“Knowledge management overview of feature selection problem in high-dimensional financial data: cooperative co-evolution and MapReduce perspectives”

AUTHORS	A N M Bazlur Rashid https://orcid.org/0000-0002-8672-5023 Tonmoy Choudhury https://orcid.org/0000-0002-7745-0048

ARTICLE INFO	A N M Bazlur Rashid and Tonmoy Choudhury (2019). Knowledge management overview of feature selection problem in high-dimensional financial data: cooperative co-evolution and MapReduce perspectives. *Problems and Perspectives in Management*, *17*(4), 340-359. doi:10.21511/ppm.17(4).2019.28
DOI	http://dx.doi.org/10.21511/ppm.17(4).2019.28
RELEASED ON	Thursday, 26 December 2019
RECEIVED ON	Friday, 30 August 2019
ACCEPTED ON	Wednesday, 20 November 2019

JOURNAL	"Problems and Perspectives in Management"
ISSN PRINT	1727-7051
ISSN ONLINE	1810-5467
PUBLISHER	LLC “Consulting Publishing Company “Business Perspectives”
FOUNDER	LLC “Consulting Publishing Company “Business Perspectives”

NUMBER OF REFERENCES	NUMBER OF FIGURES	NUMBER OF TABLES
143	7	2

BUSINESS PERSPECTIVES

LLC "CPC "Business Perspectives"
Hryhorii Skovoroda lane, 10,
Sumy, 40022, Ukraine

www.businessperspectives.org

A N M Bazlur Rashid (Australia), **Tonmoy Choudhury** (Australia)

KNOWLEDGE MANAGEMENT OVERVIEW OF FEATURE SELECTION PROBLEM IN HIGH-DIMENSIONAL FINANCIAL DATA: COOPERATIVE CO-EVOLUTION AND MAPREDUCE PERSPECTIVES

Abstract

The term "big data" characterizes the massive amounts of data generation by the advanced technologies in different domains using 4Vs – volume, velocity, variety, and veracity - to indicate the amount of data that can only be processed via computationally intensive analysis, the speed of their creation, the different types of data, and their accuracy. High-dimensional financial data, such as time-series and space-time data, contain a large number of features (variables) while having a small number of samples, which are used to measure various real-time business situations for financial organizations. Such datasets are normally noisy, and complex correlations may exist between their features, and many domains, including financial, lack the al analytic tools to mine the data for knowledge discovery because of the high-dimensionality. Feature selection is an optimization problem to find a minimal subset of relevant features that maximizes the classification accuracy and reduces the computations. Traditional statistical-based feature selection approaches are not adequate to deal with the curse of dimensionality associated with big data. Cooperative co-evolution, a meta-heuristic algorithm and a divide-and-conquer approach, decomposes high-dimensional problems into smaller sub-problems. Further, MapReduce, a programming model, offers a ready-to-use distributed, scalable, and fault-tolerant infrastructure for parallelizing the developed algorithm. This article presents a knowledge management overview of evolutionary feature selection approaches, state-of-the-art cooperative co-evolution and MapReduce-based feature selection techniques, and future research directions.

Keywords big data, optimization, computational techniques, meta-heuristics, problem decomposition, parallel programming, knowledge discovery

JEL Classification M11, M15, C61, C63

Received on: 30th of August, 2019
Accepted on: 20th of November, 2019

A N M Bazlur Rashid, Ph.D. Student, School of Science, Edith Cowan University, Australia.

Tonmoy Choudhury, Ph.D., Lecturer, School of Business and Law, Edith Cowan University, Australia.

INTRODUCTION

Modern technologies produce tons of new data about individuals, industries, finance, economics, health sciences, and so on; the volume of new data nearly doubles every two years (IBM, 2018). IBM has reported that 90% of the world's data was created in the previous two years, with more than 2.5 exabytes of data produced daily. Financial time-series and space-time are examples of high-dimensional data used to mine and measure the real-time business conditions for financial organizations or for data mining (Gao & Tsay, 2019; Wu, Liu, & Yang, 2018) in supply chain (Habib & Hasan, 2019; Tseng, Wu, Lim, & Wong, 2019; Voyer, Dean, Pickles, & Robar, 2018). In health science (Tursunbayeva, Bunduchi, Franco, & Pagliari, 2016), high-throughput technologies, such as microarrays, generate DNA microarray datasets

http://dx.doi.org/10.21511/ppm.17(4).2019.28

having more than 500,000 genes in gene arrays or mass spectrometry creates high-dimensional datasets regarding living cells having a range of 300,000 m/z values (Aliferis, Statnikov, & Tsamardinos, 2006). These high-throughput data are known as "big data" and can be defined in terms of 4Vs: volume (size of the data), velocity (speed of data generation), variety (diverse types of data – structured, semi-structured, or unstructured), and veracity (uncertain or imprecise data) (Laney, 2001; Zhou, Chawla, Jin, & Williams, 2014).

The availability of large-scale data provides new opportunities for the research community to find new insights. Knowledge management (Ali, Rattanawiboonsom, Hassan, & Nedelea, 2019; Bakanauskienė, Bendaravičienė, & Juodelytė, 2018; Chalikias, Kyriakopoulos, Skordoulis, & Koniordos, 2014; Grytten & Minde, 2019; Gupta, 2016; Illiashenko et al., 2018; Yee, Tan, & Ramayah, 2017) or knowledge discovery (Ketcha, Johannesson, & Bocij, 2015) from these large-scale data is a challenging task because the massive volume and high-dimensionality lead to computational difficulties (Bolon-Canedo et al., 2018). High-dimensional data suffer from both the curse of dimensionality (an enormous number of features (also called "variables" or "attributes") in the dataset (Clarke et al., 2008)) and the curse of dataset sparsity (tiny samples in the dataset (Somorjai, Dolenko, & Baumgartner, 2003)). For example, a microarray dataset consists of 3,816 features for each sample, with a sample size of only 158 (Stoeckel & Fung, 2005). Identification of biomarkers from high-dimensional biological datasets can assist in improving the diagnostic process and treatment of diseases. Similary, an organization can decide to purchase the options on the future exchange rates to reduce the effect of currency exchange fluctuations rates on corporate finance (Fan & Li, 2006). However, it requires a systematic search technique for finding the relevant biomarkers or deciding to purchase the options from a large set of features. Due to these challenges, existing high-dimensional data analysis techniques experience the problems like overfitting, erroneous classification, and high computational cost. Hence, most of the available techniques, including conventional statistical methods and machine learning strategies are not suitable for these type of datasets (Yamada et al., 2018). Therefore, advanced knowledge and information processing systems are required to overcome these challenges (Deepak, Mahesh, & Medi, 2019).

Dimensionality reduction is one way to deal with the curse of dimensionality by representing the data using a reduced set of features. Dimensionality reduction is of two types: feature extraction and feature selection (Xue, Zhang, Browne, & Yao, 2016). Feature extraction normally creates new features from the original feature set, while feature selection (FS) finds a subset of the original features. G. Kim, Y. Kim, Lim, and H. Kim (2010) define the FS problem as finding a set of minimum number of relevant features that describes the dataset. In high-dimensional datasets, features have complex interactions between them, extracting features is generally not suitable. FS is the alternative approach for these datasets. One objective of the FS process is to improve the classification's (Mura, Daňová, Vavrek, & Dubravska, 2017) accuracy with respect to the sensitivity (possibility of the prediction to be positive) and specificity (possibility of the prediction to be negative) (Dash & Liu, 1997, 2003).

Several methods are available in the literature based on different metrics, such as entropy, probability distribution, information theory, or the accuracy of a predictive model. However, users of these techniques need to understand their technical details to apply them correctly (Liu & Yu, 2005). Approaches to FS are two-fold: individual evaluation (individual features (Bakanauskienė, Bendaravičienė, & Barkauskė, 2017) are ranked based on their relevancy) and subset evaluation (depends on a particular search technique to produce a subset of features). FS methods are also classified into three categories: filter methods, wrapper methods, and embedded methods (Xue et al., 2016).

The cooperative co-evolutionary algorithm (CCEA), a meta-heuristic algorithm, handles the multiple populations, evaluates the fitness function in terms of the subjective fitness landscape, collaborates the individuals from different populations, and divides a large problem into smaller sub-problems to evolve and execute independently (Derrac, Garcia, & Herrera, 2010; Potter & de Jong, 2000). Further,

the MapReduce programming model (a open-source platform) is a parallel programming model that communicates with Hadoop Distributed File System (HDFS) and executes the computations. It was originally introduced by Google research for building the search indices, distributed computing, and large-scale data (Dean & Ghemawat, 2008, 2010). MapReduce can to handle the large-scale data in a distributed environment using *map* and *reduce* features with available resources in parallel. Moreover, MapReduce provides fault tolerance, data locality, scalability, ease of programming, and flexibility (Hashem, Anuar, Gani, Yaqoob, Xia, & Khan, 2016).

A survey on evolutionary computation (EC) approaches for FS indicates that genetic algorithm (GA) and genetic programming (GP) are the most commonly used EC techniques applied to FS problems (Xue et al., 2016). Similarly, Bhattacharya, Islam, and Abawajy (2016), Stanovov, Brester, Kolehmainen, and Semenkina (2017) have argued for the need to use EC in big data. Further, a survey on CCEAs includes the prospects of CCEA in big data optimization (Ma, Li, Zhang, Tang, Liang, Xie, & Zhu, 2018). From the existing literature, studies involving the combination of CCEA (Khan & Kakabadse, 2014) and the MapReduce model are an emerging area of research, and the existing works are limited (Ding, Lin, Chen, Zhang, & Hu, 2018; Ding, Jie. Wang, & Jia. Wang, 2016). This paper presents a knowledge management overview of evolutionary FS approaches and FS approaches based on CCEA and the MapReduce model with future research directions for FS problems.

The rest of the paper is organized as follows. Section 1 describes FS fundamentals and classification of evolutionary FS approaches. Section 2 includes CCEA. Section 3 illustrates the MapReduce technique. Section 4 discusses the state-of-the-art FS approaches based on different techniques. Finally, a summary of the paper is presented in the conclusion section.

1. LITERATURE REVIEW

1.1. Fundamentals of feature selection

Many real-world problems consist of a large number of features. However, some of these features may be irrelevant or redundant and may degrade the performance of data mining and machine learning algorithms. FS is an approach to choose the relevant features and reduce the dimensionality of the data for improving the learning process and algorithmic performance. FS techniques have been used to identify the biomarkers (i.e., important genes) from high-dimensional biological datasets (Ahmed, Zhang, & Peng, 2013), searching for words or phrases in text mining (Aghdam, Ghasem-Aghaee, & Basiri, 2009), or selecting the important visual subjects (e.g., color, shape, pixel, texture, etc.) in image analysis (A. Ghosh, Datta, & S. Ghosh, 2013). Figure 1 shows a general FS process consisting of four main steps (Dash & Liu, 1997).

The first step of a FS process is using a search technique (e.g., GA, greedy search, or best first search) to find the subsets of features. Next, various subset evaluation measures, such as distance measures, dependency measures, or classification accuracy are applied to evaluate the goodness of the subsets of features. A stopping criterion (e.g., number of generations) is used to terminate the FS process. Lastly, a validation (Grandon, Ramirez-Correa, & Luna, 2019) procedure is be used to test the validity of the selected subset.

FS is challenging in terms of computation owing to the increased number of features, advanced techniques of data collection, and complexities of problems. Given a dataset consists of k features, there can be 2^k possible solutions, which ultimately makes the FS a difficult and computationally intensive task (Guyon & Elisseeff, 2003). With a large enough k, an exhaustive search for FS becomes infeasible from such a dataset. Several search techniques, for example, greedy search, complete search, random search, and heuristic search can be applied to FS procedures (Liu, Tang, & Zeng, 2015). However, many of the FS approaches are limited by high computational cost or stagnation in local optima (Liu, Wang, Chen, Dong, Zhu, & Wang, 2011). FS is also difficult because of complex feature interactions, which can exist among

http://dx.doi.org/10.21511/ppm.17(4).2019.28

Source: Developed by the authors based on Xue, Zhang, Browne, and Yao (2016).

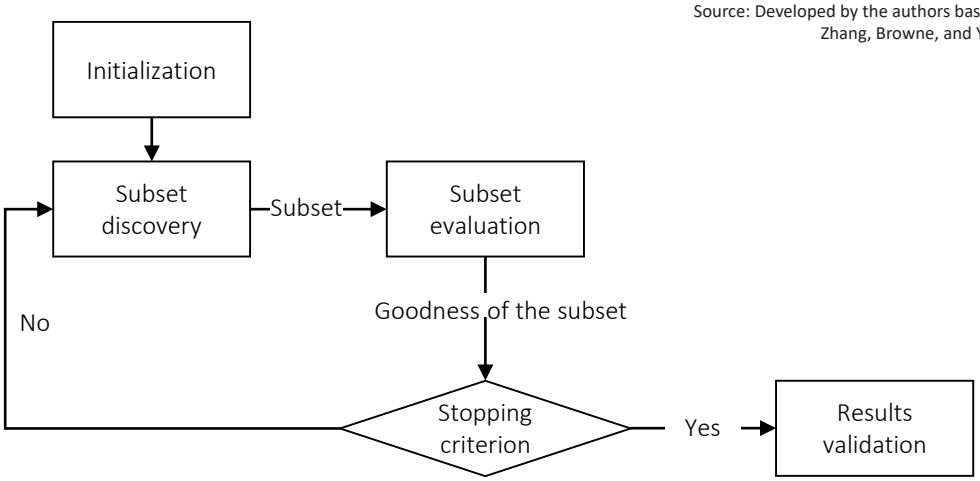


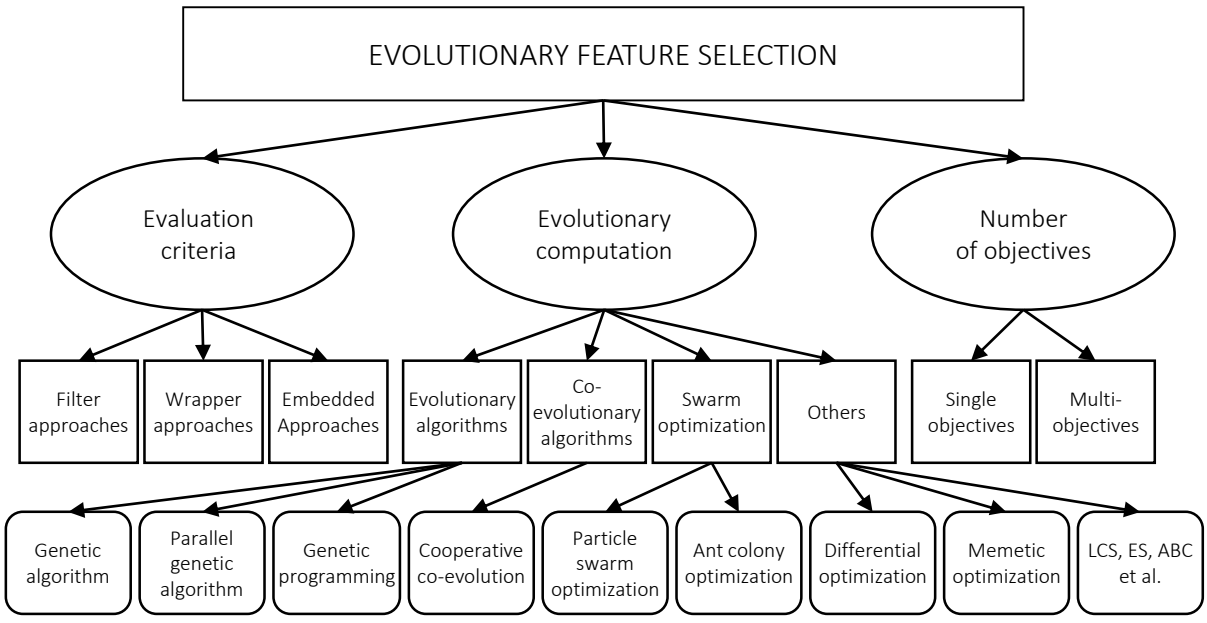
Figure 1. General feature selection process

features in a variety of ways. A weak feature in terms of its target can become redundant when used independently, while the exact same feature may improve the classification performance when used together with a few complementary features. A balanced selection or removal of this kind of features is an important task. Hence, FS techniques evaluating the subsets of features together rather than evaluating the features independently are more suited for feature interactions. FS aims to maximize the classification accuracy while minimizing the number of selected features. Factors, such as evaluation criteria and search techniques, are important in FS for exploring the search space efficiently and for evaluating the quality of the selected features (Xue et al., 2016).

1.2. Classification of evolutionary feature selection methods

From literature, several FS approaches incorporate the different techniques, such as fuzzy set

Source: Developed by the authors based on Xue, Zhang, Browne, and Yao (2016).



Note: LCS – learning classifier system, ES – evolutionary strategies, ABC – artificial bee colony

Figure 2. Overall categories of evolutionary computation for feature selection

theory, rough set theory, neural networks, and metaheuristics, resulting in many different ways to classify the FS methods. Figure 2 presents an overall classification of evolutionary FS methods based on three criteria: evaluation criteria, search techniques, and objectives.

Based on the evaluation criteria, there are three types of FS methods: filter methods, wrapper methods, and embedded methods. Filter methods are independent of a classifier or learning algorithm. Initially, each feature is scored based on some measures and then features are ranked using such techniques as *T*-test or *P*-test. Finally, based on a threshold value, a subset of features from the top-ranked features is selected (Levner, 2005). Unlike filter methods, wrapper methods involve a specific classification algorithm for evaluating the goodness of the selected subset of features. The classification algorithm is considered as a "black box" in wrapper methods (Xue et al., 2016). The difference between wrapper methods and filter methods lies in using a classification algorithm. Since wrapper methods evaluate each subset of features in terms of classification performance, this often results in a better performance. However, wrapper methods are computationally more expensive than filter methods (Dash & Liu, 1997). The third FS method is the embedded method that combines the filter and wrapper methods, i.e., FS and classification model formation are performed in a single process (Boroujeni, Stantic, & Wang, 2017). EC techniques, such as GP and learning classifier systems (LCSs), can carry out the embedded approaches of FS (Lin, Ke, Chien, & Yang, 2008).

2. COOPERATIVE CO-EVOLUTIONARY ALGORITHMS

The cooperative co-evolutionary approach was originally introduced by Potter and de Jong (1994) to solve the large-scale complex optimization problems (Rentsen, Zhou, & Teo, 2016) through a divide-and-conquer strategy and by evolving the interacting co-adapted sub-problems. The cooperative co-evolution achieves the promising performance in optimizing many real-world problems, such as function optimization (Potter & de Jong, 1994), designing artificial neural networks (Potter & de Jong, 1995), occurrence of *Red Queen* dynamics (Pagie & Hogeweg, 2000), and machine

Source: Developed by the authors based on Shi and Gao (2017).

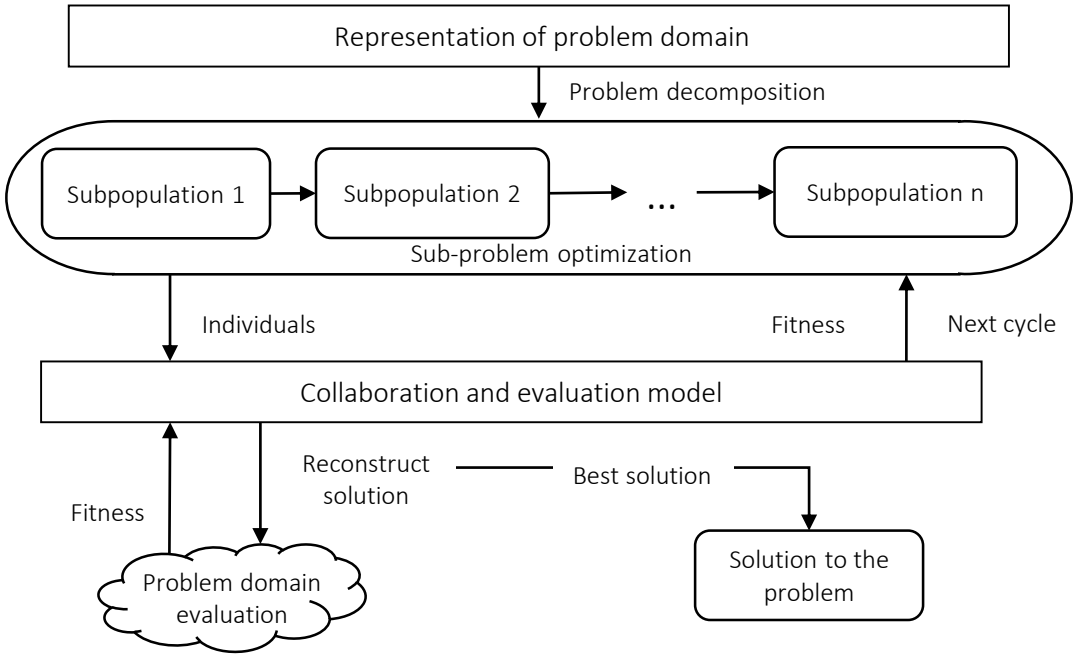


Figure 3. A general architecture of cooperative co-evolutionary algorithm

http://dx.doi.org/10.21511/ppm.17(4).2019.28

Source: Developed by the authors.

```
Algorithm 1 Cooperative Co-Evolutionary Algorithm (CCEA)
Require: n: number of variables, N: the population size, CR: crossover rate, MR:
mutation rate, G: number of generations.
1. Start of CCEA algorithm.
2. RPRESENT the problem domain.
3. Decompose the problem into a fixed or dynamic number of sub-problems and assign
   to subpopulations.
4. For each sub-problems s Do
   4.1 Randomly INITIALIZE subpopulation pop (s) of N individuals.
5. End of For
6. For each sub-problems s Do
   6.1 EVALUATE individuals in one subpopulation collaborating with other
    individuals from other subpopulations and assign fitness values to the
    individuals being evaluated.
7. End of For
8. Set count ← 0.
9. While termination condition (until G) is not met Do
   9.1 count ← count + 1.
   9.2 For each sub-problem s Do
       9.2.1 SELECT parents from the population.
       9.2.2 Apply GENETIC OPERATORS (if GA is used to evolve) on the selected
        parents to get offspring population.
             9.2.2.1 RECOMBINE (CROSSOVER) parents to generate new individuals
              subject to CR.
             9.2.2.2 MUTATE the individuals after crossover subject to MR.
       9.2.3 EVALUATE new individuals and assign fitness values.
       9.2.4 UPDATE CONTEXT VECTOR with the best individuals from each of the
        subpopulation.
       9.2.5 Decide SURVIVAL individuals for each subpopulation for new generation.
       9.2.6 Display best individual from each subpopulation for each generation.
   9.3 End of For
10. End of While
11. Return the best individual from each of the subpopulation over all generations.
12. End of CCEA algorithm.
```

Figure 4. An outline of cooperative co-evolutionary algorithm

learning applications (Juillé & Pollack, 1996). A general architecture and an outline of cooperative co-evolutionary algorithm (CCEA) are shown in Figure 3 and Figure 4. The CCEA consists of three main steps (Shi & Gao, 2017).

2.1. Problem decomposition

A decomposition strategy is used to decompose a complex problem into several sub-problems based on the structure of the problem (i.e., separable or non-separable problem) with appropriate granularity (Shi & Gao, 2017). The decomposition strategies are classified as static (decomposes a problem before the evolutionary process starts and decomposed sub-problems are fixed (Bucci & Pollack, 2005)) or dynamic (decomposes a problem at the beginning, but sub-problems have the ability to self-adaptively tune to proper collaboration levels at the time of evolutionary process (Omidvar, Li, Mei, & Yao, 2014)). Differential grouping (Omidvar, Li, Mei, & Yao, 2014) and random grouping (Yang, Tang, & Yao, 2008a) strategies have been used extensively for solving the complex optimization problems (both separable and non-separable problems). Improvements of both of the grouping methods are extended differential grouping (XDG) (Sun, Kirley, & Halgamuge, 2015), DG2 (Omidvar, Yang, Mei, Li, & Yao, 2017), recursive differential grouping (RDG) (Sun, Kirley, & Halgamuge, 2018), improvement of RDG inspired by DG2 (Sun, Omidvar, Kirley, & Li, 2018), multilevel CC framework (MLCC) (Yang, Tang, & Yao, 2008b), and random based dynamic grouping (RDG) (Song, Yang, Chen, & Zhang, 2016) to overcome the problems, for example, indirect and dynamic identification of variable interactions, nonlinearity detection of variable interactions, self-adaptive group size, and tackling large-scale MOPs, etc.

2.2. Sub-problems evolution

Once the decomposition is performed, each sub-problem is assigned to a population and an

evolutionary optimizer (the same or different) is used to evolve them. Evolutionary processes (initialization, fitness evaluation, selection, recombination, mutation, and survivor selection) are performed by populations independently (Shi & Gao, 2017). Sub-problems are evolved sequentially (only one population performs the evolutionary process per generation, while other populations are frozen (Potter, 1997)) or in parallel (all populations perform the evolutionary processes per generation concurrently (Wiegand, 2004). Evolutionary optimizers, such as GAs, are widely used to evolve the different subcomponents of CCEA after the decomposition of a problem into sub-problems. However, the most effective optimizer to CCEA in the literature found is the differential evolution (DE) (Storn & Price, 1997), which is a parallel direct search method and an EA technique. To improve the performance of DE, different variants of DE, such as self-adapting control parameters for DE (jDE) (Brest, Greiner, Boskovic, Mernik, & Zumer, 2006), neighborhood search differential evolution (NSDE) (Yang et al., 2008), self-adaptively NSDE (SaNSDE) (Yang, Yao, & He, 2008), self-adaptive strategy and control parameters for DE (SSCPDE) (Fan & Yan, 2015), and self-adaptive DE with zoning evolution of control parameters and adaptive mutation strategies (ZEPDE) (Fan & Yan, 2016), have been proposed in the literature.

2.3. Collaboration and evaluation

The fitness of an individual is evaluated by a collaborative mechanism that selects a collaborator from each of the populations. The performance of the collaboration is the fitness value to the individual. At the collaboration step, a population of the complete solution is formed by combining the collaborators to each individual of the current population and at the end of a CCEA process, the final solution to the problem is built by combining the individuals with the best collaboration (Shi & Gao, 2017). A number of collaboration strategies have been studied in the literature, including less greedy strategy (Potter, 1997), 1+1 collaboration model (Potter & de Jong, 2000), blended population algorithm (Sofge, De Jong, & Schultz, 2002), 1+N collaboration model (Bucci & Pollack, 2005), archive-based collaboration (Panait, Luke, & Harrison, 2006), N+N collaboration (Hoverstad, 2007), and Reference Sharing (RS) (Shi & Gao, 2017), all of which are significant collaboration models.

3. THE MAPREDUCE PROGRAMMING MODEL

Hadoop frameworks are built with a distributed storage location, the Hadoop distributed file system (HDFS) (Hadoop Apache, 2018), and the MapReduce programming model (Dean & Ghemawat, 2008, 2010). HDFS is a Java-based distributed file system that offers reliable, scalable, and fault-tolerant storage and computation processes for big data with faster access. The input data are divided into blocks in HDFS that can be processed in parallel without any need for communication between the data blocks. MapReduce has two main functions: *map* and *reduce*. *Map* and *reduce* functions are combined in a divide-and-conquer approach in which the *map* function works in parallel with the data blocks, whereas the *reduce* function collects and combines the intermediate result into a final output (Ferrucci, Salza, & Sarro, 2017). The MapReduce model is based on the data flow of (*key*, *value*) pairs. In general, a master node divides the initial input into several blocks identified as (*key*, *value*) pairs. The input, usually stored in HDFS, is split into (*key*, *value*) pairs and distributed through the *map* function to several slave nodes for working in parallel and executing the same task on a different block of input independently from each other. The mapper generates an intermediate list of (*key*, *value*) pairs, which is *shuffled* using a shuffling process. The MapReduce library groups these pairs together by the same *key* and passes to reducers. Finally, the reducer

Source: Developed by the authors.

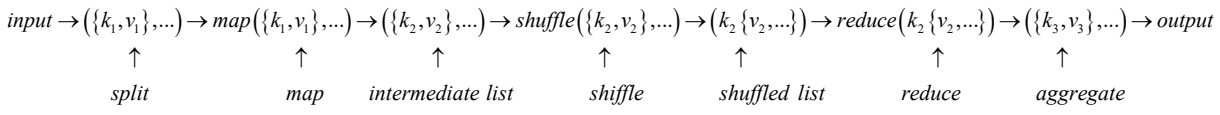


Figure 5. A typical MapReduce workflow shuffled list

http://dx.doi.org/10.21511/ppm.17(4).2019.28

Source: Developed by the authors based on Peralta et al. (2015)

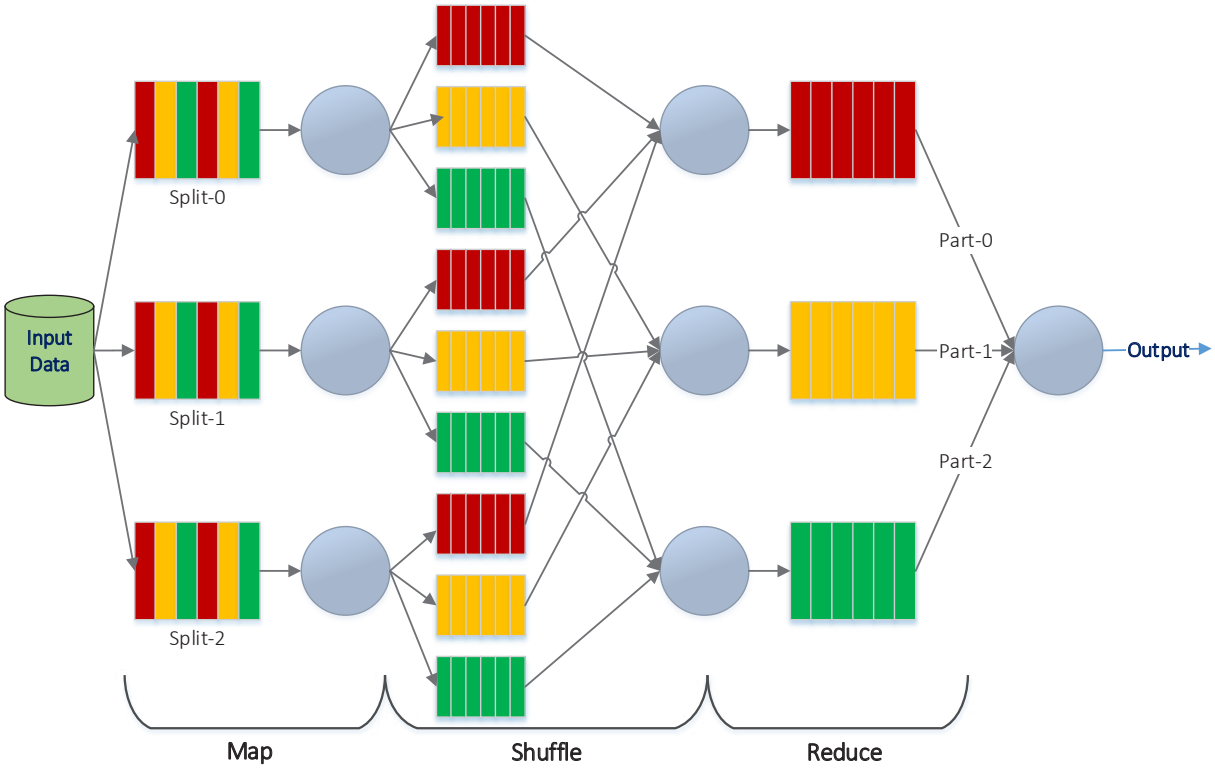


Figure 6. The basic flowchart of a MapReduce model

aggregates the different groups and produces new (*key*, *value*) pairs as final output to store in HDFS (Peralta et al., 2015; Sinha & Jana, 2018). The transition of (*key*, *value*) pair in MapReduce is depicted in Figure 5 and Figure 6 presents the basic flowchart of a MapReduce model.

MapReduce offers a parallel, fault-tolerant, and scalable framework for processing a large volume of distributed datasets. However, it increases overheads in terms of time during the execution of multiple and useless operations and iterative program execution because in each iteration, the data are written back to the HDFS (Sinha & Jana, 2018). One possible solution to this problem is to reduce the data store operations, for example, using an island model (Pierreval & Paris, 2000) for the MapReduce implementation, where the island model limits the data store only in the island migration phase (Ferrucci, Salza, & Sarro, 2017). Another solution to the overhead problem can be the use of MapReduce on Spark (Zaharia, Chowdhury, Franklin, Shenker, & Stoica, 2010) that helps to improve the performance of the iterative execution. Spark uses resilient distributed datasets (RDDs) that are read-only collections of objects distributed into different nodes. RDDs can be rebuilt if lost and it can ultimately be cached into memory, thereby providing a faster execution. However, Spark needs a lot of memory. There are numerous fields of applications of the MapReduce model, for example, big data analysis (Shim, 2012), bioinformatics (Taylor, 2010), and text mining (Balkir, Foster, & Rzhetsky, 2011). The MapReduce model provides the framework for implementing the *map* and *reduce* functions for applications to be executed in parallel. However, these two functions are problem-specific and need to be designed on a case-by-case basis.

4. STATE-OF-THE-ART FEATURE SELECTION TECHNIQUES

4.1. Evaluation criteria-based feature selection approaches

Based on the feature evaluation criteria, common classification algorithms, for instance, support vec-

tor machines (SVMs), *K*-nearest neighbor (KNN), Naïve Bayes (NB), decision trees (DT), etc. are used to evaluate the features in wrapper-based methods (Liu, Motoda, Setiono, & Zhao, 2010). Further, correlation measures, distance measures, information theory-based measures, or consistency measures are used for filter-based methods (Dash & Liu, 1997); one example is Relief (Kira & Rendell, 1992), which evaluates the feature relevance by distance measures. A distance measure-based feature evaluation (Wang, Pedrycz, Q. Zhu, & W. Zhu, 2015) and a minimum redundancy maximum relevance (mRMR) (Peng et al., 2005), based on mutual information incorporating the evolutionary computation (EC) techniques, are the examples that fall into the category of feature subset evaluation (i.e., wrapper methods). A FS method (Mao & Tsang, 2013) uses the optimization of multivariate performance measures, but it creates a huge search space involving the high-dimensional data. Traditional statistical approaches, such as logistic regression, cart classification (CART), regression tree, *T*-test, or hierarchical clustering, perform comparatively better and are simple, but are not suitable to high-dimensional data (Tan, Fisher, Rosenblatt, & Garner, 2009). Recently, sparse approaches have become popular to deal with FS involving the datasets with millions of features. An example of this approach is a sparse logistic regression method, where automatic weight is assigned to each relevant feature and low weights close to zero are assigned to irrelevant features (Tan, Tsang, & Wang, 2013). Sparse techniques, in terms of performance, have high efficiency and these techniques tend to learn simple models because of the bias to features with high weights. Further, these statistical sparse techniques typically make the assumptions about the probability distribution of the data.

4.2. Evolutionary computation-based feature selection approaches

Based on the search technique, very few existing works on FS are based on exhaustive search because these methods are computationally more expensive (Liu, Motoda, Setiono, & Zhao, 2010). A variety of heuristic search techniques, such as greedy search algorithms, sequential forward selection (SFS) (Whitney, 1971), and sequential backward selection (SBS) (Marill & Green, 1963) have been applied in the FS process as an alternative to the exhaustive search. However, SFS and SBS methods are limited by the nesting effect, i.e., selection or removal of a feature cannot be performed in a reverse way in the subsequent steps. An attempt to solve this problem, the "plus-*l*-take-away-*r*" approach (Strearns, 1976) was proposed by applying SFS *l* times and SBS *r* times. Nevertheless, the estimation of approximate values of *l* and *r* in practice is difficult. Approaches such as sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS) methods claim that they perform better than static sequential methods (Pudil, Novovicova, & Kittler, 1994).

FS is a typical combinatorial optimization problem. EC or evolutionary algorithms (EAs) have been used effectively for FS problems. A GA-based FS technique, which adopts the domain knowledge of financial distress prediction, divides the features into groups and each group uses a GA for finding the subsets of features (Lian, Liang, Yeh, & Huang, 2014). A GP-based hyper-heuristics wrapper FS (Hunt, Neshatian, & Zhang, 2012) finds the subset of features from UCI Machine Learning Repository datasets. A FS approach uses particle swarm intelligence (PSO) (Lane, Xue, Liu, & Zhang, 2013) to integrate the statistical feature clustering information during the PSO search to select the subset of features on benchmark datasets. An improved ant colony optimization (ACO)-based FS method (Zhao, Li, Yang, Ma, Zhu, & Chen, 2014) was used for online detection of foreign fiber in cotton. A self-adaptive differential evolution (DE) approach (A. Ghosh, Datta, & S. Ghosh, 2013) for FS involves the hyperspectral remotely sensed image datasets, where subsets of features are evaluated using a wrapper method with a fuzzy *k*-nearest neighbor classifier. A correlation-based memetic algorithm (MA) (GA plus a local search) FS technique uses the symmetrical uncertainty for large-scale gene expression datasets (Kannan & Ramaraj, 2010). To optimize FS and consolidation in music classification, evolutionary strategies (ESs) are applied (Vatolkin, Theimer, & Rudolph, 2009). A multi-objective artificial bee colony (ABC) filter method of FS based on a fuzzy mutual information fitness evaluation criteria has been proposed and tested on six benchmark datasets from UCI machine learning repository (Hancer, Xue, Zhang, Karaboga, & Akay, 2015). An improved artificial immune system (AIS) based on the opposite sign test and nearest neighbor classifier for FS method

http://dx.doi.org/10.21511/ppm.17(4).2019.28

(Wang, Chen, & Angelia, 2014) evaluates the datasets from UCI, KEEL repository, and microarray datasets. A hybrid estimation of distribution algorithm (EDA)-based filter-wrapper FS method (Shelke, Jayaraman, Ghosh, & Valadi, 2013) finds the subsets of features to build a robust quantitative structure-activity relationship (QSAR). Finally, a hybrid approach of FS using PSO and tabu search (TS) (Shen et al., 2008) selects the genes for tumor classification using the gene expression data.

Traditional GAs require high computational time to find the satisfactory solutions when they are applied to complex problems and they suffer from the risk of premature convergence to local optima. To make it scalable, parallel genetic algorithms (PGAs) have been proposed (Luque & Alba, 2011). PGA divides the whole population into multiple sub-populations and evolves them using the multiple processors concurrently. A PGA consists of several of GAs, which perform the execution on a part of population or independent sub-population with or without requiring any communication between them. PGAs can increase the population diversity that may lead to performance improvements plus reduced computational time (Chen, Lin, Tang, & Xia, 2016). Implementation of PGAs is based on global parallelization (master-slave), coarse-grained (island or distributed), or fine-grained (grid or cellular) types (Luque & Alba, 2011). Applications of PGAs to FS problems include a PGA of FS method to analyze complex systems (Mokshin, Saifudinov, Sharnin, Trusfus, & Tutubalin, 2018), a PGA-based attribute subset selection method using the rough set theory and MapReduce for intrusion detection in computer networks (El-Alfy & Alshammari, 2016), a coarse-grained PGA method for FS involving the benchmark datasets (Chen, Lin, Tang, & Xia, 2016), a web-based PGA tool for wrapper FS for biomedical datasets (Soufan, Kleftogiannis, Kalnis, & Bajic, 2015), and a PGA FS to predict geometric mean diameter of soil (Besalatpour, Ayoubi, Hajabbasi, Jazi, & Gharipour, 2014).

4.3. Cooperative co-evolutionary algorithms based feature selection approaches

Existing FS research based on CCEA is limited. The first one is a FS method for a pedestrian detection system (Guo, Cao, Xu, & Hong, 2007), where for each feature type, a sub-population is allocated individually. Based on the population size (small or large), this approach suffers from premature convergence and high computations. To avoid this, they proposed a sub-population size adjustment strategy to manage the proportion of features. The method has been compared with GA, random selection, and greedy approaches (AdaBoost algorithm) and has obtained a better subset of fea-

Table 1. Feature selection techniques based on cooperative co-evolution

Source: Developed by the authors.

References	Methodology used	Purpose	Data size
Guo, Cao, Xu, and Hong (2007); Cao, Xu, Wei, and Guo (2011)	One sub-population for each feature group, sub-population size adjustment strategy	Determine whether a candidate region contains a pedestrian	Minimum 1,000 to maximum 5,000 samples and 400 features each
Derrac, Garcia, and Herrera (2009)	3-population, CHC algorithm, 1-NN as multi-classifier, majority voting	Attribute reduction using instance and FS in a single process	Minimum 101 to maximum 1,728 samples, and minimum 4 to maximum 60 features
Derrac, Garcia, and Herrera (2010)	3-population, CHC algorithm, k-NN classification, majority voting	Attribute reduction using instance and FS in a single process	Maximum 6,435 to minimum 360 samples, and minimum 36 to maximum 90 features
Tian, Li, and Chen (2010)	Dual population, ranked-based selection, Pareto optimality, decaying radius selection clustering (DRSC)	Simultaneous network identification and prominent features by compact RBFNN model	20,000 samples and 180 features
Ebrahimpour, Nezamabadi-Pour, and Eftekhari (2018)	Random vertical decomposition, BGSA, information gain weights and Pearson correlation coefficients	Dealing with the small sample size and an enormously huge number of features (e.g., microarray datasets)	21 samples and 22,283 features (microarray datasets)

tures (from 400 features) with higher detection rate. This same work has been reproduced (Cao, Xu, Wei, & Guo, 2011) involving a different experimental environment and a higher number of negative samples, and obtained similar results. Table 1 presents a summary of the state-of-the-art FS techniques based on CCEAs.

A GA-based CCEA (CoCHC) for instance selection (IS) and FS (Derrac, Garcia, & Herrera, 2009) used three populations: one for IS, one for FS, and for IS and FS together. This method is computationally less expensive owing to FS and IS tasks being performed in a single process; however, it requires further verification for datasets with large number of features and noisy instances together with irrelevant features. Derrac, Garcia, and Herrera (2010) proposed another CCE technique (IFS-CoCo) based on three populations concept and *k*-NN classification for feature and instance selection. They performed the experiments over a wide range of datasets and obtained the improved results over other evolutionary feature and instance selection algorithms. Datasets they used for experiments range from having a sample size of 6,435 with 36 features to a dataset containing 360 samples with only 90 features. Hence, this method requires further validation in terms of high-dimensional datasets.

A dual-population-based CCEA (Tian, Li, & Chen, 2010) trains a hybrid machine learning algorithm called the radial basis function neural network (RBFNN) for FS and network identification on 26 real-world classification problems. The proposed method performed the simultaneous implementation of processing hidden layer structure and FS of the RBFNN using a divide-and-cooperative mechanism. Experiments performed on 26 datasets with a maximum of 20,000 samples, 180 features, and 26 different classes and it obtained better accuracy and decreased the number of features to tackle multi-objective (Inotai et al., 2018) optimization (Goberna, Jeyakumar, Li, & Vicente-Pérez, 2018) in comparison to other methods. The FS based on CCE (CCFS) techniques (Ebrahimpour, Nezamabadi-Pour, & Eftekhari, 2018) deals with small sample size and an enormously huge number of features (e.g., microarray datasets). They divided datasets vertically in a random manner and used a binary gravitational search algorithm (BGSA) in each of the subsolution spaces. Information gain weights and Pearson correlation coefficients were used to evaluate the fitness function. Experiments were performed on seven binary microarray datasets and were evaluated against nine state-of-the-art FS techniques. In terms of accuracy, sensitivity, specificity, and a several selected features, CCFS has achieved the significant results compared to other methods.

Several FS studies are performed based on the cooperative (Shi, Li, & Teo, 2015) concepts, but not using CCEA are a multiple population cooperative GA-based FS approach (Li, Zhang, & Zengl, 2009), a fuzzy model-based wrapper FS method on two cooperative ant colonies (Vieira, Sousa, & Runkler, 2010), a cooperative particle swarm optimization (PSO) technique-based integrative feature and instance selection approach (FS-CPSO) (Ahmad & Pedrycz, 2011), a cooperative binary particle swarm optimization (CBPSO) approach of integrative feature and instance selection (FISCBPSO) to deal with the problem of nearest neighbor (NN) classification for high dimensional data (Sakinah & Ahmad, 2014), a cooperative subset search and instance learning-based FS (Brahim & Limam, 2016), and cooperative game-theory based FS approaches (Gore & Govindaraju, 2016; Mortazavi & Moattar, 2016).

4.4. MapReduce-based feature selection approaches

Several works on distributed FS are available in the literature, where different subsets of features were processed concurrently using the parallel processing. The parallel processing might increase the efficiency of search techniques for relevant features, but it required the dataset to store in each of the computing units. Hence, these approaches are not efficient when the dataset size is increased (Guillen, Sorjamaa, Miche, Lendasse, & Rojas, 2009). To improve the efficiency of parallel processing, MapReduce-based scalable FS techniques have been introduced where datasets are split into chunks. Singh et al. (2009) proposed a scalable embedded FS method based on the estimate of logistic regression model's performance on the subsets of the training dataset. Peralta et al. (2015) proposed a wrapper FS-based EA on MapReduce platform. Filter-based FS methods

http://dx.doi.org/10.21511/ppm.17(4).2019.28

Source: Developed by the authors.

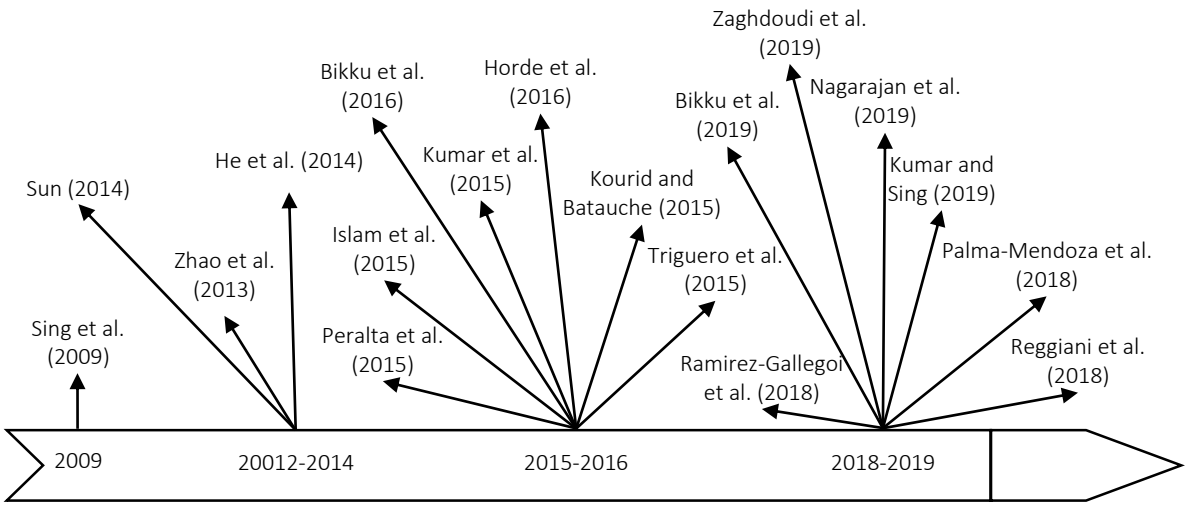


Figure 7. Feature selections techniques based on MapReduce

using MapReduce (Ramírez-Gallego et al., 2018; Sun, 2014) have used different evaluation metrics, such as mutual information or preservation to address the column subset selection problem (CSSP) and the distribution of data by features. Figure 7 presents a summary of FS techniques based on MapReduce.

A Hadoop MapReduce-based FS method for traditional rough sets (He, Cheng, Zhuang, & Shi, 2014) uses a positive approximation as an accelerator. A CPU-based MapReduce parallel gene selection model (Islam, Jeong, Bari, Lim, & Jeon, 2015) uses the sampling techniques and between-groups to the within-groups sum of square (BW) ratio, where BW ratio specifies the variances among gene expression values. After the subset of features is selected, MRkNN techniques are used to run multiple *k*NN in parallel in the MapReduce model. The effectiveness of this method has been tested using four real and three synthetic datasets, and in terms of accuracy and scalability, it performed better. Kourid and Batouche (2015) proposed a biomarker identification method based on a large-scale FS and MapReduce model by combining *K*-means clustering and signal-to-noise ratio with a Binary Particle Swarm Optimization technique (BPSO). Such approach for analyzing microarray data requires high computation time. A similar method based on the MapReduce model (Kumar, Rath, Swain, & Rath, 2015) for FS and classification of microarray data (NCBI) uses the statistical test analysis of variance (ANOVA) for gene selection and kNN classification. Methods based on ANOVA require testing the assumptions of independence and normality that may not work for FS problems with complex interactions among features and these methods are computationally expensive. Triguero, Peralta, Bacardit, García, and Herrera (2015) proposed an IS method based on distributed partitioning and an advanced IR technique (SSMA-SFLSDE) for nearest neighbor classification.

The FS and decision-making method based on Hadoop MapReduce model (Bikku, Rao, & Akepogu, 2016) suffers from problems, such as high-latency to store intermediate results on disk and the overhead of map jobs common to the Hadoop MapReduce framework. To reduce the computation time, FS algorithms are executed in parallel using the ANN embedded method in Hadoop framework (Hodge, O'Keefe, & Austin, 2016). A filter-based method (Reggiani, Le Borgne, & Bontempi, 2018) tackles the forward FS algorithm minimal Redundancy Maximal Relevance (*mRMR*) using MapReduce on Apache Spark. Here, an alternative encoding system (representing features in row level) customizes the feature score function on MapReduce to improve the performance in comparison to conventional encoding. These methods need further verification with the state-of-the-art FS techniques because they did not compare their accuracy of the proposed method with other conventional and alternative methods based on *mRMR*. Moreover, they

Table 2. Feature selection techniques based on cooperative co-evolution and MapReduce

Source: Developed by the authors.

References	Techniques used	Purpose	Data size
Ding, Jie. Wang, and Jia. Wang (2016)	Hierarchical co-evolution, ensemble Pareto dominance, layered niche neighborhood, MapReduce	Knowledge reduction for big data analysis	5 million samples and 60 attributes
Ding, Lin, Chen, Zhang, and Hu (2018)	Multiagent-consensus MapReduce, co-evolutionary quantum PSO with self-adaptive memeplexes, four-layer neighborhood radius framework, rough set theory	Attribute reduction for big data applications	Samples from 10,000 to 5,000,00 and variable number of features with low to high dimensions

have used the artificial datasets for the experiments. Palma-Mendoza et al. (2018) proposed a distributed ReliefF-based FS method (DiReliefF) in Apache Spark. The assumptions about the estimation sample, for instance, tiny samples with few hundreds of instances to estimate the class separability problems in millions of samples, need further verification.

4.5. Co-evolution and MapReduce-based feature selection approaches

To the best of our knowledge, works involving the combination of CEA and the MapReduce model are an emerging area of research and the existing works are limited. Table 2 presents a summary of the state-of-the-art FS techniques based on the combination of CCEA and MapReduce.

Ding, Jie. Wang, and Jia. Wang (2016) proposed a knowledge reduction method based on a hierarchical co-evolutionary MapReduce (HCMPKR) with ensemble Pareto dominance. A layered niche neighborhood radius is used to split the whole population into *N* sub-populations and to self-adaptively divide into attribute approximate space with interacting attributes. Elitist leaders from the Pareto front use an ensemble approach of reduction Pareto equilibrium perform cooperative game subsets in various niche conic subsets. MapReduce technique were used for knowledge reduction using the elitist leaders. Experiments performed on four real datasets and four synthetic datasets having a maximum of 60 attributes, 45 class variables, and 5 million samples where datasets were duplicated for generating big data from the UCI repository. The performance of this approach was compared with the state-of-the-art techniques and resulted in better performance. An attribute reduction method based on a multiagent-consensus MapReduce model for big data applications has been proposed using a co-evolutionary quantum PSO with self-adaptive memeplexes to group the particles into different memeplexes (Ding et al., 2018). A four-layer neighborhood radius framework with a compensatory scheme splits the attribute sets into subspace maintaining attributes interacting properties and maps to the MapReduce model. The attribute reduction is performed based on rough set theory, and the ensemble co-evolutionary MapReduce optimization is performed by five varieties of agents. Experiments were conducted on 16 benchmark datasets including three biomedical datasets, four public microarray datasets, four NIPS 2003 FS challenge datasets, and four large-scale synthetic datasets generated by WEKA. The proposed approach of attribute reduction achieves better results in most cases based on classification accuracy in comparison to algorithms, such as RACOFS, mRMR, and MRMS.

CONCLUSION

Feature or variable selection in high-dimensional big data is a challenging task and it improves the classification accuracy. Despite of numerous feature selection algorithms, including traditional or statistical methods, they cannot meet the demands of optimizing large-scale high-dimensional datasets. Most feature selection algorithms emphasize the datasets containing a large number of samples, but only a few studies are available on high-dimensional data, such as financial big data. Big data optimization, such as feature selection requires a large number of computations, especially when the case is high-di-

http://dx.doi.org/10.21511/ppm.17(4).2019.28

mensional. Evolutionary optimization is therefore an obvious selection to tackle these types of problem. Moreover, evolutionary optimization on big data for feature selection works is limited. Cooperative co-evolution, a meta-heuristic evolutionary algorithm uses the divide-and-conquer strategy to decompose a high-dimension problem into a number of lower-dimension sub-problems, which are optimized independently. Thus, it improves the optimization performance. Further, MapReduce, a parallel programming model can help to reduce computations of the developed distributed cooperative co-evolutionary algorithm parallelizing it. Hence, feature selection techniques involving co-evolutionary algorithms and MapReduce is an emerging area of research and yet to be fully explored for knowledge management or knowledge discovery.

ACKNOWLEDGMENT

This research is supported by ECU Higher Degree by Research Scholarship and ECU School of Science Research Scholarship.

REFERENCES

1. Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert Systems with Applications, 36*(3), 6843-6853. https://doi.org/10.1016/j.eswa.2008.08.022
2. Ahmad, S. S. S., & Pedrycz, W. (2011). Feature and Instance Selection Via Cooperative PSO. *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC),* 2127-2132.
3. Ahmed, S., Zhang, M. J., & Peng, L. F. (2013). Enhanced Feature Selection for Biomarker Discovery in LC-MS Data using GP. *2013 Ieee Congress on Evolutionary Computation (Cec),* 584-591.
4. Ali, M. M., Rattanawiboonsom, V., Hassan, F., & Nedelea, A. M. (2019). Knowledge Management at Higher Educational Institutes in Bangladesh: The case study of self-assessed processes of two educational Institutions. *Ecoforum Journal, 8*(1). Retrieved from http://www.ecoforumjournal.ro/index.php/eco/article/view/901/572
5. Aliferis, C. F., Statnikov, A., & Tsamardinos, I. (2006). Challenges in the Analysis of Mass-Throughput Data: A Technical Commentary from the Statistical Machine Learning Perspective. *Cancer Informatics, 2,* 117693510600200004. https://doi.org/10.1177/117693510600200004
6. Bakanauskienė, I., Bendaravičienė, R., & Barkauskė, L. (2017). Features of Employer Attractiveness on Lithuanian Business Organizations: Employees' Perceptions. *Management of Organizations: Systematic Research, 77*(1), 7-23. https://doi.org/10.1515/mosr-2017-0001
7. Bakanauskienė, I., Bendaravičienė, R., & Juodelytė, N. (2018). Organizational values in human resource management context: case of Lithuania. *Human resources management and ergonomics (HRM&E). Zvolen, Slovakia: Technical university in Zvolen, 12*(1), 6-20. Retrieved from https://www.vdu.lt/cris/handle/20.500.12259/60051
8. Balkir, A. S., Foster, I., & Rzhetsky, A. (2011). A distributed look-up architecture for text mining applications using mapreduce. *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis.*
9. Besalatpour, A. A., Ayoubi, S., Hajabbasi, M. A., Jazi, A. Y., & Gharipour, A. (2014). Feature Selection Using Parallel Genetic Algorithm for the Prediction of Geometric Mean Diameter of Soil Aggregates by Machine Learning Methods. *Arid Land Research and Management, 28*(4), 383-394. https://doi.org/10.1080/15324982.2013.871599
10. Bhattacharya, M., Islam, R., & Abawajy, J. (2016). Evolutionary optimization: A big data perspective. *Journal of Network and Computer Applications, 59,* 416-426. https://doi.org/10.1016/j.jnca.2014.07.032
11. Bikku, T., Rao, N. S., & Akepogu, A. R. (2016). Hadoop based Feature Selection and Decision Making Models on Big Data. *Indian Journal of Science and Technology, 9*(10), 1-6. https://doi.org/10.17485/ijst/2016/v9i10/88905
12. Bolon-Canedo, V., Rego-Fernandez, D., Peteiro-Barral, D., Alonso-Betanzos, A., Guijarro-Berdinas, B., & Sanchez-Marono, N. (2018). On the scalability of feature selection methods on high-dimensional data. *Knowledge and Information Systems, 56*(2), 395-442. https://doi.org/10.1007/s10115-017-1140-3
13. Boroujeni, F. R., Stantic, B., & Wang, S. (2017). An Embedded Feature Selection Framework for Hybrid Data. *Databases Theory and Applications, 10538,* 138-150. https://doi.org/10.1007/978-3-319-68155-9_11

14. Brahim, A. B., & Limam, M. (2016). A hybrid feature selection method based on instance learning and cooperative subset search. *Pattern Recognition Letters, 69,* 28-34. https://doi.org/10.1016/j.patrec.2015.10.005

15. Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *Ieee Transactions on Evolutionary Computation, 10*(6), 646-657. https://doi.org/10.1109/TEVC.2006.872133

16. Bucci, A., & Pollack, J. B. (2005). On identifying global optima in cooperative coevolution. *GECCO 2005: Genetic and Evolutionary Computation Conference, 1-2,* 539-544.

17. Cao, X. B., Xu, Y. W., Wei, C. X., & Guo, Y. P. (2011). Feature subset selection based on co-evolution for pedestrian detection. *Transactions of the Institute of Measurement and Control, 33*(7), 867-879. https://doi.org/10.1177/0142331209103041

18. Chalikias, M., Kyriakopoulos, G., Skordoulis, M., & Koniordos, M. (2014). *Knowledge Management for Business Processes: Employees' Recruitment and Human Resources' Selection: A Combined Literature Review and a Case Study.* Cham.

19. Chen, Z., Lin, T., Tang, N. J., & Xia, X. (2016). A Parallel Genetic Algorithm Based Feature Selection and Parameter Optimization for Support Vector Machine. *Scientific Programming.* https://doi.org/10.1155/2016/2739621

20. Clarke, R., Ressom, H. W., Wang, A., Xuan, J., Liu, M. C., Gehan, E. A., & Wang, Y. (2008). The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nature Reviews Cancer, 8,* 37. https://doi.org/10.1038/nrc2294

21. Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*(1), 131-156. https://doi.org/10.1016/S1088-467X(97)00008-5

22. Dash, M., & Liu, H. (2003). Consistency-based search in feature selection. *Artificial Intelligence, 151*(1-2), 155-176. https://doi.org/10.1016/S0004-3702(03)00079-1

23. Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Commun. ACM, 51*(1), 107-113. https://doi.org/10.1145/1327452.1327492

24. Dean, J., & Ghemawat, S. (2010). MapReduce: A Flexible Data Processing Tool. *Communications of the Acm, 53*(1), 72-77. https://doi.org/10.1145/1629175.1629198

25. Deepak, M., Mahesh, G., & Medi, N. K. (2019). Knowledge Management Influence on Safety Management Practices: Evidence from Construction Industry. *International Journal of Knowledge Management (IJKM), 15*(4), 16-37. https://doi.org/10.4018/IJKM.2019100102

26. Derrac, J., Garcia, S., & Herrera, F. (2009). A first study on the use of coevolutionary algorithms for instance and feature selection. *Hybrid Artificial Intelligence Systems,* 557-564. https://doi.org/10.1007/978-3-642-02319-4_67

27. Derrac, J., Garcia, S., & Herrera, F. (2010). IFS-CoCo: Instance and feature selection based on cooperative coevolution with nearest neighbor rule. *Pattern Recognition, 43*(6), 2082-2105. https://doi.org/10.1016/j.patcog.2009.12.012

28. Ding, W., Lin, C., Chen, S., Zhang, X., & Hu, B. (2018). Multiagent-consensus-MapReduce-based attribute reduction using co-evolutionary quantum PSO for big data applications. *Neurocomputing, 272,* 136-153. https://doi.org/10.1016/j.neucom.2017.06.059

29. Ding, W., Wang, Jie., & Wang, Jia. (2016). A hierarchical-coevolutionary-MapReduce-based knowledge reduction algorithm with robust ensemble Pareto equilibrium. *Information Sciences, 342,* 153-175. https://doi.org/10.1016/j.ins.2016.01.035

30. Ebrahimpour, M. K., Nezamabadi-Pour, H., & Eftekhari, M. (2018). CCFS: A cooperating coevolution technique for large scale feature selection on microarray datasets. *Computational Biology and Chemistry, 73,* 171-178. https://doi.org/10.1016/j.compbiolchem.2018.02.006

31. El-Alfy, E. M., & Alshammari, M. A. (2016). Towards scalable rough set based attribute subset selection for intrusion detection using parallel genetic algorithm in MapReduce. *Simulation Modelling Practice and Theory, 64,* 18-29. https://doi.org/10.1016/j.simpat.2016.01.010

32. Fan, J., & Li, R. (2006). Statistical challenges with high dimensionality: Feature selection in knowledge discovery. *25th International Congress of Mathematicians, ICM 2006.* Madrid, Spain. Retrieved from https://pennstate.pure.elsevier.com/en/publications/statistical-challenges-with-high-dimensionality-feature-selection

33. Fan, Q., & Yan, X. (2015). Differential evolution algorithm with self-adaptive strategy and control parameters for P-xylene oxidation process optimization. *Soft Computing, 19*(5), 1363-1391. https://doi.org/10.1007/s00500-014-1349-y

34. Fan, Q., & Yan, X. (2016). Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies. *Ieee Transactions on Cybernetics, 46*(1), 219-232. https://doi.org/10.1109/TCYB.2015.2399478

35. Ferrucci, F., Salza, P., & Sarro, F. (2017). Using Hadoop MapReduce for Parallel Genetic Algorithms: A Comparison of the Global, Grid and Island Models. *Evolutionary Computation, XX*(X), 1-33. https://doi.org/10.1162/evco_a_00213

36. Gao, Z., & Tsay, R. S. (2019). A Structural-Factor Approach to Modeling High-Dimensional Time Series and Space-Time Data. *Journal of Time Series Analysis, 40*(3), 343-362. https://doi.org/10.1111/jtsa.12466

37. Ghosh, A., Datta, A., & Ghosh, S. (2013). Self-adaptive differential

http://dx.doi.org/10.21511/ppm.17(4).2019.28

evolution for feature selection in hyperspectral image data. *Applied Soft Computing, 13*(4), 1969-1977. https://doi.org/10.1016/j.asoc.2012.11.042

38. Goberna, M. A., Jeyakumar, V., Li, G., & Vicente-Pérez, J. (2018). Guaranteeing highly robust weakly efficient solutions for uncertain multi-objective convex programs. *European Journal of Operational Research, 270*(1), 40-50. https://doi.org/10.1016/j.ejor.2018.03.018

39. Gore, S., & Govindaraju, V. (2016). Feature Selection Using Cooperative Game Theory and Relief Algorithm. In A. Skulimowski & J. Kacprzyk (Eds.), *Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions* (pp. 401-412). https://doi.org/10.1007/978-3-319-19090-7_30

40. Grandon, E. E., Ramirez-Correa, P. E., & Luna, J. S. (2019). E-Business Applications Model in Large Companies: An Empirical Validation. *Interciencia, 44*(4), 210-217. Retrieved from https://www.interciencia.net/wp-content/uploads/2019/05/03_210_Com_Grandon_v44_04.pdf

41. Grytten, O. H., & Minde, K. B. (2019). *Generational links between entrepreneurship, management and puritanism*. Retrieved from https://openaccess.nhh.no/nhh-xmlui/handle/11250/2600232?locale-attribute=no

42. Guillen, A., Sorjamaa, A., Miche, Y., Lendasse, A., & Rojas, I. (2009). Efficient Parallel Feature Selection for Steganography Problems. In J. Cabestany, F. Sandoval, A. Prieto & J. M. Corchado (Eds.), *Bio-Inspired Systems: Computational and Ambient Intelligence*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-02478-8_153

43. Guo, Y. P., Cao, X. B., Xu, Y. W., & Hong, Q. (2007). Co-evolution based feature selection for pedestrian detection. *2007 IEEE International Conference on Control and Automation*. Guangzhou, China. https://doi.org/10.1109/ICCA.2007.4376871

44. Gupta, R. (2016). Marketing Management is a Trust Worthy Paradigm of Corporate Branding. *International Journal of Information, Business and Management, 8*(1), 46-50. Retrieved from https://ijibm.elitehall.com/IJIBM_Vol8No1_Feb2016.pdf

45. Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research, 3,* 1157-1182. Retrieved from https://www.researchgate.net/publication/221996079_An_Introduction_of_Variable_and_Feature_Selection

46. Habib, M. M., & Hasan, I. (2019). Supply Chain Management (SCM) – Is it Value Addition towards Academia? *IOP Conference Series: Materials Science and Engineering, 528*(1), 012090. https://doi.org/10.1088/1757-899X/528/1/012090

47. Hadoop Apache. (2018). *HDFS Architecture*. Retrieved from http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html

48. Hancer, E., Xue, B., Zhang, M. J., Karaboga, D., & Akay, B. (2015). A Multi-Objective Artificial Bee Colony Approach to Feature Selection Using Fuzzy Mutual Information. *2015 IEEE Congress on Evolutionary Computation (CEC),* 2420-2427. Retrieved from https://homepages.ecs.vuw.ac.nz//~xuebing/Papers/moabc-CEC2015.pdf

49. Hashem, I. A. T., Anuar, N. B., Gani, A., Yaqoob, I., Xia, F., & Khan, S. U. (2016). MapReduce: Review and open challenges. *Scientometrics, 109*(1), 389-422. https://doi.org/10.1007/s11192-016-1945-y

50. He, Q., Cheng, X. H., Zhuang, F. Z., & Shi, Z. Z. (2014). Parallel Feature Selection Using Positive Approximation Based on MapReduce. *11th International Conference on Fuzzy Systems and Knowledge Discovery (Fskd),* 397-402. https://doi.org/10.1109/FSKD.2014.6980867

51. Hodge, V. J., O'Keefe, S., & Austin, J. (2016). Hadoop neural network for parallel and distributed feature selection. *Neural Networks, 78,* 24-35. https://doi.org/10.1016/j.neunet.2015.08.011

52. Hoverstad, B. A. (2007). Revisiting the personal satellite assistant: neuroevolution with a modified enforced sub-populations algorithm. *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation.* Retrieved from https://www.researchgate.net/publication/220743467_Revisiting_the_personal_satellite_assistant_neuroevolution_with_a_modified_enforced_sub-populations_algorithm

53. Hunt, R., Neshatian, K., & Zhang, M. (2012). *A Genetic Programming Approach to Hyper-Heuristic Feature Selection*. Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-34859-4_32

54. IBM. (2018). *IBM big data analytics: insights without limits*. Retrieved from https://www.ibm.com/au-en/it-infrastructure/solutions/big-data

55. Illiashenko, S. M., Strielkowski, W., Letunovska, N. Y., Bozhkova, V. V., Prokopenko, O. V., Tielietov, O. S.,…, & Hryshchenko, O. F. (2018*). Innovative management: theoretical, methodical, and applied grounds.* Retrieved from https://essuir.sumdu.edu.ua/handle/123456789/67454

56. Inotai, A., Brixner, D., Maniadakis, N., Dwiprahasto, I., Kristin, E., Prabowo, A.,…, & Kalo, Z. (2018). Development of multi-criteria decision analysis (MCDA) framework for off-patent pharmaceuticals – an application on improving tender decision making in Indonesia. *BMC health services research, 18*(1), 1003. https://doi.org/10.1186/s12913-018-3805-3

57. Islam, A. K. M. T., Jeong, B. S., Bari, A. T. M. G., Lim, C. G., & Jeon, S. H. (2015). MapReduce based parallel gene selection method. *Applied Intelligence, 42*(2), 147-156. https://doi.org/10.1007/s10489-014-0561-x

58. Juillé, H., & Pollack, J. B. (1996). Co-evolving Intertwined Spirals.

Proceedings of the Fifth Annual Conference on Evolutionary Programming, 461-468. Retrieved from http://www.demo.cs.brandeis.edu/papers/ep96.pdf

59. Kannan, S. S., & Ramaraj, N. (2010). A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm. *Knowledge-Based Systems, 23*(6), 580-585. https://doi.org/10.1016/j.knosys.2010.03.016

60. Ketcha, A., Johannesson, J., & Bocij, P. (2015). Tacit knowledge acquisition and dissemination in distance learning. *European Journal of Open, Distance and E-learning, 18*(2). Retrieved from https://www.eurodl.org/index.php?p=archives&sp=brief&year=2015&halfyear=2&article=692

61. Khan, N., & Kakabadse, N. K. (2014). CSR: the co-evolution of grocery multiples in the UK (2005–2010). *Social Responsibility Journal, 10*(1), 137-160. https://doi.org/10.1108/SRJ-06-2012-0069

62. Kim, G., Kim, Y., Lim, H., & Kim, H. (2010). An MLP-based feature subset selection for HIV-1 protease cleavage site analysis. *Artificial Intelligence in Medicine, 48*(2), 83-89. https://doi.org/10.1016/j.artmed.2009.07.010

63. Kira, K., & Rendell, L. A. (1992). A Practical Approach to Feature-Selection. *Machine Learning, 92,* 249-256. Retrieved from https://sci2s.ugr.es/keel/pdf/algorithm/congreso/kira1992.pdf

64. Kourid, A., & Batouche, M. (2015). Biomarker Discovery Based on Large-Scale Feature Selection and MapReduce. *IFIP International Conference on Computer Science and Its Applications* (pp. 81-92). https://doi.org/10.1007/978-3-319-19578-0_7

65. Kumar, M., Rath, N. K., Swain, A., & Rath, S. K. (2015). Feature Selection and Classification of Microarray Data using MapReduce based ANOVA and K-Nearest Neighbor. *Procedia Computer Science, 54,* 301-310. https://doi.org/10.1016/j.procs.2015.06.035

66. Lane, M. C., Xue, B., Liu, I., & Zhang, M. (2013). Particle Swarm Optimisation and Statistical Clustering for Feature Selection. In *Australasian Joint Conference on Artificial Intelligence,* 214-220. https://doi.org/10.1007/978-3-319-03680-9_23

67. Laney, D. (2001). *3D Data Management: Controlling Data Volume, Velocity, and Variety*. Retrieved from http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

68. Levner, I. (2005). Feature selection and nearest centroid classification for protein mass spectrometry. *BMC Bioinformatics, 6,* 68. https://doi.org/10.1186/1471-2105-6-68

69. Li, Y. M., Zhang, S. J., & Zeng, X. P. (2009). Research of multi-population agent genetic algorithm for feature selection. *Expert Systems with Applications, 36*(9), 11570-11581. https://doi.org/10.1016/j.eswa.2009.03.032

70. Lin, F., Liang, D., Yeh, C.-C., & Huang, J.-C. (2014). Novel feature selection methods to financial distress prediction. *Expert Systems with Applications, 41*(5), 2472-2483. https://doi.org/10.1016/j.eswa.2013.09.047

71. Lin, J. Y., Ke, H. R., Chien, B. C., & Yang, W. P. (2008). Classifier design with feature selection and feature extraction using layered genetic programming. *Expert Systems with Applications, 34*(2), 1384-1393. https://doi.org/10.1016/j.eswa.2007.01.006

72. Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *Ieee Transactions on Knowledge and Data Engineering, 17*(4), 491-502. https://doi.org/10.1109/TKDE.2005.66

73. Liu, H., Motoda, H., Setiono, R., & Zhao, Z. (2010). Feature Selection: An Ever Evolving Frontier in Data Mining. *Proceedings of the Fourth International Workshop on Feature Selection in Data Mining, 10,* 4-13. Retrieved from http://proceedings.mlr.press/v10/liu10b/liu10b.pdf

74. Liu, Y. N., Wang, G., Chen, H. L., Dong, H., Zhu, X. D., & Wang, S. J. (2011). An Improved Particle Swarm Optimization for Feature Selection. *Journal of Bionic Engineering, 8*(2), 191-200. https://doi.org/10.1016/S1672-6529(11)60020-6

75. Liu, Y., Tang, F., & Zeng, Z. Y. (2015). Feature Selection Based on Dependency Margin. *Ieee Transactions on Cybernetics, 45*(6), 1209-1221. https://doi.org/10.1109/Tcyb.2014.2347372

76. Luque, G., & Alba, E. (2011). *Parallel genetic algorithms: Theory and real world applications*. Springer. Retrieved from https://link.springer.com/book/10.1007%2F978-3-642-22084-5

77. Ma, X., Li, X., Zhang, Q., Tang, K., Liang, Z., Xie, W., & Zhu, Z. (2018). A Survey on Cooperative Co-evolutionary Algorithms. *Ieee Transactions on Evolutionary Computation,* 1-1. https://doi.org/10.1109/TEVC.2018.2868770

78. Mao, Q., & Tsang, I. W. H. (2013). A Feature Selection Method for Multivariate Performance Measures. *Ieee Transactions on Pattern Analysis and Machine Intelligence, 35*(9), 2051-2063. https://doi.org/10.1109/Tpami.2012.266

79. Marill, T., & Green, D. (1963). On the effectiveness of receptors in recognition systems. *Ieee Transactions on Information Theory, 9*(1), 11-17. https://doi.org/10.1109/TIT.1963.1057810

80. Mokshin, V., Saifudinov, I., Sharnin, L., Trusfus, M., & Tutubalin, P. (2018). A parallel genetic algorithm of feature selection for analysis of complex system. *Journal of Physics: Conference Series, 1096*(1). https://doi.org/10.1088/1742-6596/1096/1/012089

81. Mortazavi, A., & Moattar, M. H. (2016). Robust Feature Selection from Microarray Data Based on Cooperative Game Theory and Qualitative Mutual Information. *Advances in Bioinformatics, 2016,* 1058305. https://doi.org/10.1155/2016/1058305

http://dx.doi.org/10.21511/ppm.17(4).2019.28

82. Mura, L., Daňová, M., Vavrek, R., & Dubravska, M. (2017). Economic Freedom-Classification of its Level and Impact on the Economic Security. *Ad Alta: Journal of Interdisciplinary Research, 7*(2), 154-157. Retrieved from https://www.researchgate.net/publication/324484966_ECONOMIC_FREEDOM-CLASSIFICATION_OF_ITS_LEVEL_AND_IMPACT_ON_THE_ECONOMIC_SECURITY

83. Omidvar, M. N., Li, X., Mei, Y., & Yao, X. (2014). Cooperative co-evolution with differential grouping for large scale optimization. *Ieee Transactions on Evolutionary Computation, 18*(3), 378-393. http://dx.doi.org/10.1109/TEVC.2013.2281543

84. Omidvar, M. N., Yang, M., Mei, Y., Li, X. D., & Yao, X. (2017). DG2: A Faster and More Accurate Differential Grouping for Large-Scale Black-Box Optimization. *Ieee Transactions on Evolutionary Computation, 21*(6), 929-942. https://doi.org/10.1109/Tevc.2017.2694221

85. Pagie, L., & Hogeweg, P. (2000). Information integration and red queen dynamics in coevolutionary optimization. *Proceedings of the 2000 Congress on Evolutionary Computation, 1-2,* 1260-1267. Retrieved from http://www-binf.bio.uu.nl/pdf/Pagie.cec00.pdf

86. Palma-Mendoza, R. J., Rodriguez, D., & de-Marcos, L. (2018). Distributed ReliefF-based feature selection in Spark. *Knowledge and Information Systems, 57*(1), 1-20. https://doi.org/10.1007/s10115-017-1145-y

87. Panait, L., Luke, S., & Harrison, J. F. (2006). Archive-based cooperative coevolutionary algorithms. *Gecco 2006: Genetic and Evolutionary Computation Conference, 1-2,* 345-352. https://doi.org/10.1145/1143997.1144060

88. Peng, H. C., Long, F. H., & Ding, C. (2005). Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *Ieee Transactions on Pattern Analysis and Machine Intelligence, 27*(8), 1226-1238. https://doi.org/10.1109/Tpami.2005.159

89. Peralta, D., del Rio, S., Ramirez-Gallego, S., Triguero, I., Benitez, J., & Herrera, F. (2015). Evolutionary Feature Selection for Big Data Classification: A MapReduce Approach. *Mathematical Problems in Engineering, 2015,* 245139. https://doi.org/10.1155/2015/246139

90. Pierreval, H., & Paris, J. (2000). Distributed evolutionary algorithms for simulation optimization. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 30*(1), 15-24. https://doi.org/10.1109/3468.823477

91. Potter, M. A. (1997). *The design and analysis of a computational model of cooperative coevolution.* George Mason University Fairfax, VA, USA. Retrieved from https://cs.gmu.edu/~mpotter/pubs/thesis2.pdf

92. Potter, M. A., & de Jong, K. A. (1994). A Cooperative Coevolutionary Approach to Function Optimization. *Parallel Problem Solving from Nature – PPSN III*, 249-257. https://doi.org/10.1007/3-540-58484-6_269

93. Potter, M. A., & de Jong, K. A. (1995). Evolving neural networks with collaborative species. *Proceedings of the Summer Computer Simulation Conference,* 340-345. The Society for Computer Simulation, San Diego, California. Retrieved from https://www.researchgate.net/publication/2788312_Evolving_Neural_Networks_With_Collaborative_Species

94. Potter, M. A., & de Jong, K. A. (2000). Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evolutionary Computation, 8*(1), 1-29. https://doi.org/10.1162/106365600568086

95. Pudil, P., Novovicova, J., & Kittler, J. (1994). Floating Search Methods in Feature-Selection. *Pattern Recognition Letters, 15*(11), 1119-1125. https://doi.org/10.1016/0167-8655(94)90127-9

96. Ramírez-Gallego, S., Mouriño-Talín, H., Martínez-Rego, D., Bolón-Canedo, V., Benítez, J. M., Alonso-Betanzos, A., & Herrera, F. (2018). An Information Theory-Based Feature Selection Framework for Big Data Under Apache Spark. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48*(9), 1441-1453. https://doi.org/10.1109/TSMC.2017.2670926

97. Reggiani, C., Le Borgne, Y. A., & Bontempi, G. (2018). Feature Selection in High-Dimensional Dataset Using MapReduce. Paper presented at the *Benelux Conference on Artificial Intelligence* (pp. 101-115). Cham: Springer. https://doi.org/10.1007/978-3-319-76892-2_8

98. Rentsen, E., Zhou, J., & Teo, K. L. (2016). A global optimization approach to fractional optimal control. *Journal of Industrial and Management Optimization (JIMO), 12*(1), 73-82. https://doi.org/10.3934/jimo.2016.12.73

99. Sakinah, S., & Ahmad, S. (2014). Feature and Instances Selection for Nearest Neighbor Classification via Cooperative PSO. *2014 4th World Congress on Information and Communication Technologies (WICT)*, 45-50. https://doi.org/10.1109/WICT.2014.7077300

100. Shelke, K., Jayaraman, S., Ghosh, S., & Valadi, J. (2013). Hybrid Feature Selection and Peptide Binding Affinity Prediction using an EDA based Algorithm. *2013 Ieee Congress on Evolutionary Computation (Cec),* 2384-2389. https://doi.org/10.1109/CEC.2013.6557854

101. Shen, Q., Shi, W., & Kong, W. (2008). Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data. *Computational Biology and Chemistry, 32*(1), 53-60. https://doi.org/10.1016/j.compbiolchem.2007.10.001

102. Shi, M., & Gao, S. (2017). Reference sharing: a new collaboration model for cooperative coevolution. *Journal of Heuristics, 23*(1), 1-30. https://doi.org/10.1007/s10732-016-9322-9

103. Shi, Y. J., Li, R., & Teo, K. L. (2015). Cooperative enclosing control for multiple moving targets by a group of agents. *International Journal of Control, 88*(1), 80-89. https://doi.org/10.1080/00207179.2014.938447

104. Shim, K. (2012). MapReduce algorithms for big data analysis. Proceedings *of the Vldb Endowment, 5*(12), 2016-2017. https://doi.org/10.14778/2367502.2367563

105. Singh, S., Kubica, J., Larsen, S., & Sorokina, D. (2009). Parallel Large Scale Feature Selection for Logistic Regression. *Proceedings of the 2009 SIAM International Conference on Data Mining* (pp. 1172-1183). https://doi.org/10.1137/1.9781611972795.100

106. Sinha, A., & Jana, P. K. (2018). A hybrid MapReduce-based k-means clustering using genetic algorithm for distributed datasets. *Journal of Supercomputing, 74*(4), 1562-1579. https://doi.org/10.1007/s11227-017-2182-8

107. Sofge, D., De Jong, K., & Schultz, A. (2002). A blended population approach to cooperative coevolution for decomposition of complex problems. *CEC'02: Proceedings of the 2002 Congress on Evolutionary Computation, 1-2,* 413-418. https://doi.org/10.1109/CEC.2002.1006270

108. Somorjai, R. L., Dolenko, B., & Baumgartner, R. (2003). Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics, 19*(12), 1484-1491. https://doi.org/10.1093/bioinformatics/btg182

109. Song, A., Yang, Q., Chen, W. N., & Zhang, J. (2016). A Random-Based Dynamic Grouping Strategy for Large Scale Multi-objective Optimization. *2016 IEEE Congress on Evolutionary Computation (CEC),* 468-475. https://doi.org/10.1109/CEC.2016.7743831

110. Soufan, O., Kleftogiannis, D., Kalnis, P., & Bajic, V. B. (2015). DWFS: A Wrapper Feature Selection Tool Based on a Parallel Genetic Algorithm. *Plos One, 10*(2). https://doi.org/10.1371/journal.pone.0117988

111. Stanovov, V., Brester, C., Kolehmainen, M., & Semenkina, O. (2017). Why don't you use Evolutionary Algorithms in Big Data? *IOP Conference Series: Materials Science and Engineering, 173*(1), 012020. https://doi.org/10.1088/1757-899X/173/1/012020

112. Stoeckel, J., & Fung, G. (2005). SVM Feature Selection for Classification of SPECT Images of Alzheimer's Disease Using Spatial Information. *Proceedings of the Fifth IEEE International Conference on Data Mining.* https://doi.org/10.1109/ICDM.2005.141

113. Storn, R., & Price, K. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341-359. https://doi.org/10.1023/A:1008202821328

114. Strearns, S. D. (1976). On selecting features for pattern classifiers. *Proceedings of the International Conference on Pattern Recognition (ICPR),* 71-75.

115. Sun, Y., Kirley, M., & Halgamuge, S. K. (2015). Extended differential grouping for large scale global optimization with direct and indirect variable interactions. *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation.* Retrieved from https://yuansuny.github.io/files/Cpaper_XDG.pdf

116. Sun, Y., Kirley, M., & Halgamuge, S. K. (2018). A Recursive Decomposition Method for Large Scale Continuous Optimization. *Ieee Transactions on Evolutionary Computation, 22*(5), 647-661. https://doi.org/10.1109/TEVC.2017.2778089

117. Sun, Y., Omidvar, M. N., Kirley, M., & Li, X. (2018). Adaptive threshold parameter estimation with recursive differential grouping for problem decomposition. *Proceedings of the Genetic and Evolutionary Computation Conference,* 889-896. Kyoto, Japan. https://doi.org/10.1145/3205455.3205483

118. Sun, Z. (2014). Parallel Feature Selection Based on MapReduce. Paper presented at the *Computer Engineering and Networking* (pp. 299-306). Cham. https://doi.org/10.1007/978-3-319-01766-2_35

119. Tan, M. K., Tsang, I. W., & Wang, L. (2013). Minimax Sparse Logistic Regression for Very High-Dimensional Feature Selection. *Ieee Transactions on Neural Networks and Learning Systems, 24*(10), 1609-1622. https://doi.org/10.1109/Tnnls.2013.2263427

120. Tan, N. C., Fisher, W. G., Rosenblatt, K. P., & Garner, H. R. (2009). Application of multiple statistical tests to enhance mass spectrometry-based biomarker discovery. *BMC Bioinformatics, 10*(1), 144. https://doi.org/10.1186/1471-2105-10-144

121. Taylor, R. C. (2010). An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. *BMC Bioinformatics, 11*(S1). https://doi.org/10.1186/1471-2105-11-S12-S1

122. Tian, J., Li, M., & Chen, F. (2010). Dual-population based coevolutionary algorithm for designing RBFNN with feature selection. *Expert Systems with Applications, 37*(10), 6904-6918. https://doi.org/10.1016/j.eswa.2010.03.031

123. Triguero, I., Peralta, D., Bacardit, J., García, S., & Herrera, F. (2015). MRPR: A MapReduce solution for prototype reduction in big data classification. *Neurocomputing, 150,* 331-345. https://doi.org/10.1016/j.neucom.2014.04.078

124. Tseng, M.-L., Wu, K.-J., Lim, M. K., & Wong, W.-P. (2019). Data-driven sustainable supply chain management performance: A hierarchical structure assessment under uncertainties. *Journal of Cleaner Production, 227,* 760-771. https://doi.org/10.1016/j.jclepro.2019.04.201

http://dx.doi.org/10.21511/ppm.17(4).2019.28

125. Tursunbayeva, A., Bunduchi, R., Franco, M., & Pagliari, C. (2016). Human resource information systems in health care: a systematic evidence review. *Journal of the American Medical Informatics Association, 24*(3), 633-654. https://doi.org/10.1093/jamia/ocw141

126. Vatolkin, I., Theimer, W., & Rudolph, G. (2009). Design and Comparison of Different Evolution Strategies for Feature Selection and Consolidation in Music Classification. *2009 IEEE Congress on Evolutionary Computation, 1-5,* 174-181. https://doi.org/10.1109/Cec.2009.4982945

127. Vieira, S. M., Sousa, J. M. C., & Runkler, T. A. (2010). Two cooperative ant colonies for feature selection using fuzzy models. *Expert Systems with Applications, 37*(4), 2714-2723. https://doi.org/10.1016/j.eswa.2009.08.026

128. Voyer, J., Dean, M. D., Pickles, C. B., & Robar, C. R. (2018). Leveraging System Dynamics Modeling to Help Understand Humanitarian Food Supply During Disaster Response. *Journal of Strategic Innovation & Sustainability, 13*(4), 52-70. https://doi.org/10.33423/jsis.v13i4.92

129. Wang, K. J., Chen, K. H., & Angelia, M. A. (2014). An improved artificial immune recognition system with the opposite sign test for feature selection. *Knowledge-Based Systems, 71,* 126-145. https://doi.org/10.1016/j.knosys.2014.07.013

130. Wang, S., Pedrycz, W., Zhu, Q., & Zhu, W. (2015). Subspace learning for unsupervised feature selection via matrix factorization. *Pattern Recognition, 48*(1), 10-19. https://doi.org/10.1016/j.patcog.2014.08.004

131. Whitney, A. W. (1971). A direct method of nonparametric measurement selection. *Ieee Transactions on Computers, 100*(9), 1100-1103. https://doi.org/10.1109/T-C.1971.223410

132. Wiegand, R. P. (2004). *An analysis of cooperative coevolutionary algorithms.* George Mason University. Retrieved from http://www.tesseract.org/paul/papers/rpw-dissertation-univ.pdf

133. Wu, M., Liu, K., & Yang, H. (2018). Supply chain production and delivery scheduling based on data mining. *Cluster Computing.* https://doi.org/10.1007/s10586-018-1894-8

134. Xue, B., Zhang, M. J., Browne, W. N., & Yao, X. (2016). A Survey on Evolutionary Computation Approaches to Feature Selection. *Ieee Transactions on Evolutionary Computation, 20*(4), 606-626. https://doi.org/10.1109/Tevc.2015.2504420

135. Yamada, M., Tang, J. L., Lugo-Martinez, J., Hodzic, E., Shrestha, R., Saha, A.,…, & Chang, Y. (2018). Ultra High-Dimensional Nonlinear Feature Selection for Big Biological Data. *Ieee Transactions on Knowledge and Data Engineering, 30*(7), 1352-1365. https://doi.org/10.1109/Tkde.2018.2789451

136. Yang, Z. Y., Tang, K., & Yao, X. (2008a). Large scale evolutionary optimization using cooperative coevolution. *Information Sciences, 178*(15), 2985-2999. https://doi.org/10.1016/j.ins.2008.02.017

137. Yang, Z. Y., Tang, K., & Yao, X. (2008b). Multilevel Cooperative Coevolution for Large Scale Optimization. *2008 IEEE Congress on Evolutionary Computation, 1-8*, 1663-1670. https://doi.org/10.1109/Cec.2008.4631014

138. Yang, Z. Y., Tang, K., & Yao, X. (2008c). Self-adaptive differential evolution with neighborhood search. *Proceedings of the IEEE Congress on Evolutionary Computatio (CEC 2008).* Hong Kong, China. Retrieved from https://www.researchgate.net/publication/221007846_Self-adaptive_Differential_Evolution_with_Neighborhood_Search

139. Yang, Z., Yao, X., & He, J. (2008). Making a Difference to Differential Evolution. In P. Siarry & Z. Michalewicz (Eds.), *Advances in Metaheuristics for Hard Optimization* (pp. 397-414). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-72960-0_19

140. Yee, Y. M., Tan, C. L., & Ramayah, T. (2017). Connect the Silos: Knowledge Management, Absorptive Capacity, Leadership Styles, Organisational Cultures. Paper presented at the *International Conference on Intellectual Capital and Knowledge Management and Organisational Learning* (pp. 310-315). Retrieved from http://toc.proceedings.com/37737webtoc.pdf

141. Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: cluster computing with working sets. *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing.* Boston, MA. Retrieved from https://www.usenix.org/legacy/events/hotcloud10/tech/full_papers/Zaharia.pdf

142. Zhao, X. H., Li, D. L., Yang, B., Ma, C., Zhu, Y. G., & Chen, H. L. (2014). Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton. *Applied Soft Computing, 24,* 585-596. https://doi.org/10.1016/j.asoc.2014.07.024

143. Zhou, Z., Chawla, N. V., Jin, Y., & Williams, G. J. (2014). Big data opportunities and challenges: Discussions from data analytics perspectives. *Ieee Computational Intelligence Magazine, 9*(4), 62-74. Retrieved from https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/cim14.pdf