Research Article

Parallel Attribute Reduction Algorithm for Complex Heterogeneous Data Using MapReduce

Tengfei Zhang ^{1,2}, Fumin Ma ³, Jie Cao,³ Chen Peng,⁴ and Dong Yue ^{1,2}

¹College of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

²Jiangsu Engineering Lab. of Big Data Analysis & Control for Active Distribution Network, Nanjing 210023, China

³College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210023, China

⁴School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200072, China

Correspondence should be addressed to Tengfei Zhang; tfzhang@126.com and Dong Yue; medongy@vip.163.com

Received 20 May 2018; Accepted 20 August 2018; Published 27 September 2018

Academic Editor: Liang Hu

Copyright © 2018 Tengfei Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Parallel attribute reduction is one of the most important topics in current research on rough set theory. Although some parallel algorithms were well documented, most of them are still faced with some challenges for effectively dealing with the complex heterogeneous data including categorical and numerical attributes. Aiming at this problem, a novel attribute reduction algorithm based on neighborhood multigranulation rough sets was developed to process the massive heterogeneous data in the parallel way. The MapReduce-based parallelization method for attribute reduction was proposed in the framework of neighborhood multigranulation rough sets. To improve the reduction efficiency, the hashing Map/Reduce functions were designed to speed up the positive region calculation. Thereafter, a quick parallel attribute reduction algorithm using MapReduce was developed. The effectiveness and superiority of this parallel algorithm were demonstrated by theoretical analysis and comparison experiments.

1. Introduction

With the rapid development of information technology, especially in the aspects of sensing, communication, network, and calculation, the amount of accumulated data in many fields is increasing at striking speeds. The inestimable value in big data has become a common understanding in academia and industry [1] and has garnered great attention in many countries; thus, big data technology development was announced as a national strategy by many countries [2]. However, most of the vast array of data that comes to us may be chaotic, irrelevant, and redundant. How to extract and express such implicit information in the form of explicit knowledge hidden in the given complex information systems has been an active area of research in the past number of decades. In practice, rough sets [3] have been widely used as a mathematical tool to deal with uncertain data. As one of the core research contents of rough set theory, attribute reduction can remove redundant attributes and reduce data dimensions under the premise of stable dependencies

between decision attributes and conditional attributes in a decision table. Scholars have designed a large number of attribute reduction algorithms in recent years, which can generally be divided into three methods [4]: positive region [5–7], discernible matrix [8–11], and information entropy [12–15]. However, the execution of these algorithms is a typical serial operation. Although these kinds of algorithms with serial operation are possible to handle small data efficiently, their computational complexity, which depends on attribute number $|C|$ and sample size $|U|$, may inevitably lead to lower efficiency and/or even complete failure when facing massive data.

In order to solve this problem, some scholars have proposed parallel algorithms for high-dimensional or large-scale data. Based on the divide and conquer strategy, Xiao et al. [16] used parallel computing to divide the reduction task into multiple processors that process simultaneously. Rough entropy was used to measure attribute significance by Lv et al. [17], and a parallel minimum reduction set algorithm was proposed. However, the dataset should be loaded

at once into the memory when implementing these algorithms. To make up for the defects of the aforementioned parallel algorithms, the Google File System- (GFS-) based distributed file system and the MapReduce parallel programming model were utilized by Qian et al. [18, 19]. The decision table was divided into several subdecision tables, and thus, a large amount of data did not need to be loaded into a single memory bank when calculating attribute reduction. Moreover, the machines in the cluster can cooperate with each other to solve problems that the single machine could not address. Zhang et al. [20] proposed a parallel method for computing rough set approximations based on the MapReduce technique to deal with the massive data. The equivalence classes for each subdecision table were calculated parallelly in Map step; thereafter, these equivalence classes were combined in Reduce step if their information sets are the same. Qian et al. [21] further analyzed that the key to improving the reduction efficiency is the effective computation of equivalence classes and attribute significance. Consequently, a structure of $\langle \text{key}, \text{value} \rangle$ pair to speed up the computation of equivalence classes and attribute significance was designed and the traditional attribute reduction process was parallelized based on MapReduce mechanism. However, these abovementioned algorithms were based on the Pawlak's classic rough set model with an equivalence relation, which is only suitable for categorical data.

To break the limit of the equivalence relation, Lin [22] proposed the concept of the neighborhood model and adopted the neighborhood relation instead of the equivalent relation, which can directly deal with numerical data through the neighborhood granulation in the universe. The monotonic relation between the positive region and attribute set in the neighborhood rough set model was proved by Hu et al. [23, 24], and an attribute reduction algorithm with lower computational complexity, which is suitable for heterogeneous data including categorical and numerical attributes, was put forward. Qian et al. [25] extended Pawlak's rough set model to a multigranulation rough set (MGRS) model, where the set approximations are defined by using multiple equivalence relations on the universe. Based on the pessimistic multigranulation rough set model, Sang and Qian [26] analyzed a granular space selection under multiple granular spaces, defined the importance measure of the granular space, and designed a granular space reduction algorithm. Subsequently, Lin et al. [27] expanded the neighborhood rough set model to multiple granular spaces and proposed the concept of a neighborhood multigranulation rough set (NMGRS) model by constructing the universe through a hierarchical division of attribute set sequences. Furthermore, Yong et al. [28] hashed a dataset by dividing data into a series of hash buckets according to the Euclidean distance, which dramatically decreased the calculation time and reduced time complexity to $O(m|U|)$ for getting positive regions. A quick and efficient attribute reduction algorithm with the time complexity of $O(m^2|U|)$ was also given. In addition, Qian et al. discussed local rough set to deal with big data [29, 30]. However, these aforementioned algorithms based on different extended rough set models were still involved in the serial computation.

To the best of our knowledge, it is still a challenging task to perform parallel attribute reduction on complex and massive data. In particular, the existing algorithms could not effectively deal with the complex heterogeneous data, which include categorical and numerical attributes, in the parallel way from the multiple granular computing perspectives. Due to the rampant existence of heterogeneous datasets in real-life applications, it is therefore necessary to investigate effective parallel approaches to deal with this issue. For the purpose of parallelizing the traditional attribute reduction algorithm for complex heterogeneous data, the neighborhood multigranulation rough set model was considered in this paper, and the parallelization points of the hashing, positive region calculating, and boundary objects pruning are analyzed based on MapReduce mechanism. Thereafter, a fast parallel attribute reduction algorithm is developed. The effectiveness and superiority of this parallel algorithm were demonstrated by theoretical analysis and comparison experiments.

Different from available algorithms, the contribution of this paper is twofold. (1) Motivated by the aforementioned MapReduce technology, hash algorithm, and neighborhood multigranulation rough set model, the parallelization methods of multiple granular spaces and hashing Map/Reduce functions for heterogeneous data are brought to light; (2) a neighborhood multigranulation rough set model-based parallel attribute reduction algorithm using MapReduce, which has never been done before, is proposed.

The paper is organized as follows. Section 2 outlines some preliminary knowledge. In Section 3, we present the parallelization strategies of multiple granular spaces for heterogeneous data and the parallel fast attribute reduction algorithm based on the neighborhood multigranulation rough set model. Next, experiments are conducted to evaluate the efficiency of the proposed algorithm in Section 4. Finally, in Section 5, we present the conclusion and the future work.

2. Preliminary Knowledge

In this section, 1-type and 2-type neighborhood multigranulation rough sets and MapReduce programming model will be briefly described.

2.1. Neighborhood Multigranulation Rough Set. Neighborhood rough set model uses neighborhood relation to replace equivalence relation, which can directly process numerical data and heterogeneous data. For further processing heterogeneous data from the perspective of multiple granular spaces and multiple levels of granularity, neighborhood rough set theory has been extended from single attribute subset to multiple attribute subsets. Two types of neighborhood multigranulation rough set models have been developed [27].

2.1.1. 1-Type Neighborhood Multigranulation Rough Sets (1-Type NMGRS)

Definition 1 (see [25]). Let $\langle U, \Delta \rangle$ be a nonempty metric space; $U = \{x_1, x_2, \dots, x_n\}$ is a nonempty finite set of objects, called the universe. A closed ball taking $x \in U$ as its center

and δ as its radius is called the δ neighborhood of x and is defined as follows:

$$n(x) = \{y \in U \mid \Delta(x, y) \leq \delta\}, \quad (1)$$

where $\delta \geq 0$, and Δ is a distance function. For two points in the universe, $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ and $x_j = \{x_{j1}, x_{j2}, \dots, x_{jn}\}$, the distance function can usually use the Euclidean distance formula.

$$\Delta(x_i, x_j) = \sqrt{\sum_{l=1}^n (x_{il} - x_{jl})^2}. \quad (2)$$

Definition 2 (see [25]). Let $\langle U, \Delta \rangle$ be a nonempty metric space. When categorical and numerical attributes coexist, let B_1 and B_2 be categorical and numerical attributes, respectively. The neighborhood of x can be defined as follows:

$$\begin{aligned} n_{B_1}(x) &= \{x_i \in U \mid \Delta_{B_1}(x, x_i) = 0\}, \\ n_{B_2}(x) &= \{x_i \in U \mid \Delta_{B_2}(x, x_i) \leq \delta\}, \\ n_{B_1 \cup B_2}(x) &= \{x_i \in U \mid (\Delta_{B_1}(x, x_i) = 0) \wedge (\Delta_{B_2}(x, x_i) \leq \delta)\}. \end{aligned} \quad (3)$$

Definition 3 (see [25]). Given a decision system, $NIS = (U, C \cup D, V, f, N)$, where $C = \{c_1, c_2, c_3, \dots, c_m\}$ is the set of condition attributes and D is the set of decision attributes, and $C \cap D = \emptyset$. $V = \cup_{a \in R} V_a$ is a set of attribute values and V_a is a domain of the attribute a . $f : U \times R \rightarrow V$ is a function such that $f(x, a) \in V_a$ for every $a \in C$ and $x \in U$. N is the neighborhood relation. Let $A \subseteq C$ be a categorical attribute set and $B \subseteq C$ be a numerical attribute set, so $A \cup B \subseteq C$ is an mixed attribute set; U/A , U/B , and $U/(A \cup B)$ represent two partitions and a covering of the universe U , respectively. For any $X \subseteq U$, the optimistic multigranulation lower and upper approximations of X with respect to A and B in U are defined as follows:

$$\begin{aligned} \underline{(A+B)}^O(X) &= \{x \in U \mid n_A(x) \subseteq X \vee n_B(x) \subseteq X\}, \\ \overline{(A+B)}^O(X) &= \sim \underline{(A+B)}^O(\sim X), \end{aligned} \quad (4)$$

whereas the pessimistic multigranulation lower and upper approximations of X are defined as follows:

$$\begin{aligned} \underline{(A+B)}^P(X) &= \{x \in U \mid n_A(x) \subseteq X \wedge n_B(x) \subseteq X\}, \\ \overline{(A+B)}^P(X) &= \sim \underline{(A+B)}^P(\sim X). \end{aligned} \quad (5)$$

2.1.2. 2-Type Neighborhood Multigranulation Rough Sets (2-Type NMGRS). Comparing with that just the single neighborhood relation was used in the 1-type neighborhood multigranulation rough sets, multiple neighborhood relations were fully considered in the 2-type neighborhood multigranulation rough sets, which were denoted by 2-type NMGRS by Qian et al. [25].

Definition 4 (see [27]). Given a decision system $NIS = (U, C \cup D, V, f, N)$, let $N = \{n_1, n_2, \dots, n_m\}$ be a neighborhood m -relation on the universe U induced by A_i and B_i , where A_i is a categorical attribute subset, and B_i is a numerical attribute subset. For any $X \subseteq U$, the optimistic lower and upper approximations of X in U are defined as follows:

$$\begin{aligned} \sum_{i=1}^m \overline{N_i}^O(X) &= \left\{x \in U \mid \left(n_{(A_1+B_1)}(x) \subseteq X\right) \vee \dots \vee \right. \\ &\quad \left. \left(n_{(A_i+B_i)}(x) \subseteq X\right) \vee \dots \vee \left(n_{(A_m+B_m)}(x) \subseteq X\right)\right\}, \\ \sum_{i=1}^m \underline{N_i}^O(X) &= \sim \sum_{i=1}^m \overline{N_i}^O(\sim X), \end{aligned} \quad (6)$$

whereas the pessimistic multigranulation lower and upper approximations of X are defined as follows:

$$\begin{aligned} \sum_{i=1}^m \overline{N_i}^P(X) &= \left\{x \in U \mid \left(n_{(A_1+B_1)}(x) \subseteq X\right) \wedge \dots \wedge \right. \\ &\quad \left. \left(n_{(A_i+B_i)}(x) \subseteq X\right) \wedge \dots \wedge \left(n_{(A_m+B_m)}(x) \subseteq X\right)\right\}, \\ \sum_{i=1}^m \underline{N_i}^P(X) &= \sim \sum_{i=1}^m \overline{N_i}^P(\sim X). \end{aligned} \quad (7)$$

Definition 5 (see [31]). Given a decision system $NIS = (U, C \cup D, V, f, N)$, let $B = \{B_1, B_2, \dots, B_m\}$ be an attribute subset of C , $Y = \{Y_1, Y_2, \dots, Y_r\}$ is a partition of universe U induced by decision attribute D , and $\delta = \{\delta_1, \delta_2, \dots, \delta_n\}$ is a set of n neighborhood radii. The attribute dependency of B_i for decision class Y with the neighborhood radius δ_j is defined as follows:

$$\gamma_{B_i}^{\delta_j}(Y) = \frac{|\text{POS}_{B_i}^{\delta_j}(Y)|}{|U|} = \frac{|\cup \underline{N}_{B_i}^{\delta_j}(Y_k)|}{|U|}. \quad (8)$$

Definition 6 (see [31]). Given a decision system $NIS = (U, C \cup D, V, f, N)$, let $B = \{B_1, B_2, \dots, B_m\}$ be an attribute subset of C , $Y = \{Y_1, Y_2, \dots, Y_r\}$ is a partition of universe U induced by decision attribute D , and $\delta = \{\delta_1, \delta_2, \dots, \delta_n\}$ is a set of n neighborhood radii. $\forall a \in B_i$, if $\gamma_{(B_i-\{a\})}^{\delta_j}(Y) < \gamma_{B_i}^{\delta_j}(Y)$, then the attribute a is necessary for B_i ; else, if $\gamma_{(B_i-\{a\})}^{\delta_j}(Y) = \gamma_{B_i}^{\delta_j}(Y)$, then regardless of whether the attribute a is removed from B_i , the decision positive region of the system is unchanged; in other words, the attribute a is redundant for B_i .

Given the attribute subset $R \subseteq B_i$, $\forall a \in R$, if $\gamma_{(R-\{a\})}^{\delta_j}(Y) < \gamma_R^{\delta_j}(Y)$, and $\gamma_R^{\delta_j}(Y) = \gamma_{B_i}^{\delta_j}(Y)$, then R is a relative reduction of B_i with the neighborhood radius δ_j .

Definition 7 (see [19]). Given a decision system $NIS = (U, C \cup D, V, f, N)$, let $S = \bigcup_{i=1}^m S_i = \bigcup_{i=1}^m (U_i, C \cup D, V, f, N)$, $\forall j$, $k \in \{1, 2, \dots, m\}$ and $j \neq k$, if $U = \bigcup_{i=1}^m U_i$ and $U_j \cap U_k = \emptyset$, the decision system NIS can be divided into m -subdecision systems, and S_i is called the subsystem of S .

2.2. MapReduce Programming Model. MapReduce is a parallel processing framework that breaks down large tasks into many small tasks. With the small tasks independent of each other, big tasks and small tasks are just different in size. The MapReduce parallel programming model also breaks down the computational process into two main stages: the Map stage and the Reduce stage.

In the MapReduce model, the whole dataset is split into many splits in natural sequence and then is passed to the Map stage. Data in the MapReduce programming model can be represented as $\langle \text{key}, \text{value} \rangle$ pairs. The Map function takes pairs $\langle K_1, V_1 \rangle$ as input and generates a set of intermediate $\langle K_2, V_2 \rangle$ pairs. The Reduce function groups together all intermediate values V_2 associated with the same K_2 and then merges together a set of values for each K_2 to form a possibly smaller set of values and finally outputs $\langle K_3, V_3 \rangle$ pairs. The Map and Reduce functions are given as follows:

Map: $\langle K_1, V_1 \rangle \rightarrow \langle K_2, V_2 \rangle$.

Reduce: $\langle K_2, [V_2] \rangle \rightarrow \langle K_3, V_3 \rangle$.

Here, K_i and V_i ($i = 1, \dots, 3$) represent the user-defined data types; $[]$ is used to denote a list.

3. Parallel Attribute Reduction

Algorithm for NMGRS

Aiming at the numerical or the heterogeneous data, many attribute reduction algorithms based on neighborhood multigranulation rough sets have been developed. However, it is still a challenging task to parallelize these attribute reduction algorithms for massive heterogeneous data. Motivated by the works of Qian et al. [21] and Yong et al. [28], quick parallelization strategies to speed up the computation of neighborhood classes and positive regions are proposed, and a parallel attribute reduction algorithm is designed in this section.

3.1. Parallelization Strategies. To parallelize the attribute reduction algorithm based on the neighborhood multigranulation rough set model, the MapReduce model was adopted. Thus, it is the key point that how to design the Map and Reduce functions for quickly getting neighborhood classes and positive regions. The work of Yong et al. [28] demonstrated that the neighborhood of a sample can only exist in its adjacent hash buckets or its own hash bucket. Therefore, to find possible neighborhoods, it is only necessary to group the samples according to their hash values. So, in the Map function, the hash value of each sample could be firstly

TABLE 1: Decision table.

U	c_1	c_2	c_3	c_4	D
x_1	0.10	0.20	0.61	0.20	Yes
x_2	0.13	0.22	0.56	0.10	Yes
x_3	0.14	0.23	0.40	0.31	No
x_4	0.16	0.41	0.30	0.16	No

calculated, and then the hash values and sample IDs are output. In the Reduce function, the sample IDs in the hash bucket are merged according to the same hash value.

Thus, the Map and Reduce functions for hash buckets calculation are designed as follows.

Example 1. We take the decision table shown in Table 1 as an example to illustrate the calculation process of Algorithms 1 and 2, where the decision attribute is listed in the last column.

According to Definition 7, the decision information system was divided into $S_1 = (U_1, C \cup D, V, f, N)$ and $S_2 = (U_2, C \cup D, V, f, N)$, where $U_1 = \{x_1, x_2\}$, and $U_2 = \{x_3, x_4\}$. The neighborhood radius is given as 0.08, and the condition attribute subset is given as $B = \{c_1, c_2\}$.

The Map process:

The $\langle \text{KEY}_{HM}, \text{VALUE}_{HM} \rangle$ pairs that output from Map 1 are $\langle 0,1 \rangle$, and $\langle 1,2 \rangle$.

The $\langle \text{KEY}_{HM}, \text{VALUE}_{HM} \rangle$ pairs that output from Map 2 are $\langle 1,3 \rangle$, and $\langle 3,4 \rangle$.

The Reduce process:

The $\langle \text{KEY}_{HR}, \text{VALUE}_{HR} \rangle$ pair that outputs from Reduce 1 is $\langle 0, \{1\} \rangle$.

The $\langle \text{KEY}_{HR}, \text{VALUE}_{HR} \rangle$ pair that outputs from Reduce 2 is $\langle 1, \{2,3\} \rangle$.

The $\langle \text{KEY}_{HR}, \text{VALUE}_{HR} \rangle$ pair that outputs from Reduce 3 is $\langle 3, \{4\} \rangle$.

After Algorithms 1 and 2, samples were hashed into three hash buckets with hash values of 0, 1, and 3.

Next, we calculated the positive regions under the current subset. According to Definition 4, the process of neighborhood computation and positive region judgment based on the multigranulation neighborhood rough sets could be divided into two parts. First, calculate the neighborhood of x under one condition attribute subset and then take the intersection of the positive region set of multiple condition subsets (applicable to the pessimistic neighborhood multigranulation rough set model) or take the union of the positive region set (applicable to the optimistic neighborhood multigranulation rough set model).

As to the neighborhood calculation of a single condition attribute subset, according to the work in literature [26], whether the sample belongs to the positive region can be judged by a distance function after traversing the hash bucket where a neighborhood probably exists. The hash value of a sample was calculated by the Map function firstly, and then, possible hash buckets can be searched in the output files (named after the hash value) by Algorithm 2. Consequently, the sample can be judged whether it belongs to the positive region or not by scanning the found hash buckets of this

Input: condition attribute subset, C ; a data split S_i

Output: $\langle KEY_{HM}, VALUE_{HM} \rangle$ // let KEY_{HM} be the set of hash value of each sample, and $VALUE_{HM}$ be the set of sample ID

begin

$\langle KEY_{HM}, VALUE_{HM} \rangle = \emptyset$

for each $x_i \in U$ do

let $key = hash(x_i)$;

// $hash(x_i) = \lceil \Delta(x_0, x_i) / \delta \rceil = key$, where x_0 is a special sample in universe U , which is satisfied with $\forall_{a \in C, x_i \in U} f(x_0, a) = \min [f(x_i, a)]$, and f is the information function.

let $value$ = the ID of x_i

$\langle KEY_{HM}, VALUE_{HM} \rangle = \langle KEY_{HM}, VALUE_{HM} \rangle \cup \langle key, value \rangle$

end for

end

ALGORITHM 1: Hash-Map function.

Input: $\langle KEY_{HM}, VALUE_{HM} \rangle$

Output: $\langle KEY_{HR}, VALUE_{HR} \rangle$ // let KEY_{HR} be the set of different hash value key' , and $VALUE_{HR}$ be the set of sample IDs subset $value'$ with the same hash value key' .

begin

$\langle KEY_{HR}, VALUE_{HR} \rangle = \emptyset$

for $\langle key, value \rangle$ in $\langle KEY_{HM}, VALUE_{HM} \rangle$ do

if key is not appeared in $\langle KEY_{HR}, VALUE_{HR} \rangle$

$\langle key', value' \rangle = \langle key, value \rangle$

else

if $key = key'_k$

$\langle KEY_{HR}, VALUE_{HR} \rangle = \langle KEY_{HR}, VALUE_{HR} \rangle - \langle key', value' \rangle$

$value'_k = value'_k \cup value$ // combine samples with the same hash value, obtain the hash bucket

end if

end if

$\langle KEY_{HR}, VALUE_{HR} \rangle = \langle KEY_{HR}, VALUE_{HR} \rangle \cup \langle key', value' \rangle$

end for // output with multi-file; a file named after a hash value is a hash bucket

end

ALGORITHM 2: Hash-Reduce function.

sample according to the distance function and decision attribute, where key values of samples in the positive region were assigned by 1, while key values of samples in the boundary were assigned by 0. The positive region for the whole universe U could be obtained by combining each positive region of S_i in the Reduce function.

Map and Reduce functions for neighborhood calculation by a single condition attribute subset are designed as follows.

The positive region of the whole universe U under single attribute granularity can be obtained by this algorithm through the intersection or union of the single granularity positive region sequence. Thus, the significant attribute could be calculated and added to current reduction subsets.

Example 2. We continue using the decision table in Table 1 and the same conditions as in Example 1 to illustrate the operation process.

The Map process:

The $\langle KEY_M, VALUE_M \rangle$ pairs that output from Map 1 are $\langle 0, 1 \rangle$, and $\langle 0, 2 \rangle$.

The $\langle KEY_M, VALUE_M \rangle$ pairs that output from Map 2 are $\langle 0, 3 \rangle$, and $\langle 1, 4 \rangle$.

The Reduce process:

The $\langle KEY_R, VALUE_R \rangle$ pair that outputs from Reduce 1 is $\langle 0, \{1, 2, 3\} \rangle$.

The $\langle KEY_R, VALUE_R \rangle$ pair that outputs from Reduce 2 is $\langle 1, \{4\} \rangle$.

The above operation results show that the positive and boundary region of the current universe can be acquired by Algorithms 3 and 4, where the positive region includes $\{4\}$ and the boundary region includes $\{1, 2, 3\}$.

According to the monotonic proof of the work of Ma et al. [32] that the situation of one sample belonging to a certain positive region will not be changed when additional attributes are added. In other words, in this case, it is unnecessary that these samples are repeatedly calculated in the second Reduce stage. So the Reduce function could focus the boundary samples.

The Map and Reduce functions for updating positive regions are designed as follows.

Example 3. Here, we are taking Table 1 and the results of Example 2 to illustrate the operation process on parallel boundary set updating and forming a new decision table.

Input: Single condition attribute subset, C ; the hash bucket B ; and a data split S_i

Output: $\langle KEY_M, VALUE_M \rangle$
 // key^* represents whether the sample belongs to the positive region; let key^* of the sample in positive region be 1, and key^* of the sample in boundary be 0; $value^*$ represents all samples' ID that have the same key^*

```

begin
   $\langle KEY_M, VALUE_M \rangle = \emptyset$ 
  for each  $x_i \in U$  do
    let  $key^*_i = 0$  // assuming that this sample does not belong to the positive region under  $C$ 
    let  $value^*$  = the ID of  $x_i$  and given that  $x_i \in B_k$ 
    for each  $x_j \in B_{k-1} \cup B_k \cup B_{k+1}$  // traversing the hash bucket where a neighborhood probably exists
      if  $x_j$  is the neighborhood of  $x_i$ , but they have different decision attribute values
        let  $key^*_i = 1$ 
      else
        break
    end if
  end for
   $\langle KEY_M, VALUE_M \rangle = \langle KEY_M, VALUE_M \rangle \cup \langle key^*_i, value^*_i \rangle$ 
end for
end

```

ALGORITHM 3: Single condition attribute subset neighborhood-Map.

Input: $\langle KEY_M, VALUE_M \rangle$

Output: $\langle KEY_R, VALUE_R \rangle$
 // let KEY_R be the set of different key^* , and $VALUE_R$ be the set of sample IDs subset $value^*$ with the same key^* .

```

begin
   $\langle KEY_R, VALUE_R \rangle = \emptyset$ 
  for  $\langle key^*, value^* \rangle$  in  $\langle KEY_M, VALUE_M \rangle$  do
    if  $key^*$  is not appeared in  $\langle KEY_R, VALUE_R \rangle$ 
       $\langle key^*, value^* \rangle = \langle key^*, value^* \rangle$ 
    else
      if  $key^* = key^*_k$ 
         $\langle KEY_R, VALUE_R \rangle = \langle KEY_R, VALUE_R \rangle - \langle key^*_k, value^*_k \rangle$ 
         $value^*_k = value^*_k \cup value^*$  // combine samples with the same  $key^*$ 
      end if
    end if
     $\langle KEY_R, VALUE_R \rangle = \langle KEY_R, VALUE_R \rangle \cup \langle key^*, value^* \rangle$ 
  end for
end

```

ALGORITHM 4: Single condition attribute subset neighborhood-Reduce.

It can be seen from Example 2 that the current positive region set is $\{4\}$, and thus, this update should remove the sample with the ID of 4 from the decision table.

The Map process:

The $\langle KEY_{UM}, VALUE_{UM} \rangle$ pairs that output from Map 1 are $\langle 1, (1 \ 0.10 \ 0.20 \ 0.61 \ 0.20 \ \text{Yes}) \rangle$ and $\langle 1, (2 \ 0.13 \ 0.22 \ 0.56 \ 0.10 \ \text{Yes}) \rangle$.

The $\langle KEY_{UM}, VALUE_{UM} \rangle$ pair that outputs from Map 2 is $\langle 1, (0.14 \ 0.23 \ 0.40 \ 0.31 \ \text{No}) \rangle$.

The Reduce process:

The $\langle KEY_{UR}, VALUE_{UR} \rangle$ pairs that output from Reduce1 are $\langle 1 \ (0.10 \ 0.20 \ 0.61 \ 0.20 \ \text{Yes}) \rangle$, $\langle 2 \ (0.13 \ 0.22 \ 0.56 \ 0.10 \ \text{Yes}) \rangle$, and $\langle 0.14 \ (0.23 \ 0.40 \ 0.31 \ \text{No}) \rangle$.

After the above operation, the boundary region updating was finished, and the actual storage situation is shown as follows:

S_1 : 1 0.10 0.20 0.61 0.20 Yes

2 0.13 0.22 0.56 0.10 Yes

S_2 : 0.14 0.23 0.40 0.31 No

The above results show that the subsequent attribute reduction can be processed directly on the basis of the whole dataset without any extra splitting.

3.2. Parallel Attribute Reduction Algorithm. On the basis of parallel algorithms given in Section 3.1, a neighborhood multigranulation rough set-based parallel attribute reduction algorithm using MapReduce is presented. For

```

Input: a data split  $S_i$ 
Output:  $\langle KEY_{UM}, VALUE_{UM} \rangle$ 
//  $key^{\wedge}$  represents the sample does not belongs to the positive region;  $value^{\wedge}$  represents the sequence of the sample's
attribute value
begin
 $\langle KEY_{UM}, VALUE_{UM} \rangle = \emptyset$ 
for each  $x_i \in S_i$  do
  if  $x_i \notin POS$   $key^{\wedge}_i = 1, value^{\wedge}_i = x_i$ 
  else  $key^{\wedge}_i = 0, value^{\wedge}_i = x_i$ 
  end if
 $\langle KEY_{UM}, VALUE_{UM} \rangle = \langle KEY_{UM}, VALUE_{UM} \rangle \cup \langle key^{\wedge}_i, value^{\wedge}_i \rangle$ 
end for
end

```

ALGORITHM 5: U_REDUCE-Map.

TABLE 2: Datasets.

	Dataset	Record number	Attribute number	Class number
1	DS1	30,700,000	35	19
2	DS2	2,458,285	68	3
3	DS3	5,000,000	18	2
4	DS4	3,850,505	52	2
5	DS5	1,025,010	11	10
6	DS6	4,898,421	41	23

convenience, this algorithm is denoted as PARA_NMG in this paper.

3.3. Algorithm: Time Complexity Analysis. It is assumed that the neighborhood decision information system has $|U|$ -samples and m -condition attributes. The positive region calculation is still the key step for the proposed PARA_NMG algorithm. In step 2.1, the calculation method in literature [28] is used to calculate the positive region of each attribute set, the time complexity of which is $O(m|U|)$. As to step 2.3, suppose that there are l -attributes eventually selected, with each attribute is added into the reduction subset, $|U|/l$ -samples will convert from the boundary sample to the positive region (in term of probability). Therefore, the time complexity of serial calculation is $O(m|U|(l+1))$. Furthermore, the MapReduce model was used in the PARA_NMG algorithm to parallelize the attribute reduction algorithm; assuming that there are k -nodes, the time complexity of the algorithm is $O(m|U|(l+1)/k)$, which is superior to the time complexity of $\max(O(m^2|U|(\sum_{t=1}^m \ln c_t + \ln d)/k), O(m^2|U|/k))$ in literature [22] and time complexity of $O(|C|^2|U|^2/k)$ in literature [33].

4. Experiment Analysis

In this section, we conducted some numerical experiments to assess the efficiency of our proposed algorithm. The experiments were implemented on a PC cluster of nine nodes, where one was set as a master node and the rest were configured as slave nodes. Each node is equipped with Inter Core i5-2400M CPU (four cores in all, each 3.1 GHz), 4 GB RAM

TABLE 3: Reduction results of the two algorithms.

Dataset	PAAR_PR	PARA_NMG
DS1	18, 26, 29, 1, 22, 3, 6, 10, 7, 9, 7, 17	18, 26, 29, 1, 22, 3, 6, 10, 7, 9, 4, 16
DS2	53, 25, 50, 14, 66	32, 67, 53, 35, 14
DS3	/	1, 17, 6
DS4	/	7, 19, 57, 2, 61, 5
DS5	/	4, 8, 2, 10
DS6	12, 6, 5, 3, 1, 23, 33, 36, 2, 24, 40	12, 6, 5, 3, 1, 23, 33, 36, 8, 11, 29, 34, 39, 27

memory, and the software of Ubuntu 14.0, Hadoop 2.6.0, and Java 1.6.20. All algorithms were coded in Java.

To illustrate the efficiency of our proposed PARA_NMG algorithm, the representative parallel algorithm for reduction algorithm based on positive region, which was denoted as PAAR_PR, proposed in literature [21] was used for comparisons. The difference is that the PAAR_PR algorithm is based on the classical rough set model, while the PARA_NMG algorithm is based on the neighborhood multigranulation rough set model.

To test the efficiencies of above two algorithms on different types of data, the experiments were carried out with the real datasets *Soybean*, *US Census Data* (1990), *Susy*, *PAMAP2 Physical Activity Monitoring*, and *Poker Hand* from UCI Machine Learning Repository [34] and another dataset *KDD99*. Here, *Soybean* and *US Census Data* (1990) are categorical datasets, *Susy* and *PAMAP2 Physical Activity Monitoring* are numerical datasets, and *Poker Hand* and *KDD99* are heterogeneous datasets. To create a big data environment, the dataset *Soybean* was duplicated 100,000 times as a new dataset. For convenience, these above six datasets were denoted as DS1~DS6, respectively. The characteristics of these datasets are shown in Table 2.

4.1. Comparison and Analysis of Reduction Results. For neighborhood rough set model, it is important to select a proper neighborhood radius when calculating neighborhood classes. According to the work of Hu et al. [24], the reasonable neighborhood radius should be selected in the interval

```

Input:  $\langle KEY_{UM}, VALUE_{UM} \rangle$ 
Output:  $\langle KEY_{UR}, VALUE_{UR} \rangle$ 
// let  $KEY_{UR}$  be the sequence number in the case of  $key^{\wedge}=1$ , and  $VALUE_{UR}$  be the set of  $value^{\wedge}$  sequence subset.
begin
   $KEY_{UR}=0$  and  $VALUE_{UR}=\emptyset$ 
  if  $key_k^{\wedge}=1$ 
     $KEY_{UR}=KEY_{UR}++$ 
     $VALUE_{UR}=VALUE_{UR} \cup value_k^{\wedge}$ 
  end if
end

```

ALGORITHM 6: U_REDUCE-Reduce.

```

Input:  $NIS = (U, C \cup D, V, f, N), \beta$ 
Output: reduction  $reduct$ 
begin
  Step 1: initialize  $reduct \leftarrow \emptyset, Q \leftarrow U, Max\_POS = \emptyset$ ;
  Step 2: if  $(C-reduct) = \emptyset$ , go to Step 3; while if  $(C-reduct) \neq \emptyset$ , execute the following loop operations:
    Step 2.1: for each condition attribute  $c_k \in (C-reduct)$ , use algorithms 1~4 to calculate the positive region  $POS_k$  of  $reduct \cup \{c_k\}$ ;
    Step 2.2: compare the positive region  $POS_k$  of condition attribute subset  $reduct \cup \{c_k\}$  after each attribute  $c_k$  added. Find current
      maximum positive region  $Max\_POS$ ; if  $|Max\_POS| = \emptyset$  then keep attribute reduction invariant, and go to Step 3.
      Otherwise, add  $c_k$  into  $reduct$ ,  $reduct = reduct \cup \{c_k\}$ ;
    Step 2.3: update boundary sample set  $Q$  with algorithms 5~6,  $Q = Q - Max\_POS$ , and return to Step 2.1
  Step 3: output reduction  $reduct$ 
end.

```

ALGORITHM 7: PARA_NMG algorithm.

[0.1, 0.3). Qian et al. [29] analyzed the monotonicity of positive region with the neighborhood radius, and they found that the classification accuracy will be deduced with neighborhood radius increase. Considering these factors and characteristics of selected datasets, the neighborhood radius for our PARA_NMG algorithm was set to be 0.1 when facing numerical data. The reduction results of the PARA_NMG algorithm and the PAAR_PR algorithm are shown in Table 3.

It can be seen from Table 3 that the PAAR_PR algorithm obtained effective reduction results on categorical datasets DS1 and DS2. Notwithstanding that there are few numerical attributes in DS6, the equivalence classes could be obtained, so the PAAR_PR algorithm is still practicable. However, for numerical datasets DS3 and DS4 and heterogeneous dataset DS5, PAAR_PR could not get the reduction results because equivalent classes could not be obtained in these datasets. Thus, the applicability of PAAR_PR algorithm depends on the characteristics of datasets. Comparatively speaking, the PARA_NMG algorithm was not limited by data types when calculating attribute reduction on different datasets. Considering the rampant existence of heterogeneous datasets in real-life applications, the neighborhood multigranulation rough set-based PARA_NMG algorithm has better applicability.

In addition, for datasets DS1, DS2, and DS6, although the attribute reduction results were all obtained, there was a little difference between the selected attribute subsets by both algorithms. To further analyze the two algorithms' effects on reduction results from the perspective of classification

TABLE 4: Classification accuracies with two algorithms' reduction results.

Classifier	Dataset	PAAR_PR	PARA_NMG
SMO	DS1	82.1037	86.4802
	DS2	84.5962	87.6572
	DS6	91.5151	97.9569
Naive Bayes	DS1	75.4451	84.7014
	DS2	87.4395	93.1425
	DS6	67.0616	75.6389
NBM	DS1	57.4696	77.7349
	DS2	87.6570	89.6472
	DS6	97.1868	97.9569
Logistic	DS1	83.6540	88.0672
	DS2	87.7754	90.7764
	DS6	99.9873	99.9898
LWL	DS1	91.7377	96.5833
	DS2	70.9932	80.4851
	DS6	67.0616	67.0616
J48	DS1	90.5104	97.9103
	DS2	82	82
	DS6	97.1868	98.1967
MultiClassClassifier	DS1	81.6033	82.8716
	DS2	95.1595	98.5344
	DS6	96.1977	99.9781

TABLE 5: Average computational time of two algorithms (s).

Dataset	Algorithm	Number of nodes							
		1	2	3	4	5	6	7	8
DS1	PARA_NMG	14,958	8499	6420	4953	4132	3596	3317	2968
	PAAR_PR	1724	894	607	492	418	364	312	272
DS2	PARA_NMG	55,423	28,717	19,515	17,820	14,031	11,996	10,226	8854
	PAAR_PR	8732	4961	3731	2960	2591	2140	2003	1815
DS3	PARA_NMG	64,035	34,992	24,820	19,583	16,504	14,927	13,481	12,338
DS4	PARA_NMG	16,006	8559	6428	4694	3942	3572	3163	2663
DS5	PARA_NMG	13,226	6961	4863	4032	3157	2738	2458	2249
DS6	PARA_NMG	21,371	11,264	7714	6333	5304	4847	3961	3533
	PAAR_PR	2091	1384	1124	987	816	723	651	624

FIGURE 1: Average speedup on different number of nodes.

accuracy, seven well-known typical classifiers, namely, sequential minimal optimization (SMO), naive Bayes, naive Bayesian model (NBM), logistic regression model (LRM), locally weighted learning (LWL), J48, and MultiClassClassifier, were selected to further test the classification accuracy associated with different attribute reduction subsets. The test results are shown in Table 4.

We can see from Table 4 that classification accuracies, according to the reduction subsets of PARA_NMG algorithm, are better for most of these classifiers. In fact, the classification accuracy is the important factor that should be considered in real-life applications. So, from a practical point of view, our neighborhood multigranulation rough set-based PARA_NMG algorithm also has better applicability.

4.2. Comparative Analysis on Computational Time. To illustrate the influence of the number of computer nodes on the two algorithms' computational time, the experiments were implemented on a cluster with different number of nodes. The average running times of the two algorithms were recorded, which are shown in Table 5 as follows. For datasets DS3~DS5, only the results of PARA_NMG algorithm are given.

As can be seen from Table 5, PAAR_PR is faster than PARA_NMG because of the different rough set models were used. PAAR_PR is based on the classical rough set model, and the time complexity of the classical heuristic serial reduction algorithm based on the positive region is $O(m|U|^2)$. Conversely, PARA_NMG is based on the neighborhood multigranulation rough set model, and the time complexity of the classical serial reduction algorithm is $O(m^2|U|^2)$. To minimize the times of computation for getting positive regions, the hash function was introduced into the Map and Reduce stages for neighborhood multigranulation rough sets, and the time complexity of our parallel attribute reduction algorithm was reduced to $O(m|U|(l+1)/k)$. To some extent, the computational time of our algorithm is still comparable.

In fact, except for the computational time, the speedup is really an important performance index for evaluating the efficiency of a parallel algorithm, which is defined as follows:

$$\text{Speedup}(p) = \frac{T_1}{T_p}, \quad (9)$$

where p is the number of nodes, T_1 is the execution time at one node, and T_p is the execution time at p nodes.

The speedup of two algorithms was tested with different number of nodes. To be more intuitive, the average speedup of two algorithms on each dataset with different computer nodes is presented in Figure 1 as follows, where the x axis represents the number of computer nodes, the y axis represents the speedup, and the red star point denoted by liner represents the theoretical speedup of a parallel algorithm.

As shown in Figure 1, the parallel reduction algorithm proposed in this paper could achieve better speedup on different data types. With the number of nodes increase, the superiority of our PARA_NMG algorithm in speedup is

more and more obvious. Therefore, the PARA_NMG algorithm is more suitable for processing heterogeneous massive data parallelly on a large number of computing nodes.

5. Conclusion

Attribute reduction is one of the important research issues in rough set theory. In current big data era, traditional attribute reduction algorithms are now faced with big challenges for dealing with massive data. Most existing parallel algorithms have seldom taken granular computing into consideration, especially for dealing with complex heterogeneous data including categorical attributes and numerical attributes. To address these issues, aiming at heterogeneous data, a quick parallel attribute reduction algorithm using MapReduce in the framework of neighborhood multigranulation rough sets was developed in this paper. The hash function was introduced into the Map and Reduce stages to speed up the positive region calculation. The effectiveness and superiority of the developed algorithm were verified by comparison analysis.

However, just the static data was considered in this paper; in fact, datasets in real-world applications often vary dynamically over time. How to parallelize the incremental attribute reduction algorithm in the framework of neighborhood multigranulation rough sets is a focus for future research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61833011, 61403184, and 61533010), the major program of the Natural Science Foundation of Jiangsu Province Education Commission, China (17KJA120001), the National Key Research and Development Program of China (2017YFD0401001), and the Six Talent Peaks Project in Jiangsu Province, China (XNY-038).

References

- [1] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, pp. 146–157, 2018.
- [2] "Notice of the state council on the issuance of a platform for the development of large data," http://www.gov.cn/zhengce/content/2015G09/05/content_10137.htm.
- [3] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.
- [4] J. Qian, D. Q. Miao, Z. H. Zhang, and W. Li, "Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation," *International Journal of Approximate Reasoning*, vol. 52, no. 2, pp. 212–230, 2011.

- [5] T. F. Zhang, J. M. Xiao, and X. H. Wang, "Algorithms of attribute relative reduction in rough set theory," *Acta Electronica Sinica*, vol. 33, no. 11, pp. 2080–2083, 2005.
- [6] Z. Y. Xu, Z. P. Liu, B. R. Yang, and W. Song, "A quick attribute reduction algorithm with complexity of $\max(O(|C||U|), O(|C|^2|U/C|))$," *Chinese Journal of Computers*, vol. 29, no. 3, pp. 391–399, 2006.
- [7] F. Jing, J. Yunliang, and L. Yong, "Quick attribute reduction with generalized indiscernibility models," *Information Sciences*, vol. 397–398, pp. 15–36, 2017.
- [8] Y. Yao and Y. Zhao, "Discernibility matrix simplification for constructing attribute reducts," *Information Sciences*, vol. 179, no. 7, pp. 867–882, 2009.
- [9] Y. Yao and Y. Zhao, "Attribute reduction in decision-theoretic rough set models," *Information Sciences*, vol. 178, no. 17, pp. 3356–3373, 2008.
- [10] M. Fumin and Z. Tengfei, "Generalized binary discernibility matrix for attribute reduction in incomplete information systems," *The Journal of China Universities of Posts and Telecommunications*, vol. 24, no. 4, pp. 57–75, 2017.
- [11] J. Konecny, "On attribute reduction in concept lattices: methods based on discernibility matrix are outperformed by basic clarification and reduction," *Information Sciences*, vol. 415–416, pp. 199–212, 2017.
- [12] G. Wang, "Rough reduction in algebra view and information view," *International Journal of Intelligent Systems*, vol. 18, no. 6, pp. 679–688, 2003.
- [13] D. Ye, Z. Chen, and S. Ma, "A novel and better fitness evaluation for rough set based minimum attribute reduction problem," *Information Sciences*, vol. 222, pp. 413–423, 2013.
- [14] L. Sun, J. Xu, and Y. Tian, "Feature selection using rough entropy-based uncertainty measures in incomplete decision systems," *Knowledge-Based Systems*, vol. 36, pp. 206–216, 2012.
- [15] Q. Zhang, P. Zhang, and G. Wang, "Research on approximation set of rough set based on fuzzy similarity," *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 3, pp. 2549–2562, 2017.
- [16] D. W. Xiao, G. Y. Wang, and F. Hu, "Fast parallel attribute reduction algorithm based on rough set theory," *Computer Science*, vol. 36, no. 3, pp. 208–211, 2009.
- [17] Y. J. Lv, X. N. Liu, and L. Chen, "Rough set attribute reduction algorithm based on PGA," *Computer Science*, vol. 35, no. 3, pp. 219–221, 2008.
- [18] J. Qian, D. Q. Miao, and Z. H. Zhang, "Knowledge reduction algorithms in cloud computing," *Chinese Journal of Computers*, vol. 34, no. 12, pp. 2332–2343, 2011.
- [19] J. Qian, P. Lv, X. Yue, C. Liu, and Z. Jing, "Hierarchical attribute reduction algorithms for big data using MapReduce," *Knowledge-Based Systems*, vol. 73, pp. 18–31, 2015.
- [20] J. Zhang, T. Li, D. Ruan, Z. Gao, and C. Zhao, "A parallel method for computing rough set approximations," *Information Sciences*, vol. 194, pp. 209–223, 2012.
- [21] J. Qian, D. Miao, Z. Zhang, and X. Yue, "Parallel attribute reduction algorithms using MapReduce," *Information Sciences*, vol. 279, pp. 671–690, 2014.
- [22] T. Y. Lin, "Granular computing on binary relations I: data mining and neighborhood systems," *Rough Sets in Knowledge Discovery*, vol. 1, pp. 107–121, 1998.
- [23] Q. H. Hu, H. Zhao, and D. R. Yu, "Efficient symbolic and numerical attribute reduction with neighborhood rough sets," *Pattern Recognition & Artificial Intelligence*, vol. 21, no. 6, pp. 732–738, 2008.
- [24] Q. Hu, D. Yu, J. Liu, and C. Wu, "Neighborhood rough set based heterogeneous feature subset selection," *Information Sciences*, vol. 178, no. 18, pp. 3577–3594, 2008.
- [25] Y. Qian, J. Liang, Y. Yao, and C. Dang, "MGRS: a multi-granulation rough set," *Information Sciences*, vol. 180, no. 6, pp. 949–970, 2010.
- [26] Y. L. Sang and Y. H. Qian, "A granular space reduction approach to pessimistic multi-granulation rough sets," *Pattern Recognition & Artificial Intelligence*, vol. 25, no. 3, pp. 361–366, 2012.
- [27] G. Lin, Y. Qian, and J. Li, "NMGRS: neighborhood-based multi-granulation rough sets," *International Journal of Approximate Reasoning*, vol. 53, no. 7, pp. 1080–1093, 2012.
- [28] L. Yong, H. Wenliang, J. Yunliang, and Z. Zhiyong, "Quick attribute reduct algorithm for neighborhood rough set model," *Information Sciences*, vol. 271, pp. 65–81, 2014.
- [29] Y. Qian, X. Liang, G. Lin, Q. Guo, and J. Liang, "Local multi-granulation decision-theoretic rough sets," *International Journal of Approximate Reasoning*, vol. 82, pp. 119–137, 2017.
- [30] Y. Qian, X. Liang, Q. Wang et al., "Local rough set: a solution to rough data analysis in big data," *International Journal of Approximate Reasoning*, vol. 97, pp. 38–63, 2018.
- [31] Y. Xu, H. J. Yang, and X. Ji, "Neighborhood multi-granulation rough set model based on double granulate criterion," *Control & Decision*, vol. 30, no. 8, pp. 1469–1478, 2015.
- [32] F. M. Ma, J. W. Chen, and T. F. Zhang, "Quick attribute reduction algorithm for neighborhood multi-granulation rough set based on double granulate criterion," *Control & Decision*, vol. 32, no. 6, pp. 1121–1127, 2017.
- [33] H. Chen, T. Li, Y. Cai, C. Luo, and H. Fujita, "Parallel attribute reduction in dominance-based neighborhood rough set," *Information Sciences*, vol. 373, pp. 351–368, 2016.
- [34] UCI Machine Learning Repository <http://archive.ics.uci.edu/ml/datasets.html>.

Submit your manuscripts at
www.hindawi.com