PARALLEL EVOLUTIONARY BICLUSTERING OF SHORT-TERM ELECTRIC ENERGY CONSUMPTION

Diego P. Pinto-Roa[1,4*], Hernán Medina[4], Federico Román[4], Miguel García-Torres[1,2], Federico Divina[1], Francisco Gómez-Vela[1], Félix Morales[1], Gustavo Velázquez[1], Federico Daumas[1], José L. Vázquez-Noguera[1], Carlos Sauer Ayala[3] and Pedro E. Gardel-Sotomayor[5]

[1]ComputerEngineerDepartment, Universidad Americana, Paraguay,
[2]Division of Computer Sience, Universidad Pablo de la Ovide, Seville, Spain
[3]Facultad de Ingeniería, Universidad Nacional de Asunción, Paraguay
[4]Facultad Politécnica, Universidad Nacional de Asunción, Paraguay,
[5]Universidad Católica de Asunción, Ciudad del Este, Paraguay

ABSTRACT

The discovery and description of patterns in electric energy consumption time series is fundamental for timely management of the system. A bicluster describes a subset of observation points in a time period in which a consumption pattern occurs as abrupt changes or instabilities homogeneously. Nevertheless, the pattern detection complexity increases with the number of observation points and samples of the study period. In this context, current bi-clustering techniques may not detect significant patterns given the increased search space. This study develops a parallel evolutionary computation scheme to find biclusters in electric energy. Numerical simulations show the benefits of the proposed approach, discovering significantly more electricity consumption patterns compared to a state-of-the-art non-parallel competitive algorithm.

KEYWORDS

Biclustering, Big data, Electric energy consumption, Parallel evolutionary computation.

1. INTRODUCTION

The electricity demand is continuously growing, due to different reasons, e.g., population increment, but even lifestyle that demands more energy. For this reason, it is essential to monitor the distribution-level consumption to detect abnormal activity, such as a higher than usual demand on some systems. Doing that that would allow the system managers to carry out actions that can correct such abnormal situations.

Different data analysis techniques are usually used in the processing of these data, such as classification [1,2], forecasting [2–8], and clustering [9–12]. To this aim, biclustering can come in handy. In fact, in biclustering, it is possible to group data in two dimensions simultaneously; an example is a bicluster that can group energy consumption and time.

Biclustering has been mostly applied in bioinformatics, particularly in the context of microarray data. The primary purpose is to find subsets of genes presenting similar patterns, in terms of

David C. Wyld et al. (Eds): CMC, NCO, SOFT, CDKP, MLT, ICAITA - 2021
pp. 123-137, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111110

expression values, under a subset of experimental conditions [13]. For instance, in [14], a scatter search approach based on linear correlations among genes is used to find biclusters. In [15], the authors proposed a multi-objective evolutionary algorithm (MOEA) to detect biclusters presenting particular characteristic. Biclustering has also been applied in other fields, such as social network datasets [16] or text mining [17]. In [18], a biclustering method identifies the most suitable group of user friends in social network datasets. In [19], a multi-objective biclustering algorithm was applied for the first time to the time series of the electricity consumption in smart buildings.

The energy consumption data can be usefully treated as time series data over a period. These time-series data are extensively used in different applications such as science, engineering, finance, economics, communications, control, health care, government, among others [22,21]. Clustering time series, which can be defined as identifying the homogeneous groups of time-series data based on their similarity [22], is an important technique that can provide knowledge from raw energy data [23]. Traditional clustering techniques are not always enough in all applications because these techniques usually only look for similarities in the entire time series. In some cases, finding similarities over specific periods is significant.For example, finding consumers with shared electricity consumption characteristics at limited periods, like peak demand hours, can yield better customer segmentation. Good customer segmentation can improve the performance of demand response programs [24]. Biclustering methods could fit to study concrete periods on energy consumption time series [22]. Biclustering consists of simultaneous partitioning of samples and their attributes (features) into subsets (classes). Samples and features classified together are supposed to have high relevance to each other [19].

Although competitive techniques addressed the bicluster problem, there is a large margin to be improved as the solution space grows, as is the case with energy consumption time series. In this context, parallel computing has emerged as a promissory alternative. The parallel-evolutionary computation finds satisfactory solutions reducing the computation time in high complexity problems [26]. Consequently, in this paper, we propose developing a parallel-computing architecture extending one of the most competitive state-of-the-art techniques presented by Divina et al. [19] to the energy consumption time series.

The paper is organized as follows. Section 2 presents the relate works while Section 3 presents an introduction of the electric energy consumption data and some basic notions of biclustering and evolutionary computation. Section 4 describes the procedures of the proposed Parallel Evolutionary Biclustering, while Section 5 shows the data, the simulation setup, and results. Finally, Section 6 gives the main conclusions and future works.

2. RELATED WORKS

Understanding different patterns of energy consumption, or measuring the environmental impact of energy production, can help in the development of new strategies to respond to the growing energy demand [27] and, therefore, to have a more sustainable energy policy respectively [28].
Many papers have been published recently aiming at improving power consumption prediction and pattern discovery. To this end, machine learning (ML) approaches emerge as the most important [29-31].

As example, the work presented by Liu et al. [32], where the authors presented a support vector machine (SVM) method to forecasting and diagnose public buildings energy consumption based on different input parameters, such as historical energy consumption data, climatic factors and time-cycle factors. The work was carried out on a dataset from city of Wuhan (China), and their results showed that they were able to detect that air conditioning energy consumption was abnormal for four days in September.

In a recent work presented in [33], the authors proposed a model predictive control system with adaptive machine-learning-based building models for building automation and control applications. The results show that the proposed model reduces 58.5% cooling thermal energy consumption in the office and 36.7% cooling electricity consumption in the lecture theatre, when it is compared against their respective original control.

Artificial neural networks (ANNs) stand out as one of the most important approach among the different ML-based techniques for analysis and prediction of short-term energy consumption patterns [34-36]. As representative work, the early one presented by Nizami and Ai-Garni [37], which proposed a two-layer forward-fed ANN to study how weather-related characteristics can affect the prediction of monthly electricity consumption.

Recently, new ensemble methods are also gaining interest due to their improved results by combining several techniques. As example, the work by Divina et al. [12], where the authors proposed a new strategy based on ensemble learning in order to tackle the short-term load forecasting problem. The approach was based on the predictions produced by three base learning methods.

While clustering techniques have been less used in the literature for the analysis of energy data than others ML approaches, it is worth noting that there are several relevant works in this field. For example, the work by Diao et al. [38], where the authors proposed to identify and classify behaviour of occupants with direct energy consumption outcomes and energy time use data through unsupervised clustering. The results showed that the model was able to automatically estimating energy consumption on even larger geographic scale. Another example is presented by Perez-Chacon et al. [39], where clustering techniques was also used to identify energy consumption pattern in smart cities in a big data context. The authors proposed a parallelized method, based on the study of four clustering validity indices, to extract electric energy consumption patterns in big data time series. The method was tested using electricity consumption for the years 2011–2017 for eight buildings of a public university. Finally, in a recent work by Divina et al. [16], the authors proposed the first application of a biclustering algorithm to detect anomalies in the energy consumption patter. The algorithm was applied on data from smart buildings of a Spanish university campus. The results achieved showed that the proposed approach can help policy makers in detecting irregular situations.

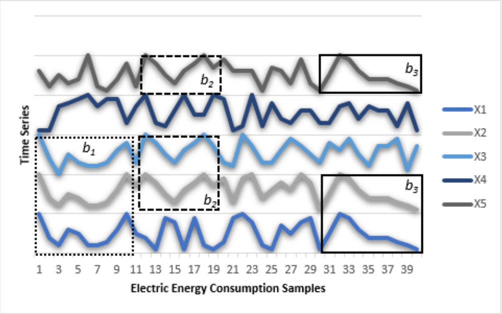
3. Background

This section introduces the electric energy consumption data and provides the basic concepts of biclustering and evolutionary computation.

3.1. Electric Energy Consumption Dataset

Electric energy consumption data are usually modeled as a time series since it consists of a discrete sequence of data points measured at equal time intervals [19]. Let $Y = \{Y_i\}_{i=1}^{N}$ be a sample of N univariate time series where $Y_i = \{Y_{i,t}\}_{t=1}^{T}$ is a univariate time series characterized by T real values. Then, the sample Y can be represented through a matrix $M_{N\times T}$. On the other hand, let us define a bicluster b as a submatrix $S_{I\times J}$, with $|I| \leq N$ and $|J| \leq T$. In the context of time series,$S_{I\times J}$ is subject to the additional condition that columns are consecutive, i.e., $J = \{j_k, j_k + 1, j_k + 2, \dots, j_k + |J| - 1\}$. Given a function $f(b)$that measures the quality of a bicluster b, the objective is to find the best B biclusters $\{b_k\}_{k=1}^{B}$ according to $f(.)$.

In our context, each time series represents a sequence of sensor data collected over time. Therefore, the data can be viewed as $N \times T$ energy consumption data matrix EM. EM is a real matrix, where each element e_{ij} represents the electric energy consumption (expressed in kWh) as measured by sensor i on sample j. We can then see a bicluster as the measurements registered by a subset of sensors over a subset of consecutive days. For example, let us look at Figure 1 a five-sensor array for 40 sample days at the left side, while three biclusters are at the right side. The bicluster arrows correspond to sensors while the columns the consecutive days where a pattern was observed.



b_1	1	2	3	4	5	6	7	8	9	10
X1	10	4	2	6	5	2	2	3	6	10
X2	20	14	12	15	14	12	12	13	16	20
X3	30	24	20	25	23	22	22	23	26	28

Virtual Error= 0.305
Row variance= 8.323
Volume= 30

b_2	12	13	14	15	16	17	18	19
X2	20	18	15	13	16	18	20	17
X3	30	28	25	23	26	28	30	27
X5	50	48	45	43	46	48	50	47

Virtual Error= 0.555
Row variance= 5.1
Volume= 24

b_3	30	31	32	33	34	35	36	37	38	39	40
X1	1	5	10	9	6	4	4	4	3	2	1
X2	11	15	20	19	16	14	14	14	13	12	11
X5	41	45	50	49	46	44	44	44	43	42	41

Virtual Error= 0.107
Row variance= 7.884
Volume= 33

Figure 1: A five-time series with 40 electric energy consumption samples.

3.2. Bicluster quality measure

This work aims to identify a subset of sensors that present a similar behaviour during a time period. Particularly, we are interested in detecting unusual peaks of consumption [19]. From the biclustering perspective, the goal is to find large high quality biclusters that represent an interesting pattern found in the data. In this work, we use as a bicluster quality measure, the Transposed Virtual Error or simply Virtual Error (*VE*) from here [40]. *VE* computes the general tendency within the bicluster along the columns. In order to define *VE*, we first introduce the concept of virtual pattern. Given a bicluster b, the virtual pattern p is defined as the set $p = \{p_j\}_{j=1}^{|J|}$, so that p_j is given by the expression:

$$p_j = \frac{1}{|I|}\sum_{i=1}^{|I|} b_{ij}, \tag{1}$$

Where b_{ij} the elements of the bicluster b, p_j represents the average value of all sensors for a specific sample. For the bicluster b_1 given in Figure 1, the average value of the three sensors is provided in the last row p.

The *VE* of bicluster b is defined as:

$$VE(b) = \frac{1}{|I|\cdot|J|}\sum_{i=1}^{|I|}\sum_{j=1}^{|J|}\left|\hat{b}_{ij} - \hat{p}_j\right|, \tag{2}$$

where $\hat{b}_{ij}$ and $\hat{p}_j$ refer to standardized value of the elements of the bicluster b and virtual condition j as follows:

$$\hat{b}_{ij} = \frac{b_{ij} - \mu(EM_i)}{\sigma(EM_i)}, 1 \leq i \leq |I|, 1 \leq j \leq |J|, \quad (3)$$

$$\hat{p}_j = \frac{p_j - \mu(p)}{\sigma(p)}, 1 \leq j \leq |J| \quad (4)$$

With b_{ij} the elements of b, and $\mu(.)$ and $\sigma(.)$ refer to the mean and standard deviation, EM_irefers to all elements of sensor i that belong to b. The standardization of the values is introduced to be able to capture the consumption pattern of a sensor regardless of the real values registered.

The measure *VE* is a measure of coherent tendency. Lower values refer to a more robust pattern in the bicluster. A value of zero means that the bicluster contains perfectly coherent patterns. Nevertheless, the use of *VE* may yield to find biclusters characterized by flat patterns. Such flat biclusters are not interesting since we aim at finding abnormal activities, which are often related to peaks in consumption patterns. To overcome this issue, and favors fluctuating and coherent patterns, the row variance (*var(b)*) of a bicluster b can be used and defined as:

$$var(b) = \frac{\sum_{i \in I, j \in J}(b_{ij} - \mu(b_i))^2}{|I| \cdot |J|}, \quad (5)$$

Finally, to find large biclusters, we introduce, as another objective function, the volume of a bicluster *V(b)*, which measures the number of elements $b_{ij} \in b$. So, we aim at finding biclusters characterized by low values of virtual error *VE*, with high row variance *var(b)* and volume *V(b)*. As an example of these values, at the right side of Figure 1, we can see *VE(b)*, *var(b)*, and *V(b)* of biclusters.

3.3. Evolutionary Computation

An evolutionary algorithm (EA) [41–43] is based on the biological concepts of evolution, for example, the survival of the fittest. Such concepts are used to evolve candidate solutions toward much better solutions. In fact, an EA is a population-based stochastic iterative strategy, where an initial population of candidate solutions is evolved to improve the quality of the solutions. Usually, the initial population consists of random solutions. Genetic operators, such as selection, crossover, and mutation, are then used in order to simulate generations to obtain a new evolutionary population. These operators distinguish them from other nature inspirations like swarm artificial and physical algorithms [19]. Each evolutionary generation consists of two main processes: (a) create new candidate solutions and (b) competition for survival.

EA generates new individuals by selecting a subset of individuals from the population, performs crossover on the subset, and then injects new genetic material by mutation. Selected individuals, called parents, will then be used to generate new solutions called offspring. The descendants inherit attributes of the ancestor via the crossover and not present via mutation. This dual approach is the main strength of EA as a search algorithm, a concept called knowledge exploitation vs. exploration.

Another fundamental concept is that EA encodes solutions to be represented in individuals by chromosome. EA refers to a solution as to the phenotype and to an individual encoding it as a genotype. The literature reports many proposals to this aim, with binary string encoding being the most widely used for the easy implementation of binary crossover and mutation [44]. In this encoding, a binary string is used, where the meaning is assigned to each bit. In this way, a solution can be encoded, and an individual can be decoded.

As previously mentioned, in each generation, some individuals are selected to generate new solutions. The selection is usually based on the quality of the solutions encoded by individuals. This means that a quality measure should be assigned to each individual. This measure is called a fitness function. The selection mechanisms usually tend to assign more probabilities of being selected to the fittest individuals. Fitness is the point of connection between EAs and the optimization problem. The fitness function design depends on the objective function of the problem. In consequence, it performs at the phenotypic level.

Once individuals have been selected, crossover and/or mutation are used to produce offspring. A crossover is used to swap genetic material between two parents, while mutation is used to introduce small random changes that can help escape local optima. These operators are applied at the genotype level.

After that offspring are generated, they compete with older individuals for surviving to the next generation. A common strategy, called elitism, is also to let the fittest individual of a generation survive to the next generation.

EAs have shown good performance in exploring huge search spaces, which is the space of all possible solutions to a problem. This capacity is due to the intrinsic parallelism achieved by the population search and the stochastic nature of EAs, which allow them to efficiently search for a solution and with the capability of escaping local optima. Initialization, selection, crossover, and mutation are stochastic procedures. In this way, EAs represent a strong alternative to greedy heuristic and competitive metaheuristic [19].

EAs have been successfully used in various problems, such as planning [44], parameter settings [45], design [46,47], knowledge extraction [48], feature selection [49], planification [50,51], simulation and identification [52], control [53] and classification [44–57].

The problem of finding a set of biclusters with some desirable features on a given matrix can be addressed as a search problem. In this case, the solution space contains all possible biclusters that can be obtained from the matrix.

4. PARALLEL BICLUSTER SEARCH

In this section, we will describe our proposed parallel version of the algorithm Sequential Covering (SC) [13], called PSC (Parallel SC), aimed at reducing the computational time required by the algorithm. In order to implement our proposal, we decided to employ a master/slave model. First, we describe the original version of the SC algorithm, and then we will provide details on its parallelization.

4.1. Sequential Covering

In brief, SC implements a sequential covering strategy, which consists of calling several times an evolutionary biclustering algorithm (EBI) until a stopping criterion is reached. SC finds biclusters with maximum volume while minimizing the effect of overlapping among biclusters. The general scheme of SC is shown in Figure 2.

```
Procedure SC
begin
Initialize L to $\emptyset$;
$wp(eij) \leftarrow \emptyset$
while (~Stopping Criterion)
    $b \leftarrow$ EBI;
    if $b \neq \emptyset$ then
    Add $b$ to $L$;
    Update $wp$ $(eij)$;
    else $b = \emptyset$
    stop;
end while
end
```

(a) Pseudocode of the SC procedure

```
Procedure EBI
begin
Initialize population;
Evaluate population;
while (~Stopping Criterion)
    Select parents;
    Cross parents;
    Mutate offspring;
    Evaluate offspring;
    Update population;
end while
returnb
end
```

(b) Pseudocode of the EBI procedure

Figure 2: General scheme of the original EBI procedure.

Given a threshold δ, EBI returns either a bicluster or nothing depending on whether the virtual error of the bicluster is lower than δ or not. The biclusters found are stored in a list L and the stopping criterion is reached after running the procedure a maximum number of times. In order to avoid overlapping among biclusters as much as possible, each element of the input energy matrix e_{ij} is given a weight w_{ij} whose value depends on the number of biclusters to which e_{ij} belongs. In particular, the weights are defined as in equation 5.

$$w_p(e_{ij}) = \begin{cases} 0 & if\ |Cov(e_{ij})| = 0 \\ \dfrac{\sum_{n\in N, m\in M} e^{|Cov(e_{mn})|}}{e^{|Cov(e_{ij})|}} & if\ |Cov(e_{ij})| > 0 \end{cases} \qquad (6)$$

In the above equation, N is the number of rows, M the number of columns, $|Cov(e_{ij})|$the number of biclusters to which e_{ij} belongs to. As shown, higher values of the weight correspond to a larger number of biclusters containing an element.

Figure 2b outlines the pseudocode of the EBI. It initializes the initial population with biclusters containing a single element and then, such population evolves employing recombination of selected pairs of parents and crossover and mutation operators. Parents are selected using a tournament approach. The crossover and mutation operators have a probability associated of p_c = 0.85 and p_m = 0.2. EBI is an elitist algorithm where the best individual survives to the next generation. The evolution stops after a maximum number of generations (g_{max}). Finally, elitism is also applied with a probability of p_e = 0.9, the best individual replaces the worst in the new population. The best individual is returned if it is a δ-bicluster.

Individuals encode a single bicluster using an array of N + T bits. The first N bits are associated to each sensor while the other T bits to each electrical energy consumption sample. Figure 3 shows an example of individual encoding a bicluster for a matrix of dataset of N = 5 sensors and

T = 40 electrical energy consumption samples. The encoded bicluster contains |*I*| = 3 rows and |*J*| = 11continuous columns, i.e., *b* = [0110100000000000011111111111000000000000000000].

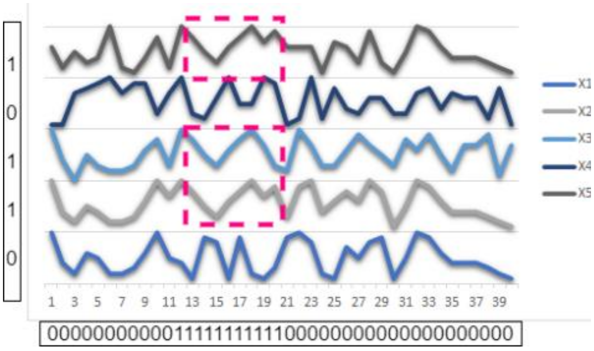


Figure 3: Example of an individual encoding a bicluster containing three rows and eleven continuous columns.

The fitness function of an individual *b*, in EBI, is defined as follows:

$$f(b) = \frac{VE(b)}{\delta} + \frac{1}{var(b)} + penalty + w\_d, \qquad (7)$$

where *VE(b)* is the virtual error of the bicluster *b* and *var (b)* is the variance associated with the sensors in the bicluster. *penalty* is computed as:

$$penalty = \sum_{i \in I, j \in J} w_p(e_{ij}) \qquad (8)$$

with *I* and *J* corresponding to the sensors and continuous days belonging to the bicluster *b*. *penalty* measures the sum of the weights of each element that belongs to *b* and its purpose is to avoid overlapping among biclusters. $w_p(e_{ij})$ is calculated using the expression (5). Finally, *w_d* is given by the expression:

$$w\_d = w_V \cdot \left(w_r \cdot \frac{\delta}{row_b} + w_c \cdot \frac{\delta}{col_b} \right) \qquad (9)$$

where w_V is a weight associated with the volume of the bicluster, row_b and col_b refer to the number of sensors and continuous days of a bicluster, respectively, and w_r and w_c correspond to the weights assigned to the number of sensors and continuous days, respectively. The search bias can vary by changing the values of the weights. The final goal of EBI is to minimize the fitness function. Following the recommendations of [38], the values were set to w_V = 1, w_r = 1 and w_c = 10.

4.2. Parallel Sequential Covering

The algorithms have problems finding global optimal when the search space is huge, as in the case of bicluster search. This is due to the unavoidable convergence in regions of good performance that not necessarily contain the global optima. To overcome this drawback, it is necessary to perform several independent runs. Each independent run has a different initial population and a different convergence region.

The simulation runs an EA sequentially or parallelly. The latter reduces the computational time by using the available computational resources.

In this context, parallelization of evolutionary algorithms emerges as a critical strategy in high-performance computing [57, 58]. Among the different parallelization strategies, we initially applied the master-slave parallelization. In this approach, a central computer, the master, distributes the tasks to the different worker computers, the slaves, to perform the tasks independently. The workers return the results to the master, who collects and processes the result. As mentioned above, Parallel Sequential Covering (PSC) is the proposed adaptation of SEBI to a master-slave architecture. Basically, PSC increases the explored search space size by running, in parallel, the SC procedure. The details about the adaptation are given in Figure 4. First, the master receives, as input, the data and runs the SC procedure on each slave. The input data contains energy matrix and evolutionary parameters. Since each SC returns a maximum number of *B* biclusters, the master will receive, at the end of the executions, a total of at most $SL \cdot B$ biclusters, with *SL* the number of slaves. Finally, the set of biclusters *L* will be composed of the union of each L_i returned.

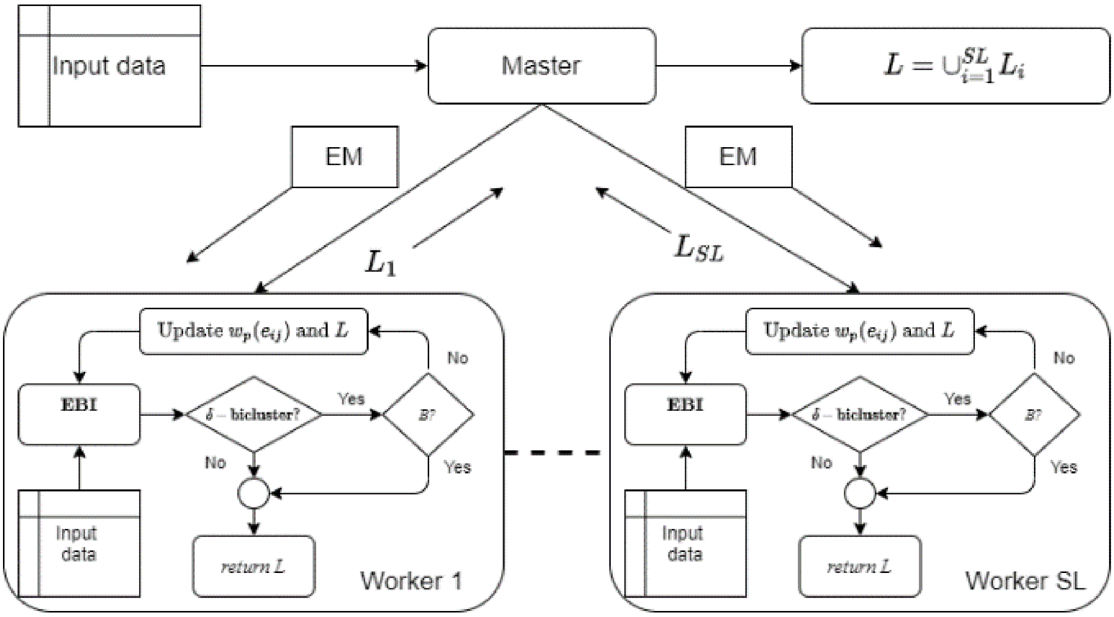


Figure 4: Workflow of the master-slave architecture of the PSC approach.

5. RESULTS AND DISCUSSION

In this section, we assess the quality of the PSC method for finding biclusters on the electric energy consumption data. This dataset contains 744 samples of electric energy consumption for 184 sensors gathered in December of 2013.

When the search space size is large, the choice of a correct set of parameter values is critical and might need many trials. Though what is essential is the description of a precise fitness function, as the parameter values will only help reach the goal defined by that function, i.e., the evolutionary algorithm might need much more time to converge. The results might not be good. Additionally, if we have limited time but enough computer resources, it is possible to perform parallel tests. We have run various preliminary resultsfor set the parameters of the algorithms. Table 1 shows the parameters setting. It is important to note that every time that PS calls EBI, it runs $PS{\times}g_{max}$ evaluations of potential biclusters, and the size of the search space is about 2^{n+m}, with n samples and m sensors. Thus, the number of candidate biclusters evaluated by EBI could be too small compared to the number of potential biclusters when n and m are large. Therefore, the role of genetic operators and the fitness function is a key factor in correctly guiding the search toward reasonable solutions.

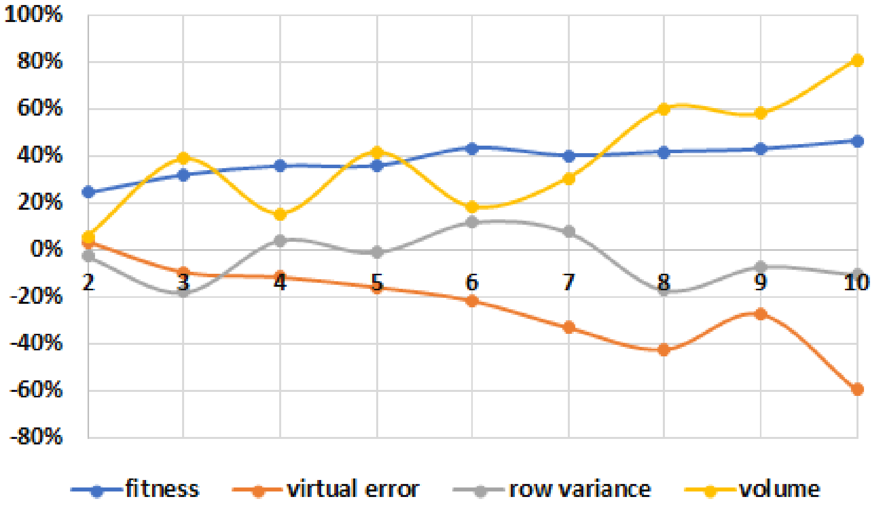
Table 1. Parameter settings of PSC.

Parameter	*Symbol*	*Value*
Population Size	PS	200
Number of generations	g_{max}	30
Crossover probability	p_c	0.85
Mutation probability	p_m	0.20
Elitism Probability	p_e	0.9
Weight for volume	w_V	1
Weight for sensors	w_r	1
Weight for samples day	w_c	10
Tournament Size	ts	4
Biclusters to find	B	10
Delta Biclusters	δ	10
Number of workers	SL	2 to 10
Number of independent runs	NR	30

The executions have performed a computer with Intel(R) Core (TM) i5-2520M CPU @ 2.50GHz, 4Gb RAM as the master node, and a computer with AMD A8-7410 APU processor with 8Gb RAM as worker nodes, both with Linux operating systems. The simulation was performed on a local area network. The automatization of PSC was via Apache Hadoop and Apache Spark version spark-2.3.0-bin-hadoop2.7. One core per computer worker was configured to perform the SC algorithm. All algorithms were implemented in Java language using the javac 1.8.0_201 version. Each parallel run consumed an average of 30 minutes, totaling a computation time of 150 hours (NR x 30 min x *SL*), including the SC run.

Algorithms	***Workers***	***Fitness***	***VE***	***var***	***V***
SC	1	1.86E-03	1.40E-01	1.15E+03	4.45E+02
PSC	2	1.40E-03	1.35E-01	1.12E+03	4.71E+02
	3	1.26E-03	1.53E-01	9.43E+02	6.18E+02
	4	1.19E-03	1.56E-01	1.20E+03	5.14E+02
	5	1.19E-03	1.62E-01	1.14E+03	6.31E+02
	6	1.05E-03	1.70E-01	1.29E+03	5.27E+02
	7	1.11E-03	1.86E-01	1.24E+03	5.82E+02
	8	1.08E-03	1.99E-01	9.54E+02	7.14E+02
	9	1.06E-03	1.78E-01	1.07E+03	7.04E+02
	10	9.94E-04	2.23E-01	1.03E+03	8.06E+02

(a) Table of PSC and SC experimental results



(b) Relative performance of PSC over SC

Figure 5. Experimental result of PSC and SC

Table 2 shows the average values of the different metrics. The fitness score is highly correlated with virtual error (-0.71) and volume (-0.69), while to a lesser degree with a row variation (0.06). In general, when the bicluster fitness decreases, the volume and row variance tend to increase. On the other hand, the row variance does not show a definite pattern about fitness. With these results, we can note that the algorithm finds biclusters with high-quality in volume at the cost of slightly increasing some virtual error.Figure 5 presents a graphical representation of these explanations. This figure shows the relative and normalized values of the fitness, virtual error, row variance, and volume metrics. The relative fitness with k workers is calculated as $f_k = {(f_1 - f_k)}/{f_1}$, likewise the other metrics.

The virtual error measure has a low correlation with row variation (-0.25) and a high correlation with volume (0.86). The above implies that virtual error and volume are in a trade-off relationship. With the worsening of the virtual error, the volume improves and vice versa. This result depends on the input parameters that define the algorithm's course towards some sub-region of the efficient frontier. The row variation and volume have a medium correlation of -0.59. The row variation influences the volume moderately.

Another critical aspect observed is the impact of parallelization. As the number of worker nodes increases, then the fitness score also improves. This improvement is a consequence of further exploration of the search space with the same computational time.

Figure 6 presents two examples of bicluster calculated using the PSC algorithm. In these biclusters, we can see patterns of electric energy consumption. The sensors 58, 73, and 112 and samples block 656 to 685 constitute Bicluster A, while sensors 13, 47, and 52 and samples block 557 to 585 for Bicluster B. In the figure also we can see the virtual error, row variance, and volume of biclusters. The patterns show abrupt changes of sensor groups at different times of the month. Such descriptions of electricity consumption are crucial to energy policy or decisionmaking. In this context, the developed tool is promising as a pattern description system for the electricity sector.

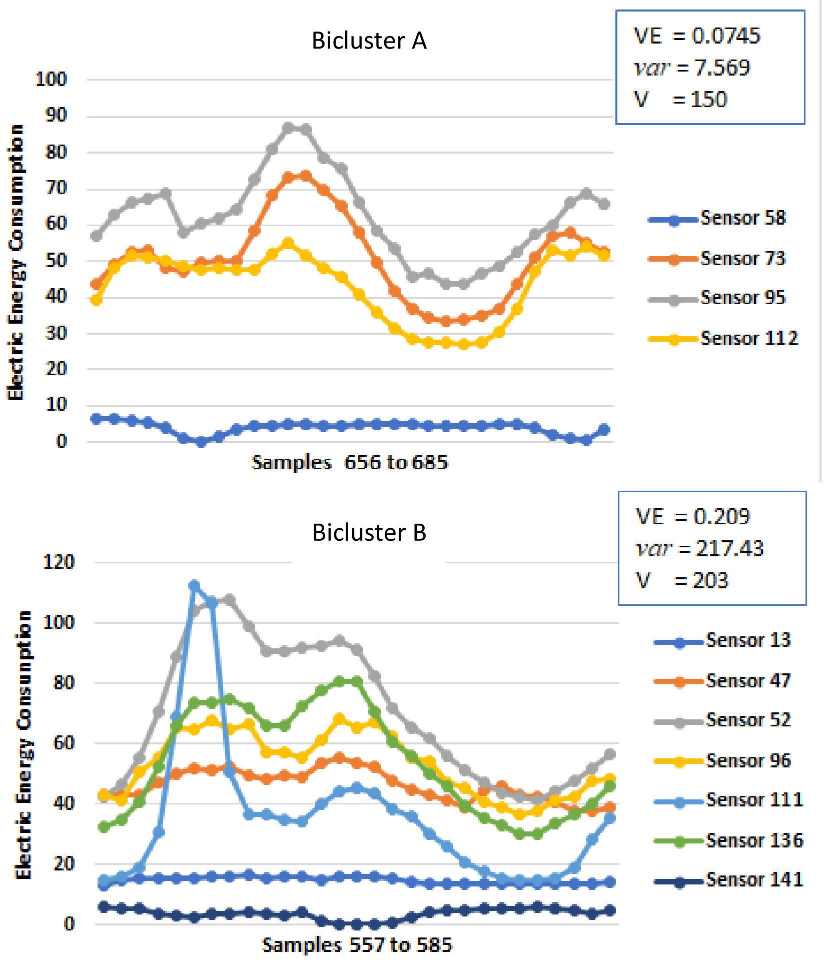


Figure 6: Examples of two biclusters calculated by PSC algorithm.

6. Conclusions and Future Work

In this paper, we have proposed, implemented, and validated a parallel alternative to an evolutionary biclustering algorithm. The proposed parallel architecture is based on the master-worker scheme to discover biclusters called Parallel Sequential Covering (PSC). The PSC was tested to locate and describe hidden patterns in a set of time series of an electrical system. Experimental results indicate that the PSC approach is promising for the study scenario, outperforming the state-of-the-art non-parallel evolutionary algorithm. A key aspect is that the quality of the solutions improves with the increasing number of workers. The algorithm converges to bicluster regions with high volume and vector variation as well as low virtual errors. These characteristics are the ones desired in the problem in question. Experiments also indicate that volume and virtual error are objective functions that conflict with each other. Therefore, we consider approaching a Parallel Pareto multi-objective optimization scheme as the next step in this research line.

Acknowledgments

This work was supported by the CONACYT, Paraguay, under Grant PINV18-661.

REFERENCES

[1] Li, X.; Bowers, C.P.; Schnier, T. Classification of energy consumption in buildings with outlier detection. IEEE Transactions on Industrial Electronics 2009, 57, 3639–3644.

[2] Wei, Y.; Zhang, X.; Shi, Y.; Xia, L.; Pan, S.; Wu, J.; Han, M.; Zhao, X. A review of data-driven approaches for prediction and classification of building energy consumption. Renewable and Sustainable Energy Reviews 2018, 82, 1027–1047.

[3] Hernandez, L.; Baladron, C.; Aguiar, J.M.; Carro, B.; Sanchez-Esguevillas, A.J.; Lloret, J.; Massana, J. A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings. IEEE Communications Surveys & Tutorials 2014, 16, 1460–1495.

[4] Moon, J.; Jung, S.; Rew, J.; Rho, S.; Hwang, E. Combination of short-term load forecasting models based on a stacking ensemble approach. Energy and Buildings 2020, 216, 109921.

[5] Somu, N.; MR, G.R.; Ramamritham, K. A deep learning framework for building energy consumption forecast. Renewable and Sustainable Energy Reviews 2021, 137, 110591.

[6] Jain, R.K.; Smith, K.M.; Culligan, P.J.; Taylor, J.E. Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy. Applied Energy 2014, 123, 168–178.

[7] Divina, F.; Gilson, A.; Goméz-Vela, F.; García Torres, M.; Torres, J.F. Stacking ensemble learning for short-term electricity consumption forecasting. Energies 2018, 11, 949.

[8] Divina, F.; Garcia Torres, M.; Goméz Vela, F.A.; Vazquez Noguera, J.L. A comparative study of time series forecasting methods for short term electric energy consumption prediction in smart buildings. Energies 2019, 12, 1934.

[9] Capozzoli, A.; Lauro, F.; Khan, I. Fault detection analysis using data mining techniques for a cluster of smart office buildings. Expert Systems with Applications 2015, 42, 4324–4338.

[10] Fan, C.; Xiao, F.; Wang, S. Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques. Applied Energy 2014, 127, 1–10.

[11] Zhan, S.; Liu, Z.; Chong, A.; Yan, D. Building categorization revisited: A clustering-based approach to using smart meter data for building energy benchmarking. Applied Energy 2020, 269, 114920.

[12] Rajabi, A.; Eskandari, M.; Ghadi, M.J.; Li, L.; Zhang, J.; Siano, P. A comparative study of clustering techniques for electrical load pattern segmentation. Renewable and Sustainable Energy Reviews 2020, 120, 109628.

[13] Anitha, S.; Chandran, C. Review on Analysis of Gene Expression Data Using Biclustering Approaches. Bonfring Int. J. Data Min. 2016, 6, 16.

[14] Nepomuceno, J.A.; Troncoso, A.; Aguilar-Ruiz, J.S. Scatter search-based identification of local patterns with positive and negative correlations in gene expression data. Appl. Soft Comput. 2015, 35, 637–651.

[15] Divina, F.; Pontes, B.; Giráldez, R.; Aguilar-Ruiz, J.S. An effective measure for assessing the quality of biclusters. Comput. Biol. Med. 2012, 42, 245–256

[16] Gnatyshak, D.V.; Ignatov, D.I.; Semenov, A.; Poelmans, J. Analysing online social network data with biclustering and triclustering. Concept Discovery in Unstructured Data. In Proceedings of the 2nd International Workshop, CDUD 2012, Leuven, Belgium, 6–10 May 2012; pp. 30–39

[17] Li, F.; Li, M.; Guan, P.; Ma, S.; Cui, L. Mapping publication trends and identifying hot spots of research on internet health information seeking behavior: A quantitative and co-word biclustering analysis. J. Med. Internet Res. 2015, 17.

[18] Sun, Z.; Han, L.; Huang, W.; Wang, X.; Zeng, X.; Wang, M.; Yan, H. Recommender systems based on social networks. J. Syst. Softw. 2015, 99, 109–119.

[19] Divina, F., Goméz Vela, F.A. and García Torres, M., 2019. Biclustering of smart building electric energy consumption data. Applied Sciences, 9(2), p.222.

[20] Liao, T.W. Clustering of time series data—a survey. Pattern recognition 2005, 38, 1857–1874.

[21] Rani, S.; Sikka, G. Recent techniques of clustering of time series data: a survey. International Journal of Computer Applications 2012, 52.

[22] Divina, F.; Goméz Vela, F.A.; García Torres, M. Biclustering of Smart Building Electric Energy Consumption Data. Applied Sciences 2019, 9. doi:10.3390/app9020222.

[23] Ruiz, L.; Pegalajar, M.; Arcucci, R.; Molina-Solana, M. A time-series clustering methodology for knowledge extraction in energy consumption data. Expert Systems with Applications 2020, 160, 113731.

[24] Lee, E.; Kim, J.; Jang, D. Load profile segmentation for effective residential demand response program: Method and evidence from Korean pilot study. Energies 2020, 13, 1348.

[25] Busygin, S.; Prokopyev, O.; Pardalos, P.M. Biclustering in data mining. Computers & Operations Research 2008, 35, 2964–2987.

[26] Gendreau, M., & Potvin, J. Y. (Eds.). (2010). Handbook of metaheuristics (Vol. 2, p. 9). New York: Springer.

[27] Campillo, J., Wallin, F., Torstensson, D., and Vassileva,I. (2012). Energy demand model design for forecasting electricity consumption and simulating demand responsescenarios insweden. In4th International Conference in Applied Energy 2012, July 5-8, 2012. Suzhou, China.

[28] Medina, A., Cámara, A., and Monrobel, J.-R. (2016). Measuring the socioeconomic and environmental effects of energy efficiency in-vestments for a more sustainable spanish economy. Sustainability, 8(10):1039.

[29] Abdel-Aal, R. E. and aZ, A.-G. (1997). Forecasting monthly electric energy consumption in eastern saudiarabia using univariatetime-series analysis. Energy, 22:1059–1069

[30] Newsham, G. R., &Birt, B. J. (2010, November). Building-level occupancy data to improve ARIMA-based electricity use forecasts. In Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building (pp. 13-18).

[31] Jain, R. K., Smith, K. M., Culligan, P. J., & Taylor, J. E. (2014). Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy. Applied Energy, 123, 168-178.

[32] Liu, Y., Chen, H., Zhang, L., Wu, X., & Wang, X. J. (2020). Energy consumption prediction and diagnosis of public buildings based on support vector machine learning: A case study in China. Journal of Cleaner Production, 272, 122542.

[33] Yang, S., Wan, M. P., Chen, W., Ng, B. F., & Dubey, S. (2020). Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization. Applied Energy, 271, 115147.

[34] Zheng, H., Yuan, J., & Chen, L. (2017). Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. Energies, 10(8), 1168.

[35] Chitsaz, H., Shaker, H., Zareipour, H., Wood, D., &Amjady, N. (2015). Short-term electricity load forecasting of buildings in microgrids. Energy and Buildings, 99, 50-60.

[36] Kelo, S., &Dudul, S. (2012). A wavelet Elman neural network for short-term electrical load prediction under the influence of temperature. International Journal of Electrical Power & Energy Systems, 43(1), 1063-1071.

[37] Nizami, S. J., & Al-Garni, A. Z. (1995). Forecasting electric energy consumption using neural networks. Energy policy, 23(12), 1097-1104.

[38] Diao, L., Sun, Y., Chen, Z., & Chen, J. (2017). Modeling energy consumption in residential buildings: A bottom-up analysis based on occupant behavior pattern clustering and stochastic simulation. Energy and Buildings, 147, 47-66.

[39] Pérez-Chacón, R., Luna-Romera, J. M., Troncoso, A., Martínez-Álvarez, F., & Riquelme, J. C. (2018). Big data analytics for discovering electricity consumption patterns in smart cities. Energies, 11(3), 683.

[40] Pontes, B.; Giráldez, R.; Aguilar-Ruiz, J.S. Quality Measures for Gene Expression Biclusters. PLoS ONE 2015, p. e0115497.

[41] Eiben, A.E.; Smith, J.E. Introduction to Evolutionary Computing; Springer-Verlag, 2003.

[42] Bäck, T.; Fogel, D.B.; Michalewicz, Z. Evolutionary Computation 1: Basic Algorithms and Operators; Institute of Physics Publishing, 2000.

[43] Yao, X., Evolutionary computation: A gentle introduction. In Evolutionary Optimization; Kluwer Academic Publishers, 2002; chapter 2, pp. 27–53.

[44] Goldberg, D.E.; Robert, L. Alleles, loci and the travelling salesman problem. Proceedings of 1st Int. Conf. on Genetic Algorithms; Grefenstette, J.e., Ed. Lawrence Erlbaum Associates, Hillsdale, 1985, pp. 154–159.

[45] Rocha, M.; Cortez, P.; Neves, J. Evolution of neural networks for classification and regression. Neurocomputing 2007, 70, 2809–2816. Neural Network Applications in Electrical Engineering Selected papers from the 3rd International Work-Conference on Artificial Neural Networks (IWANN

2005), https://doi.org/10.1016/j.neucom.2006.05.023.

[46] Bentley, P.J.; Corne, D.W. Creative evolutionary systems; Morgan Kaufmann Publishers Inc., 2001.

[47] Zheng, F.; Zecchin, A.C.; Newman, J.P.; Maier, H.R.; Dandy, G.C. An Adaptive Convergence-Trajectory Controlled Ant Colony Optimization Algorithm with Application to Water Distribution System Design Problems. IEEE Transactions on Evolutionary Computation 2017, 21, 773–791. doi:10.1109/TEVC.2017.2682899.

[48] Cheng, S.; Ma, L.; Lu, H.; Lei, X.; Shi, Y. Evolutionary computation for solving search-based data analytics problems. Artif. Intell. Rev. 2021, 54, 1321–1348. doi:10.1007/s10462-020-09882-x.

[49] Nakisa, B.; Rastgoo, M.N.; Tjondronegoro, D.; Chandran, V. Evolutionary computation algorithms for feature selection of EEG-based emotion recognition using mobile sensors. Expert Systems with Applications 2018, 93, 143–155. https://doi.org/10.1016/j.eswa.2017.09.062.

[50] Yamada, T.; Nakano, R. A Genetic Algorithm Applicable to Large-Scale Job-Shop Problems. Parallel Problem Solving from Nature, 2; R. Männer.; Manderick, B., Eds.; Elsevier Science Publishers, B. V.: Amsterdam, 1992.

[51] Corne, D.; Ross, P.; Fang, H.L. Fast Practical Evolutionary Timetabling. Lecture Notes in Computer Science, vol 865 (Evolutionary Computing AISB Workshop, Leeds, UK, April 1994). Springer-Verlag, 1994, pp. 251–263.

[52] Gehlhaar, D.K.; Verkhivker, G.M.; Rejto, P.A.; Sherman, C.J.; Fogel, D.B.; Fogel, L.J.; Freer, S.T. Molecular recognition of the inhibitor AG-1343 by HIV-1 protease: conformationally flexible docking by evolutionary programming. Chemistry and Biology 1995, 2, 317–324.

[53] Spencer, G.F. Automatic Generation of Programs for Crawling and Walking. Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93; Forrest, S., Ed.; Morgan Kaufmann: University of Illinois at Urbana-Champaign, 1993; p. 654.

[54] Fogel, D.B. Evolving Behavious in the Iterated Prisoner's Dilemma. Evolutionary Computation 1993, 1, 77–97.

[55] Divina, F.; Marchiori, E. Evolutionary Concept Learning. GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference; Morgan Kaufmann Publishers: New York, 2002; pp. 343–350.reless networks. In Proceedings of the IEEE 1st International Conference on Broadnets Networks (BroadNets'04). IEEE, Los Alamitos, CA, 210–217. https://doi.org/10.1109/BROADNETS.2004.8

[56] Divina, F., and Aguilar-Ruiz, J.S., 2007, July. A multi-objective approach to discover biclusters in microarray data. In Proceedings of the 9th annual conference on Genetic and evolutionary computation (pp. 385-392).

[57] Erick Cantú-Paz, David E. Goldberg, Efficient parallel genetic algorithms: theory and practice, Computer Methods in Applied Mechanics and Engineering, Volume 186, Issues 2–4, 2000, Pages 221-238, ISSN 0045-7825, https://doi.org/10.1016/S0045-7825(99)00385-0.

[58] Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). Evolutionary algorithms for solving multi-objective problems (Vol. 5, pp. 79-104). New York: Springer.