Developing a Distributed Self Adaptive Genetic Algorithm with Migration for Identification of Electrical Customer Patterns

Vahid Golmah[1], Golsa Mirhashemi[2]
[1]Ferdowsi University of Mashhad, Mashhad, Iran,
[2]Department of Engineering; Islamic Azad University; Neyshabur branch; Neyshabur; Iran
Email:v.golmah@um.ac.ir (Corresponding author),g.mirha6emi@gmail.com

Receive Date: 26 August 2024 Revise Date: 15 November 2024 Accept Date: 27 January 2025

Abstract

Data visualization is a key component of undirected data mining that it transforms data, information, and knowledge into visual view. In this paper, we formulate data visualization problem as a quadratic assignment problem (DV-QAP). The QAP is an NP-Hard problem and has high complexity that it is more acute for data visualization problem because it has intense dependencies among variables and big search space. Therefore, the exact approaches are inefficient to solve DV-QAP and we introduce a new technique called Distributed Self Adaptive Genetic Algorithm with Migration (DSAGAM) that their parameters adjust to increases the exploration and exploitation. This paper focuses on the effect of controlling the migration process and adjusting parameters with respect to the fitness to explore such big search spaces to improve solutions quality. Then we demonstrate the efficiency of the model for a real data set compared with the SGA, SAMGA, IGA and Sammon's mapping approaches.

Key words: Data visualization, Adaptive genetic algorithms, Quadratic assignment problem and Clustering.

1. Introduction

Due to the information technology improvement and the growth of Internet, enterprises are able to collect and to store huge amount of data. The primary challenge is how to make the database a competitive intelligence by converting seemingly meaningless data into useful information. Therefore, Machine learning and data mining became a vital part of modern problem-solving solutions for many applications including of health care[1-3], Learning[4-6], Circuit design[7], Voice recognition[8], Business[9], Transportation[10] etc.

Although, concept of "Data mining" is defined by different forms in researches, but a complete and popular definition of it is[11-13]: The analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner. The relationships and summaries derived through a data mining exercise are often referred to as models or patterns.

Big data experts[14, 15] believe that 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.

Data Visualization is the process of transforming data, information, and knowledge into visual form, making use of humans' natural visual capabilities. 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

Problem (QAP). The QAP is a combinatorial optimization problem first introduced by Koopmans and Beckmann[22] as the plant layout problem to locate equipment to locations so that the transportation cost among the departments is minimized. The QAP is an NP-Hard problem and it is considered to be one of the most difficult problems to be solved optimally[23]. It is very difficult to solve it optimally once the number of departments exceeds 15. On the other hand, data visualization usually deal with numerous points to assign data. Therefore, discrete optimization techniques have high complexity and single methods are not efficient to solve them. Several researches [24-28] suggest using of parallel optimization for complex issues with nonlinear relations that parallel Genetic Algorithm(GA) is the most efficient of them.

The GA is a meta-heuristic approach based on the evolution in nature it has been applied to solve complex combination optimization problems because of its higher steadiness and global Optimization. Although, GAs in their elementary forms are not competitive with other heuristic algorithms such as the simulated annealing and algorithm and the optimum individual protecting algorithm for these problems, but improved versions have been more efficient. The main goal of this paper is to apply several possible enhancements to Genetic Algorithm (GA) to explore search spaces to improve solutions quality for data visualization problem that is transformed to the Quadratic Assignment Problem (QAP) by implementing them on a real data and comparing quality of their solutions. Then this comparison is extended to cover genetic algorithms and Sammon's mapping as a popular method for data visualization also.

The remainder of this paper is organized as follows: Section 2 describes Genetic Algorithm; Section 3 introduces a short review of adaptive GAs; the proposed dimensions in the data set while preserving the relevant inherent properties. The complexity of these problems is high since they locate data points in a high-dimensional space in a lower-dimensional space such that a relevant measure of distance is preserved. Popular techniques used to solve data visualization problems include Principal component analysis (PCA) [16, 17], multidimensional scaling (MDS)[18] and Sammon maps (SM) [19, 20].

Principal component analysis (PCA) and multi dimensional scaling (MDS) are the most widely used methods for performing linear projection. Linear projection means the projection of data is conducted by multiplying each component of the original vector with a scalar. These methods are capable only of handling data that is inherently linear in nature which all available variables are used and the data is transformed using a linear transformation to a reduced dimension space. Thus, PCA and MDS are not efficient when one is dealing with highly nonlinear data. In addition, final visualization map of MDS is difficult to perceive, when one is handling high-dimensional and highly unsymmetrical data set[21]. Sammon's mapping has been one of the most successful nonlinear MMDS methods since its advent in 1969, which improves the quality of scaling by focusing on small distances in data space: explicitly the Sammon mapping tries to capture the local structure of the data while paying less attention to the global structure of the data. Sammon mapping is computationally demanding especially when one is handling huge numbers of data points. Even with today's fast computers, nonlinear optimization techniques are usually slow and inefficient for large data sets. In addition, it requires re computation when new data points add.

The discrete optimization techniques are efficient approach for data visualization by transforming it to a Quadric Assignment binary tournament, n-size tournament and linear ranking.

(5) **Offspring generation**: The offspring generation obtains by the processes of crossover and mutation. New individuals generates until a fixed maximum number of individuals reaches.

(6) **Termination criterion**: If the stop criterion is satisfied, the algorithm ends and the best chromosome, together with the corresponding result, gives as output. Otherwise, the algorithm iterates again steps 3–5.

The process of GA is controlled by several parameters, e.g. genetic operators (this component contains selection, crossover and mutation), initial population, etc. These parameters largely determine the success and efficiency of GA in solving a specific problem so that GA often suffers from premature convergence. Unfortunately, these parameters interact with each other in a complicated way and determining which parameter set is best to use for a specific problem can be a complex task requiring much trial and error [30]. This approach of parameter selection obviously requires a lot of computation, which sometimes is larger than the time used for solving a particular problem by GA itself. The experiments that involved several test functions and parameter combinations took approximately 1.5 CPU years. A prominent example of exhaustively testing several combinations of parameters was shown in the study done by Zubel et al. [31].

A new breed of GA called adaptive GAs[32], fix the parameters for the GA operators dynamically to adapt to the current problem. Adapting evolutionary algorithms to the environment has been investigated in detail [33] and is reported to be an essential feature that makes the algorithms closer to natural evolution. It has been further pointed out that GA would be accurate and efficient only if the genetic parameters are appropriately adapted to the nature of the problem. The adaptive model is detailed in Section 4; problem is defined in Section 5; detailed experiments by proposed model and other approaches are discussed in Section 6 where the results obtained are also given. Analysis of results and use of them to data visualization presented in Section 7; Section 8 discusses the Concluding remarks and future research directions.

2. Genetic algorithm

GA is stochastic search techniques based on analogy to Darwinian natural selection. Individuals who fit the environment best should have a better chance to propagate their offspring. By the same reason, solutions that have the best ‘‘fitness’’ should receive higher probability to search their ‘‘neighbors’’. The main advantage of GA lies in its powerful implicit parallelism. In Holland’s theory, a GA implicitly evaluates a number of patterns larger than population size without additional computational time and memory. The overall structure of GA can be described as follows [29]:

(1) **Encoding:** For any GA, a chromosome encoding is needed to describe each chromosome in the population. The encoding method determines how the problem is structured in the algorithm and the genetic operators that are used. Each chromosome is made up of a sequence of genes from certain alphabet, which can consist of binary digits (0 and 1), floating-point numbers, integers, symbols (i.e., A, B, C, D), etc. Each chromosome represents a solution for the problem.

(2) **Initial population:** An initial population of size P can generate randomly. The length of each chromosome in a population should be the same.

(3) **Fitness evaluation**: The fitness computes for each chromosome in the current generation.

(4) **Selection**: At each iteration, the best chromosomes choses for reproduction by one among three different methods, i.e.,

models are faster and keep the chromosomes variety. Keeping chromosomes variety is the most important factor to obtain solution with high quality in evolutionary computations [34].

As illustrated in Fig.1, in the island model, a population divide into sub-populations, islands. Then, the genetic operations executes in each island independently. In this GA model, a number of subpopulations evolve in parallel and in a cooperative framework. Selection, crossover and mutation occur in each subpopulation independently. In addition, individuals could migrate between islands (or subpopulations).

parameters have also to be depending on the scheme of crossover and mutation methods.

Using of parallel systems in the execution of genetic algorithms has led to parallel genetic algorithms. The parallel GA classifies into centralized and distributed models. The centralized models aim to speed up by evaluating chromosomes in parallel; while the distribute models, known also as island models, divide the chromosome pool into several sub-pools and perform genetic operations independently in parallel. However, sub-pools may exchange with each other their chromosomes, usually elite chromosomes; that are called migration. The distributed

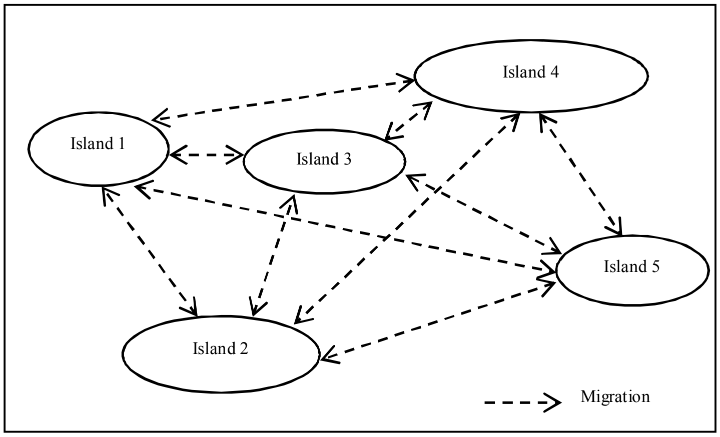

Fig.1 An island Genetic Algorithm.

different population sizes have been proposed expecting that one or several of them would yield good results[37]. Although, this method is not truly adaptive in the sense that the appropriate number of individuals is not learnt but is obtained by trial and error. This method is not feasible as it is not realistic to perform large number of blind searches in parallel. Some of the applications of the parameter-less genetic algorithms [38] and multi-objective rule mining using Genetic Algorithms are discussed in[39]. However, it is even more difficult to accept these kinds of approaches in problems like topology optimization

An integral part of an island GA is the migration policy which governs the exchange of information between islands [35]. A migration policy specifies: communications topology, migration rate, selection mechanism, replacement strategy [36]. The present work is an attempt to improve the performance of IGA by incorporating appropriate adaptive techniques and migration policies to data visualization problem based on QAP.

3. Relevant Works

When the optimal population size is not known in a problem, parallel searches with this drawback, Takashima *et al.*[46, 47]have proposed a Self-Adaptive Island GA (SAIGA) procedure for adapting multiple genetic parameters in which several island runs are executed in parallel that reaches very close to global optimal solutions. However, this algorithm does not efficiently utilize the genetic characteristics of the individuals obtained from different islands - only the elite individuals could immigrate between islands. The procedure executes all the islands in parallel throughout the optimization process; this is computationally very expensive and most of them would become redundant and ineffective after the initial stage in the context of topology optimization.

Srinivasa *et al.*[48] proposed a Self-Adaptive Migration model Genetic Algorithm (SAMGA) that increases the explorative power of search for individuals in bad regions of the state space, and increases exploitative power of search for individuals in the high fitness regions of the state space. This model uses a competitive search to increase the fitness of one population and makes the search in other populations with lower fitness more explorative and rigorous. But it ignores migration policies and its adaptive parameters are not general for any problem. The model proposed in the current research applies for data visualization problems where there are intense dependencies among variables, and parameters adjust themselves based on the problem circumstances. Four parameters adjust by the algorithm including of: crossover rate, mutation rate, migration rate, and the survival rate of individuals. The only similarity between [48] and proposed model is that both the methods chooses adaptively where evaluating the objective function is computationally expensive and adjusting one or two parameters alone does not give the desirable improvement.

The difficulties present in manual adaptation solve through much adaptive GAs (AGAs) that can dynamically adjust the parameter values. Theoretical aspects and implementation methodologies of various self-adaptive procedures have been discussed and compared in detail [40].An adaptive GA, in which mutation rate for an individual is encoded in the gene of an individual, is proposed in [41], with the hope that finally individuals with good mutation rate survive. However, only individuals with low mutation rate survive in the later phases of the search. An adaptive GA that determines mutation and crossover rate of an individual by its location in a two dimensional lattice plane is proposed in [42]. The algorithm keeps diversity of these parameters by limiting the number of individuals in each lattice.

A meta-GA is a method of using GA to fix the right parameters for the GA. However, in this process, the number of evaluations needed is high and the process is expensive. Murata et al. [43] have proposed an Agent Oriented Self-Adaptive Genetic Algorithm (A-SAGA) that combines meta-GA [44] and GA with distributed environment search scheme [45] where the distributed environment GA is a kind of parallel GA. This A-SAGA performs well once the algorithm trains according to the nature of the problem in hand and appropriate genetic parameters identify. The training of the algorithm is equivalent to solving the complete optimization problem, and becomes computationally very costly. To overcome

Migration(DSAGAM) method to increase the number of individuals in the population residing in a relatively high fitness region of a given search space, thus improving exploitation. Moreover, variation migration policies apply in this model. The focus in this paper is on applying several possible enhancements to Genetic Algorithm (GA) to explore such big search spaces to improve solutions quality for data visualization problem that transforms to the Quadratic Assignment Problem (QAP) by controlling the migration process and adjusting parameters by establishing a relation to the fitness. This hybrid model summarizes in Algorithm 1. Then results are compared to SGA, SAMGA, IGA and Sammon's mapping as a popular method for data visualization for a real data set.

In our hybrid proposed model, only individuals who meet the specified conditions could migrate. Note that effects of the cooperative evolution among islands depend on the topology where the migration apply. Although there may be many different communication topologies, we focus on the typical three topologies in this paper. Fig.2 shows these topologies with 7 nodes: star, line and ring.

4. Model

By parallel processing, Island Genetic Algorithms are able to speed up the evolution process. This implies distributing subpopulations among many processors, which requires a new migration operator added to the configuration. It is now known that migration policy is quite significant in providing diversified population, and hence in providing higher quality solutions, and possibly faster convergence.

The migration of individuals from one population to another during the evolution control by several parameters, mainly:

1. Communications topology, which determines the migration paths between islands,
2. Migration rate, which determines the frequency of migration,
3. Selection mechanism to decide which individuals will migrate, and
4. Replacement strategy to decide which individual of the destination island will be replaced by the migrant.

In order to obtain good quality solutions, we propose a new self-adaptive island genetic algorithm called Distributed Self Adaptive Genetic Algorithm with

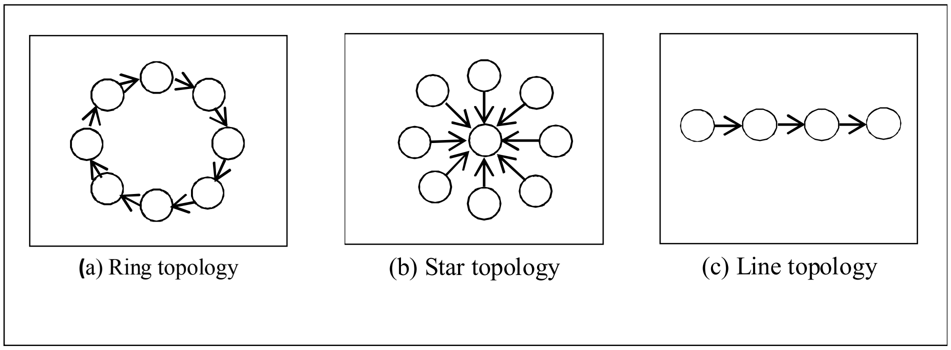
(a) Ring topology (b) Star topology (c) Line topology

Fig.2 The most popular communication topologies

In the model, individuals screen and examine at both the source and the destination to qualify for migration. The source gives or denies a visa based on local qualification criteria and the destination grants or denies a residency based on local qualification criteria. The applied form of qualification criteria is through the individual's relative fitness that changes dynamically during the evolution. Every individual in a very subpopulation is ranked locally;. On the destination island, an immigrant is accepted as a new member of the an individual qualifies for a visa at the source island if it is ranked first within its subpopulation population if its fitness value is better than a threshold set by that receiving island. Therefore, each node invokes the migration process at a variant rate, to nominate an individual for migration. The nominated individual checks against source criteria to get a visa, and then sent to a selected destination subpopulation based on the communication topology. In the destination node, this individual checks against the local criteria, to be accepted or denied.

Algorithm .1. Distributed Self Adaptive Genetic Algorithm with Migration

```
Require: $E = \{p_1, p_2, \dots, p_{np}\}$ be an ecosystem with np;
Populations($p_2, \dots, p_{np} \subset S$);
$p_{ij}$ : j$^{th}$ individual of population $p_i$;
$n_i$: Size of population $p_i$;
$pc_i$: Crossover probability of population $p_i$;
$pmg_i$: Migration probability of population $p_i$;
$pm_i$: Mutation probability of population $p_i$;
$pk_i$: Percent of population $p_i$ that transfer to next generation without any change;
$\bar{f_i}$: Average fitness of population $p_i$;
$bs\bar{f}$: The best average fitness of populations in ecosystem;
$bd\bar{f}$: The worst average fitness of populations in ecosystem;
$f(p_{ij})$: fitness of j$^{th}$ individual of population $p_i$ ($p_{ij}$);
$bf(p_i)$: fitness of the best individual of population $p_i$;
$\bar{f}$: average fitness of ecosystem;
for$i = 1$ to $np$do
    Determine the number of individuals in the population $p_i$ by arbitrary selecting from [100,500];
    Set initial mutation probability of population $p_i$, $pm_i$, to 0.0001;
    Set initial crossover probability of population $p_i$, $pc_i$, to 0.5;
    Set initial percent of population $p_i$ that transfers to the next generation without any change, $pk_i$, to 0.5;
    for j=1 to $n_i$do
        Initialize chromosomes of individuals in the population $p_i$ to random values, $p_{ij}$;
        Evaluate fitness of all the individuals of the population $p_i$, $f(p_{ij})$;
    end for
    $bf(p_i) \Leftarrow$ |fitness of the best individual of population $p_i$|;
    $bs\bar{f} \Leftarrow$ |The best average fitness of populations in ecosystem|;
    $bd\bar{f} \Leftarrow$ |The worst average fitness of populations in ecosystem|;
    $fsum \Leftarrow \sum_{i=1}^{np} \sum_{j=1}^{n_i} |f(p_{ij})|$;
    $pmg_i \Leftarrow bf(p_i)/fsum$;
    if $bf(p_i)$ unchanged in 10 consecutive generation then
        The source population, based on migration topology, gives a visa to its the best individual
        The destination population grants a residency with $pmg_i$ probability which i is source population based on migration topology.
    end if
    update $n_i$
    $\bar{f} \Leftarrow fsum/\sum_{i=1}^{np} n_i$;
    $\bar{f_i} \Leftarrow \sum_{j=1}^{n_i} p_{ij}/n_i$;
    Transfer $pk_i$% of individuals of population $p_i$ to next generation without any change
    $pm_i \Leftarrow 0.0001 + \left|\left((\bar{f_i} - bd\bar{f})/(bs\bar{f} - bd\bar{f})\right) - 1\right| * 0.0001$ ;
    $pc_i \Leftarrow (|(\bar{f_i} - bd\bar{f})/(bs\bar{f} - bd\bar{f})| * 0.25) + 0.5$;
    $pk_i \Leftarrow \left((|\bar{f_i} - \bar{f}|)/\bar{f}\right)$;
    if ($n_i$ = 0) then
    Delete population Pi (extinction)
    end if
```

To have an efficient and flexible Genetic Algorithm, we need to adjust parameters of the algorithm using the knowledge gained for any region in the ecosystem, to improve the solutions quality by increasing the exploitive and explorative power of the algorithm. As shown in Algorithm .1, four parameters are defined to change adaptively. The first changing parameter across process is the migration rate to update the migration probability of the best individual in each subpopulation. Migration probability in i^{th}subpopulation updates by,

$$pmg_i = bf(p_i)/fsum \tag{1}$$

The probability of migration parameter changes based on the best fitness of each subpopulation compared to the sum of fitness in ecosystem. Using this update, probability of migration from subpopulations in which the best individuals locate, increases. Therefore, the best individuals in the ecosystem have a higher chance to migrate. Migration occurs when the average fitness of subpopulation is static across evolution process. The best individuals, after receiving visa to migrate based on telecommunication topology, move to destination subpopulation. We use three popular telecommunication topologies i.e. star, line and ring; then analyze resulted solutions. This procedure utilizes the good genetic characteristics of ecosystem individuals, in order to reduce computational cost.

Each individual who has a fitness value higher than the average fitness of destination subpopulation receives visa for migrating to destination subpopulation. To reduce the computational time in process, we use assignment policy in the migration scheme. Therefore, migration only causes move of individuals between subpopulations while the total size of the ecosystem remains constant across evolution.

The next two genetic operators are crossover and mutation. Crossover enhances the ability of evolution by exchanging the information of the parent generation; while mutation enhances the ability of evolution by introducing new characters into the strings. In fact, the fitness value of a string may become worse when either the crossover rate or mutation rate is too high. update probability of genes in uniform crossover were determined according to the concept of [49]. Consequently, the crossover rate of ith subpopulation in our study was determined analogously by:

$$pc_i = \left(\left|\left(\bar{f_i} - bd\bar{f}\right) / \left(bs\bar{f} - bd\bar{f}\right)\right| * 0.25\right) + 0.5 \tag{2}$$

Where, $\bar{f_i}$ is Average fitness of population p_i , $bd\bar{f}$ is The worst average fitness of populations in ecosystem, $bs\bar{f}$ is The best average fitness of populations in ecosystem and probability of crossover operation on all individuals is 1 and the crossover operation is controlled by updating probability of genes. A random number between 0 and 1 is selected for each bit of the chromosome. 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.

Using of this update we can see that, if the Average fitness of individuals in a population is worse from the best average fitness of populations in ecosystem, then the probability of crossover is increased. In an indirect way, update on the probability of crossover links to fitness of population. From update on Average fitness of individuals in a population, it is clear that probability of crossover increases for bad subpopulations and decreases for good

Table. , for crossover probability greater than 0.75 we do not see improvement in resulted solutions. Therefore, restrict the interval of p_c to [0.5, 0.75].

The next changing parameter across the process is the mutation rate which updates the mutation probability of individuals in each subpopulation. Since mutation operator destroys the structure of good chromosomes. In addition, mutation brings a new individual into the ecosystem with new attributes; we determine a small mutation rate across evolution to increase the explorative power of model. Mutation probability in i^{th} subpopulation is chosen arbitrary from [0.0001, 0.0002] and updated by

$$pm_i = 0.0001 + \left|\left(\left(\bar{f_i} - bd\bar{f}\right) / \left(bs\bar{f} - bd\bar{f}\right)\right) - 1\right| * 0.0001 \qquad (3)$$

Mutation probability parameter changes the size of each subpopulation. It acts based on the average fitness of subpopulation compared to the difference between the best and the worst average fitness of subpopulation and hence search is more explorative and exploitative. The constant amounts in Equation 2 and 3 determines by a single run of GA on data with different population size ([500, 5000]). As it is shown in Table 1

subpopulations in the ecosystem. Obviously, higher the mutation rate, more explorative is the search. The factor of 0.0001 chooses arbitrarily as it is a considerably small factor to update probability of mutation.

This update for crossover and mutation operator is important to increase the mutation and crossover rates in bad regions and to decrease the min good regions. Therefore, this update increases the explorative power of search for individuals in bad regions of the state space and decreases exploitative power of search for individuals in the high fitness regions of the state space.

On the other hand, transferring some individuals to the next generation without any change increases the exploitative power and decreases the time of selection, crossover and mutation. Therefore, the forth changing parameter for this algorithm is the number of individuals that transfer to the next generation without any change. This parameter is updated by,

$$pk_i = \left(\left(\left|\bar{f_i} - \bar{f}\right|\right)/\bar{f}\right) \qquad (4)$$

Table. 1. Results or genetic algorithm with different initial population size and crossover probability

Initial population											
5000	**4500**	**4000**	**3500**	**3000**	**2500**	**2000**	**1500**	**1000**	**500**		
0.1512	0.1651	0.1751	0.1665	0.1765	0.1986	0.1676	0.2133	0.2717	0.2913	**0.5**	Crossover probability
0.1675	0.1752	0.1712	0.1623	0.1834	0.1934	0.245	0.2165	0.2613	0.3154	**0.55**	
0.1612	0.1835	0.1734	0.1524	0.1945	0.1964	0.2137	0.2123	0.2432	0.3366	**0.6**	
0.1545	0.1741	0.1743	0.1608	0.1865	0.1837	0.275	0.2229	0.2426	0.3234	**0.65**	
0.1531	0.1668	0.1623	0.1676	0.1778	0.1754	0.1936	0.2334	0.2448	0.2721	**0.7**	
0.1621	0.1778	0.1556	0.1718	0.1776	0.1724	0.1867	0.2484	0.2353	0.2856	**0.75**	
0.1735	0.1735	0.1534	0.1834	0.1813	0.1707	0.1839	0.2425	0.2369	0.3123	**0.8**	
0.1645	0.1615	0.1639	0.1965	0.1823	0.1834	0.1636	0.2236	0.2437	0.2817	**0.85**	
0.1639	0.1602	0.184	0.1923	0.1947	0.267	0.2674	0.2346	0.2475	0.2612	**0.9**	

Where, $\bar{\mathrm{f}}$ is average fitness of ecosystem. Using this update, subpopulations with higher average fitness can transfer more individuals to the next generation without any change to increase the exploitative power and speed of the process.

5. Problem Definition

Let M be a set with n point 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 [50]. We use discrete optimization techniques to solve this problem. Therefore, we approximate the continuous q-dimensional space by a lattice N which each cell has a center point. Thus, the data visualization problem can be similar to assigning m points to$n$ cells (center points). Decision variables are:

$$x_{ik} = \begin{cases} 1, if\ the\ ith\ instance\ is\ assigned\ to\ lattice\ point\ k \in N \\ 0, otherwise \end{cases} \tag{5}$$

Therefore, the problem can be formulated as:

$$\begin{aligned} & min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k \in N} \sum_{l \in N} F(D_{i,j}^{old}, D_{k,l}^{new}) x_{ik} x_{jl} \\ & subject\ to \sum_{k \in N} x_{ik} = 1, \quad \forall i \\ & x_{ik} \in \{0,1\} \end{aligned} \tag{6}$$

In which $D^{old} \in R^n \times R^n$ is a matrix measuring the distances between nn 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 deviation between the differences between the instances in the original space and the new q-dimensional space. Choices for F include 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 reformulates as:

$$\begin{aligned} & Min \frac{1}{\sum_{i=1}^{m} \sum_{\substack{j=1 \\ j>i}}^{n} d(i,j)} \sum_{i=1}^{m} \sum_{\substack{j=1 \\ j>i}}^{m} \sum_{k=1}^{n} \sum_{l=1}^{n} \left(\frac{(d(i,j) - d^*(k,l))^2}{d(i,j)} \right) x_{ik} x_{jl} \\ & subject\ to \sum_{k=1}^{n} x_{ik} = 1, \forall i \\ & x_{ik} \in \{0,1\} \end{aligned} \tag{7}$$

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

The number of cells determines by the requirements of the visualization method. A problem with sparse points will require a larger grid than one with clustered points. 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 any pair of points in a given data set. Let the greatest distance in the chosen lattice N be b.We multiply all original distances between points in M byb/a, so that our lattice scales to the given problem. The problem of assigning m points to n lattice points cannot be treated as a linear assignment problem because in a linear assignment problem the cost of assigning of one point to a lattice point does not depend on the assigning of the other points. However, this is not the case for data visualization problems. Therefore, this problem is a quadric assignment problem (QAP) in which the objective function is a convex and it can have many local solutions. On the other hand, the space of this problem is very large and decision variables are dependent.

Any solution method can be used to solve this problem but it should be effective in circumstances of problem. Here, we use the proposed Genetic Algorithm. Each chromosome is represented by n genes, g_i; chromosomes are shown in . n is the number of instances and each gene in the chromosome, g_i, shows the i^{th} data point in the original space assigned to g_i cell in lattice. The fitness of each chromosome is calculated by the subjective function in equation (7).

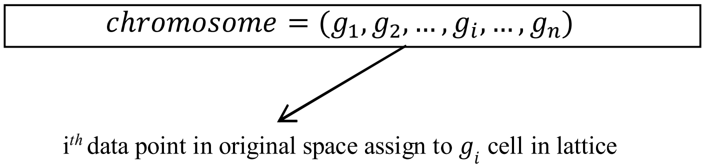
$chromosome = (g_1, g_2, \dots, g_i, \dots, g_n)$
i^{th} data point in original space assign to g_i cell in lattice

Fig.3 Chromosome representation

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

6. Experiment and Discussion

To verify the DSAGAM for data visualization problem, we used electrical loads data taken from the dispatching center in Esfahan (Iran). The data set used in this problem includes 24-hour electrical loads of 366 days starting from 20 March 2008 in Esfahan. Therefore, Each data array ($n = 366$) reflects the load behavior associated to a day demand throughout the year. The dimension of the output map is considered as two ($q = 2$), to be visible easily. By using the heuristic method in SOM toolbar of MATLAB[51], the output map is assumed a 60×60 grid square.

Each population of the ecosystem run in parallel on different processors since the algorithm is a parallel genetic algorithm. The algorithm is coded here in Microsoft Visual C# 2008, and run on a single processor desktop computer with 2 Core 1.6 GHz machine CPU and 1GB RAM under WINXP platform, which was similar to all algorithms. The stochastic nature of a GA optimization requires multiple runs to ensure reproducibility. Thus, the different GA methods run 10 times, then average of solutions of each time regarded as result of any method to compare.

In the our experiments, the SGA had a population with random initial sizes selected uniformly from [500,5000] and IGA, SAMGA and DSAGMA had 10 populations with random initial sizes selected uniformly from [50,500]. For SGA and IGA the probability of uniform crossover chosen was 0.7 and mutation rate 0.01.

Each population in the ecosystem is regarded a node of the linked list. Any node consist the population parameters like probability replacement of genes in crossover, rate migration, rate mutation, and the number of unchanged individuals. Further, the population contains an array of bits, which represent the individuals of the population. The genotype to phenotype conversion and fitness evaluation performs by a fitness function given as

$$\frac{1}{\sum_{i=1}^{366}\sum_{\substack{j=1\\ j>i}}^{3600} d(i,j)}\sum_{i=1}^{366}\sum_{\substack{j=1\\ j>i}}^{366}\sum_{k=1}^{3600}\sum_{l=1}^{3600}\Big(\big((d(i,j)-d^{*}(k,l))^{2}/d(i,j)\big)\Big)\,x_{ik}x_{jl} \qquad (8)$$

Where, individuals must satisfy following constraints:

$$\text{s.t: } \sum_{k=1}^{3600} x_{ik}=1,\ \forall i \qquad (9)$$

$$x_{ik}\in\{0,1\}$$

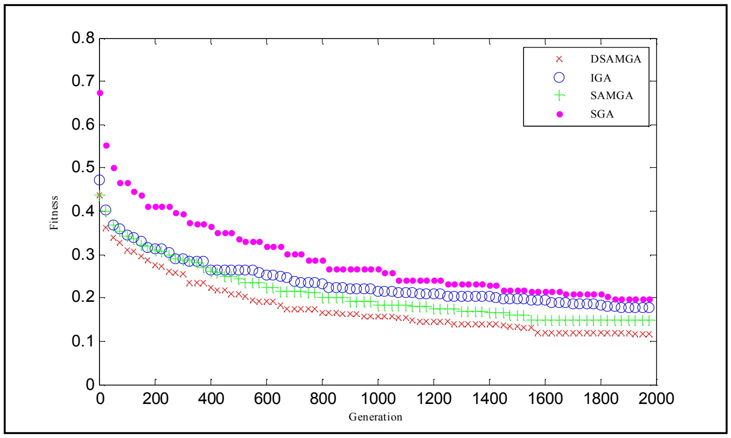

Fig.4 Convergence of DSAGAM and other GAs for data visualization problem.

Fig.4 shows the fitness value after 2000 iterations of our proposed algorithm (DSAGAM), IGA, SAMGA and SGA for data visualization problem in terms of generations. The same communication (line) topology uses in IGA, SAMGA and DSAGAM. SGA struggles with some local optimal solution that it causes to progress of GA be very slow. Moreover, the performance of SGA is the worst. IGA and SAMGA have similar performances in start with generation K less than 100. However, SAMGA is about 5-6% better than IGA in the later generations. This is because the IGA is not able to tune the solution for better stiffness. Therefore, it could not reach the global optimal solution and there is a risk to trap in local solutions. The progress of SAMGA is slow in the initial stages, because it searches appropriate genetic parameters initially and improve resulted solutions in later stages by identification of optimum parameters. The DSAGAM converges much faster than other methods and clearly outperforms all other GA approaches in the present work. Although progress of SAMGA and DSAGAM is similar initially, effective parameter updating results in improved solutions in the DSAGAM. This experiment proves that the exploitative and explorative power of the DSAGAM is higher than SAMGA.

The results using DSAGAM with different communication topologies are shown in Fig.5. It shows that different topologies have different performances. Therefore, DSAGAM with more efficient communication topology can reach a better optimal solution.

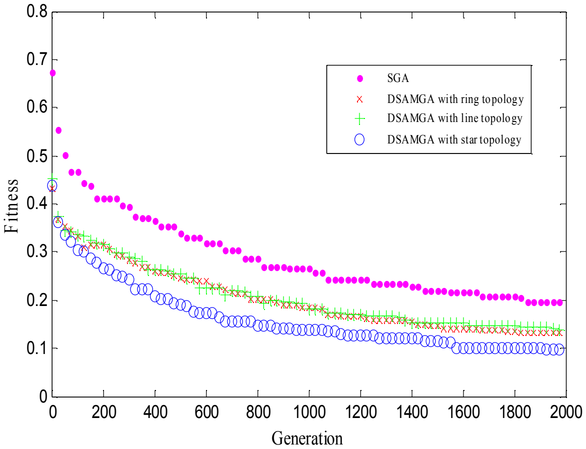

Fig.5 Convergence of the DSAGAM with different communication topologies for data visualization problem.

Fitness values corresponding to DSAGAM with line, star, and ring, IGA, SAMGA and SGA are respectively 0.14, 0.1, 0.13, 0.17, 0.15 and 0.2., as shown in Fig.5. It can be clearly observed that the DSAGAM with star topology has been better than other Gas and thus offering 28, 23, 41, 33 and 50% improvement in the fitness values over DSAGAM with line, ring, IGA, SAMGA and SGA respectively. Moreover, the SAMGA reaches to a local solution after 2000 generations while the DSAGAM with star topology reaches to it within 553 generations. Therefore, the DSAGAM with star topology is able to reduce the computational effort by a fraction of about 1/3.6 compared to the SAMGA. However, the IGA and SGA could not reach this value and the DSAGAM with ring and line topology reach this value after 1740 and 1860 generations, respectively.

In order to apply Sammon's mapping we used MDSCALE toolbar of MATLAB. In Sammon's mapping, the learning rate is set to 0.2 and the maximum number of iteration is 2000 according to [52]. Based on Eq. (8), the fitness of resulted solution from Sammon's mapping is 0.19 . It shows that DSAGAM improve at least 20% the solutions quality compared to Sammon's mapping. As error in the Sammon's mapping is minimized by the steepest descent procedure, it easily tropes in local minima. Besides, the run time of DSAGAM does not exceed from 0.25h, while the Sammon's mapping could not give solution earlier than 0.75h. Because, the distance calculations of $N(N\_1)/2$ times result in the computational complexity of Sammon's mapping $O(N^2)$ is large.

7. Analyzing Daily Electrical Load

In section 5, we concluded that DSAGAM with star topology having a fitness value of 0.1 results in the most precise solution among other approaches in projecting the data to a 2-dimensional map problem. Therefore, we start from this point to analyze electrical load patterns.

Resulted map should contain the necessary information to evaluate the association 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. An efficient approach for it is finding and extracting of load pattern from historical data. The load pattern recognition reduce power costs and facilitate energy management.

The first phase of data analysis is data labeling to better understanding of data and results. We suggest assignment a numeric label to daily load as 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 directly presents on the map.

The selection of the number of clusters is a significant task in data visualization problem that it should choose by a reasonable manner. The appropriate number of clusters help visual view of user and improve his/her understanding about hidden relations. As shown in Fig. 6, daily load behavior classify into 11 clusters while days located in any cluster have similar demand patterns. The clusters and days corresponding to each one are given in Table 2. As shown, days can be divided into different classes. The most attractive region is region 1 which includes cold days in year. Electrical load pattern in these days is the same 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 from region 1. This disparity is due to seasonal temperature changes and different appliances used. These days belong to warm days. Electrical load in these days increases significantly.

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

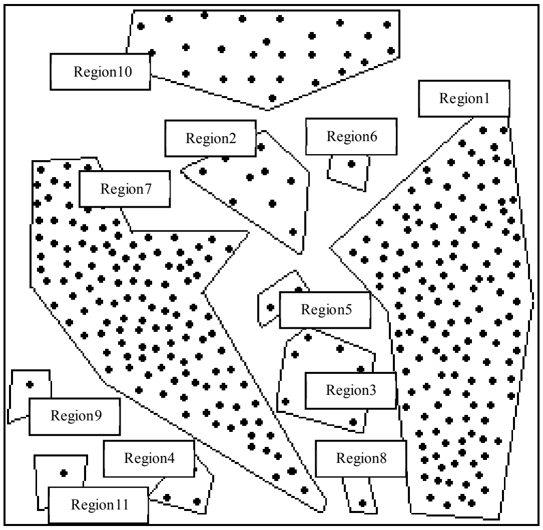

Fig. 6 Best chromosome map resulted from QAP-GA method.

Table 2. Classified days with 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

8. Conclusions

We propose a Distributed Self Adaptive Genetic Algorithm with Migration (DSAGAM) for the data visualization problem that its parameters automatically adjust themselves across evolution to increase the exploration power of the algorithm in low fitness regions and to increase the exploitation in high fitness regions. Moreover, our proposed approach has efficient solution for escaping of local optimization by parallel optimization and migration. Because, if a subpopulation struggle a local optima, migration of individuals to it could escape them. Since the migration method has a significant impact on the convergence performance and the improvement of the solutions in the evolution process, we use different migration strategies and analyze the performance of each of them. The efficient communication topology and updating parameters cause to deeper exploitation of the promising regions of the search space, and preserves them from destroying their structure. Results of experiments on a real problem of data visualization for identification and segmentation of load consumer based on their pattern behavior, show DSAGAM with different communication topologies has better performance compared to SGA, IGA, SAMGA and Sammon's mapping. Moreover, it is more reliable in finding the optimal solutions, while uses the lowest number of generations in finding the optimal solution. However, we improve our proposed DSAGAM by its combination with other meta-heuristic methods. On the other hand, although we evaluate on a popular and real problem, but applying of it on other problems could propose as future work. Moreover, our proposed approach need again computation when new data points are added but it could fulfill by preservation of previous computations that it propose as future research.

Acknowledgement

The authors are grateful to the anonymous referees for their constructive comments and suggestions. This research was partially supported by are search project grant by Esfahan Regional Electric Company (EREC) in Iran.

References

[1] 1 Li, Q., Liu, Y., Zhang, Z., Liu, J., Yuan, Y., Wang, K., and He, R.: 'Learning with incomplete labels of multisource datasets for ECG classification', Pattern Recognition, 2024, 150, pp. 110321

[2] 2 Liu, C., Cui, Y., Li, X., and Yao, M.: 'microeco: an R package for data mining in microbial community ecology', FEMS microbiology ecology, 2021, 97, (2), pp. fiaa255

[3] 3 Alves, C.L., and Rodrigues, F.A.: 'Diagnosis of neurological diseases based on data mining and complex network', Anais, 2019

[4] 4 Sun, X., Li, B., Sutcliffe, R., Gao, Z., Kang, W., and Feng, J.: 'Wse-MF: A weighting-based student exercise matrix factorization model', Pattern Recognition, 2023, 138, pp. 109285

[5] 5 Penteado, B.E., Paiva, P.M.P., Zupelari, M.M., Lopes, N.B.F., Ferrari, D.V., and Isotani, S.: 'Educational data mining methods: opportunities for teaching and learning in Audiology', in Editor (Ed.)^(Eds.): 'Book Educational data mining methods: opportunities for teaching and learning in Audiology' (2017, edn.), pp.

[6] 6 Paiva, P.M.P., Penteado, B., Zupelari, M.M., Lopes, N.B.F., Costa Filho, O.A., Isotani, S., and Ferrari, D.V.: 'Educational data mining for understanding students' behavior in a virtual learning environments: a case study', in Editor (Ed.)^(Eds.): 'Book Educational data mining for understanding students' behavior in a virtual learning environments: a case study' (2017, edn.), pp.

[7] 7 Cano, A., and Krawczyk, B.: 'Evolving rule-based classifiers with genetic programming on GPUs for drifting data streams', Pattern Recognition, 2019, 87, pp. 248-268

[8] 8 Dajer, M.E., Tsutsumi, M., Pimienta, R.A., de Olivera, V.H.C., Isotani, S., Delbem, A.C.B., Hachiya, A., and Tsuji, D.H.: 'Preliminary study of Data Mining analysis of high speed Kymography and voice data', Areté, 2018, 18, (2), pp. 11-20

[9] 9 De Nadai Fernandes, E.A., Tagliaferro, F.S., Azevedo-Filho, A., and Bode, P.: 'Organic coffee discrimination with INAA and data mining/KDD techniques: new perspectives for coffee trade', Accreditation and quality assurance, 2002, 7, pp. 378-387

[10] 10 Rodrigues, L., Rodrigues, L., and Bacchi, M.R.P.: 'Performance evaluation of forecasting models based on time series and machine learning techniques: an application to light fuel consumption in Brazil', International Journal of Energy Sector Management, 2022, 16, (4), pp. 636-658

[11] 11 Paletta, F.C., and Tavares, L.W.: 'Complex network research data mining', Anais, 2020

[12] 12 Mabu, A.M., Prasad, R., and Yadav, R.: 'Mining gene expression data using data mining techniques: A critical review', Journal of Information and Optimization Sciences, 2020, 41, (3), pp. 723-742

[13] 13 Mohd Selamat, S.A., Prakoonwit, S., Sahandi, R., Khan, W., and Ramachandran, M.: 'Big data analytics—A review of data□mining models for small and medium enterprises in the transportation sector', Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018, 8, (3), pp. e1238

[14] 14 Koukouvinis, P., Rodriguez, C., Hwang, J., Karathanassis, I., Gavaises, M., and Pickett, L.: 'Machine Learning and transcritical sprays: A demonstration study of their potential in ECN Spray-A', International Journal of Engine Research, 2022, 23, (9), pp. 1556-1572

[15] 15 Niebel, T., Rasel, F., and Viete, S.: 'BIG data–BIG gains? Understanding the link between big data analytics and innovation', Economics of Innovation and New Technology, 2019, 28, (3), pp. 296-316

[16] 16 Hasugian, P.M., and Nadeak, B.: 'Bibliometric Analysis On Techniques For Data Visualization', Jurnal Info Sains: Informatika dan Sains, 2024, 14, (01), pp. 425-433

[17] 17 Silva, R.B., Oliveira, D.d., Santos, D.P.d., Santos, L.F.D., Wilson, R.E., and Bêdo, M.V.N.: 'Criteria for choosing the number of dimensions in a principal component analysis: An empirical assessment', Anais, 2020

[18] 18 Mair, P., Groenen, P.J., and de Leeuw, J.: ‘More on multidimensional scaling and unfolding in R: smacof version 2’, Journal of Statistical Software, 2022, 102, pp. 1-47

[19] 19 Du, M., and Yuan, X.: ‘A survey of competitive sports data visualization and visual analysis’, Journal of Visualization, 2021, 24, pp. 47-67

[20] 20 Stryhal, J., and Plavcová, E.: ‘On using self□organizing maps and discretized Sammon maps to study links between atmospheric circulation and weather extremes’, International Journal of Climatology, 2023, 43, (6), pp. 2678-2698

[21] 21 Nanga, S., Bawah, A.T., Acquaye, B.A., Billa, M.-I., Baeta, F.D., Odai, N.A., Obeng, S.K., and Nsiah, A.D.: ‘Review of dimension reduction methods’, Journal of Data Analysis and Information Processing, 2021, 9, (3), pp. 189-231

[22] 22 Koopmans, T., and Beckmann, M.: ‘Assignment problems and the location of economic activities’, Econometrica, 1957, 25, pp. 53–76

[23] 23 Verel, S., Thomson, S., and Rifki, O.: ‘Where the Really Hard Quadratic Assignment Problems Are: the QAP-SAT instances’, arXiv preprint arXiv:2403.02783, 2024

[24] 24 Ande, A., Subhash, V., and Natarajan, V.: ‘tachyon: Efficient Shared Memory Parallel Computation of Extremum Graphs’, in Editor (Ed.)^(Eds.): ‘Book tachyon: Efficient Shared Memory Parallel Computation of Extremum Graphs’ (Wiley Online Library, 2023, edn.), pp. e14784

[25] 25 Shao, X., Dai, X., and He, X.: ‘Noise robustness and parallel computation of the inverse compositional Gauss–Newton algorithm in digital image correlation’, Optics and Lasers in Engineering, 2015, 71, pp. 9-19

[26] 26 Yan, B., and Regueiro, R.A.: ‘Three□dimensional discrete element method parallel computation of Cauchy stress distribution over granular materials’, International Journal for Numerical and Analytical Methods in Geomechanics, 2019, 43, (5), pp. 974-1004

[27] 27 Zhu, Z.-j., Ma, S.-q., Zhu, X.-Q., Lan, Q., Piao, S.-C., and Cheng, Y.-S.: ‘Parallel optimization of underwater acoustic models: A survey’, Chinese Physics B, 2022, 31, (10), pp. 104301

[28] 28 Zhong, M., Yue, Y., Zhou, L., and Zhu, J.: ‘Parallel optimization method of train scheduling and shunting at complex high□speed railway stations’, Computer□Aided Civil and Infrastructure Engineering, 2024, 39, (5), pp. 731-755

[29] 29 Xue, X., Yang, C., Mao, G., and Zhu, H.: ‘Semi-automatic ontology matching based on interactive compact genetic algorithm’, International Journal of Pattern Recognition and Artificial Intelligence, 2022, 36, (05), pp. 2257002

[30] 30 Li, Y., Wu, L., Wang, T., Gao, N., and Wang, Q.: ‘EEG signal processing based on genetic algorithm for extracting mixed features’, International Journal of Pattern Recognition and Artificial Intelligence, 2019, 33, (06), pp. 1958008

[31] 31 Zubel, M., Ottenwälder, T., Heuser, B., and Pischinger, S.: ‘Combustion system optimization for dimethyl ether using a genetic algorithm’, International Journal of Engine Research, 2021, 22, (1), pp. 22-38

[32] 32 Ren, Y., Tang, J., Yu, Y., and Li, X.: ‘A two-stage stochastic programming model and parallel Master–Slave adaptive GA for flexible Seru system formation’, International Journal of Production Research, 2024, 62, (4), pp. 1144-1161

[33] 33 Cui, Z., Zhao, B., Zhao, T., Cai, X., and Chen, J.: ‘An adaptive differential evolution algorithm based on archive reuse’, Information Sciences, 2024, pp. 120524

[34] 34 Mishra, R., and Bajpai, M.K.: ‘A priority-based self-guided serial-parallel genetic algorithm for low-dose computed tomography’, Applied Soft Computing, 2024, pp. 111520

[35] 35 Deng, W., Zhang, X., Zhou, Y., Liu, Y., Zhou, X., Chen, H., and Zhao, H.: ‘An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems’, Information Sciences, 2022, 585, pp. 441-453

[36] 36 Rahimi, I., Gandomi, A.H., Deb, K., Chen, F., and Nikoo, M.R.: ‘Scheduling by NSGA-II: Review and bibliometric analysis’, Processes, 2022, 10, (1), pp. 98

[37] 37 Teodosio, L., and Berni, F.: ‘Optimization via genetic algorithm of a variable-valve-actuation spark-ignition engine based on the integration between 1D/3D simulation codes and optimizer’, International Journal of Engine Research, 2023, 24, (4), pp. 1760-1784

[38] 38 Dushatskiy, A., Virgolin, M., Bouter, A., Thierens, D., and Bosman, P.A.: ‘Parameterless gene-pool optimal mixing evolutionary algorithms’, Evolutionary Computation, 2023, pp. 1-28

[39] 39 Przewozniczek, M.W., Dziurzanski, P., Zhao, S., and Indrusiak, L.S.: ‘Multi-Objective parameter-less population pyramid for solving industrial process planning problems’, Swarm

and Evolutionary Computation, 2021, 60, pp. 100773

[40] 40 Aygun, H., and Turan, O.: ‘Application of genetic algorithm in exergy and sustainability: A case of aero-gas turbine engine at cruise phase’, Energy, 2022, 238, pp. 121644

[41] 41 Back, T.: ‘Self-adaptation in genetic algorithms’, in Editor (Ed.)^(Eds.): ‘Book Self-adaptation in genetic algorithms’ (1992, edn.), pp. 263-271

[42] 42 Krink, T., and Ursem, R.K.: ‘Parameter control using the agent based patchwork model’, in Editor (Ed.)^(Eds.): ‘Book Parameter control using the agent based patchwork model’ (2000, edn.), pp.

[43] 43 Murata, Y., Shibata, N., Yasumoto, K., and Ito, M.: ‘Agent oriented self adaptive genetic algorithm’, Proceedings of Communications and Computer Networks (CCN), 2002, pp. 348-353

[44] 44 Weinberg, R., and Berkus, M.: ‘Computer simulation of a living cell: Part I’, International Journal of Bio-Medical Computing, 1971, 2, (2), pp. 95-120

[45] 45 Hiroyasu, T., Kaneko, M., and Hatanaka, K.: ‘A parallel genetic algorithm with distributed environment scheme’, in Editor (Ed.)^(Eds.): ‘Book A parallel genetic algorithm with distributed environment scheme’ (Citeseer, 1999, edn.), pp. 695-700

[46] 46 Takashima, E., Murata, Y., Shibata, N., and Ito, M.: ‘Asynchronous Implemention of Self Adaptive Island Genetic Algorithm’, Joho Shori Gakkai Kenkyu Hokoku, 2003, 2003, (91), pp. 65-68

[47] 47 Takashima, E., Murata, Y., Shibata, N., and Ito, M.: ‘Techniques to improve exploration efficiency of parallel self adaptive genetic algorithms by dispensing synchronization’, in Editor (Ed.)^(Eds.): ‘Book Techniques to improve exploration efficiency of parallel self adaptive genetic algorithms by dispensing synchronization’ (Citeseer, 2004, edn.), pp.

[48] 48 Srinivasa, K.G., Venugopal, K.R., and Patnaik, L.M.: ‘A self-adaptive migration model genetic algorithm for data mining applications’, Information Sciences, 2007, 177, (20), pp. 4295-4313

[49] 49 Chien-Pang Lee, W.-S.L., Yuh-Min Chen, Bo-Jein Kuo: ‘Gene selection and sample classification on microarray data based on adaptive genetic algorithm/k-nearest neighbor method’, Expert Systems with Applications, 2011, 38, pp. 4661–4667

[50] 50 Abbiw-Jackson, R., Golden, B., Raghavan, S., and Wasil, E.: ‘A divide-and-conquer local search heuristic for data visualization’, Computers and operations research, 2006, 33, (11), pp. 3070-3087

[51] 51 Widayanti, R., Madenda, S., Wibowo, E.P., and Anwar, K.: ‘SOM-SIS approach to auto summary of clustering results on university academic performance’, TELKOMNIKA (Telecommunication Computing Electronics and Control), 2023, 21, (1), pp. 104-112

[52] 52 Yang Xu, Lu Xu, and W.S.Chow, T.: ‘PPoSOM: A new variant of PolSOM by using probabilistic assignmen tfor multi dimensional data visualization’, Neuro computing, 2011, 74, pp. 2018–2027