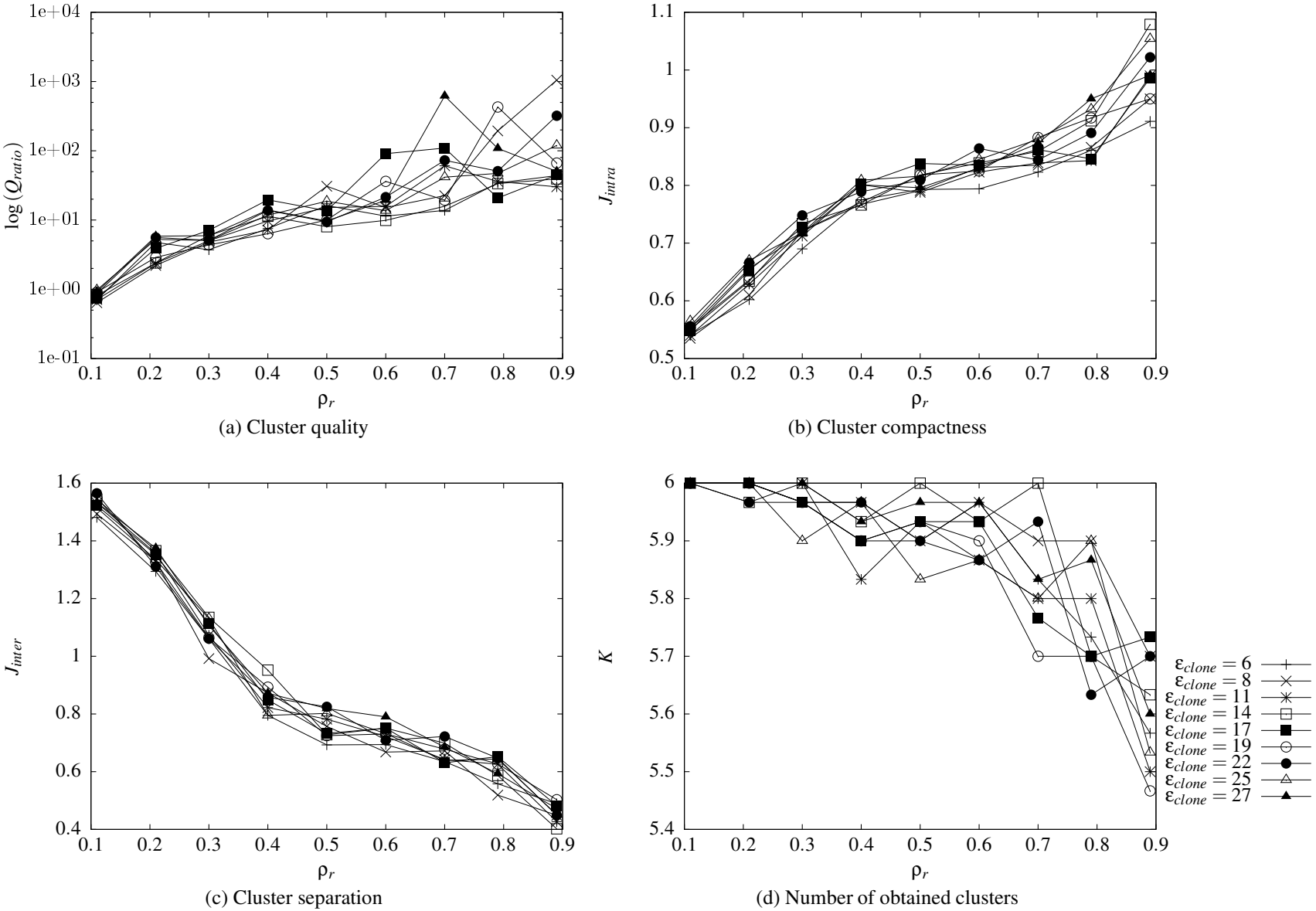
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.9 Two-spiral data set ($\mathcal{B}_{max} = 39$): Effect of the neighbourhood size with a constant ALC population size



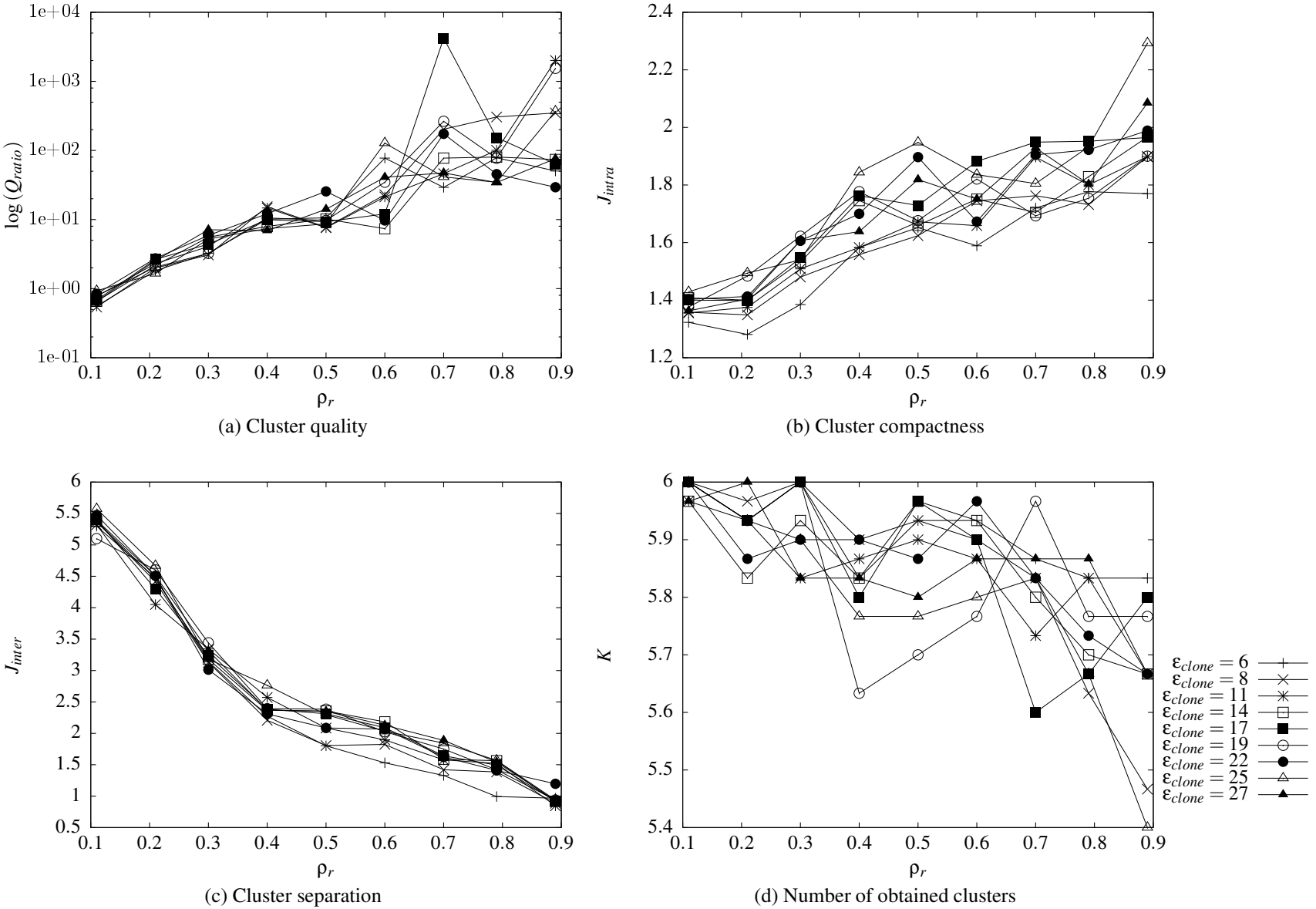
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.10 Chainlink data set ($\mathcal{B}_{max} = 24$): Effect of the neighbourhood size with a constant ALC population size



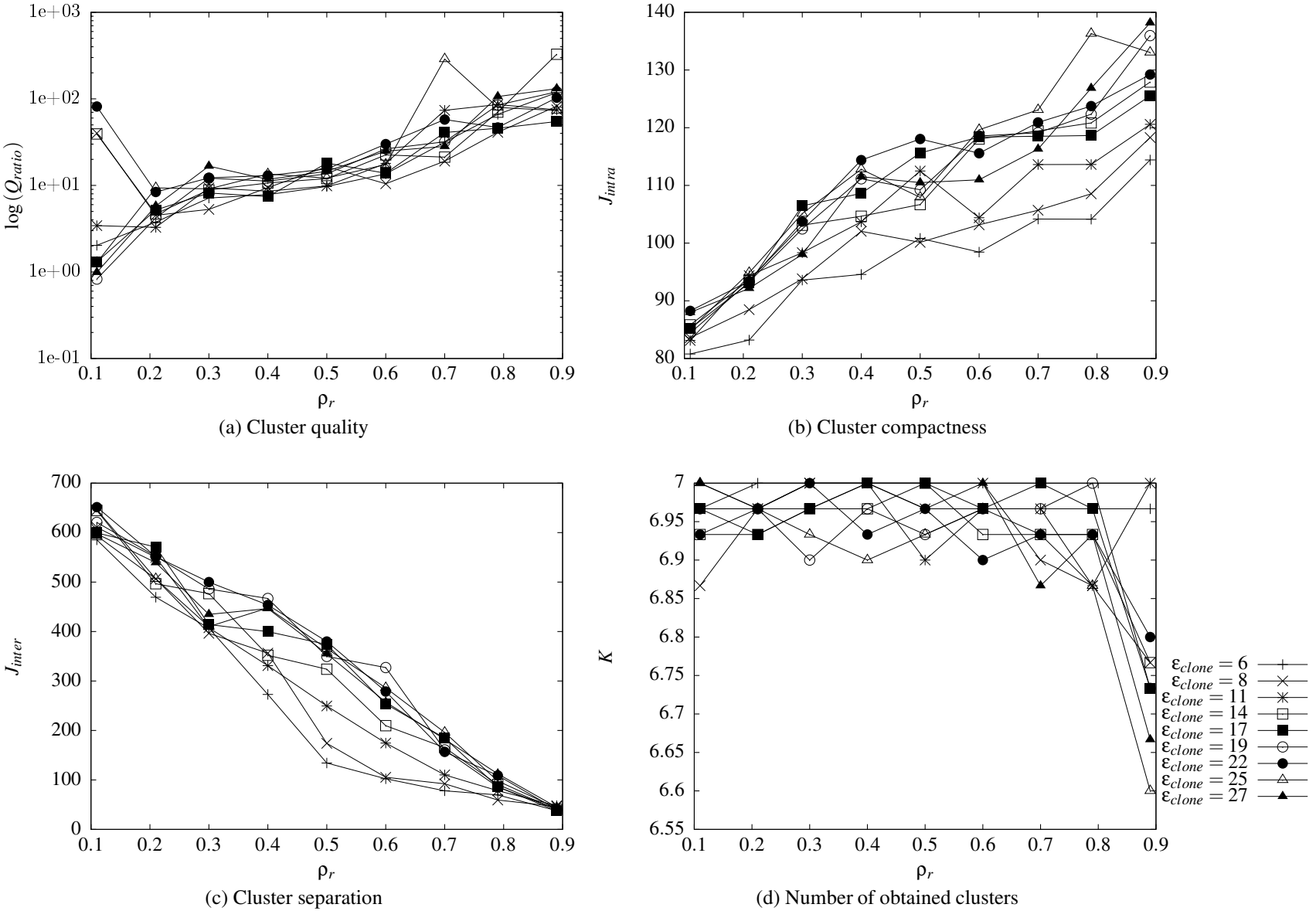
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.11 Glass data set ($\mathcal{B}_{max} = 24$): Effect of the neighbourhood size with a constant ALC population size



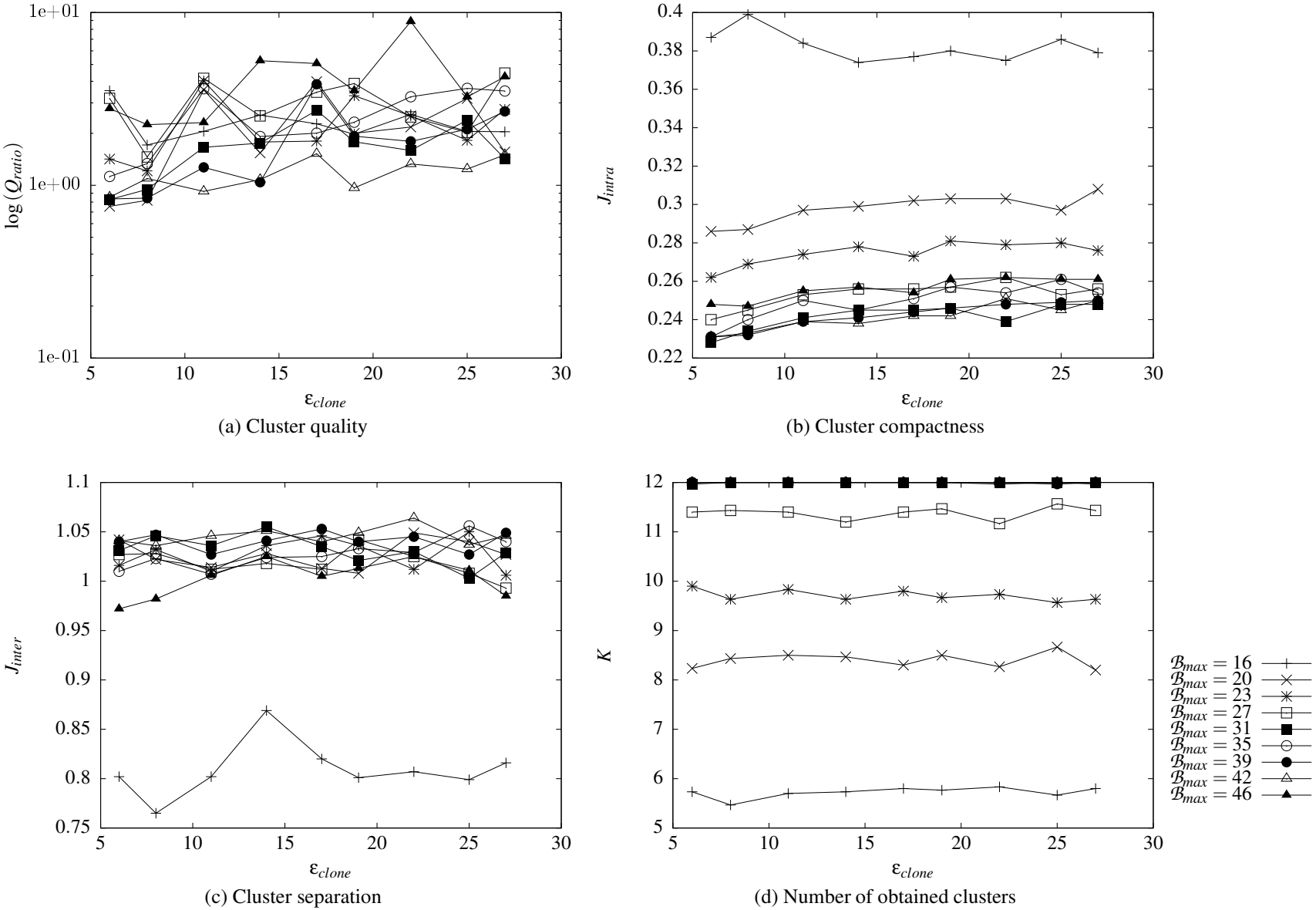
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.12 Image Segmentation data set ($\mathcal{B}_{max} = 20$): Effect of the neighbourhood size with a constant ALC population size



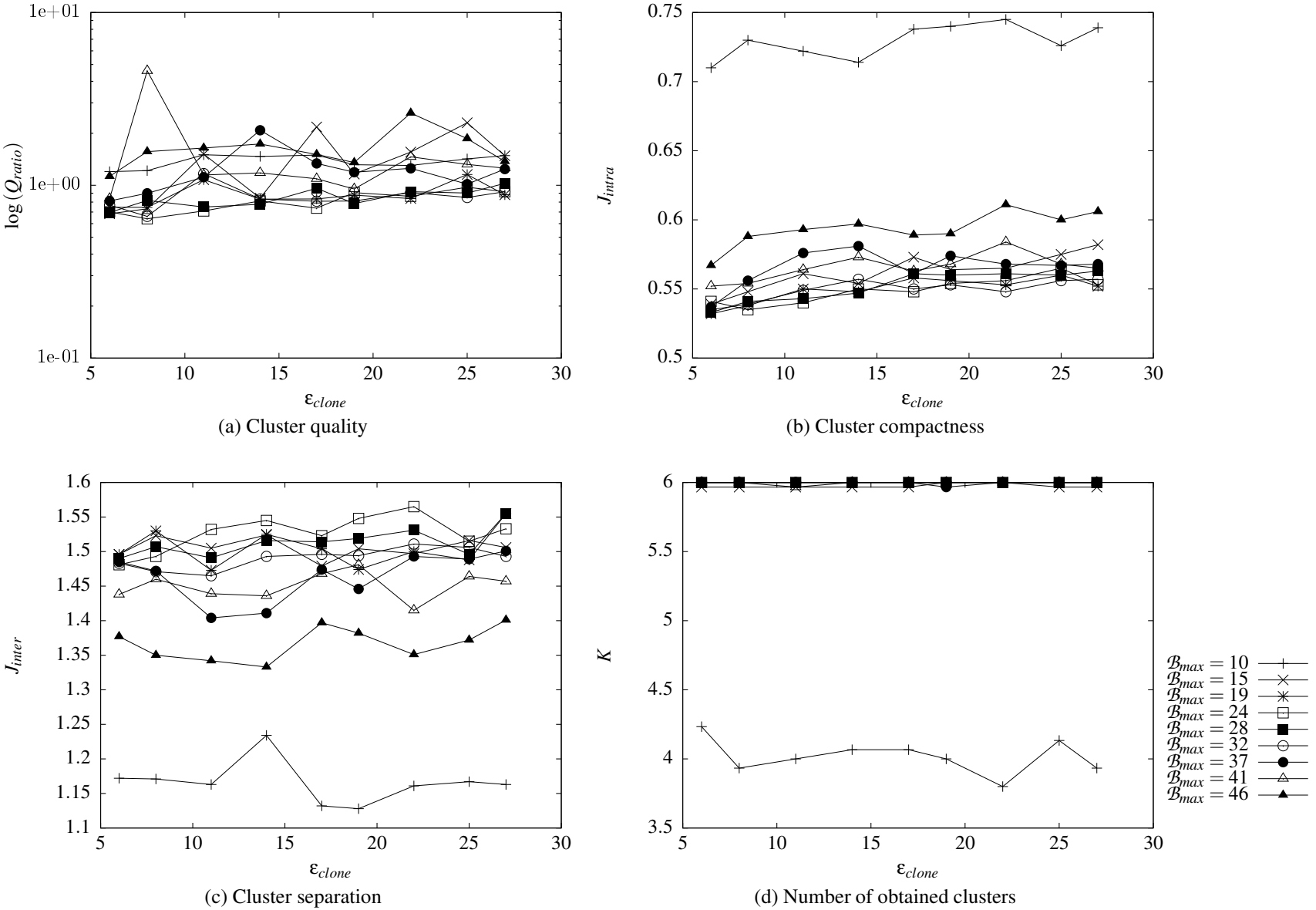
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.13 Two-spiral data set ($\rho = 3$): Effect of the clonal level threshold with a constant neighbourhood size



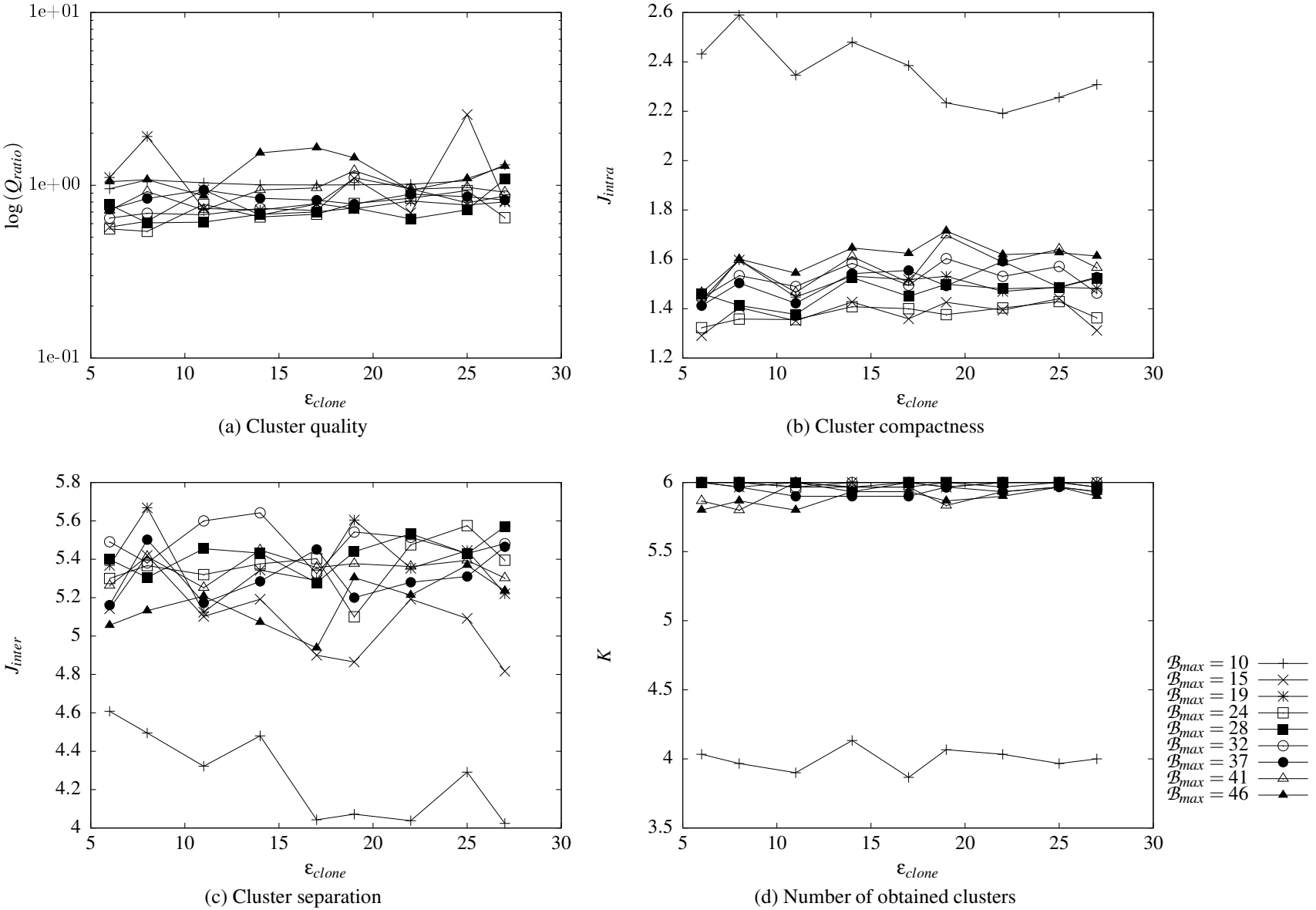
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.14 Chainlink data set ($\rho = 3$): Effect of the clonal level threshold with a constant neighbourhood size



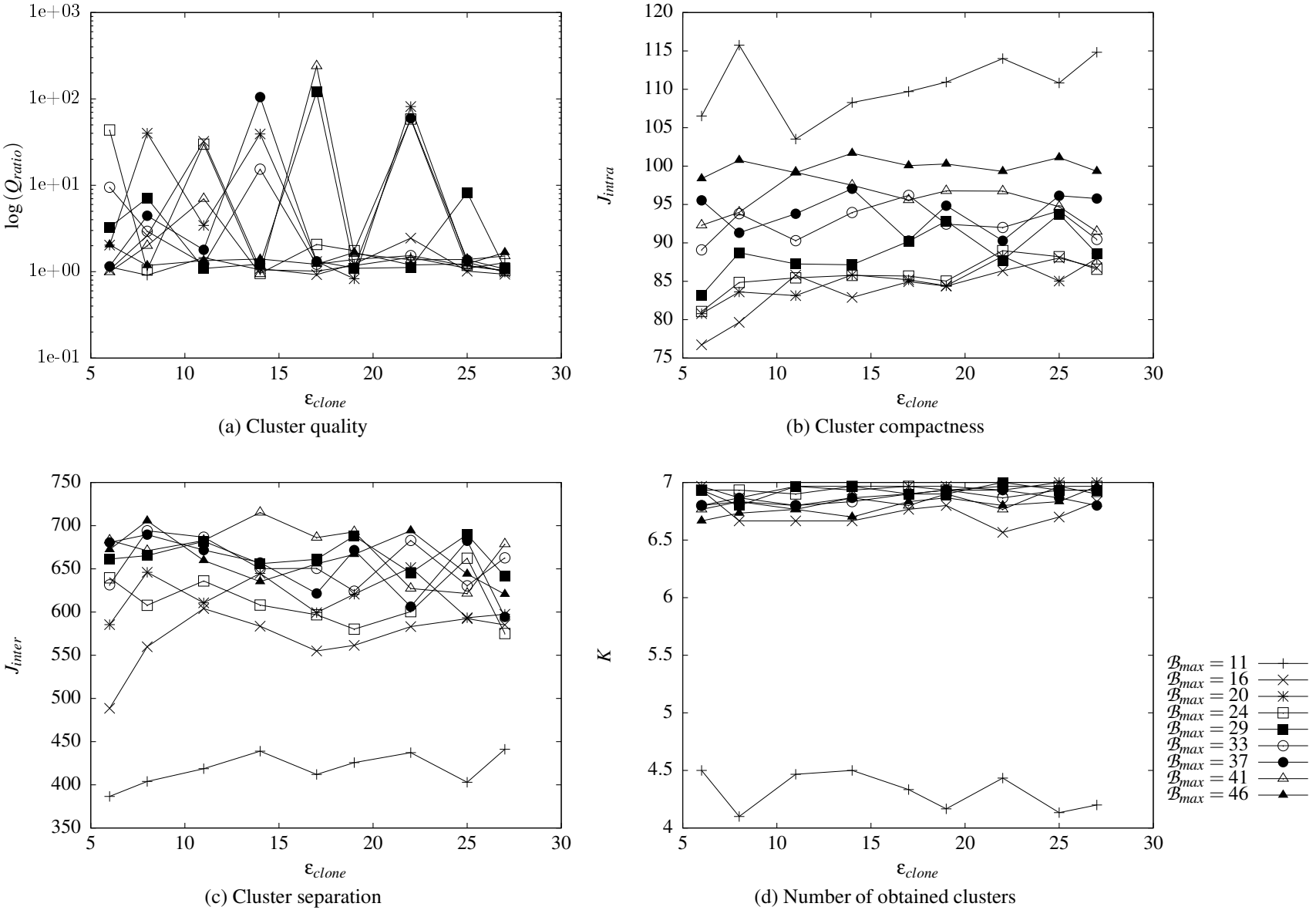
(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.15 Glass data set ($\rho = 3$): Effect of the clonal level threshold with a constant neighbourhood size



(a) Cluster quality

(b) Cluster compactness

(c) Cluster separation

(d) Number of obtained clusters

Figure 5.16 Image Segmentation data set ($\rho = 2$): Effect of the clonal level threshold with a constant neighbourhood size

network based AIS models (SMAIN, DWB and Opt-aiNet). In most cases, LNNAIS produced better or similar results with reference to the quality, compactness and separation of the clusters. Although SMAIN tends to deliver clusters of a higher quality than LNNAIS, further investigation showed that SMAIN tend to utilise a larger ALC population than LNNAIS.

A sensitivity analysis was done on the LNNAIS parameters to investigate the effect of the parameters on the clustering quality. An increase in the ALC population size increases diversity which obtains the required number of clusters and improves the clustering quality. Smaller neighbourhood sizes deliver more compact and more separated clusters when compared to larger neighbourhood sizes, and tend to obtain the required number of clusters. Therefore small neighbourhood sizes deliver clusters of a higher quality. The clonal level threshold influences the compactness of the clusters and is problem specific.

Although existing network based AIS models and LNNAIS do not require any user specified parameter of the number of required clusters to cluster the data, the techniques used by these models to determine the number of ALC networks do, however. Therefore, the following chapter investigates and proposes two alternative techniques that can be used with LNNAIS to dynamically determine the number of clusters in a data set.