Visualization and the understanding of multidimensional data using Genetic Algorithms: Case study of load patterns of electricity customers

VahidGolmah
Department of Computer Engineering, Azad University of Neyshabur
Neyshabur6621901 IRAN
v.golmah@in.iut.ac.ir

Jamshid Parvizian
Department of industrial Engineering, Isfahan University of Technology
Isfahan 8415683111 IRAN
japa@cc.iut.ac.ir

Abstract

Visualization is the process of transforming data, information, and knowledge into visual form, making use of humans' natural visual capabilities. Different methodologies are available for analyzing large multidimensional data sets and providing insights with respect to scientific, economic, and engineering applications. This problem has traditionally been formulated as a non-linear mathematical programming. In this paper, we formulate the data visualization problem as a quadratic assignment problem. However, this formulation is computationally difficult to solve optimally using an exact approach. Consequently, we investigate the use of the genetic algorithm for the data visualization problem. To examine capabilities of proposed method, we use a demand database by electricity customers, and compare the results with results by Self Organizing Maps (SOMs). This can be concluded that this approach generates higher quality output.

Keywords: Data Visualization, Genetic Algorithms, Data mining, Self Organizing Maps(SOMs).

1. Introduction

In the last twenty years, the volume of available data has increased exponentially because of the extensive use of electronic data gathering devices, such as point-of-sale remote sensing devices. Data-mining tools extract information from data by finding patterns, threads and relationships. To do this, a data-mining tool uses a hybrid of statistical and artificial intelligence methodologies such as pattern recognition, classification, categorization and learning [1, 2].

Data mining activities include both undirected and directed approaches. Directed data mining focuses on one target variable, whereas in undirected data mining, the goal is to understand the relationship amongst all of the variables. Data visualization is a key component of undirected data mining [3].

Since human vision is endowed with the classification ability for graphic figures, it would greatly help solving the problem if the data could be graphically transformed for visualization. However, human vision is only useful for figures with low dimensions [4].

The visualization of data plays an important role in data mining. This is a difficult problem since the data is usually high dimensional, that is, the number m of attributes is large, whereas the data can only be visualized in two or three dimensions. While it is possible to visualize two or three attributes at a time, a better alternative is often to map the data to two or three dimensions in a way that preserves the structure of the relationships (that is, distances) between instances [5].

More specifically, data visualization reveals relationships in data sets that are not evident from the raw data, by using mathematical techniques to reduce the number of dimensions in the data set while preserving the relevant inherent properties. The data are presented in visual form, usually in two or three dimensions. The smaller number of dimensions can be easily evaluated by human observation. The problem then involves taking a set of data points in a high-dimensional space and locating these points in a lower-dimensional space such that a relevant measure of distance is preserved [6]. This problem has traditionally been formulated as a non-linear mathematical programming problem.The major focus of this research is to apply the Genetic Algorithm to visualize data.

The rest of the paper is organized as follows. Section 2 reviews relevant works in data visualization. Section 3 introduces modeling the problem as a quadratic assignment problem. Section 4 discusses Genetic Algorithms to solve this problem in detail. Section 5 presents a case study to evaluate the technique, and compares it with Self Organizing Maps (SOM). Section 6 concludes the paper.

2. Literaturereview

The most common data visualization methods allocate a representation for each data point in a lower-dimensional space and try to optimize these representations so that the distances between them are as similar as possible to the original distances of the corresponding data items. The methods differ in that how the different distances are weighted and how the representations are optimized. Linear mapping, like principle component analysis, is effective but cannot truly reflect the data structure. Non-linear mapping, like Sammon projection [7], Multi-Dimensional Scaling (MDS) [5, 8] and Self OrganizingMap (SOM) [9, 10], requires more computation but is better at preserving the data structure.

2.1. Using linear equation to estimate nonlinear equation

The first approach involves solving the nonlinear equations relating these measurements iteratively.Commonly used techniques include linearization via Taylor series expansion [11, 12], steepest descent method [13],and Newton-type iteration [14].Although this approach can attain optimum estimation performance, it is computationally expensive, and sufficiently precise initial estimates are required to obtain the global solution. On the other hand, the second approach, which allows real-time realization and ensures global convergence, reorganizes the nonlinear equations into a set of linear equations by introducing an extra variable that is a function of the source position. The linear equations can then be solved straightforwardly by applying least squares (LS), as in the spherical interpolation (SI) technique [15, 16].

2.2. Sammon's Mapping (SM)

Sammon Jr. [7] introduced a method for nonlinear mapping of multidimensional data into a two- or three-dimensional space. This nonlinear mapping preserves approximately the inherent structure of the data and thus is widely used in pattern recognition.

Sammon indicated that one of the limitations of a nonlinear algorithm is the small number of vectors (data points) it can handle. Even with today's fast computers, nonlinear optimization techniques are usually slow and inefficient for large data sets. Discrete optimization techniques provide a possible alternative for the data visualization problem.

The procedure to minimize the mapping error is sensitive to the learning rate and is only practically useful for problems with low dimensions. De Backer *et al.* introduced a better algorithm to minimize the error function [8].

2.3.Multi-Dimensional Scaling (MDS)

MDS [5, 8] refers to a class of algorithms that visualize proximity relations of objects by distances between points in a low-dimensional Euclidean space. MDS algorithms are commonly used to visualize proximity data (i.e., pairwise dissimilarity values instead of feature vectors) by a set of representation points in a suitable embedding space. For detailed discussions on this subject, readers are referred to monographs by Borg and Groenen [5].

In MDS, the minimization problem is non-convex and sensitive to local minima. Klock and Buhmann (2000) developed a deterministic annealing approach to solve such minimization problems.

The only difference between Sammon's mapping and the nonlinear metric MDS is that (excluding the constant normalizing factor) the errors in distance preservation are normalized by the distance in the original space. Due to normalization, the preservation of small distances will be emphasized.

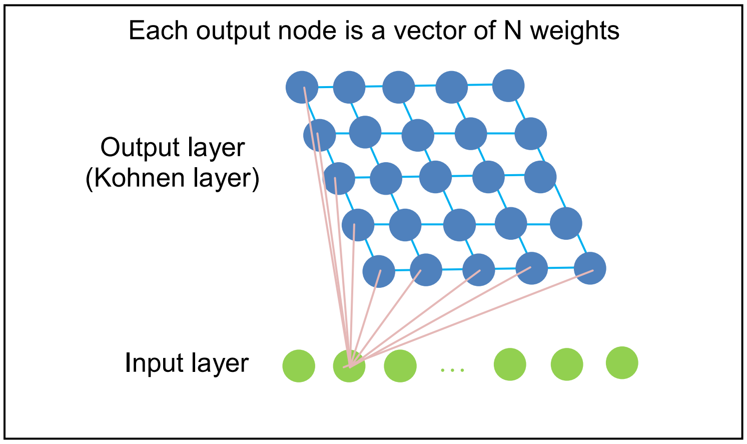

Fig. 1. The architecture of SOM in mapping to 2dimensional.

2.4. Self-OrganizingMaps (SOM)

SOM, proposed by Kohonen [9, 10], is a class of neural networks trained in an unsupervised manner, using competitive learning. It is a well-known method for mapping a high-dimensional space onto a low-dimensional one. It is popular to consider mapping onto a two-dimensional grid of neurons. The method allows putting complex data into order, based on their similarity,

and shows a map by which the features of the data can be identified and evaluated.A variety of realizations of SOM has been developed [9, 17].

The SOM architecture consists of two fully connected layers: an input layer and a Kohonen layer. Neurons in the Kohonen layer are arranged in a one- or two-dimensional lattice. Fig. 1 displays the layout of a one-dimensional map where the output neurons are arranged in a one-dimensional lattice. The number of neurons in the input layer matches the number of attributes of the objects. In the input layer each neuron has a feed-forward connection to each neuron in the Kohonen layer [18].

Kohonen's SOM and Sammon's nonlinear mapping are topology- and distance-preserving mapping techniques commonly used for multivariate data projections. However, the computations for both techniques are high.

2.5. Discrete optimization

Discrete optimization techniques provide a possible alternative for the data visualization problem. Discretizing the data visualization problem may result in a large-scale quadratic assignment problem that is very difficult to solve.

Roselyn Abbiw-Jackson[19] used the *divide and conquer*algorithm to solve the quadratic assignment problem (QAP).

3. Modeling

Let M be a set with n points in m-dimensional space.The data visualization problem is locating any p-dimensional point in M to q-dimensional space ($q<m$ and $q=2$ *or* 3) such that a relevant measure of distance is preserved. We use discrete optimization techniques to solve it. Therefore, we approximate the continuous q-dimensional space by a lattice N whileeach cell has a center point. Onthe other hand, the data visualization problem is similar to assigning m point to$n$ cell (center) points. The decision variables are given by:

$$x_{ik}=\begin{cases} 1, \text{ if thei}^{th} \text{ instance is assigned to lattice point } k \in N \\ 0, \text{ Otherwise} \end{cases} \tag{1}$$

Therefore, data visualization problem can be written as follows:

$$\begin{aligned} &\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k \in N} \sum_{l \in N} F\left(D^{old}_{i,j}, D^{new}_{i,j}\right) x_{ik} x_{jl} \\ &\text{subject to} \sum_{k \in N} x_{ik}=1, \quad \forall i \\ &x_{ik} \in \{0,1\}, \end{aligned} \tag{2}$$

Where, $D^{new} \in R^q \times R^q$ is a matrix measuring the distance between n given instances, $D^{new} \in R^q \times R^q$ is a new distance matrix between assigned instances in the $q=2$ *or* 3 dimensional space and F is a function of the deviation between the differences between the instances in the original space and the new q-dimensional space. Choices for Finclude the functions for Sammon mapping and classical scaling, and all objective functions for nonmetric scaling. By using Sammon mapping as objective function, data visualization problem can be formulated as:

$$\begin{aligned} &Min \frac{1}{\sum_{i=1}^{m} \sum_{\substack{j=1 \\ j>i}}^{n} d(i,j)} \sum_{i=1}^{m} \sum_{\substack{j=1 \\ j>i}}^{m} \sum_{k=1}^{n} \sum_{l=1}^{n} \left(\frac{(d(i,j)-d^{*}(k,l))^{2}}{d(i,j)} \right) x_{ik} x_{jl} \\ &\text{subject to} \sum_{k=1}^{n} x_{ik}=1, \quad \forall i \\ &x_{ik} \in \{0,1\} \end{aligned} \tag{3}$$

Where, $d(i,j)$ is distance (usually Euclidean distance) between original points in space of R^m and $d^*(k,l)$ is distance between lattice points in space of R^q.

The number of cells is determined with regards to requirement of data visualization method. A problem with points spread out will require a larger grid than one with points clustered together. The larger the grid, the more accurate the final result. To scale the cells (in q-dimensional space) and the given data set, we find the greatest distance between the pairs of points in given data set. Let this distance be a and the greatest distance in the chosen lattice N be b. Then, we multiply all original distances between points in M by b/a, so that our lattice is scaled to the given problem. The problem of assigning $m$points to $n$lattice points cannot be treated as a linear assignment problem because a linear assignment problem assumes that the cost of assignment of one point to a lattice point does not depend on the assignment of the other points. However, this is not the case for data visualization problems. Therefore, this problem is a quadric assignment problem (QAP) whose objective function is convex and it can have many local solutions. On the other hand, space of this problem is very large and decision variables are internally depended.

Any solution method can be used to solve this problem but it should be effective in circumstance of problem. Proposed solution method here is Genetic Algorithms (GA). GAis appropriate for solving complicatedsearch/optimization problems where the number of localoptima is large or very large.

4. Genetic algorithm

Meta-heuristic methods are preferred against other optimization methods primarily when there is a need to find good heuristic solutions to complex optimization problems with many local optima and little inherent structure to guide the search [20]. The meta-heuristic approach to solve such problem is to start by obtaining an initial solution or an initial set of solutions and then initiating an improving search guided by certain principles. The most widely used meta-heuristic is in fact GA and its variants [21-25].

GAis a nature inspired approach to large scale combinatorial optimization problems. The underlying motivation of such algorithms is an attempt to borrow ideas from the selection process in nature which develops complex and well adapted species through relatively simple evolutionary mechanisms. The basic idea is to adapt these simple evolutionary mechanisms to combinatorial optimization problems. The first genetic algorithm for optimization problems was proposed by Holland [26] in 1957.

Genetic algorithms (GA) have been found to be applicable to optimization problems that are intractable for exact solutions by conventional methods [26, 27]. In each step, a subset of the current set of solutions is selected based on their performance and these solutions are combined into new solutions. The operators used to create the new solutions are survival, where a solution is carried to the next iteration without change, crossover, where the properties of two solutions are combined into one, and mutation, where a solution is modified slightly. The same process is then repeated with the new set of solutions, Fig. 2. The crossover and mutation operators depend on the representation of the solution but not on the evaluation of its performance.

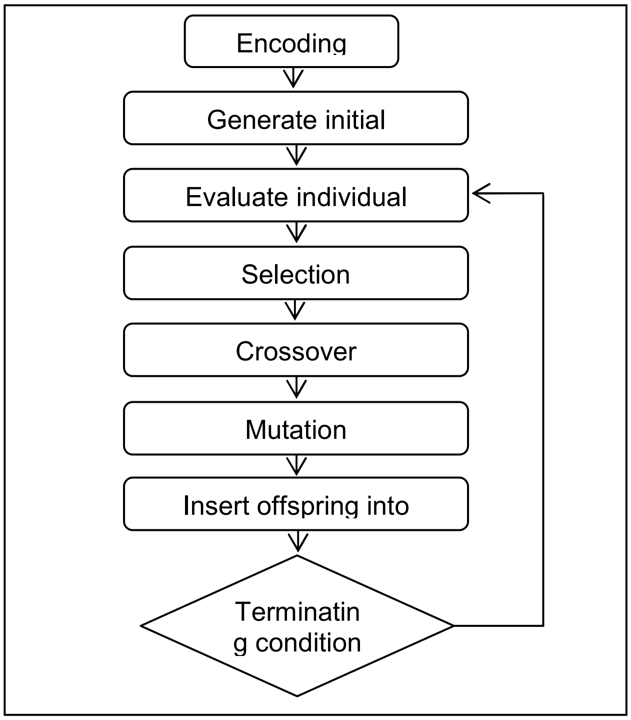

Fig. 2. The flow diagram of Genetic Algorithm.

The selection of solutions, however, does depend on the performance. The general principle is that high performing solutions (which in genetic algorithms are referred to as fit individuals) should have a better chance of both surviving and being allowed to create new solutions through crossover. For genetic algorithms and other evolutionary methods the defining element is the innovative manner in which the crossover and mutation operators define a neighborhood of the current solution. This allows the search to quickly and intelligently traverse large parts of the solution space.

4.1. Chromosome representation and decoding

In order to apply a genetic algorithm to a specific problem, a suitable encoding or representation must first be devised. In this encoding, a solution to the problem is represented by a set of parameters. These parameters (known in genetic terminology as genes) are joined together in a string of values that represents or encodes the solution to the problem. In genetic terminology, this string is referred to as a chromosome or individual. The genetic algorithm approach proposed in this paper uses a random key *U(0, m)*[28]to encode the chromosomes. In this technique, each gene is a uniform random number between 0 and m. Therefore, a chromosome is encoded as a vector of random numbers. In our algorithms, each chromosome is made of $n$genes, g_i, therefore, chromosomes in population can be shown as Fig. 3.

Where, n is number of instances and each gene in chromosome,g_i, shows that the i^{th} data point in original space assign to g_i cell in lattice.

To satisfy the constraint of model that any instance can only assign to one cell in lattice, check chromosomes across evolution. This is accomplished by removing the unfeasible chromosomes in population and product new feasible chromosome.

In order to evaluate the fitness of an individual, it is necessary to decode its chromosome into the corresponding solution to the problem; this value measures the quality or merit of the solution associated with that chromosome. Fitness function in this paper is the objective function defined in (3).

At any iteration, the genetic algorithm evolves the current population of chromosomes into a new population, using selection, crossover and mutation mechanisms.

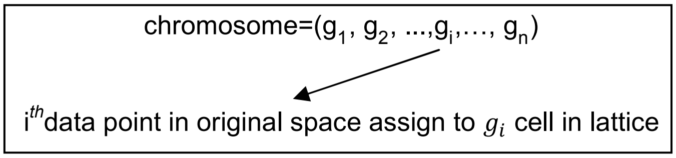
chromosome=(g_1, g_2, ...,g_i,..., g_n)
i^{th}data point in original space assign to g_i cell in lattice

Fig. 3. Chromosome representation.

4.2. Selection

Different types of selections can be implanted in the described data visualization model, but only the method called roulette wheel selection is used as the selection operator in this research.

Roulette wheel selection is an example of proportional selection operator where fitness values are normalized (e.g. by dividing each fitness value by the maximum fitness value). The probability distribution can then be seen as a Roulette wheel, where the size of each slice is proportional to the normalized selection probability of an individual. Selection can be likened to the spinning of a roulette wheel and recording which slice ends up at the top; the corresponding individual is then selected.

4.3. Crossover

Six types of crossover are used for solving the data visualization: 1-point, 2-point, 3-point, 4-point, 5-point and uniform crossover. The 1-point crossover randomly determines a cross-point in the length of the chromosome, combines the left part of the chromosome of the first parent with the right side of the chromosome of the second parent to form the first offspring. A second offspring is inversely generated. The 2-point crossover implies two cross-points. The first offspring has the beginning and the last parts of the chromosome of the first parent and the middle portion of the second one. The second offspring is again inversely generated. Another point cross over is similar to 2-point crossover. The procedure is quite different for the uniform crossover. With this type of crossover, a random number between *0* and *1* is selected for each bit of the chromosome (p_c*~U(0,1)*). If random number is smaller than p_c, then the first offspring uses the bit of the first parent at this position. Otherwise, when random number is greater than p_c,the first offspring takes the bit of the second parent at this position, Fig. 4.

Parent 1:	(8, 23, 2, …, 49)
Parent 2:	(51, 9, 37, …, 43)
Random number:	(0.32, 0.81, 0.47, …, 0.69)
	(<0.5, >0.5, <0.5, …, >0.5)
Offspring:	(8, 9, 2, …, 43)

Fig. 4. Parameterized uniform crossover example.

4.4. Mutation

The aim of mutation is to introduce new genetic material into an existing individual; that is, to add diversity to the genetic characteristics of the population. Mutation is used in support of crossover to ensure that the full range of allele is accessible for each gene. Mutation is applied at a certain probability, p_m, to each gene of the offspring, to produce the mutated offspring. The mutation probability, also referred to as the mutation rate, is usually a small value, $p_m \in [0,1]$, to ensure that good solutions are not distorted too much and over a certain level, the mutation could turn the genetic algorithm into a simple random walk, meaning a lost in the efficiency associated to the search strategy.

5. Case study

In order to compare the proposed technique (QAP-GA) and the well-knownSelf OrganizingMaps algorithm (SOM), a real-life case is considered here. Used data set in this research includes 24-hour electrical load of 366 days starting from 20 March 2008 in Esfahan. Therefore, we have 366 instances with 24 attributes, which any attribute is as an hour (a label number for all the daily load curves).

The dimension of the output map in this research is assumed to be two ($q=2$), because mostvisible media (for example: paper, monitor panel and etc.) are 2-dimensional. The output map is a 60×60 grid square.

Similarity of resulted solutions and data point in original data set is calculated by the subjective function in equation (3). This criterion is used to compare the proposed model (QAP-GA) and SOM.The methods are run on a 2Core, 1.6 GHz machine with 1 GB RAM under WINXP platform.

5.1. Visualization by using SOM

To solve the data visualization problem for our case study, we run Clementine. 12.0 built into SPSS software with the following modification of the default settings: width=60, length=60, learning date decay: exponential, Phase1 neighborhood= 32, Phase1 initial Eta=0.1, Phase1 cycles=200.

As mentioned in Section 2.4, using the SOM-based approach we can decide visually on the distribution of subjects in the m-dimensional space in accordance with their distribution among the cells of the table. However, if we have to evaluate the interlocation of subjects, we can assume that the subjects from the closer cells of the table are more similar. The combined mapping should be used in search for the answer to the question whether our assumption on the similarity of subjects is right or wrong. Regarding sensitivity of SOM technique to initial weight of neurons, it results indifferent solution in each run but the best solution 0.18147 was foundby using subjective function as similarity criteria.

5.2. Visualization by using Genetic algorithm

The Genetic Algorithm is implemented in visual studio C# (2008). A total of 100 runs are conducted by GA.A different seed for the generation of random numbers is used to start a run. A run is considered to be a success if the value obtained by the algorithm is the least in resulted solutions.

The most important and perhaps the most difficult task in binary coded GA is to find the most appropriate combination of parameters occurring in a GA which is termed as parameter fine tuning. To achieve this goal, we have carried out an extensive experiment of various possible combinations of crossover probabilities (p_c) and mutation probabilities (p_m). The stopping criterion is a maximum of 2000 generations, or if no improvement is observed in the best individual in consecutive 100 generations. We allow p_c to vary from 0.5 to 0.9 (increment 0.1), p_m from 0.001 to 0.003 (increment 0.01) and population size from 50 to 500 (increment 50). An empirical study is made on the results to find the recommendablep_c and p_m. Results of these analyses are shown in Table 1 and Table 2.

The recommended values of the p_c and p_m are shown in Table 3. We do not claim that these parametersettings are the best for any problem in general, but these values are recommended since they are found to be repeatedly giving good results for most problems, andhence they are appropriate settings tochoose when the overall performance of algorithm is considered.

The best solution using parameters of Table 3 is 0.1514. By comparing this solution and the solution of SOM technique, superiority of this solution is apparent. Therefore, the result of QAP-GA is more precise and can be used for this case.

Table 1: Results of genetic algorithm with different initial population size and number of crossover points.

Initial population size											
500	**450**	**400**	**350**	**300**	**250**	**200**	**150**	**100**	**50**		
0.28	0.29	0.29	0.3	0.31	0.32	0.34	0.36	0.39	0.41	**1**	crossover points
0.28	0.28	0.29	0.3	0.31	0.32	0.34	0.35	0.38	0.39	**2**	
0.27	0.28	0.29	0.29	0.3	0.31	0.33	0.34	0.34	0.35	**3**	
0.27	0.28	0.28	0.29	0.3	0.3	0.32	0.33	0.33	0.34	**4**	
0.27	0.27	0.28	0.29	0.29	0.29	0.31	0.31	0.32	0.33	**5**	

Table 2: Results or genetic algorithm with different initial population size and crossover probability.

Initial population											
500	**450**	**400**	**350**	**300**	**250**	**200**	**150**	**100**	**50**		
0.15	0.16	0.17	0.16	0.17	0.19	0.16	0.21	0.27	0.29	**0.5**	Crossover probability
0.16	0.17	0.17	0.16	0.18	0.19	0.2	0.21	0.26	0.31	**0.55**	
0.16	0.18	0.17	0.15	0.19	0.19	0.21	0.21	0.24	0.33	**0.6**	
0.15	0.17	0.17	0.16	0.18	0.18	0.2	0.22	0.24	0.3	**0.65**	
0.15	0.16	0.16	0.16	0.17	0.17	0.19	0.23	0.24	0.27	**0.7**	
0.16	0.17	0.15	0.17	0.17	0.17	0.18	.24	0.23	0.28	**0.75**	
0.17	0.17	0.15	0.18	0.18	0.17	0.18	0.24	0.23	0.3	**0.8**	
0.16	0.16	0.16	0.19	0.18	0.18	0.16	0.22	0.24	0.28	**0.85**	
0.16	0.16	0.18	0.19	0.19	0.2	0.2	0.2	0.24	0.26	**0.9**	

Table 3: Genetic parameters used.

Parameter	value
p_c	0.7
p_m	0.002
population size	400

6. Customer patterns classification

It came in previous section that QAP-GA map would result ina better and more precise solutioncompared to SOM technique. Therefore, we use the solution of QAP-GA to group (aggregate) and classify (disaggregate) electrical customer patterns. QAP-GA is used as a classification tool through the analysis of the influence of the form of the 366 data set arrays used to feed and train the map.

Thus, each 366 data set array reflects the load behavior associated to a day demand included in the case study (see Section 5).

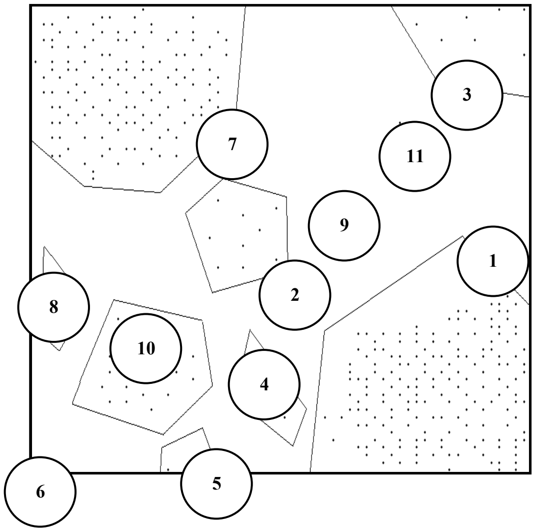

Fig. 5. Resulted map of the best chromosome resulted of QAP-GA method

It should contain the necessary information to evaluate the affiliation of each elemental demand to a cluster. From the viewpoints of the authors and technical interests (demand response and distributed generation), it is necessary to find similar load characteristics. This can be reached through field measurements performed by the customer or by commercializing to obtain, reduce, and manage energy and power costs.

An alternative labeling to the one proposed in Section 5 is used for a better understanding of results. By means of this labeling, a number is assigned to each load profile following the next criterion: the last two digits indicate the day of the month and the initial remaining ones the corresponding month (mm/dd). Thus, a label map allows the identification of daily load data assigned to each cell. The information contained in the daily load curves is directly presented to the map.

Finally, the selection of the number of clusters is another significant task. This number, a subjective value, should be a reasonable option; but in this case visual inspection helps the researcher to decide the clustering.

As shown in Fig. 5, daily load behavior can be classified to 11 clusters while located days in any cluster have similar demand pattern. These clusters and days belonging to each oneare given in Table. 4. As is shown in this table, days can be dividedinto different classes. The most attractive regions are: region 1 which includes cold days in year. Electrical load pattern in these days is the

Table 4: Classify days respect to their electrical load pattern

Days	Region
121, 115, 118, 122, 1226, 1228, 1102, 720, 721, 724, 725, 804, 809, 812, 822, 826, 903, 904, 907, 911, 912, 913, 914, 916, 917, 918, 920, 921, 924, 925, 926, 930, 1001, 1002, 1003, 1005, 1008, 1009, 1010, 1011, 1012, 1014, 1015, 1022, 1024, 1025, 1030, 1101, 1103, 1105, 1109, 1110, 1112, 1114, 1115, 1116, 1117, 1119, 1120, 1121, 1126, 1127, 1129, 1201, 1203, 1204, 1210, 1213, 1214, 1215, 1217, 1218, 1219, 722, 120, 708, 711, 714, 715, 716, 723, 728, 729, 730, 801, 802, 805, 806, 807, 808, 813, 905, 906, 1220, 1222, 1225, 901, 803, 817, 908, 915, 919, 922, 927, 1016, 1029, 1107, 1113, 1122, 1123, 1202, 1209, 1211, 201, 125, 126, 129, 205, 212, 217, 218, 219, 112, 108, 110, 111, 123, 130, 213, 1208, 726, 1026, 1124, 1207, 1216, 1229, 818, 727, 811, 902, 909, 1007, 227, 504, 705, 829, 830, 1021, 124, 214, 717, 718, 928, 1028, 1221, 1227, 114, 116, 117, 127, 206, 220, 310, 622, 701, 703, 707, 709, 710, 719, 814, 821, 823, 825, 923, 1023, 1108, 1130, 1205, 1212, 1223, 1224	1
324, 317, 318, 511, 428, 629, 303, 608, 615	2
210, 713, 815, 910, 816, 820, 311	3
1106, 824, 1004	4
827, 828	5
819	6
621, 302, 619, 623, 624, 625, 626, 627, 628, 704, 418, 326, 329, 330, 401, 405, 410, 411, 412, 413, 415, 416, 417, 419, 420, 422, 430, 506, 507, 508, 514, 516, 517, 519, 521, 523, 524, 526, 528, 529, 531, 603, 604, 617, 620, 502, 229, 305, 307, 308, 312, 313, 321, 322, 325, 328, 409, 501, 503, 512, 513, 522, 607, 610, 202, 119, 128, 131, 203, 204, 207, 208, 209, 211, 215, 216, 221, 222, 223, 224, 225, 706, 421, 331, 407, 414, 426, 509, 518, 525, 527, 601, 230, 231, 301, 404, 602, 613, 605, 611, 612, 614, 315, 304, 306, 319, 316, 505, 423, 425, 515, 616, 530, 609, 320, 327, 226, 228, 309, 314, 323, 402, 403, 408, 424, 427, 429, 431, 510, 606, 618, 630, 702	7
406, 520	8
631	9

1104, 113, 1027, 1013, 712, 810, 929, 1020, 1111, 1118, 1125, 1128, 1206, 104, 103, 105, 106, 109, 107, 1230, 101, 102, 1006, 1017, 1019	10
1018	11

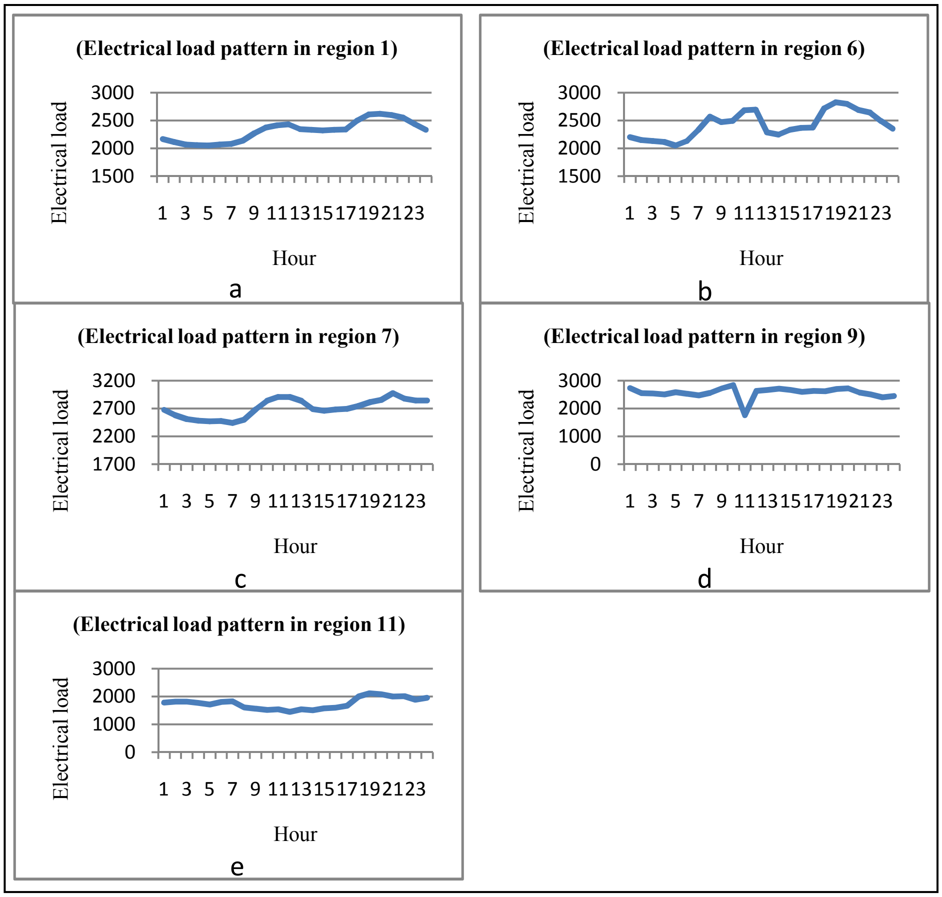

Fig. 6. Electrical load pattern in important regions

same (see Fig.6-a) with low load during the year. This region includes almost half of all days in the database.

Another important cluster is region 7. Electrical load pattern in this region is different than region 1. This disparity resulted from seasonal temperature changes and different applied electronic devices. These days belong to warm days. Electrical load in these days increases significantly (see Fig.6-b).

In addition, this map can show anomalous behavior in some specific days. For example: November 9th, 2008 in region 6 (see Fig. 6-c), September 21st, 2008 in region 9 (see Fig. 6-d) and January 7th, 2009 in region 11 (see Fig. 6-e). These days are religious vacations in the country and their electrical load pattern is different from all other days.

7. Conclusions

GA has been proved to be a robust general-purpose search technique. They have a central placein data mining applications due to their capacity to search large spaces efficiently. In this paper, we formulated data visualization as a Quadric Assignment Problem (QAP), and then presented a genetic approach (GA)to solve the resulted discrete optimization problem. To demonstrate the application of genetic algorithms on discrete optimization in data visualization, we used a data base of electricity load and comparedthe results to SOM output. Both QAP-GA and SOM can only make assignments to specific points in the lattice space and cannot assignpoints anywhere in the plane. Although the QAP-GA approach provides a good approximate solution in comparison to the SOM technique, SOM needsshorter computing time. Therefore, the QAP-GA is useful to acquire precise solutions when the running time is not important. Moreover, QAP-GA is an evolutionary methodthat starts with a random solution similarto the SOM technique; but sensitivity of QAP-GA to the starting solution is lower than SOM. Then, the output of QAP-GA is used to analyze electricity load data. There are several opportunities for future work on this topic. For example, the running time of QAP-GA can be decreased by using parallel Genetic Algorithm, and *divide and conquer* local search.

References

[1] Cabena, P., Stadler, R., Zanasi, A., *Discovering data mining: from concept to implementation*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1998.

[2] Magkos, E., Maragoudakis, M., Chrissikopoulos, V., Gritzalis, S., "Accurate and large-scale privacy-preserving data mining using the election paradigm", *Data & Knowledge Engineering*, Vol. 68,pp. 1224-1236, 2009.

[3] Berry, M. J. A., Linoff, G., *Mastering data mining*, Wiley, 2000.

[4] Chang, C. H., Ding, Z. K., "Categorical data visualization and clustering using subjective factors", *Data & Knowledge Engineering*, Vol. 53,pp. 243-262, 2005.

[5] Borg, I., Groenen, P. J. F., *Modern multidimensional scaling: Theory and applications*, Springer Verlag, 2005.

[6] Roselyn, A. J., Bruce, G., Raghavan, S., Wasil, E., "A divide-and-conquer local search heuristic for data visualization", *Computers & Operations Research*, Vol. 33,pp. 3070-3087, 2006.

[7] Sammon, J. W., "A nonlinear mapping for data structure analysis", *IEEE Transactions on computers*, Vol. 18,pp. 401-409, 1969.

[8] De Backer, S., Naud, A., Scheunders, P., "Non-linear dimensionality reduction techniques for unsupervised feature extraction", *Pattern Recognition Letters*, Vol. 19,pp. 711-720, 1998.

[9] Kohonen, T., "The self organizing map", *Neurocomputing*, Vol. 21,pp. 1-6, 1998.

[10] Kohonen, T., "Self-organization and associative memory", Vol. 1988.

[11] Torriei Don, J., "Statistical theory of passive location systems", *IEEE Trans. on AES*, Vol. 20,pp. 183-198, 1984.

[12] Spirito, M. A., "On the accuracy of cellular mobile station location estimation", *IEEE Transactions on Vehicular Technology*, Vol. 50,pp. 674-685, 2001.

[13] Caffery Jr, J., Stuber, G. L., "Subscriber location in CDMA cellular networks", *IEEE Transactions on Vehicular Technology*, Vol. 47,pp. 406-416, 1998.

[14] Hellebrandt, M., Mathar, R., Scheibenbogen, M., "Estimating position and velocity of mobiles in a cellular radio network", *IEEE Transactions on Vehicular Technology*, Vol. 46,pp. 65-71, 1997.

[15] Smith, J., Abel, J., "Closed-form least-squares source location estimation from range-difference measurements", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 35,pp. 1661-1669, 1987.

[16] Friedlander, B., "A passive localization algorithm and its accurancy analysis", *IEEE Journal of Oceanic engineering*, Vol. 12,pp. 234-245, 1987.

[17] Dzemyda, G., Kurasova, O., "Comparative analysis of the graphical result presentation in the SOM software", *Informatica*, Vol. 13,pp. 275–286, 2002.

[18] Gan, G., Ma, C., Wu, J., "Data clustering: theory, algorithms, and applications", *ASA-SIAM Series on Statistics and Applied Probability*, Vol. 20,pp. 2007.

[19] Abbiw-Jackson, R., Golden, B., Raghavan, S., Wasil, E., "A divide-and-conquer local search heuristic for data visualization", *Computers and operations research*, Vol. 33,pp. 3070-3087, 2006.

[20] Glover, F., Kochenberger, G. A., *Handbook of metaheuristics*, Springer, 2003.

[21] Sharpe, P. K., Glover, R. P., "Efficient GA based techniques for classification", *Applied Intelligence*, Vol. 11,pp. 277-284, 1999.

[22] Yang, J., Honavar, V., "Feature subset selection using a genetic algorithm", *IEEE Intelligent Systems*, Vol. 13,pp. 44-49, 1998.

[23] Kim, Y. S., Street, W. N., Menczer, F., "Feature selection in unsupervised learning via evolutionary search", Vol., pp.365-369, 2000.

[24] Fu, Z., Golden, B. L., Lele, S., Raghavan, S., Wasil, E., "Diversification for better classification trees", *Computers and operations research*, Vol. 33,pp. 3185-3202, 2006.

[25] Larrafiag, P., Poza, M., Yurramendi, Y., Murga, R., Kuijpers, C., "Structure learning of Bayesian networks by genetic algorithms: performance analysis of control parameters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18,pp. 912-926, 1996.

[26] Holland, J. H., "Adaptation in natural and artificial systems. 1975", *Ann Arbor MI: University of Michigan Press*, Vol.

[27] Goldberg, D. E., *Genetic Algorithms in Search and Optimization*, Addison-wesley, 1989.

[28] Haupt, S. E., *Practical genetic algorithms*, Wiley-Interscience, 2004.

[29] Corriveau, G., Guilbault, R., Tahan, A., "Genetic algorithms and finite element coupling for mechanical optimization", *Advances in Engineering Software*, Vol. 2009.

[30] Eiben, A. E., Michalewicz, Z., Schoenauer, M., Smith, J., "Parameter control in evolutionary algorithms", *Intelligence (SCI)*, Vol. 54,pp. 19-46, 2007.