ANS PSA 2011 International Topical Meeting on Probabilistic Safety Assessment and Analysis
Wilmington, NC, March 13-17, 2011, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2011)

Data Processing Methodologies Applied to Dynamic PRA: an Overview

Diego Mandelli*, Alper Yilmaz and Tunc Aldemir

The Ohio State University
mandelli.1@osu.edu; yilmaz.15@osu.edu
aldemir.1@mecheng.ohio-state.edu

ABSTRACT

The use of dynamic event trees (DETs) can serve as a powerful tool for the dynamic probabilistic risk assessment (DPRA) of nuclear power plants. The DETs have the capability to more accurately model the complex interactions and events which may occur during a transient. One of the challenges of DPRA through DETs is the management of the resulting very large data sets. Hence, the need for a methodology able to handle high volumes of data in terms of both cardinality (due to the high number of uncertainties included in the analysis) and dimensionality (due to the complexity of systems) arises. Hierarchical and partitional clustering methodologies are compared and evaluated with regard to their potential to analyze large scenario datasets generated by DETs using several different data sets.

Key Words: Clustering, Data Analysis, Dynamic PRA

1 INTRODUCTION

The Event Tree (ET)/Fault Tree (FT) approach [1] is the traditional tool for probabilistic safety/risk assessment (PSA/PRA) not only for nuclear systems but also for the aerospace, chemical and transportation industries. However, several concerns have been raised about the capability of the ET/FT approach to treat the coupling between the plant physical processes and triggered or stochastic logical events [2] which can have significant impacts on the consequences of upset conditions and their frequencies. Another concern is the contribution of epistemic uncertainties to the ordering of events and consequences of upset conditions. As discussed in [3], a safety methodology has to be able to:

- Model the dynamics of the system and, hence, needs to be coupled with system or plant simulators
- Model the exact time scale of the accident
- Model the change of hardware component states
- Model human interaction with the system dynamics
- Handle epistemic and aleatory uncertainties

Dynamic PSA/PRA methodologies [3] respond to these needs by using advanced system simulators to identify the timing of events and to account for the coupling between triggered and/or stochastic events.

* Corresponding author: Diego Mandelli, Nuclear Engineering Program, The Ohio State University, 201 W. 19th Ave., 43210 Columbus (OH), USA; email: mandelli.1@osu.edu.

A challenging aspect of dynamic methodologies, such as the DET methodology [4], is the large number of scenarios generated for a single initiating event. Such large amounts of information can be difficult to organize for tractable analysis. In particular, as part of the PSA/PRA framework, it is important to identify the main scenario evolutions and the main risk contributors for each initiating event. In this work, we want to address this problem of data analysis by grouping the scenarios into clusters and analyzing the properties of the scenarios of each cluster.

By scenario clustering we mean two actions:

1. Identify the scenarios that have a "similar" behavior (i.e. identify the most evident classes)
2. Decide for each scenario which class it belongs to (i.e., classification)

When dealing with nuclear transients, it is possible to analyze the set of scenarios in two possible modes:

- *End State Analysis*: Classify the scenarios into clusters based on the end state of the scenarios
- *Transient Analysis*: Classify the scenarios into clusters based on the time evolution of the scenarios

While the first mode has been widely used in the traditional ET/FT analysis, the second one is only starting to be considered in the recent years. The volume of data is not the only the only challenge we need to deal with. We also want to accomplish the following:

- Discover clusters with arbitrary shapes
- Deal with noise and outliers
- Achieve interpretability and usability

2 CLUSTERING: AN OVERVIEW

From a mathematical viewpoint, the concept of clustering [5] considered here is to find a partition $C=\{C_1, \dots, C_K\}$ of the data set $X=\{x_1, \dots, x_j, \dots, x_N\}$ where each data point x_j can be represented as a d-dimensional vector $x_j = (x_{j1}, x_{j2}, \dots, x_{jd})$ where each x_{ji} is said to be a feature (attribute, dimension or variable). A partition C of X is such that:

$$\begin{cases} C_i \neq \emptyset, \ i = 1, \dots, K \\ \bigcup_{i=1}^{K} C_i = X \end{cases} \tag{1}$$

A loose definition of clustering is the process of organizing objects into groups whose members are, in some way, similar. A cluster is therefore a collection of objects which are "similar" to each other and are "dissimilar" to the objects belonging to other clusters according to a specific distance[1] [6].

As shown in Fig. 1, the main division between clustering methodologies can be made by partitioning them in two classes:

[1] In [6] it is possible to find several types of distances. In this article we will focus on the Euclidean distance, i.e., in a d-dimensional space, the distance between two data points $x=(x_1, \dots, x_i, \dots, x_d)$ and $y=(y_1, \dots, y_i, \dots, y_d)$ is:

$$d(x,y) = \left(\sum_{i=1}^{d} (x_i - y_i)^2\right)^{1/2}$$

- Hierarchical [5]
- Partitional [5]

Hierarchical algorithms organize data into a hierarchical structure accordingly to a proximity matrix in which an entry (j, k) is some measure of the similarity (or distance) between the items to which row j and column k corresponds. Usually, the final result of these algorithms is a binary tree, also called dendrogram (see Section 5), in which the root of the tree represents the whole data set and each leaf is a data point.

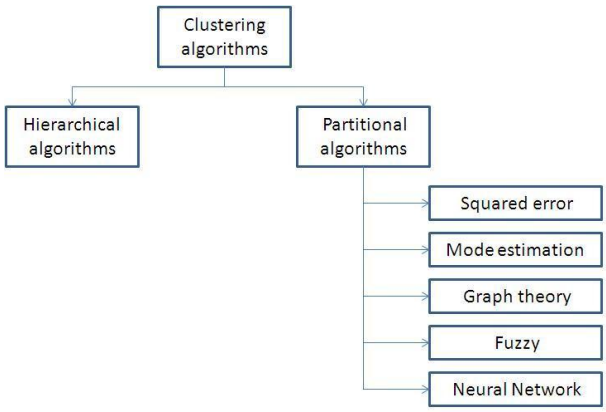


Figure 1 - Taxonomy of clustering methodologies [7].

Partitional clustering seeks for a single partition of the data sets instead of nested sequence of partitions obtained by hierarchical methodologies. Under this category it is possible to classify methodologies under five main sub-categories [7]:

- *Squared Error* assigns each point to the cluster whose center (also called centroid) is nearest. The center is the average of all the points in the cluster, that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster. The most famous and used methodology is the K-Means algorithm [7].
- *Mode Estimation* is based on the assumption that the distribution of the points in the state space can be described through a probability density function (pdf). The goal is to find the modes, i.e. the regions in the state space with higher data densities. An example of this kind of methodology is the Mean-Shift methodology [8].
- *Graph Theory* based methodologies aim to build a graph of the data set often called Minimal Spanning Tree. Clusters are determined by deleting the longest edges of the graph. Conceptually this approach is very similar to the hierarchical one.
- *Fuzzy clustering* assigns, each point a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, points on the edge of a cluster may be in the cluster to a lesser degree than points in the center of cluster. For each point x we have a coefficient $u_k(x)$ giving the degree of being in the k^{th} cluster.
- *Neural Network* methodologies are essentially inspired by the biological neural network. The learning process associated with the training of an artificial neural network (ANN) allows to

associate patterns (input variables) to clusters (output nodes) through a series of weights that are updated at each iteration.

3 CLUSTERING METHODOLOGIES

In this section we, provide an overview of describe four clustering methodologies.

3.1 Hierarchical Algorithm

Hierarchical algorithms [7] organize the data set into a hierarchical structure according to a proximity matrix. Each element $d(i, j)$ of this matrix contains the distance between the i^{th} and the j^{th} cluster center. The final result of this technique is a dendrogram. This kind of representation has the advantages of providing a very informative description and visualization of the data structure even for high values of dimensionality.

The procedure to determine the dendrogram for a data set of n points in a d-dimensional space is the following:

1. Start the analysis with a set of n clusters (i.e., each point is considered as a cluster).
2. Determine the proximity matrix M (dimension: $n \times n$): $M(i, j) = d(x_i, x_j)$ where x_i and x_j are the position of the i^{th} and the j^{th} cluster.
3. For each point p find the closest neighbor q from the proximity matrix M
4. Combine the points p and q
5. Repeat steps 2, 3 and 4 until all the points of the data set are in the same cluster

The advantage of this kind of algorithm is the nice visualization of the results that show the underlying structure of the data set. However, the computational complexity for most of the hierarchical algorithm is of the order of (n^2) (where n is the number of points in the data set).

3.2 K-Means Algorithm

As mentioned in the last section K-Means clustering algorithms [7] belong to the more general family of Squared Error algorithms. The goal is to partition n data points x_i into K clusters in which each data point maps to the cluster with the nearest mean. The stopping criterion is to find the global minimum of the error squared function $\mathcal{X}$ defined as:

$$\mathcal{X} = \sum_{i=1}^{K} \sum_{x_j \in C_i} (x_j - \mu_i)^2 \quad (2)$$

where μ_i is the centroid of the i^{th} cluster.

The procedure to determine the centroids μ_i of K clusters ($C_1, \dots, C_K$) is the following:

1. Start with a set of K random centroids distributed in the state space,
2. Assign each pattern to the closest centroid,
3. Determine the new K centroids accordingly to the point-centroid membership:

$$\mu_i = \frac{1}{N_i} \sum_{x_j \in C_i} x_j$$

where N_i corresponds to the number of data points in the i^{th} cluster,

4. Repeat 2 and 3 until convergence is met (i.e., until a minima of the function $\mathcal{X}$ is reached).

The K-Means algorithm is one of the most popular and widely used algorithms due in part to the fact that is very easy to implement and the computational time is directly proportional to the cardinality of data points. The main disadvantage is that the algorithm is sensitive to the choice of the initial partition and may converge to a local minimum of the error squared function. In addition, the number of clusters should be known ahead of time.

3.3 Fuzzy C-Means Algorithm

Fuzzy C-Means clustering [7] is a clustering methodology that is based on fuzzy sets and, hence, it allows a data point to belong to more than one cluster. Similar to the K-Means clustering, the objective is to find a partition of *C* fuzzy centers to minimize the function J defined as following:

$$\mathrm{J} = \sum_{i=1}^{K} \sum_{j=1}^{C} u_{i,j}^{m} (x_i - \mu_j)^2 \tag{3}$$

where:

- $U = [u_{ij}]$ is the fuzzy partition matrix,
- each element $u_{ij} \in [0, 1]$ of U is the membership coefficient of the j^{th} data point for the i^{th} cluster,
- $m \in [0, \infty)$ is the fuzzification parameter (usually set to $m = 2$),
- μ_j is the centroid of the j^{th} cluster center.

The procedure to determine the centroids μ_j (i=1, …, K) of C clusters is the following:

1. Initialize the $U = [u_{ij}]$ matrix,
2. Calculate the set of C centroids $\{ \mu_1 , \dots , \mu_K \}$ as following:

$$\mu_j = \frac{\sum_{i=1}^{N} u_{i,j}^{m} x_i}{\sum_{i=1}^{N} u_{i,j}^{m}}$$

3. Update the matrix $U = [u_{ij}]$ as following:

$$u_{i,j}^{m} = \frac{1}{\sum_{k=1}^{C} \left(\frac{x_i - \mu_j}{x_i - \mu_k} \right)^{\frac{2}{m-1}}}$$

4. Repeat 2 and 3 until convergence is met (i.e., until a minima of the function J is reached).

It is very easy to see that the Fuzzy C-Means clustering is very similar to the K-Means one. As for the K-Means, Fuzzy C-Means can also converge to a local minimum. Fuzzy C-Means algorithms can be useful when the boundaries among clusters are not well separated and ambiguous.

3.4 Mean-Shift algorithm

The main idea behind the Mean-Shift algorithm [8] is to consider each point x_i (i =1, …, N) of the data set as an empirical distribution density function $K(x_i)$ distributed in a d-dimensional space (blue line in Fig. 2 for the 1-D case) where regions with high data density (i.e., modes) corresponds to local maxima of the global probability density function $f_N(x)$ [8] defined as following (red line in Fig. 2 for the 1-D case):

$$f_N(x) = \frac{1}{N\,h^d}\sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right) \tag{4}$$

where each element $x_i \in \mathbb{R}^d$ and h is a scalar parameter called bandwidth which indicates the level of refinement of the cluster analysis. The function $K(x)$: $\mathbb{R}^d \mapsto \mathbb{R}$ is the distribution density associated to each data point which is also called the kernel.

1. Starting from a data point x data search all points x_i within bandwidth radius and determine the average data point $\underline{x}$ as following:

$$\underline{x} = \frac{\sum_{i=1}^{N} x_i\, K\left(\frac{x-x_i}{h}\right)}{\sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right)}$$

2. Move from x to $\underline{x}$ and repeat 1
3. Repeat 1 and 2 until convergence is met:

$$\left|\underline{x}^{(r+1)} - \underline{x}^{(r)}\right| < \varepsilon$$

   where $\underline{x}^{(r)}$ indicates $\underline{x}$ at iteration r
4. Repeat 1 through 4 for each data point

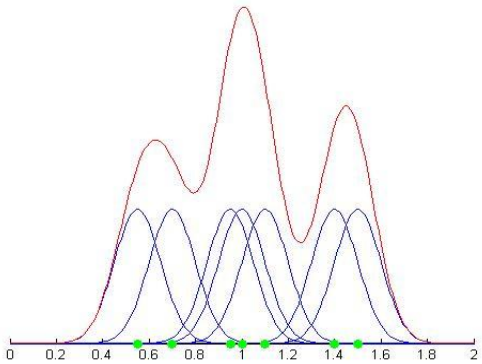


Figure 2 - Density function (red line) for points distributed in a 1-dimensional space modeled using kernels (blue lines)

The advantage of this class of algorithms is that they are able to identify clusters with arbitrary shapes and, hence, they are not limited to topological figures such as spheres or ellipsoids. Moreover, compared to K-Means (see Section 3.2) and Fuzzy C-Means (see Section 3.3) the number of clusters is not specified a-priori by the user but it is the algorithm that

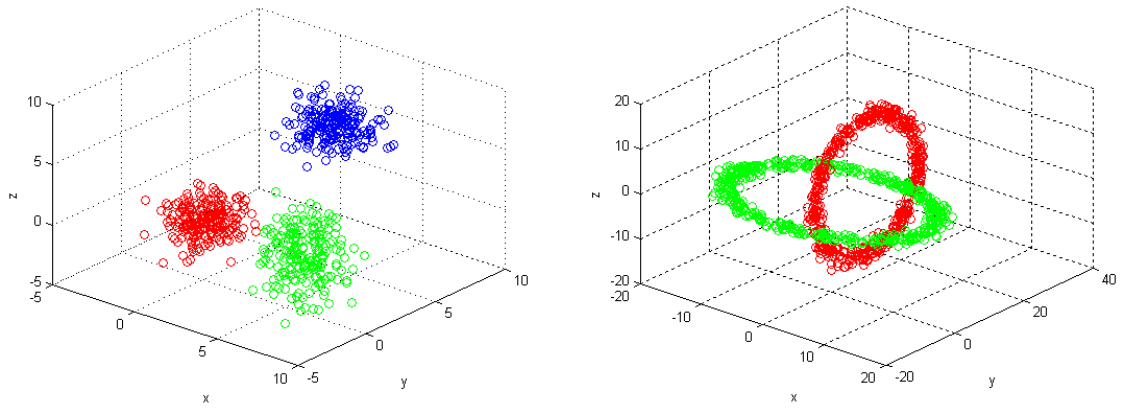
determines this number based on the areas with higher point concentration using the value of the bandwidth, *h*, chosen.

4 DATA SETS

In order to compare the four methodologies listed above we selected three different data sets as described in Figs. 3, 4 and 5:

1. A set of 300 points grouped in 3 spherical clusters (see Fig. 3a),
2. A set of 200 points distributed in 2 rings (see Fig. 3b),
3. A set of scenarios described in [9] (see Fig. 4).

We choose the first data set as applicative examples of data sets having different cluster geometries. The scope is to evaluate performances of the four algorithms for different cluster geometries.



(a) (b)

Figure 3 – Representation of the Dataset 1 (a) and 2 (b)

The last data set considers scenarios that have been generated for the analysis of a pressurized water reactor. The initiating event investigated was that of a station blackout (SBO) and the MELCOR code [10] was linked to the ADAPT tool [4] to determine the evolution of each DET scenario. The simulations using MELCOR model the transient from the occurrence of the SBO through the core melting phase and up to point of containment failure and release of radionuclides to the environment. All the 104 scenarios generated in this DET led to containment failure at some point in the scenario evolution. For the purposes of this paper, we choose 4 state variables of interests (see Fig. 4):

1. Core water level [m]: *L*,
2. System Pressure [Pa]: *P*,
3. Intact core fraction [%]: *CF*,
4. Fuel Temperature [K]: *T*.

We sampled each state variable 100 times, which gave us an accurate description of all the 104 transients. We chose to represent each scenario x_i as a multidimensional vector:

$$x_i = [L(1), \dots, L(100), P(1), \dots, P(100), CF(1), \dots, CF(100), T(1), \dots, T(100)] \quad (5)$$

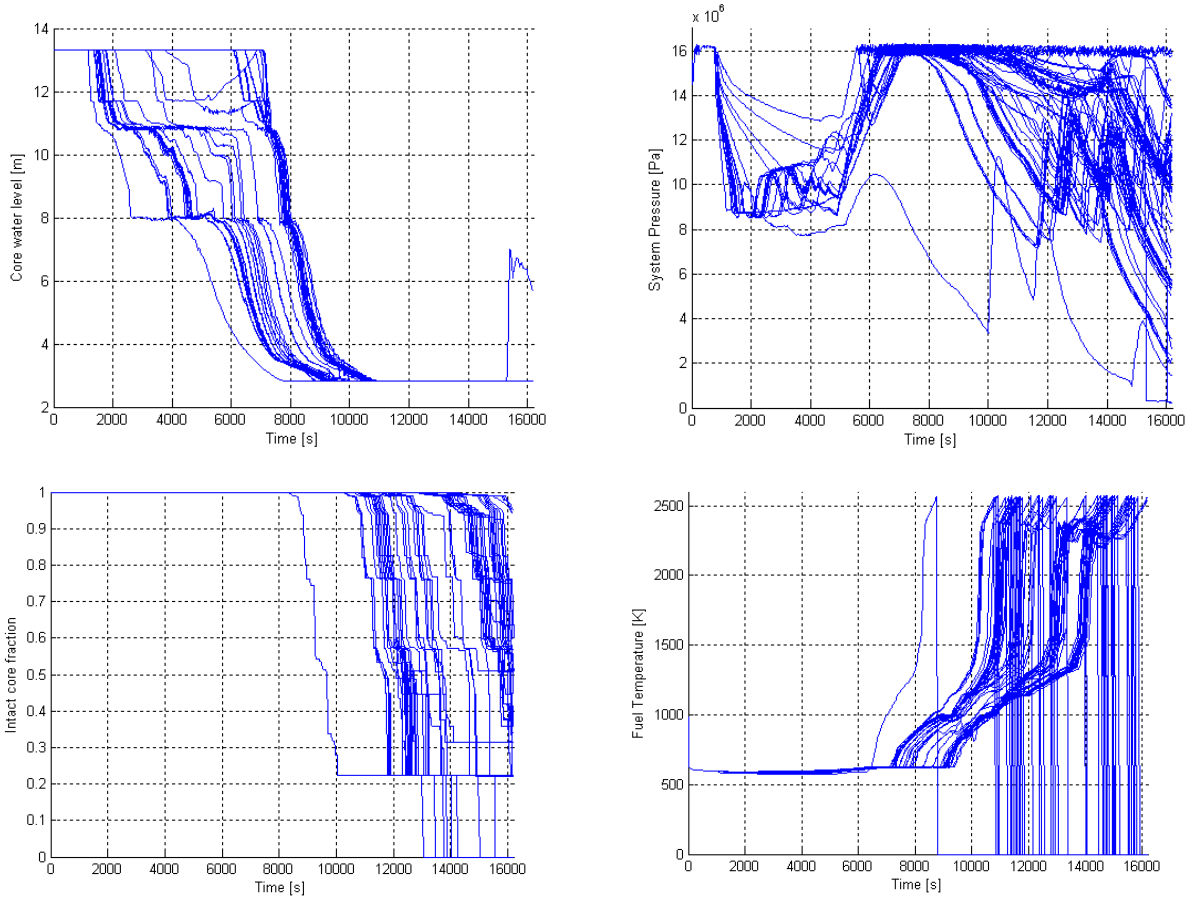


Figure 4 – Representation of the Dataset 3 [9]

5 COMPARISON OF CLUSTERING METHODOLOGIES

In order to compare the results for the methodologies presented in Section 3, for Datasets 1 and 2, we will compare the cluster centers obtained from the each methodology with the original ones.

Tables 1 and 2 show the comparison of the cluster centers for K-Means, Fuzzy C-Means and Mean-Shift using the Datasets 1 and 2 described in Section 4. For the Dataset 1 it is possible to note a general agreement between the three methodologies (Table 1). However, Table 2 shows major disagreements for the K-Means and the Fuzzy C-Means methodologies. The reason of these discrepancies for the 2 methodologies is due to the fact that the shape of the two clusters is not spherical or ellipsoidal and, hence, both the K-Means and the Fuzzy C-means algorithms have problems to find 2 clusters with a more complex shapes such as the two rings pictured in Fig.3b.

Table 1 Cluster centers comparison for Dataset 1

Original	K-Means	Fuzzy C-Means	Mean-Shift
(0,0,0)	(-0.0173,0.0177, -0.0335)	(-0.019,0.0033,-0.0311)	(-0.012, 0.0055,-0.0358)
(6,0,0)	(5.9510,-0.0008,-0.0599)	(6.381,0.128,-0.201)	(6.163,0.075,-0.0432)
(3,6,6)	(2.974,6.01,6.118)	(2.51,5.4878,5.98)	(3.02,5.893,6.011)

Table 2 Cluster centers comparison for Dataset 2 (differences are highlighted in bold)

Original	K-Means	Fuzzy C-Means	Mean-Shift
(0,0,0)	**(0.142,3.632,1.01)**	**(-0.737,3.866,0.422)**	(0.101, -0.0732,0.127)
(0,5,0)	**(0.573,15.59,-0.311)**	**(0.246,15.4894,-0.342)**	(0.121,5.182,-0.095)

We also performed the hierarchical clustering for both Datasets 1 and 2. Fig. 5 shows the dendrogram Dataset 1 where it is possible to identify the 3 clusters and the hierarchical structure based on the reciprocal distance of the points. Regarding the Dataset 2, the algorithm was not able to identify the two clusters for the same reason presented for K-Means and Fuzzy C-Means.

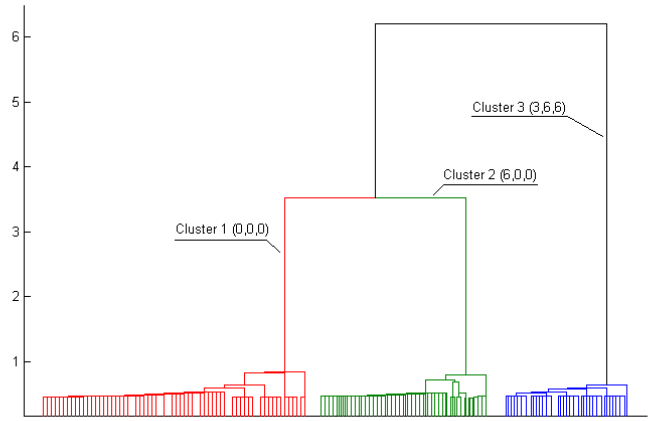


Figure 5 - Dendrogram derived from the Hierarchical clustering algorithm for Dataset 1.

For the Dataset 3 we do not have the exact solution and hence we chose one methodology as a comparison reference using the scenarios shown in Fig. 4. Mean-Shift methodology has proven to be more flexible in terms of identifying clusters with arbitrary shapes and, thus, we decided to use the cluster centers obtained from Mean-Shift as a reference. We performed the clustering using Mean-Shift with value of bandwidth equal to 20 and we obtained 8 cluster centers pictured in Fig. 6.

We then performed the clustering of the third dataset using K-Means and Fuzzy C-Means using the number of clusters (i.e., 8) obtained with the Mean-Shift as input. Results are shown in Fig. 7 and 8 for K-Means and Fuzzy C-Means, respectively. When we compared the cluster centers, we discovered that 5 out of 8 clusters had notable differences in terms of both cluster centers (pictured in Fig. 7 and 8) and scenario memberships. From the comparison of the results shown for the Datasets 1 and 2, we believe that the 8 clusters obtained using the Mean-Shift methodology have geometrical shapes that cannot be modeled using K-Means and Fuzzy C-

Means. However, all the three methodologies were able to identify outliers which are scenarios that belong to clusters having only very few elements.

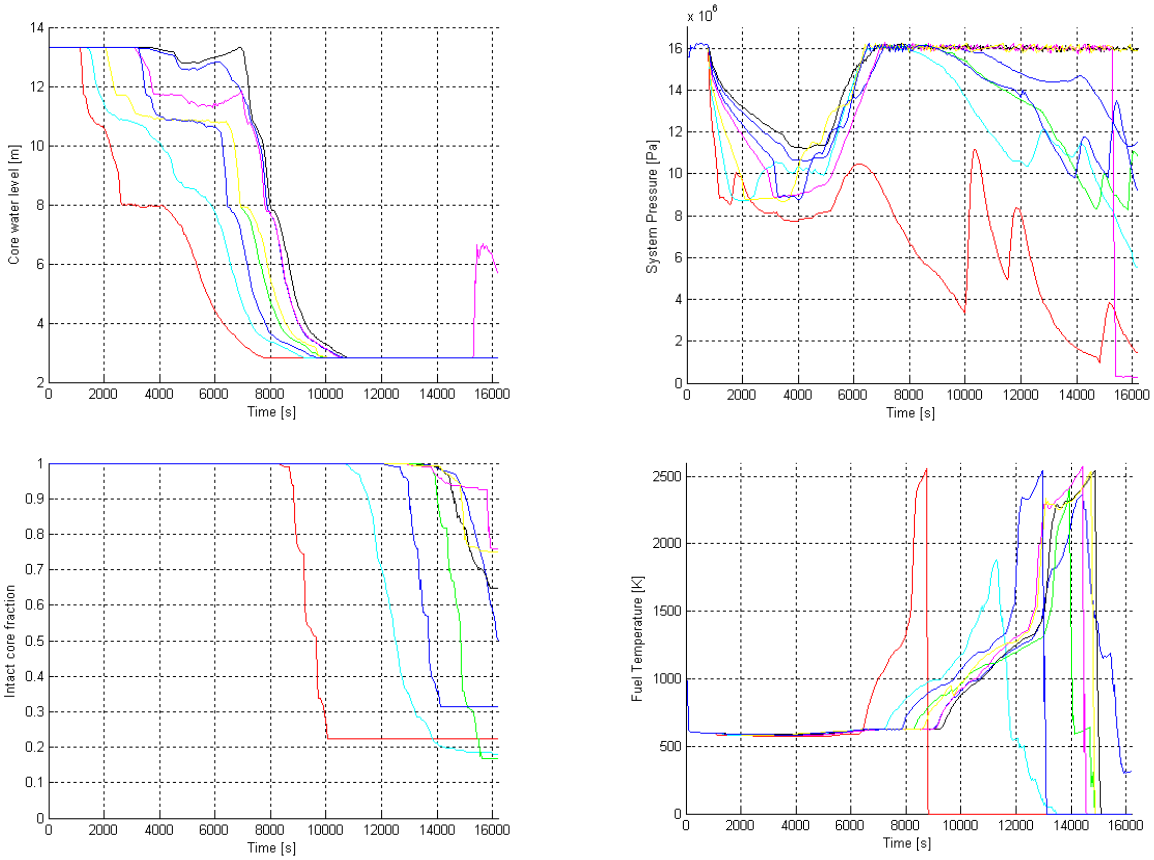


Figure 6 - Cluster centers obtained using Mean-Shift for Dataset 3

6 CONCLUSIONS

The major challenges in using dynamic PRA/PSA methodologies are the heavier computational and memory requirements compared to the classical ET analysis. Large volumes of data are generated and, hence, a large quantity of valuable information needs to be analyzed. Data clustering techniques that have been developed in the last decades offer tools to analyze and summarize large data sets. In this paper we described four different clustering methodologies: Hierarchical, K-Means, Fuzzy C-Means and Mean-Shift and we compared them using three different data sets.

Hierarchical clustering has the advantage that it is able to show the distribution of the data points through a dendrogram. This is useful when the dimensionality of the data points is greater than 3 and hence, it is not easy to graphically visualize the data set distribution. However, this ability is lost when the data points are distributed in cluster having complex geometries. K-Means and Fuzzy C-Means methodologies have the same disadvantage since they are able to

identify mainly spherical or ellipsoidal cluster of points. However, we found that Mean-Shift algorithm is able to overcome this limitation. All methodologies were able to identify outliers.

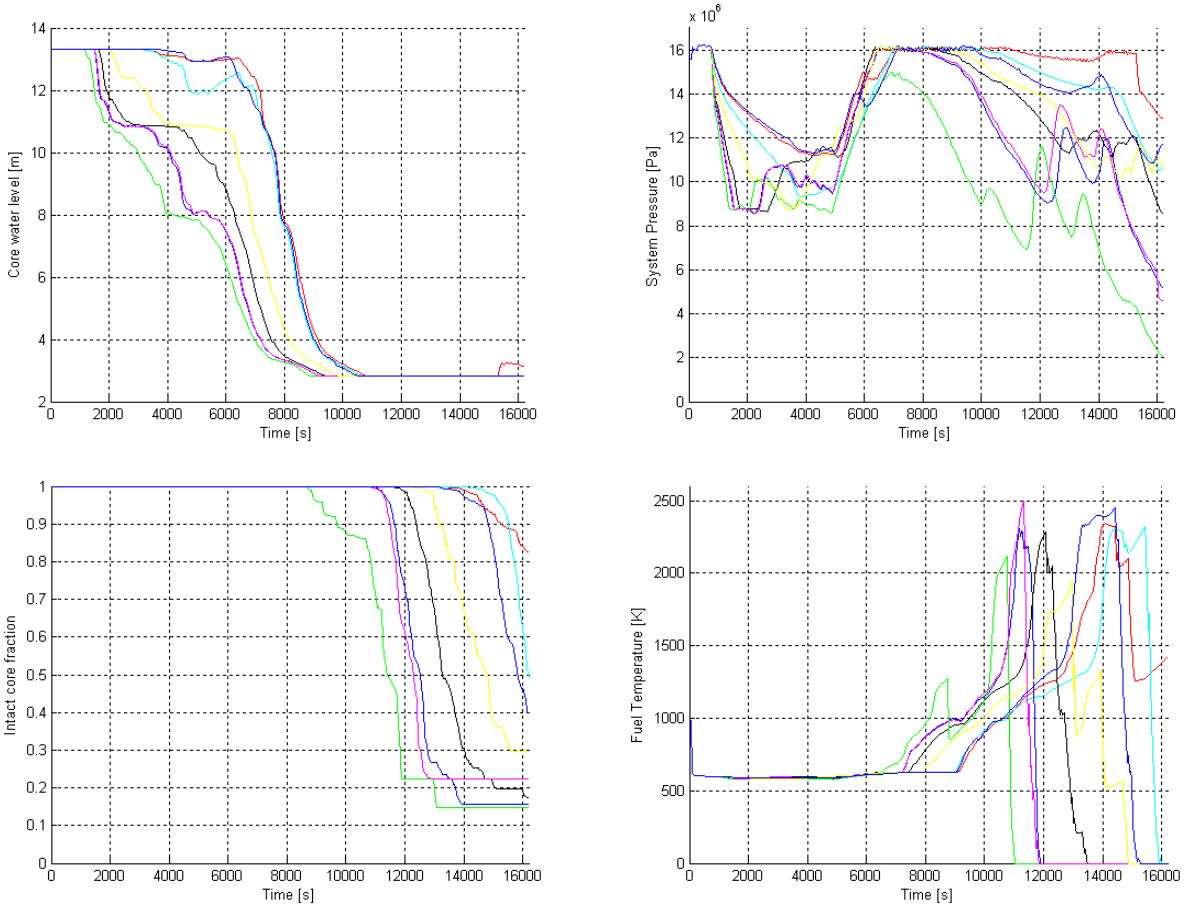


Figure 7 - Cluster Centers obtained using K-Means for Dataset 3

7 REFERENCES

[1] U.S. Nuclear Regulatory Commission, *NUREG 1150 - Severe accident risks: an assessment for five U.S. nuclear power plants*, Division of Systems Research, Office of Nuclear Regulatory Research, Washington, DC (1990).

[2] T. Aldemir, D. Miller, M. Stovsky, J. Kirschenbaum, P. Bucci, A. Fentiman, L. Mangan, and S. Arndt, *NUREG/CR 6901: Current state of reliability modeling methodologies for digital systems and their acceptance criteria for nuclear power plant assessments*. Division of Fuel, Engineering and Radiological Research, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington, DC (2006).

[3] N. Siu, "Risk assessment for dynamic systems: an overview," *Reliability Engineering and System Safety*, **43**, no. 1, pp. 43-73 (1994).

[4] A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, B. Rutt, and U. Catalyurek, "Dynamic generation of accident progression event trees," *Nuclear Engineering and Design*, **238**, no. 12, pp. 3457- 3467 (2008).

[5] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley-Interscience Publication (2000).

[6] B. Mendelson, *Introduction to Topology*, Dover Publications New York (NY), USA (1990).

[7] A. K. Jain, K. Dubes, and C. Richard, *Algorithms for clustering data*. Upper Saddle River, NJ (USA): Prentice-Hall, Inc. (1988).

[8] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, **21**, no. 1, pp. 32-40 (1975).

[9] D. Mandelli, T. Aldemir, and A. Yilmaz, "Scenario aggregation in dynamic PRA uncertainty quantification," in *Proceedings of the American Nuclear Society (ANS)*, Las Vegas (NV), **103**, pp. 371-374 (2010).

[10] R. O. Gauntt, *MELCOR Computer Code Manual, Version 1.8.5, Vol. 2, Rev. 2*. Sandia National Laboratories, NUREG/CR-6119.

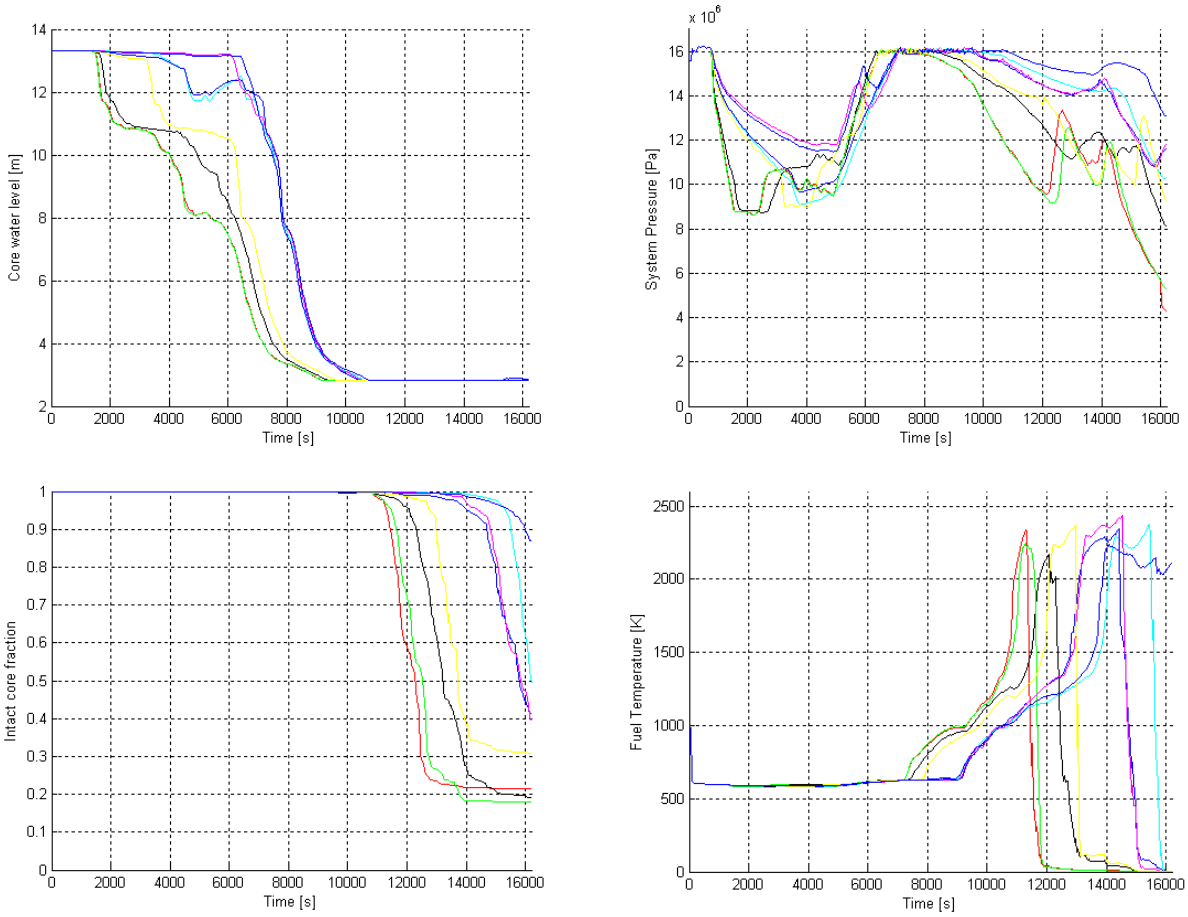


Figure 8 - Cluster **c**enters obtained using Fuzzy C-**M**eans for **D**ataset 3