Chapter 7

Clustering input data

In this chapter the concept of pattern identification on input data is investigated. What is peculiar about the benchmark problem sets proposed by both Solomon (1987) and Homberger and Gehring (1999) are the fact that they are preempting specific theoretical characteristics, unlike problems found in real applications. This is clearly illustrated when the assignment of time windows is discussed. For the problem sets $R1$, $R2$, $RC1$, and $RC2$ a percentage of customers are selected to receive time windows, say $0 < f \leq 1$. Next n random numbers from the random uniform distributions is generated on the interval $(0, 1)$, and sorted. Customers $i_1, i_2, \ldots, i_{n_1}$ are then assigned time windows, where the number of customers requiring time windows can then be approximated by $n_1 \approx f.n$. The center of the time window for customer $i_j \in \{i_1, i_2, \ldots, i_{n_1}\}$ is a uniformly distributed, randomly generated number on the interval $\left(e_0 + t_{oi_j}, l_0 - t_{i_j0}\right)$, where e_0 and l_0 denotes the opening and closing times of the depot, respectively, and t_{0i_j} and t_{i_j0} denotes the travel distance from the depot to customer i_j, and back, respectively.

For clustered problem sets $C1$ and $C2$ the process becomes questionable. Customers in each cluster are first *routed* using a *3-opt* routine as described in the previous chapter. An orientation is chosen for the route, and time windows are then assigned with the center being the arrival time at the customer. The width and density are derived in a similar fashion as for random and semi-clustered data. Although Solomon (1987) states that *"this approach permits the identification of a very good, possibly optimal, cluster-by-cluster solution which, in turn, provides an additional means of evaluating heuristic performance"*, it does not provide a credible means to evaluate real life problems where customers do not negotiate their sequence prior to stating a preferred time window.

Literature provides good references to what type of metaheuristics, or metaheuristic

configurations provide good answers to which of the six benchmark problems. When given a real data set from industry, however, one is not provided with the classification of *"this a C1 problem set"*. To therefore determine which solution algorithm to use, and which parameter setting, the routing agent first needs to classify the input data.

The idea behind *learning* is not so that an agent *can* act, but rather to *improve* an agent's ability to act in future. In the context of vehicle routing the *agent* is the routing system proposed by this thesis. The acting is the routing of vehicles, given the demand inputs, using some metaheuristic with its associated parameter settings. For a routing systems to *learn*, it must perceive certain characteristics of the inputs, for example the geographical dispersion of customers or the width of time windows provided by customers, and choose an appropriate metaheuristic, and know what parameter values to suggest in order to obtain the best route in the shortest possible time. The execution of the metaheuristic makes up the *performance* element of the agent, and have been thoroughly introduced in Chapters 4 through 6. Deciding which metaheuristic to use forms the *learning* element of the agent.

The concepts of *representation* of an agents knowledge and its *reasoning* processes that brings that knowledge to life are central to the entire field of AI. The design of a learning element is affected by three distinctive components:

- Which components of the performance element are to be learned?
- What feedback is available to learn these components?
- What representation is used for the components?

The *components* of the performance element that the agent should learn from input data provided, are the geographical distribution of customers; the relation between customer demand and vehicle capacity, and time window characteristics. In order to determine the nature of learning for the agent, the type of *feedback* available to the agent is extremely important. Russell and Norvig (2003) distinguish between three types of feedback:

Supervised learning Learning takes place by providing both input and output examples. For instance, if an agent is provided with many pictures that he is told contain buses, the agent *learns* to recognize a bus. Both the input and the output is provided.

Unsupervised learning Patterns are learned by providing input, but in the absence of specific outputs. When commuting from home to work, a person might be able to distinguish between *"good traffic days"* and *"bad traffic days"*, without ever being

given examples of either of the two. A purely unsupervised agent cannot learn as it has no information as to what constitute a desirable state, or a correct action.

Reinforcement learning The most general of the three types of feedback. Without being told by a supervisor what to do, a reinforcement learning agent must learn through *reinforcement*, for example an action that is not followed by a tip or any confirmation is interpreted as an undesirable state.

The routing agent in this thesis will typically be given a data set without knowing whether it is clustered, randomly distributed, or whether the time windows are tight. As a supervisor also do not know whether it is clustered, or not, it would also not be possible to reinforce a *correct* action taken, as the evaluation of *correctness* would be flawed. The routing agent would hence have to learn unsupervised.

Knowledge and reasoning are both required for problem solving agents to perform well in complex environments. The concept of *knowledge representation* is important as an agent would require some structure in which to put the information that it has learnt, so as to be able to revisit its knowledge base in future when decision are made. This is necessary to improve future decision making. The central component of a knowledge-based agent is its *knowledge base*, expressed as sentences in a *knowledge representation language*. Each sentence asserts something about the agent's world. There are ways to add new sentences to the knowledge base, and ways to query what is already known. In AI these two actions are referred to as Tell and Ask. Being a logical agent, when 'Ask'ed a question, the answer would be related to what the knowledge base has been 'Tell'ed previously. Also, the two tasks may involve *inference* where new sentences are derived from old ones.

7.1 Unsupervised clustering

The clustering problem is defined as partitioning a given data set into groups, or clusters, such that data points in a cluster are more similar to each other than to other points belonging to different clusters. According to Gath and Geva (1989) and Xie and Beni (1991) the criteria for the definition of *optimal partition* of the data into clusters are based on three requirements:

- Clear separation between the resulting clusters.
- Minimal volume of the clusters.

- Maximal number of data points concentrated in the vicinity of the cluster centroid, i.e. maximum cohesion.

Thus, although the environment is fuzzy, the aim of the classification is the generation of well-defined subgroups. To solve the clustering problem, a number of clustering algorithms have been proposed. One of the most important families of clustering techniques are *partitioning* clustering, with the most commonly used algorithm in this family being the k-means clustering algorithm and its numerous variants (Xu and Brereton, 2005). A main problem of the k-means clustering variants is that the algorithms require the number of clusters, c, as an input so that a data set can be clustered into c partitions.

Unsupervised clustering is the problem of discerning multiple categories in a collection of objects. The *categories* referred to are the components of the input data that the agent should learn, while *objects* refer to the input data points, i.e. the customers in the network. The learning process is unsupervised as the agent does not know whether the input data is randomly distributed, clustered, or a combination of both.

So if the number of clusters, c, is not known when learning should occur, the agent can perform a number of clustering attempts, each using a different values for c. In such a way the most appropriate value for c can be determined. Such an approach is defined as *cluster validation*. In this chapter, the behavior of a number of validation indices will be tested on benchmark data sets for the VRPTW. The objective is to establish trends that can be used to Tell the routing agent how to identify input data as belonging to either the $R1$, $R2$, $C1$, $C2$, $RC1$, or $RC2$ group of problems. The most appropriate metaheuristic can then be identified, along with its most appropriate parameter settings.

7.1.1 Fuzzy c-means clustering

One of the variants of the k-means clustering algorithm, fuzzy c-means (FCM) clustering, attempts to find the most characteristic point in each cluster $v_i \in \boldsymbol{V} = \{v_1, \ldots, v_c\}$, which can be considered as the *center* of cluster i and then grade the membership for each node $x_j \in \boldsymbol{X} = \{x_1, \ldots, x_n\}$ in cluster i. The member allocation is achieved by minimizing the commonly used *membership weighted with-in cluster error* objective function defined in (7.1)

$$J_e(\boldsymbol{U}, \boldsymbol{V}) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 \tag{7.1}$$

where d_{ij} is the Euclidean distance between object j and the i^{th} center, and u_{ij} is the fuzzy membership of object j belonging to the i^{th} cluster. The FCM is then described

by Algorithm 7.1. The algorithm requires the number of classes, a fuzzy factor and a

Algorithm 7.1: Fuzzy c-means clustering

Input: Number of classes, c

Input: Fuzzy factor, $m > 1$

Input: Convergence threshold, $\varepsilon > 0$

Randomly select c nodes to initialize centers matrix $\boldsymbol{V}^0$,

$k \leftarrow 0$

$J_e^k \leftarrow \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m(k)} \left(d_{ij}^k\right)^2$

repeat

  for $i \in \{1, \ldots, c\}, j \in \{1, \ldots, n\}$ **do**

    $u_{ij}^k = \left(\sum_{r=1}^{c} \left[\left(\frac{d_{ij}^k}{d_{rj}^k} \right)^{\frac{2}{m-1}} \right] \right)^{-1}$ **if** *For any* $r \in \{1, \ldots, c\}$, $d_{rj}^k = 0$ **then**

      $u_{rj}^k = 1$

      for $i, r \in \{1, \ldots, c\}, i \neq r$ **do**

        $u_{ij}^k = 0$

      endfor

    endif

  endfor

  for $i \in \{1, \ldots, c\}$ **do**

    $\boldsymbol{V}_i^{k+1} = \frac{\sum_{j=i}^{n} u_{ij}^{m(k)} x_j}{\sum_{j=i}^{n} u_{ij}^{m(k)}}$

  endfor

  $k \leftarrow k + 1$

  $J_e^k \leftarrow \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m(k)} \left(d_{ij}^k\right)^2$

until $\left\| J_e^k - J_e^{k+1} \right\| < \varepsilon$

convergence threshold as input. The centers matrix $\boldsymbol{V}$ is then initialized using a random selection of c nodes from the node set $\{1, \ldots, n\}$. The iteration count is zeroed before the membership matrix $\boldsymbol{U}^k$ is calculated. A new centers matrix is calculated, before the convergence of the objective function is tested. Xu and Brereton (2005) notes that when the fuzzy factor m approaches 1, the FCM is similar to the standard k-means clustering. When m approaches infinity, however, the clustering of the FCM is at its fuzziest: each node is assigned equally to each cluster. The authors also note that the FCM is but a local search

algorithm, and at best will find a local minimum, and is therefore sensitive to the random initial guess for $\boldsymbol{V}^0$. Figure 7.1 illustrates the clustering of one of the $C1$ problem sets provided by Gehring and Homberger (1999), $C1$-2-1, the first of their problem sets with 200 customers. The small circles indicate the customer nodes, while asterisks indicate the center of the cluster. All nodes clustered together are linked with gray lines. In establishing the clusters, a fuzzy factor of $m = 3$, convergence threshold of $\varepsilon = 1.0 \times 10^{-5}$, and an iteration limit of $k^{\max} = 1000$ is used. A number of validation indices are subsequently considered to evaluate the clustering.

7.1.2 Validation indices

A validation index is a single real value that describes the quality of a cluster partition. Some of the validation indices are only concerned with the membership value of the final clustering partition. Although Bolshakova and Azuaje (2003) do not apply the *Silhouette* index on fuzzy clusters, this thesis propose that for a given cluster $i \in \{1, \ldots, c\}$, assign to each node j a quality measure s_j, known as the *silhouette width*, defined in (7.2)

$$s_j = \frac{b_j - a_j}{\max\{a_j, b_j\}} \tag{7.2}$$

where a_j is the average distance between the j^{th} node and all the other nodes included in the i^{th} cluster, and b_j the average distance between node j and all the other nodes *not* in cluster i. Here a node j is assigned to cluster i if $u_{ij} = \max\limits_{i \in \{1,\ldots,c\}} \{u_{ij}\}$. The value of s_j will range in the region $[-1, 1]$. A value close to 1 indicates node j to be well clustered, i.e. appropriately assigned to cluster i. A value for s_i in the region of zero indicates that node j may well be assigned to a neighboring cluster, and a value close to -1 indicates node j to be misclassified, i.e. assigned to the wrong cluster. For cluster i one may then determine a silhouette value S_i, defined by (7.3)

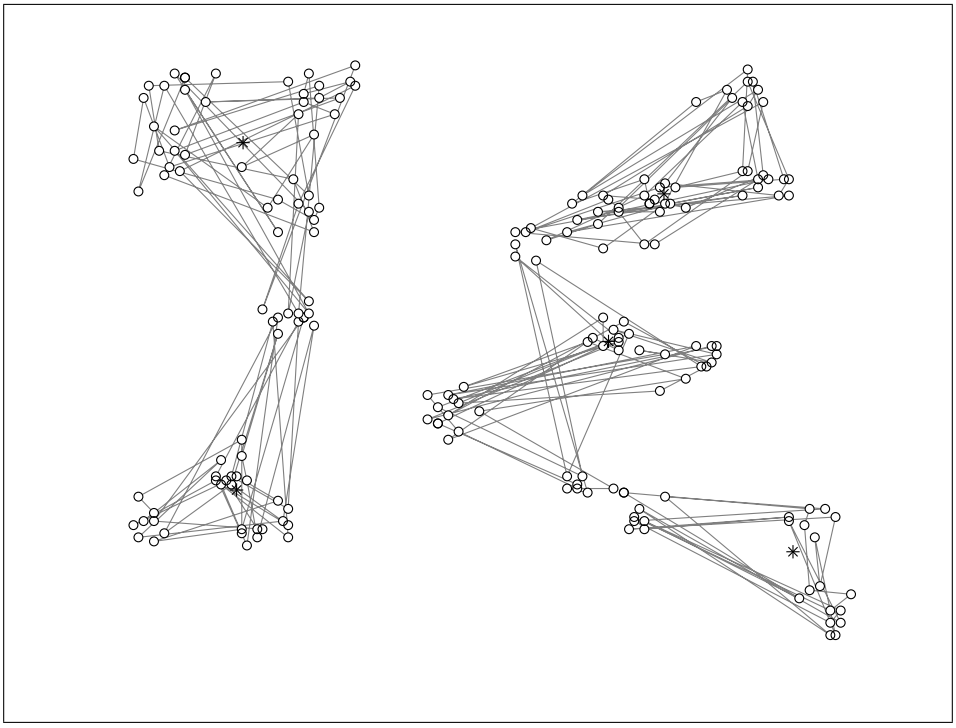
$$S_i = \frac{1}{m} \sum_{j=1}^{m} s_j \tag{7.3}$$

where m is the number of samples in cluster i. The global silhouette value V_s as defined by (7.4) is an effective index.

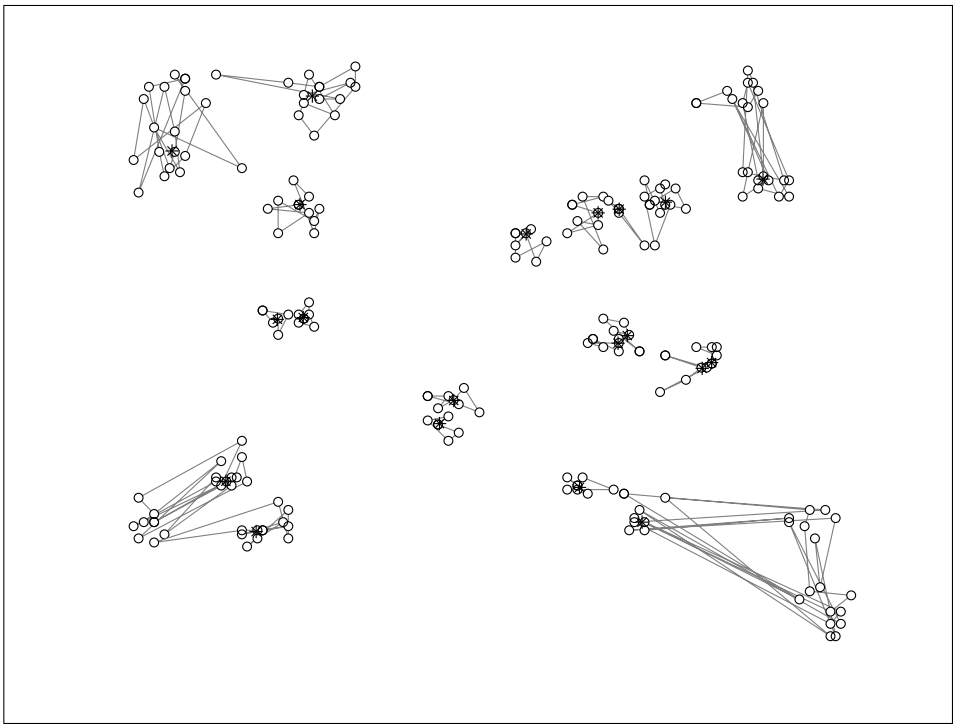
$$V_s = \frac{1}{c} \sum_{i=1}^{c} S_i \tag{7.4}$$

The *Partition Coefficient* index is defined by (7.5)

$$V_{PC} = \frac{1}{n} \left(\sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^2 \right) \tag{7.5}$$



(a) 5 Clusters



(b) 20 Clusters

Figure 7.1: Clustering the $C1$-2-1 problem set

where u_{ij} is the fuzzy membership for node j belonging to cluster i, and n the number of nodes in the input data, excluding the depot. The value of V_{PC} is in the range $[1/c, 1]$. An index close to 1 indicates good cluster separation, while a low index value indicates fuzzier clustering. An index of $V_{PC} = 1/c$ indicates that there is no clustering tendency. The disadvantages of V_{PC} are the lack of direct connection to a geometrical property, and the monotonic decreasing tendency with c.

The *Partition Entropy* index is defined by (7.6)

$$V_{PE} = -\frac{1}{n}\left(\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}\log\left(u_{ij}\right)\right) \tag{7.6}$$

The value of V_{PE} is in the range $[0, \log c]$. In contrast to PC, a low value of V_{PE} indicates good cluster separation. Unfortunately the same disadvantages as for V_{PC} hold for V_{PE} in that there is not direct connection to a geometrical property, and the index has a monotonic decreasing tendency with c. The following indices involve not only the membership value, but also the actual data set.

In the following indices the numerical taxonomy of Bezdek (1974) is used. Xie and Beni (1991) introduced an index that give weight to both compactness, and separation. First the *fuzzy deviation* of node j from cluster i, denoted by d_{ij} is determined as the Euclidean distance between node j and cluster i, weighted by the fuzzy membership of node j belonging to cluster i. The sum of the squares of the fuzzy deviations of each node j is referred to as the *variance* of cluster i, denoted by σ_i. The total variation of the data set with respect to the given fuzzy c-partition is referred to as σ. The compactness of the partition is the ratio between the total variation of the data set to the size of the data set, expressed as $\frac{\sigma}{n}$. The centers between all cluster center combinations $i, r \in \{1, \ldots, c\}, i \neq r$ is calculated, and the minimum inter-center distance is denoted by $d_{\min}$. The separation of clusters is then determined by $s = d_{\min}^2$. A high value of s indicates well-separated clusters. The index is the minimum value for $\frac{\sigma}{n \cdot s}$, or more explicitly written in (7.7).

$$V_{XB} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^2 \left\|\boldsymbol{x}_j - \boldsymbol{v}_i\right\|^2}{n\left(\min_{i,r\in\{1,\ldots,c\}, i\neq r}\left\{\left\|\boldsymbol{v}_i - \boldsymbol{v}_r\right\|^2\right\}\right)} \tag{7.7}$$

Pal and Bezdek (1995) extend the Xie-Beni index for cases where the fuzzy factor $m \neq 2$,

and the extended index V_{XB}^{+} is defined in (7.8)

$$V_{XB}^{+} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} \left\|\boldsymbol{x}_j - \boldsymbol{v}_i\right\|^2}{n\left(\min_{i,r\in\{1,\ldots,c\},i\neq r}\left\{\left\|\boldsymbol{v}_i - \boldsymbol{v}_r\right\|^2\right\}\right)} \tag{7.8}$$

Kwon (1998) also investigates the Xie-Beni index, and proposes an index that eliminates the monotonically decreasing tendency as the number of clusters increases and approaches n, the number of nodes in the data set. The index is denoted by V_K and is defined in (7.9).

$$V_K = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{2} \left\|\boldsymbol{x}_j - \boldsymbol{v}_i\right\|^2 + \frac{1}{c}\sum_{i=1}^{c}\left\|\boldsymbol{v}_i - \overline{\boldsymbol{v}}\right\|^2}{n\left(\min_{i,r\in\{1,\ldots,c\},i\neq r}\left\{\left\|\boldsymbol{v}_i - \boldsymbol{v}_r\right\|^2\right\}\right)} \tag{7.9}$$

The second term in the numerator is an *ad hoc* punishing function used to eliminate the decreasing tendency when c becomes large and close to n. The center of the data set is denoted by $\overline{v}$.

The *Fukuyama-Sugeno* index (as cited by Kim et al. (2003); Rao and Srinivas (2006); Xu and Brereton (2005)) is defined by (7.10)

$$V_{FS} = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}\left(\left\|\boldsymbol{x}_j - \boldsymbol{v}_i\right\|^2 - \left\|\boldsymbol{v}_i - \overline{\boldsymbol{v}}\right\|^2\right) \tag{7.10}$$

The weighted membership value is multiplied by the difference between the distance between nodes and its cluster centers, and the distance between cluster centers and the data center. A small value represents a well-separated and compact cluster.

The *Compose Within and Between Scattering* index was introduced by Rezaee et al. (1998) and is defined by (7.11).

$$V_{CWB} = \alpha Scat(c) + Dis(c) \tag{7.11}$$

where

$$Scat(c) = \frac{\frac{1}{c}\sum_{i=1}^{c}\left[\sigma(\boldsymbol{v}_i)^T \cdot \sigma(\boldsymbol{v}_i)\right]^{\frac{1}{2}}}{\left[\sigma(\boldsymbol{X})^T \cdot \sigma(\boldsymbol{X})\right]^{\frac{1}{2}}} \tag{7.12}$$

$$Dis(c) = \frac{D_{\max}}{D_{\min}}\sum_{i=1}^{c}\left(\sum_{r=1}^{c}\left\|\boldsymbol{v}_i - \boldsymbol{v}_r\right\|\right)^{-1} \tag{7.13}$$

$$\sigma(\boldsymbol{X}) = \frac{1}{n}\sum_{j=1}^{n}\left\|\boldsymbol{x}_j - \overline{\boldsymbol{v}}\right\|^2 \tag{7.14}$$

$$\sigma(\boldsymbol{v}_i) = \frac{1}{n}\sum_{j=1}^{n} u_{ij}\left\|\boldsymbol{x}_j - \boldsymbol{v}_i\right\|^2 \qquad \forall i \in \{1,\ldots,c\} \tag{7.15}$$

$$D_{\max} = \max_{i,r \in \{1,\ldots,r\}, i \neq r} \{\boldsymbol{v}_i - \boldsymbol{v}_r\} \tag{7.16}$$

$$D_{\min} = \min_{i,r \in \{1,\ldots,r\}, i \neq r} \{\boldsymbol{v}_i - \boldsymbol{v}_r\} \tag{7.17}$$

$$\alpha = Dis(c_{\max}) \tag{7.18}$$

The V_{CWB} tends to find an optimum between compactness and separation. $Scat(c)$ denotes the average scattering (compactness) for the c clusters, while $Dis(c)$ denotes the distance between cluster centers (separation). With $Scat(c)$ taking on much smaller values than $Dis(c)$, a scaling factor α is introduced to balance the two terms' opposite trends. $D_{\max}$ and $D_{\min}$ are the maximum and minimum distances between clusters. The authors perform the validation over cluster partitions with values $2 \leq c \leq c_{\max}$. In the application of this thesis a cluster is considered to be more than 5 nodes, hence $c_{\max} = \frac{n}{5}$.

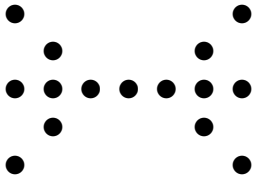
7.2 Evaluating fuzzy membership parameters

Three test sets for clusters have been found in literature, and one set is proposed in this thesis. Test sets are used to determine the *effectiveness* of a clustering algorithm as a function of the fuzzy factor m.

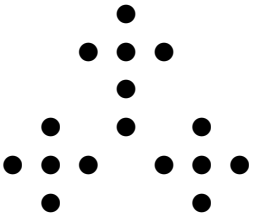
Kwon (1998) suggests the data sets illustrated in Figures 7.2(a) through 7.2(c) with two, three, and four clusters, respectively. A fourth data set, having five clusters, is proposed in this thesis and is illustrated in Figure 7.2(d).

All data sets are validated with an iteration limit of $k = 10000$ and a convergence threshold of $\varepsilon = 1 \times 10^{-12}$. The first three data sets provided by Kwon (1998) were tested for $c = \{2, 3, \ldots, 10\}$ clusters, while the fourth data set is tested for $c = \{2, 3, \ldots, 30\}$ clusters. Results of the cluster validation is provided in Appendix D in Tables D.1 through D.4. Incorrect predictions for the number of clusters in a data set are boxed. Through observation it can be seen that the best results are obtained with the fuzzy factor in the region $1.5 \leq m \leq 2.0$. The best performing validation indices are the Xi-Beni index, V_{XB}, and the enhanced Xi-Beni index, V_{XB}^{+}. As expected, these two indices perform very similar in close proximity of $m = 2.0$, and become identical in the value $m = 2.0$.

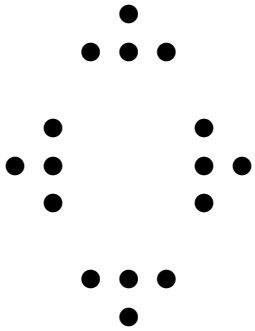
It is therefor proposed that either V_{XB} or V_{XB}^{+} be used when benchmark data sets' clustering is validated. Furthermore, a fuzzy factor of $m \in \boldsymbol{M} = \{1.5, 1.6, 1.7, 1.8, 1.9, 2.0\}$ is proposed.



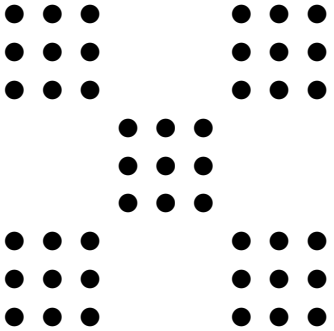
(a) Data set with two clusters (Kwon, 1998)



(b) Data set with three clusters (Kwon, 1998)



(c) Data set with four clusters (Kwon, 1998)



(d) Data set with five clusters

7.3 Validation of benchmark data

Cluster validation is performed for each of the ten problems of each of the six benchmark sets. Although somewhat expected, it is interesting to report that the results for all problems in a given data set are *exactly* the same. Table E.1 therefor does not report the results for each problem, but rather for each class.

Each sub-table shows the optimal number of clusters for a specific fuzzy factor, m, as well as the corresponding validation index value for both the Xi-Beni index, V_{XB}, and the extended Xi-Beni index, V_{XB}^{+}. It is noticeable that the optimal number of clusters is much lower than expected, especially for data sets $C1$ (4 clusters) and $C2$ (2 clusters). One might have expected a number in the region of 20 when referring to Figure 7.1.

The index values for the problem sets $R1$ and $R2$ are also lower than expected, indicating good clusteredness and separation. The index values are significantly (approximately double) higher than the values for clustered problem sets, but one might have expected values indicating much worse clusteredness.

7.4 Conclusion

In this chapter, fuzzy c-means clustering is introduced as a mechanism to establish the level of geographical clusteredness of vehicle routing benchmark problem sets. The two values of interest in the cluster validation is the optimal number of clusters identified, and the validation index value. The latter provides insight to the level of clusteredness of a data set, for example the index values for the set $RC1$ (semi-clustered) is between that of the set $C1$ (clustered), and set $R1$ (random).

In the next chapter, these values will be used, along with a time window width analysis, to train a neural network so that new data sets could be tested to determine which problem set it resembles best.