Knowledge-Based Systems xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Parallel computing of approximations in dominance-based rough sets approach

Shaoyong Li [a,b], Tianrui Li [a,c,*], Zhixue Zhang [b], Hongmei Chen [a,c], Junbo Zhang [a,c]

[a] School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China
[b] Naval Aviation Institute, PLA, Huludao 125001, China
[c] Sichuan Provincial Key Laboratory of Higher Education of Cloud Computing and Intelligent Technique, Chengdu 610031, China

ARTICLE INFO

Article history:
Received 2 November 2014
Received in revised form 16 March 2015
Accepted 1 May 2015
Available online xxxx

Keywords:
Rough sets
Approximation
Parallel computing
Dominance relation
Multi-core

ABSTRACT

Computation of approximation is a critical step for applying rough sets methodologies in knowledge discovery and data mining. As an extension of classic rough sets theory, Dominance-based Rough Sets Approach (DRSA) can process information with preference-ordered attribute domain and then can be applied in multi-criteria decision analysis and other related works. Efficiently computing approximations is helpful for reducing the time of making decisions based on DRSA. Parallel computing is an effective way to speed up the process of computation. In this paper, several strategies of decomposition and composition of granules in DRSA are proposed for computing approximations in parallel and the corresponding parallel algorithm is designed. A numerical example is employed to validate the feasibility of these strategies. The experimental evaluations on a multi-core environment showed that the parallel algorithm can obviously reduce the time of computing approximations in DRSA.

1. Introduction

Rough sets theory proposed by Pawlak is an effective mathematic tool to process uncertain, incomplete and inconsistent information, which has been applied in many fields related to data mining and knowledge discovery [13,15]. In order to meet the requirements of real applications, many scholars have modified or extended Classic Rough Sets Theory (CRST) and proposed some useful approaches such as Decision-Theoretic Rough Sets (DTRS) [24], Variable Precision Rough Sets (VPRS) [32], Tolerance-based Rough Sets (TRS) [6], Dominance-based Rough Sets Approach (DRSA) [4], Multi-Granulation Rough Sets (MGRS) [19], and Composite Rough Sets (CRS) [25]. The essence of rough sets methodologies is to approximate a rough set with two crisp sets. One of these two crisp sets is the maximal crisp set included by the rough set, called lower approximation and another is the minimal crisp set including the rough set, called upper approximation. Hence, computation of approximations in rough sets methodologies is a vital step for applying them in solving real-world problems.

People have various preference in their daily life, which is the main reason why information systems from the real-world often include some attributes with preference-ordered domains (in the decision analysis research, these attributes are called criteria). However, CRST cannot process information with preference-ordered domains effectively [14,16]. For this reason, Greco et al. proposed DRSA by replacing the indiscernibility relation in CRST with a dominance relation. The dominance relation is built based on those available information which has preference-ordered attribute domains. For example, (x_1, x_2) belonging to the dominance relation D_P means that the object x_1 is at least no worse than x_2 with respect to the attribute set P. Since DRSA has its advantage of processing information with preference-ordered domains, it has been applied to multi-criteria analysis widely.

Due to ubiquitous information-sensing mobile devices, remote sensing, software logs, cameras, microphones and wireless sensor networks as well as the increasing world's technological per-capita capacity to store information, data sets grow in size continuously. In research fields of data mining and knowledge discovery, many traditional approaches cannot work efficiently with processing big data sets. Hence, some exceptional approaches and technologies are required to efficiently process big data within tolerable elapsed times.

In rough sets theory, many scholars have attempted to employ some techniques and strategies to accelerate the process of

* Corresponding author at: School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China.
E-mail addresses: meterer@163.com (S. Li), trli@swjtu.edu.cn (T. Li), zzx7113@126.com (Z. Zhang), hmchen@swjtu.edu.cn (H. Chen), jbzhang@my.swjtu.edu.cn (J. Zhang).

http://dx.doi.org/10.1016/j.knosys.2015.05.003
0950-7051/

knowledge acquisition for different characteristics of big data. A lot of incremental approaches based on rough sets methodologies for dynamically maintaining knowledge are reported in recent years [2,3,5,8,10–12,18,20,21,23,26,27,29]. These incremental approaches use the incremental learning technique to avoid recomputing from scratch in order to reduce time of updating knowledge. Parallel programming is another way to speed up process of knowledge acquisition. Zhang et al. designed a parallel algorithm by using MapReduce for computing approximations of rough sets under the indiscernible relation [28]. Following that, they compared the parallel algorithms of computing approximations of rough sets on different MapReduce runtime systems [30]. Zhang et al. also proposed a parallel algorithm for computing approximations of CRS on GPU [31]. Li et al. proposed two parallel approaches for computing approximations in DRSA on multi-core and GPU platforms [7,9], respectively. Qian et al. proposed several parallel attribute reduction algorithms for improving the reduction efficiency [17].

In DRSA, there are two types of sets (upward and downward unions of decision classes) regarded as concepts approximated for a preference-ordered decision attribute. Using approximations to describe these concepts is the core of DRSA. Computation of approximations is a basic step for applying DRSA to mine useful knowledge from data with preference-ordered attribute domains. Hence, to accelerate computing approximations is benefit for improving the efficiency of data mining based on DRSA. We have studied the update of approximations in DRSA under dynamic data environment and proposed several incremental approaches in our previous works [8,10,11]. For massive data, it is well-known that parallel computing (programming) is an effective way to speed up the efficiency of computation. Parallel computing means that multiple process elements are used simultaneously to solve a problem. This is accomplished by partitioning the problem into independent parts so that each process element can carry out its part of algorithm concurrently with the others. The process elements are diverse, e.g., a single computer with multiple processors, several networked computers, special hardware, etc. [1]. With enlightenment of the information granule decomposition method introduced by Skowron and Stepaniuk [22], we investigated some strategies to decompose process of computing basic knowledge granules, concept granules and approximations in DRSA, which are based on the precondition that the information system is divided into many subsystems [9].

In this paper, we further study the parallel approach and the corresponding parallel algorithm for computing approximations in DRSA based on those strategies reported in [9]. This parallel approach is based on classification of information systems. The main idea is to decompose computation of approximations based on decomposition of computing basic knowledge granules and concept granules (upward and downward unions of decision classes), which ensures that each process element can do its part of work concurrently. Based on these interim results computed in parallel, we compose the final results (approximations of concept granules). According to these strategies, we design a parallel algorithm for computing approximations in this paper. A numerical example is employed to illustrate the feasibility of these parallel strategies. Experimental evaluations show that the parallel algorithm can effectively reduce computational time under multi-core environment.

The remainder of this paper is organized as follows. We present some basic notions of DRSA in Section 2. We introduce a parallel framework for computing approximations of DRSA in Section 3. In Section 4, a numerical example is employed to illustrate the feasibility of the parallel strategies. In Section 5, a parallel algorithm is developed and its performance is evaluated by some experiments. This paper ends with conclusions and further research topics in Section 6.

2. Preliminaries

In this section, we briefly review some concepts and notations of DRSA which have been introduced in [4,11].

An information system can be represented in the form of a data table, whose separate rows refer to distinct objects, and whose columns refer to attributes considered. Each cell of the table indicates an evaluation of the object placed in that row according to the attribute in the corresponding column.

A data table is a four-tuple $S = (U, A, V, f)$, where U is a non-empty finite set of objects, called the universe. $A = C \cup \{d\}$. C is a non-empty finite set of condition attributes, and d is a decision attribute. V is regarded as the domain of all attributes. $f : U \times A \to V$ is an information function such that $f(x,a) \in V_a,\ \forall a \in A$ and $x \in U$, where V_a is the domain of attribute a.

Usually, an attribute with preference-ordered domain is called as a criterion and an attribute without preference-ordered domain is called as a regular attribute. Here, each attribute in A can be regarded as a criterion.

$\forall a \in C$, there is a weak preference relation on U with respect to a, denoted by

$$\succeq_a = \{(x,y) \in U \times U | f(x,a) \geqslant f(y,a)\}.$$

$\forall x, y \in U, (x,y) \in \succeq_a$ means "x is at least as good as y with respect to a".

$\forall P \subseteq C$ and $P \neq \emptyset$, if $x \succeq_a y$ for all $a \in P$, it is said that x dominates y with respect to P, denoted by xD_Py, where D_P is a dominance relation on the universe U with respect to P,

$$D_P = \{(x,y) \in U \times U | f(x,a) \geqslant f(y,a), \forall a \in P\}.$$

$\forall x \in U$, there are two sets of it in DRSA as follows:

- A set of objects dominating x with respect to P, called P-dominating set, $D_P^+(x) = \{y \in U | yD_Px\}$.
- A set of objects dominated by x with respect to P, called P-dominated set, $D_P^-(x) = \{y \in U | xD_Py\}$.

P-dominating and P-dominated sets are called as basic knowledge granules in DRSA. In our previous work [11], we had defined a dominance matrix to represent the dominance relation on the universe. Following that, we redefine P-dominating and P-dominated sets.

In [11], a character variant

$$r_{i,j}^a = \begin{cases} 1, & f(x_i,a) \geqslant f(x_j,a) \\ 0, & f(x_i,a) < f(x_j,a) \end{cases} \quad (1)$$

was employed to indicate the weak preference relation between x_i and x_j with respect to a. Then a matrix

$$R^a = \left(r_{i,j}^a\right)_{|U|\times|U|} = \begin{pmatrix} r_{1,1}^a & \cdots & r_{1,|U|}^a \\ \vdots & \ddots & \vdots \\ r_{|U|,1}^a & \cdots & r_{|U|,|U|}^a \end{pmatrix} \quad (2)$$

was used to reflect the preference relation on U with respect to a.

For $P \subseteq C$, the dominance relation on the universe U with respect to P was presented by the matrix

$$R^P = \sum_{a\in P} R^a = \left(\phi_{i,j}^P\right)_{|U|\times|U|} = \begin{pmatrix} \phi_{1,1}^P & \cdots & \phi_{1,|U|}^P \\ \vdots & \ddots & \vdots \\ \phi_{|U|,1}^P & \cdots & \phi_{|U|,|U|}^P \end{pmatrix} \quad (3)$$

Please cite this article in press as: S. Li et al., Parallel computing of approximations in dominance-based rough sets approach, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.05.003

where $\phi_{i,j}^P = \sum_{a\in P} r_{i,j}^a$. If $\phi_{i,j}^P = |P|$, then x_i dominates x_j. Hence, P-dominating and P-dominated sets of the object $x_i \in U$ were written as follows, respectively.

$$D_P^+(x_i) = \left\{x_j \in U | \phi_{i,j}^P = |P|\right\} \tag{4}$$

$$D_P^-(x_i) = \left\{x_j \in U | \phi_{i,j}^P = |P|\right\} \tag{5}$$

U is divided by d into a family of decision classes with preference-ordered. Let $Cl = \{Cl_n, n \in T\}, T = \{1, \ldots, t\}$, be a collection of decision classes. $\forall r, s \in T$ such that $r > s$, the objects from Cl_r are preferred to the objects from Cl_s. The concepts to be approximated in DRSA are upward unions and downward unions of decision classes such that

$$Cl_n^{\geqslant} = \bigcup_{s\geqslant n} Cl_s, \quad Cl_n^{\leqslant} = \bigcup_{s\leqslant n} Cl_s, \quad \forall n, s \in T.$$

$x \in Cl_n^{\geqslant}$ means "x belongs to at least class Cl_n", and $x \in Cl_n^{\leqslant}$ means "x belongs to at most class Cl_n". Assumed that $d_1 < d_2 < \cdots < d_t$, upward unions and downward unions of decision classes can also be defined as follows:

$$Cl_n^{\geqslant} = \{x \in U | f(x,d) \geqslant d_n\}, \quad Cl_n^{\leqslant} = \{x \in U | f(x,d) \leqslant d_n\}, \quad \forall n \in T.$$

Notice that $Cl_1^{\geqslant} = Cl_t^{\leqslant} = U, Cl_t^{\geqslant} = Cl_t$ and $Cl_1^{\leqslant} = Cl_1$.

For $P \subseteq C, n \in T$, the lower and upper approximations of $Cl_n^{\geqslant}$ are defined as:

$$\underline{P}(Cl_n^{\geqslant}) = \{x \in U | D_P^+(x) \subseteq Cl_n^{\geqslant}\} \tag{6}$$

$$\overline{P}(Cl_n^{\geqslant}) = \{x \in U | D_P^-(x) \cap Cl_n^{\geqslant} \neq \emptyset\} \tag{7}$$

The lower and upper approximations of $Cl_n^{\leqslant}$ are defined as:

$$\underline{P}\left(Cl_n^{\leqslant}\right) = \left\{x \in U | D_P^-(x) \subseteq Cl_n^{\leqslant}\right\} \tag{8}$$

$$\overline{P}\left(Cl_n^{\leqslant}\right) = \left\{x \in U | D_P^+(x) \cap Cl_n^{\leqslant} \neq \emptyset\right\} \tag{9}$$

3. A parallel framework for computing approximations in DRSA

In DRSA, computations of basic knowledge granules and concept granules are previous steps of computing approximations. The traditional (serial) algorithm for computing basic knowledge granules in DRSA needs to scan across the universe. In this section, we attempt to decompose computation of basic knowledge granules based on classification of the information system. Then, we propose the strategies of decomposition and composition of computing approximations in DRSA. Our basic idea of computing approximations in parallel may be reflected in a scheme as shown in Fig. 1.

In the following, we elaborate strategies of decomposition of basic knowledge granules, concept granules and approximations.

3.1. Strategies of decomposing granules in DRSA

In the scheme of computing approximations in parallel, the starting point is to divide an information system into some smaller subsystems. We assume that an information system S may be divided into m smaller subsystems, i.e., $S = \bigcup_{k=1}^m S_k$, where $S_k = (U_k, A, V, f)$ called a subsystem of S. Clearly, the following items hold.

1. $U = \bigcup_{k=1}^m U_i$.
2. If $k \neq l$, then $U_k \cap U_l = \emptyset$.

The subsets of concept granules in DRSA can be computed based on these subsystems. As shown in Fig. 2, we can obtain the corresponding subsets of concept granules from S_1 to S_m, respectively. For example, according to S_1, we can get $Cl_n^{\geqslant 1}$ and $Cl_n^{\leqslant 1}$ which are subsets of $Cl_n^{\geqslant}$ and $Cl_n^{\leqslant}$ on U_1. $Cl_n^{\geqslant 1}$ and $Cl_n^{\leqslant 1}$ are marked as CG_1 in Fig. 2.

Let $U_k \subset U$, the subsets of concept granules on U_k are defined as follows:

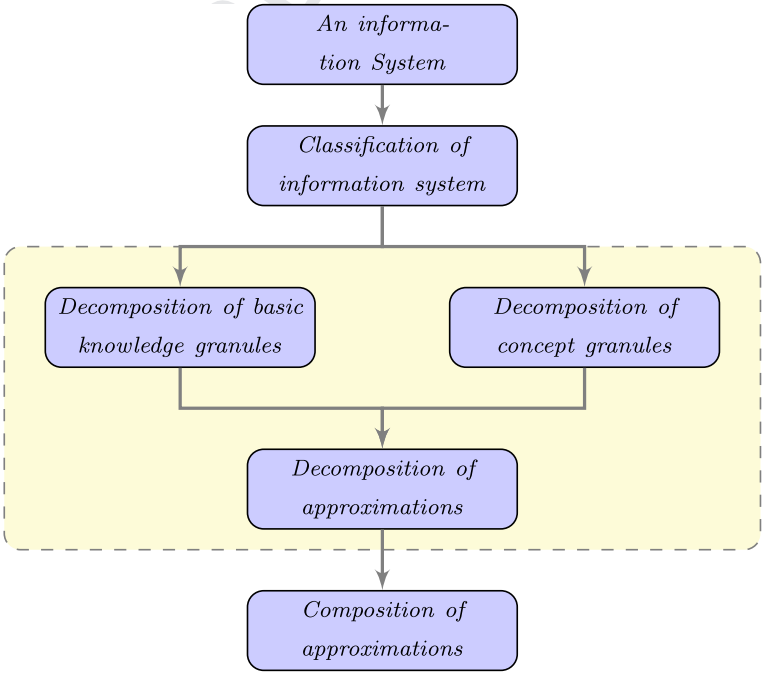

Fig. 1. The scheme of parallel computing approximations in DRSA.

Please cite this article in press as: S. Li et al., Parallel computing of approximations in dominance-based rough sets approach, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.05.003

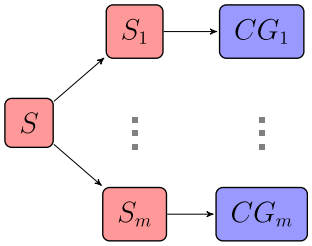

Fig. 2. The decomposition of concept granules in DRSA (CG_k includes $Cl_n^{\geqslant k}$ and $Cl_n^{\leqslant k}$, $k = 1, \ldots, m$).

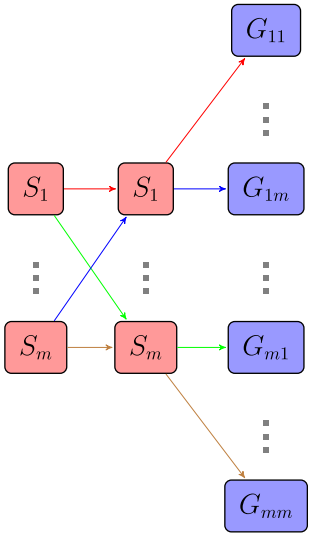

Fig. 3. The decomposition of basic knowledge granules in DRSA (G_{kl} represents $D_P^-(x_i)^l$ and $D_P^+(x_j)^k$, where $x_i \in U_k, x_j \in U_l$, $k = 1, \ldots, m$ and $l = 1, \ldots, m$.).

$$Cl_n^{\geqslant k} = \{x \in U_k | f(x,d) \geqslant d_n\} \tag{10}$$

$$Cl_n^{\leqslant k} = \{x \in U_k | f(x,d) \leqslant d_n\} \tag{11}$$

$Cl_n^{\geqslant k}$ and $Cl_n^{\leqslant k}$ are subsets of $Cl_n^{\geqslant}$ and $Cl_n^{\leqslant}$ on U_k, respectively.

Fig. 3 shows the basic idea of decomposing basic knowledge granules in DRSA. In Fig. 3, G_{11} represents $D_P^-(x_i)^1$ and $D_P^+(x_i)^1$, where $x_i \in U_1$. G_{11} can be computed based on the subsystem S_1. G_{1m} represents $D_P^-(x_i)^m$ and $D_P^+(x_j)^1$, where $x_i \in U_1, x_j \in U_m$. Computation of G_{1m} depends on two subsystems S_1 and S_m. $\forall k,\ l = 1, \ldots, m,\ x_i \in U_k$ and $x_j \in U_l, D_P^-(x_i)^l$ and $D_P^+(x_j)^k$ are defined as follows:

$$D_P^-(x_i)^l = \{x_j \in U_l | x_i D_P x_j\} \tag{12}$$

$$D_P^+(x_j)^k = \{x_i \in U_k | x_i D_P x_j\} \tag{13}$$

Due to the classification of the information system, the dominance matrix can be assembled by some smaller matrices. Hence, the dominance matrix

$$R^P = \begin{pmatrix} R_{1,1}^P & \cdots & R_{1,m}^P \\ \vdots & \ddots & \vdots \\ R_{m,1}^P & \cdots & R_{m,m}^P \end{pmatrix} \tag{14}$$

where $R_{k,l}^P = \left(\phi_{i,j}^P\right)_{|U_k| \times |U_l|}$ reflects the dominance relation from U_k to U_l, and $\phi_{i,j}^P$ indicates whether there is the dominance relation between $x_i \in U_k$ and $x_j \in U_l$. According to $R_{k,l}^P$, we can rewrite Eqs. (12) and (13) as follows:

$$D_P^-(x_i)^l = \{x_j \in U_l | \phi_{i,j}^P = |P|\} \tag{15}$$

$$D_P^+(x_j)^k = \{x_i \in U_k | \phi_{j,i}^P = |P|\} \tag{16}$$

Based on the discussion above, we give definitions of lower and upper approximations of $Cl_n^{\geqslant k}$ and $Cl_n^{\leqslant k}$ related to U_l by imitating definitions of lower and upper approximations of upward and downward unions of decision classes, respectively.

Definition 1. Given $k, l \in \{1, \ldots, m\}$, lower and upper approximations of $Cl_n^{\geqslant k}$ related to U_l are defined respectively as follows:

$$\underline{P}\left(Cl_n^{\geqslant k}\right)^l = \left\{x \in U_l | D_P^+(x)^k \subseteq Cl_n^{\geqslant k}\right\} \tag{17}$$

$$\overline{P}\left(Cl_n^{\geqslant k}\right)^l = \left\{x \in U_l | D_P^-(x)^k \cap Cl_n^{\geqslant k} \neq \emptyset\right\} \tag{18}$$

Analogously, lower and upper approximations of $Cl_n^{\leqslant k}$ related to U_l are defined respectively as follows:

$$\underline{P}\left(Cl_n^{\leqslant k}\right)^l = \left\{x \in U_l | D_P^-(x)^k \subseteq Cl_n^{\leqslant k}\right\} \tag{19}$$

$$\overline{P}\left(Cl_n^{\leqslant k}\right)^l = \left\{x \in U_l | D_P^+(x)^k \cap Cl_n^{\leqslant k} \neq \emptyset\right\} \tag{20}$$

$\underline{P}\left(Cl_n^{\geqslant k}\right)^l, \overline{P}\left(Cl_n^{\geqslant k}\right)^l, \underline{P}\left(Cl_n^{\leqslant k}\right)^l$ and $\overline{P}\left(Cl_n^{\leqslant k}\right)^l$ can be regarded as the parts decomposed from approximations.

3.2. Strategies of composing approximations in DRSA

In order to discuss the composition of approximations, we give Lemmas 1 and 2 at first.

Lemma 1. $Cl_n^{\geqslant} = \bigcup_{k=1}^m Cl_n^{\geqslant k}$ and $Cl_n^{\leqslant} = \bigcup_{k=1}^m Cl_n^{\leqslant k}$.

Proof. $\because Cl_n^{\geqslant} = \{x \in U | f(x,d) \geqslant d_n\} = \left\{x \in \bigcup_{k=1}^m U_k | f(x,d) \geqslant d_n\right\} = \bigcup_{k=1}^m \{x \in U_k | f(x,d) \geqslant d_n\} = \bigcup_{k=1}^m Cl_n^{\geqslant k}. \therefore Cl_n^{\geqslant} = \bigcup_{k=1}^m Cl_n^{\geqslant k}$.

Similarly, $Cl_n^{\leqslant} = \bigcup_{k=1}^m Cl_n^{\leqslant k}$ also holds. □

Lemma 2. *Given $k, l \in \{1, \ldots, m\}$, the following items hold.*

1. $\forall x_i \in U_k, D_P^-(x_i) = \bigcup_{l=1}^m D_P^-(x_i)^l$;
2. $\forall x_j \in U_l, D_P^+(x_i) = \bigcup_{k=1}^m D_P^+(x_i)^k$.

Proof. $\because x_i \in U_k \subset U, \therefore D_P^-(x_i) = \left\{x_j \in U | \phi_{i,j}^P = |P|\right\} = \left\{x_j \in \bigcup_{l=1}^m U_l | \phi_{i,j}^P = |P|\right\} = \bigcup_{l=1}^m \left\{x_j \in U_l | \phi_{i,j}^P = |P|\right\} = \bigcup_{l=1}^m D_P^-(x_i)^l$.

Similarly, the item 2 holds. □

Lemmas 1 and 2 show approaches for composing concept granules and basic knowledge granules with their subsets, respectively. Based on Lemmas 1 and 2, we present the following proposition which can be employed to compose approximations.

Proposition 1. *For $k, l \in \{1, \ldots, m\}$, the following items hold.*

1. $\underline{P}(Cl_n^{\geqslant}) = \bigcup_{l=1}^m \left[\bigcap_{k=1}^m \underline{P}\left(Cl_n^{\geqslant k}\right)^l\right], \overline{P}(Cl_n^{\geqslant}) = \bigcup_{k,l=1}^m \overline{P}\left(Cl_n^{\geqslant k}\right)^l$;
2. $\underline{P}\left(Cl_n^{\leqslant}\right) = \bigcup_{l=1}^m \left[\bigcap_{k=1}^m \underline{P}\left(Cl_n^{\leqslant k}\right)^l\right], \overline{P}\left(Cl_n^{\leqslant}\right) = \bigcup_{k,l=1}^m \overline{P}\left(Cl_n^{\leqslant k}\right)^l$.

Proof. $\because \underline{P}(Cl_n^{\geqslant}) = \{x \in U | D_P^+(x) \subseteq Cl_n^{\geqslant}\} = \{x \in \bigcup_{l=1}^m U_l | \bigcup_{k=1}^m D_P^+(x)^k \subseteq \bigcup_{k=1}^m Cl_n^{\geqslant k}\} = \bigcup_{l=1}^m \{x \in U_l | D_P^+(x)^1 \subseteq Cl_n^{\geqslant 1} \wedge \cdots \wedge D_P^+(x)^m \subseteq Cl_n^{\geqslant m}\} = \bigcup_{l=1}^m [\bigcap_{k=1}^m \{x \in U_l | D_P^+(x)^k \subseteq Cl_n^{\geqslant k}\}] = \bigcup_{l=1}^m \left[\bigcap_{k=1}^m \underline{P}\left(Cl_n^{\geqslant k}\right)^l\right], \therefore \underline{P}(Cl_n^{\geqslant}) = \bigcup_{l=1}^m \left[\bigcap_{k=1}^m \underline{P}\left(Cl_n^{\geqslant k}\right)^l\right]$.

$\because \overline{P}(Cl_n^{\geqslant}) = \{x \in U | D_P^-(x) \cap Cl_n^{\geqslant} \neq \emptyset\} = \{x \in \bigcup_{l=1}^m U_l | \bigcup_{k=1}^m D_P^-(x)^k \cap \bigcup_{k=1}^m Cl_n^{\geqslant k} \neq \emptyset\} = \bigcup_{l=1}^m \{x \in U_l | \bigcup_{k=1}^m D_P^-(x)^k \cap \bigcup_{k=1}^m Cl_n^{\geqslant k} \neq \emptyset\} = \bigcup_{l=1}^m \{x \in U_l | D_P^-(x)^1 \cap Cl_n^{\geqslant 1} \neq \emptyset \vee \cdots \vee D_P^-(x)^m \cap Cl_n^{\geqslant m} \neq \emptyset\} = \bigcup_{l=1}^m \left[\bigcup_{k=1}^m \{x \in U_l | D_P^-(x)^k \cap Cl_n^{\geqslant k} \neq \emptyset\}\right] = \bigcup_{l=1}^m \left[\bigcup_{k=1}^m \overline{P}\left(Cl_n^{\geqslant k}\right)^l\right] = \bigcup_{k,l=1}^m \overline{P}\left(Cl_n^{\geqslant k}\right)^l, \therefore \overline{P}(Cl_n^{\geqslant}) = \bigcup_{k,l=1}^m \overline{P}\left(Cl_n^{\geqslant k}\right)^l$.

Similarly, the item 2 holds. □

Based on these strategies mentioned above, we introduce a parallel framework of computing approximations in DRSA. This framework is shown in Fig. 4. From Fig. 4, we can see that the computational process can be divided into m tasks which can be done concurrently before the sixth column. Hence, these tasks can be executed on different processors in parallel. We call the task which relates with $G'_{11}, \ldots, G'_{1m}$ and CG_1 as the first task. In the first task, we compute approximations of concept granules' subsets at first. They are marked as $LA_{11}, \ldots, LA_{1m}$ and $UA_{11}, \ldots, UA_{1m}$ in Fig. 4, respectively. Then, we compute four sets which are some necessary interim results for composing approximations in DRSA as follows: $\bigcup_{k=1}^m \overline{P}\left(Cl_n^{\geqslant k}\right)^l, \bigcup_{k=1}^m \overline{P}\left(Cl_n^{\leqslant k}\right)^l, \bigcap_{k=1}^m \underline{P}\left(Cl_n^{\geqslant k}\right)^l$ and $\bigcap_{k=1}^m \underline{P}\left(Cl_n^{\leqslant k}\right)^l$. In Fig. 4, LA_1 represents $\bigcap_{k=1}^m \underline{P}\left(Cl_n^{\geqslant k}\right)^l$ and $\bigcap_{k=1}^m \underline{P}\left(Cl_n^{\leqslant k}\right)^l$. UA_1 represents $\bigcup_{k=1}^m \overline{P}\left(Cl_n^{\geqslant k}\right)^l$ and $\bigcup_{k=1}^m \overline{P}\left(Cl_n^{\leqslant k}\right)^l$. Analogously, other tasks can obtain their corresponding interim results. From the fifth column to the sixth column, it shows that LA and UA (lower and upper approximations in DRSA) are assembled with $LA_1, \ldots, LA_m$ and $UA_1, \ldots, UA_m$, respectively. By Proposition 1, LA is the union of $LA_1, \ldots, LA_m$ and UA is the union of $UA_1, \ldots, UA_m$.

4. A numeric illustration

Example 1. Assume that the information system S shown in Table 1 can be divided into two subsystems S_1 and S_2. $U_1 = \{x_1, x_2, x_3, x_4\}$ and $U_2 = \{x_5, x_6, x_7\}$ are the object sets of S_1 and S_2, respectively. Let $P = \{a_1, a_2, a_3\}$.

Firstly, we compute the subsets of decision class unions related to S_1 and S_2, respectively. The results are shown as follows:

$Cl_1^{\geqslant 1} = U_1, \quad Cl_2^{\geqslant 1} = \{x_2, x_3\}, \quad Cl_3^{\geqslant 1} = \emptyset; \quad Cl_1^{\leqslant 1} = \{x_1, x_4\},$
$Cl_2^{\leqslant 1} = U_1, \quad Cl_3^{\leqslant 1} = U_1.$
$Cl_1^{\geqslant 2} = U_2, \quad Cl_2^{\geqslant 2} = \{x_6, x_7\}, \quad Cl_3^{\geqslant 2} = \{x_7\}; \quad Cl_1^{\leqslant 2} = \{x_5\},$
$Cl_2^{\leqslant 2} = \{x_5, x_6\}, \quad Cl_3^{\leqslant 2} = U_2.$

Next, we compute four dominance matrices: $R_{1,1}^P$ on S_1, $R_{2,2}^P$ on S_2, $R_{1,2}^P$ from S_1 to S_2 and $R_{2,1}^P$ from S_2 to S_1, respectively. They are shown as follows:

Table 1
An information table.

U	a_1	a_2	a_3	d	U	a_1	a_2	a_3	d
x_1	2	1	3	1	x_5	1	2	3	1
x_2	2	1	2	2	x_6	2	2	1	2
x_3	3	1	1	2	x_7	3	1	2	3
x_4	2	3	1	1					

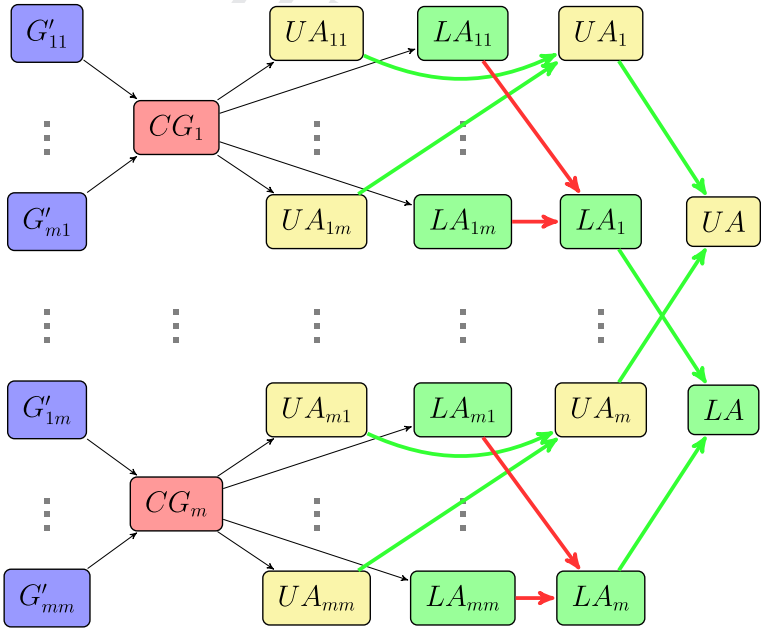

Fig. 4. A parallel framework of computing approximations in DRSA (G'_{kl} represents $D_P^-(x_i)^l$ and $D_P^+(x_i)^l$, where $x_i \in U_k$. UA_{kl} represents $\overline{P}\left(Cl_n^{\geqslant k}\right)^l$ and $\overline{P}\left(Cl_n^{\leqslant k}\right)^l$. LA_{kl} represents $\underline{P}\left(Cl_n^{\geqslant k}\right)^l$ and $\underline{P}\left(Cl_n^{\leqslant k}\right)^l$. UA_l represents $\bigcup_{k=1}^m \overline{P}\left(Cl_n^{\geqslant k}\right)^l$ and $\bigcup_{k=1}^m \overline{P}\left(Cl_n^{\leqslant k}\right)^l$. LA_l represents $\bigcap_{k=1}^m \underline{P}\left(Cl_n^{\geqslant k}\right)^l$ and $\bigcap_{k=1}^m \underline{P}\left(Cl_n^{\leqslant k}\right)^l$. UA represents $\overline{P}(Cl_n^{\geqslant})$ and $\overline{P}\left(Cl_n^{\leqslant}\right)$. LA represents $\underline{P}(Cl_n^{\geqslant})$ and $\underline{P}\left(Cl_n^{\leqslant}\right)$. $k, l = 1, \ldots, m$).

Please cite this article in press as: S. Li et al., Parallel computing of approximations in dominance-based rough sets approach, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.05.003

$$R^P_{1,1}=\begin{pmatrix}3&3&2&2\\2&3&2&2\\2&2&3&2\\2&2&2&3\end{pmatrix},\quad R^P_{2,2}=\begin{pmatrix}3&2&2\\1&3&2\\1&2&3\end{pmatrix},$$

$$R^P_{1,2}=\begin{pmatrix}2&2&2\\1&2&2\\1&2&2\\2&3&1\end{pmatrix},\quad R^P_{2,1}=\begin{pmatrix}2&2&2&1\\2&2&2&2\\2&3&3&2\end{pmatrix}.$$

Based on these dominance matrices, we compute the subsets of basic knowledge granules, respectively. The results are shown as follows:

$D_P^-(x_1)^1=\{x_1,x_2\},\ D_P^+(x_1)^1=\{x_1\},\ D_P^-(x_1)^2=\emptyset,\ D_P^+(x_1)^2=\emptyset;$

$D_P^-(x_2)^1=\{x_2\},\ D_P^+(x_2)^1=\{x_1,x_2\},\ D_P^-(x_2)^2=\emptyset,\ D_P^+(x_2)^2=\{x_7\};$

$D_P^-(x_3)^1=\{x_3\},\ D_P^+(x_3)^1=\{x_3\},\ D_P^-(x_3)^2=\emptyset,\ D_P^+(x_3)^2=\{x_7\};$

$D_P^-(x_4)^1=\{x_4\},\ D_P^+(x_4)^1=\{x_4\},\ D_P^-(x_4)^2=\{x_6\},\ D_P^+(x_4)^2=\emptyset;$

$D_P^-(x_5)^1=\emptyset,\ D_P^+(x_5)^1=\emptyset,\ D_P^-(x_5)^2=\{x_5\},\ D_P^+(x_5)^2=\{x_5\};$

$D_P^-(x_6)^1=\emptyset,\ D_P^+(x_6)^1=\{x_4\},\ D_P^-(x_6)^2=\{x_6\},\ D_P^+(x_6)^2=\{x_6\};$

$D_P^-(x_7)^1=\{x_2,x_3\},\ D_P^+(x_7)^1=\emptyset,\ D_P^-(x_7)^2=\{x_7\},\ D_P^+(x_7)^2=\{x_7\}.$

Thirdly, we compute approximations of concept granules' subsets obtained above according to Definition 1, respectively. The results are shown as follows:

$\underline{P}\left(Cl_2^{\geqslant 1}\right)^1=\{x_3\},\ \underline{P}\left(Cl_3^{\geqslant 1}\right)^1=\emptyset;\ \overline{P}\left(Cl_2^{\geqslant 1}\right)^1=\{x_1,x_2,x_3\},\ \overline{P}\left(Cl_3^{\geqslant 1}\right)^1=\emptyset;$

$\underline{P}\left(Cl_1^{\leqslant 1}\right)^1=\{x_4\},\ \underline{P}\left(Cl_2^{\leqslant 1}\right)^1=U_1;\ \overline{P}\left(Cl_1^{\leqslant 1}\right)^1=\{x_1,x_2\},\ \overline{P}\left(Cl_2^{\leqslant 1}\right)^1=U_1.$

$\underline{P}\left(Cl_2^{\geqslant 1}\right)^2=U_1,\ \underline{P}\left(Cl_3^{\geqslant 1}\right)^2=\{x_1,x_2,x_3\};\ \overline{P}\left(Cl_2^{\geqslant 1}\right)^2=\overline{P}\left(Cl_3^{\geqslant 1}\right)^2=\{x_2,x_3\};$

$\underline{P}\left(Cl_1^{\leqslant 1}\right)^2=\{x_1,x_2,x_3\},\ \underline{P}\left(Cl_2^{\leqslant 1}\right)^2=U_1;\ \overline{P}\left(Cl_1^{\leqslant 1}\right)^1=\overline{P}\left(Cl_2^{\leqslant 1}\right)^2=\emptyset.$

$\underline{P}\left(Cl_2^{\geqslant 2}\right)^1=\{x_5,x_7\},\ \underline{P}\left(Cl_3^{\geqslant 2}\right)^1=\{x_5,x_7\};\ \overline{P}\left(Cl_2^{\geqslant 2}\right)^1=\{x_7\},\ \overline{P}\left(Cl_3^{\geqslant 2}\right)^1=\emptyset;$

$\underline{P}\left(Cl_1^{\leqslant 2}\right)^1=\{x_5,x_6\},\ \underline{P}\left(Cl_2^{\leqslant 2}\right)^1=U_2;\ \overline{P}\left(Cl_1^{\leqslant 2}\right)^1=\{x_6\},\ \overline{P}\left(Cl_2^{\leqslant 2}\right)^1=\{x_6\}.$

$\underline{P}\left(Cl_2^{\geqslant 2}\right)^2=\{x_6,x_7\},\ \underline{P}\left(Cl_3^{\geqslant 2}\right)^2=\{x_7\};\ \overline{P}\left(Cl_2^{\geqslant 2}\right)^2=\{x_6,x_7\},\ \overline{P}\left(Cl_3^{\geqslant 2}\right)^2=\{x_7\};$

$\underline{P}\left(Cl_1^{\leqslant 2}\right)^2=\{x_5\},\ \underline{P}\left(Cl_2^{\leqslant 2}\right)^2=\{x_5,x_6\};\ \overline{P}\left(Cl_1^{\leqslant 2}\right)^2=\{x_5\},\ \overline{P}\left(Cl_2^{\leqslant 2}\right)^2=\{x_5,x_6\}.$

Finally, we assemble approximations in DRSA with the results obtained in previous step according to Proposition 1.

$\underline{P}(Cl_2^{\geqslant})=\{x_3,x_7\},\quad \underline{P}(Cl_3^{\geqslant})=\{x_7\};\quad \overline{P}(Cl_2^{\geqslant})=\{x_1,x_2,x_3,x_6,x_7\},$
$\overline{P}(Cl_3^{\geqslant})=\{x_2,x_3,x_7\}.$

$\underline{P}\left(Cl_1^{\leqslant}\right)=\{x_5\},\quad \underline{P}\left(Cl_2^{\leqslant}\right)=\{x_1,x_2,x_3,x_4,x_5,x_6\};$

$\overline{P}\left(Cl_1^{\leqslant}\right)=\{x_1,x_2,x_5,x_6\},\quad \overline{P}\left(Cl_2^{\leqslant}\right)=\{x_1,x_2,x_3,x_4,x_5,x_6\}.$

In Example 1, clearly, $\underline{P}(Cl_1^{\geqslant})=\overline{P}(Cl_1^{\geqslant})=\underline{P}\left(Cl_3^{\leqslant}\right)=\overline{P}\left(Cl_3^{\leqslant}\right)=U.$

5. A parallel algorithm for computing approximations in DRSA

In this section, we design a parallel algorithm based on the framework of computing approximations in DRSA mentioned above. Before introducing the parallel algorithm, we give algorithms for computing subsets of basic knowledge granules and concept granules as well as approximations of concept granules' subsets on each of subsystems, respectively.

Algorithm 1. An algorithm for computing subsets of basic knowledge granules

Require: Any two subsystems, S_a and S_b;
Ensure The subsets of P-dominating and P-dominated sets of objects in S_a with respect to S_b;
Function PartGranules(S_a, S_b)
  for all $x \in S_a$ **do**
    $G^+(x)\leftarrow\emptyset, G^-(x)\leftarrow\emptyset, s1\leftarrow 0, s2\leftarrow 0$ ▷ Initial subsets of P-dominating and P-dominated sets of objects in S_a, $G^+(x)$ and $G^-(x)$.
    for all $y \in S_b$ **do**
      for all $a \in C$ **do**
        If $f(y,a) > f(x,a)$ **then** $s1 \leftarrow s1+1$
        else if $f(y,a) < f(x,a)$ **then** $s2 \leftarrow s2+1$
        else $s1 \leftarrow s1+1, s2 \leftarrow s2+1$
        end if
      end for
      If $s1 == |C|$ **then** $G^+(x) \leftarrow G^+(x) \cup \{y\}$
      end if
      If $s2 == |C|$ **then** $G^-(x) \leftarrow G^-(x) \cup \{y\}$
      end if
    end for
  end for
  return $[G^+(S_a), G^-(S_a)]$ ▷ Notice that $G^+(S_a)=\{G^+(x)|x\in S_a\}, G^-(S_a)=\{G^-(x)|x\in S_a\}$
end function

Algorithm 1 is to compute subsets of P-dominating and P-dominated sets of all objects in subsystem S_a related to subsystem S_b. Its time complexity is $O(|S_a||S_b||C|)$.

Algorithm 2. An algorithm for computing subsets of concept granules

Require: A subsystem S_b;
Ensure: The subsets of upward and downward unions of decision classes with respect to S_b;
function: PartUnions(S_b)
  for all $n \in T$ **do**
    $G_n^{\geqslant}\leftarrow\emptyset, G_n^{\leqslant}\leftarrow\emptyset$ ▷ Initial subsets upward and downward unions of decision classes, $G_n^{\geqslant}$ and $G_n^{\leqslant}$.
    for all $x \in S_b$ **do**
      If $f(x,d) > d_n$ **then** $G_n^{\geqslant} \leftarrow G_n^{\geqslant} \cup \{x\}$
      else if $f(x,d) < d_n$ **then** $G_n^{\leqslant} \leftarrow G_n^{\leqslant} \cup \{x\}$
      else $G_n^{\geqslant} \leftarrow G_n^{\geqslant} \cup \{x\}, G_n^{\leqslant} \leftarrow G_n^{\leqslant} \cup \{x\}$
      end if
    end for
  end for
  return $[G^{\geqslant}, G^{\leqslant}]$ ▷ Notice that $G^{\geqslant}=\{G_n^{\geqslant}|n\in T\}, G^{\leqslant}=\{G_n^{\leqslant}|n\in T\}$
end function

Algorithm 2 is to compute subsets of downward and upward unions of decision classes related to subsystem S_b. Its time complexity is $O(|S_b||T|)$.

Please cite this article in press as: S. Li et al., Parallel computing of approximations in dominance-based rough sets approach, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.05.003

Algorithm 3. An algorithm for computing approximations of concept granules' subsets

Require: The subsets of upward union or downward unions of decision classes N, the subset of P-dominating set or P-dominated set $G(S_a)$;
Ensure: Subsets of approximations K;
function PartAppr$(N, G(S_a), B)$ ▷ Notice that $G(S_a) = \{G(x)|x \in S_a\}, N = \{N_n|n \in T\}$
for all $N_n \in N$ **do**
$K_n \leftarrow \emptyset$
for all $x \in S_a$ **do**
If $B == 1$ **and** $G(x) \subseteq N_n$ **then**
$K_n \leftarrow K_n \cup \{x\}$
end if
If $B == 0$ **and** $G(x) \cap N_n \neq \emptyset$ **then**
$K_n \leftarrow K_n \cup \{x\}$
end if
end for
end for
return K ▷ Notice that $K = \{K_n|n \in T\}$
end function

Algorithm 3 is to compute upper approximations of concept granules' subsets N related to subsystem S_a when $B = 1$ and lower approximations of concept granules' subsets N related to subsystem S_a when $B = 0$. Its time complexity is $O(|N||S_a|)$.

Combined with Algorithms 1–3, we design a parallel algorithm (Algorithm 4) for computing approximations of DRSA.

Algorithm 4. A parallel algorithm for computing approximations of DRSA

Require: $S = \bigcup_{i=1}^{m} S_i$;
Ensure: $\overline{P}(Cl_n^{\geqslant}), \overline{P}\left(Cl_n^{\leqslant}\right), \underline{P}(Cl_n^{\geqslant}), \underline{P}\left(Cl_n^{\leqslant}\right), \forall n \in T$;
function ParComApp(S)
for all $n \in T$ **do**
$\overline{P}(Cl_n^{\geqslant}) \leftarrow \emptyset, \overline{P}\left(Cl_n^{\leqslant}\right) \leftarrow \emptyset, \underline{P}(Cl_n^{\geqslant}) \leftarrow \emptyset, \underline{P}\left(Cl_n^{\leqslant}\right) \leftarrow \emptyset$
end for
parfor $i = 1 \rightarrow m^2$ **do** ▷ This loop module can be executed in parallel.

$$\begin{bmatrix} \underline{P}\left(Cl^{\geqslant i-int(i/k)}\right)^{int(i/k)+1}, \\ \underline{P}\left(Cl^{\leqslant i-int(i/k)}\right)^{int(i/k)+1}, \\ \overline{P}\left(Cl^{\geqslant i-int(i/k)}\right)^{int(i/k)+1}, \\ \overline{P}\left(Cl^{\leqslant i-int(i/k)}\right)^{int(i/k)+1} \end{bmatrix} \leftarrow \text{PartApproximations}$$

$(U_{int(i/k)+1}, U_{i-int(i/k)})$ ▷ k is the number of cores employed.
for all $n \in T$ **do**

$$\overline{P}(Cl_n^{\geqslant}) \leftarrow \overline{P}(Cl_n^{\geqslant}) \cup \overline{P}\left(Cl_n^{\geqslant i-int(i/k)}\right)^{int(i/k)+1}$$

$$\overline{P}\left(Cl_n^{\leqslant}\right) \leftarrow \overline{P}\left(Cl_n^{\leqslant}\right) \cup \overline{P}\left(Cl_n^{\leqslant i-int(i/k)}\right)^{int(i/k)+1}$$

$$\underline{P}(Cl_n^{\geqslant}) \leftarrow \underline{P}(Cl_n^{\geqslant}) \cup \underline{P}\left(Cl_n^{\geqslant i-int(i/k)}\right)^{int(i/k)+1}$$

$$\underline{P}\left(Cl_n^{\leqslant}\right) \leftarrow \underline{P}\left(Cl_n^{\leqslant}\right) \cup \underline{P}\left(Cl_n^{\leqslant i-int(i/k)}\right)^{int(i/k)+1}$$

end for
end parfor
end function
function PartApproximations(S_a, S_b)
$[D_P^+(S_a), D_P^-(S_a)] \leftarrow\leftarrow$PartGranules$(S_a, S_b)$
$[Cl^{\geqslant}, Cl^{\leqslant}] \leftarrow$ PartUnions(S_b)
$\underline{P}(Cl^{\geqslant}) \leftarrow$ PartAppr$(Cl^{\geqslant}, D_P^+(S_a),1)$
$\underline{P}(Cl^{\leqslant}) \leftarrow$ PartAppr$(Cl^{\leqslant}, D_P^-(S_a),1)$
$\overline{P}(Cl^{\geqslant}) \leftarrow$PartAppr$(Cl^{\geqslant}, D_P^-(S_a),0)$
$\overline{P}(Cl^{\leqslant}) \leftarrow$ PartAppr$(Cl^{\leqslant}, D_P^+(S_a),0)$
return $[\underline{P}(Cl^{\geqslant}), \underline{P}(Cl^{\leqslant}), \overline{P}(Cl^{\geqslant}), \overline{P}(Cl^{\leqslant})]$
end function

Algorithm 4 is to compute approximations of DRSA in parallel on a k-cores computational platform. Its time complexity is $O\left(m^2\left[\frac{1}{k^2}|U||U||C| + \frac{1}{k}|U||T| + \frac{1}{k}|U||N||T|\right] + |T|\right) \approx O\left(m^2\left[\frac{1}{k^2}|U||U||C| + \frac{1}{k}|U||N||T|\right]\right)$, where m and k are the numbers of universe's subsets and cores, respectively. Indeed, the time complexity equals with that of the serial algorithm for computing approximations of DRSA when $k = 1$ and $m = 1$.

6. Experimental analysis

In this section, we do experiments for comparing the parallel algorithm on multi-core environments and a serial algorithm (the tradition algorithm) on singleton core environment. The platform is Intel(R) Xeon(R) CPU E5620 2.40 GHz (8 cores processor) and the operation system is Windows (professional version). The programs employed in the experiments are coded in Matlab(R2010b), and a data set, EEG Eye State, is downloaded from UCI. The data includes 14980 objects, 15 attributes and 2 classes.

Firstly, we partition EEG Eye State into ten data sets of equivalent size and call them as data set 1, data set 2, ..., data set 10. Then, we construct ten experimental data sets such as the data set 1, union of data sets 1 and 2, ..., union of data sets 1 to 10. Then, we run the parallel algorithm and its counterpart on each of experimental data sets. In order to balance the amount of computation on each core of the multi-core processor, we divide the universe into several smaller universes in same size under different multi-core environments, respectively. Table 2 reflects the corresponding classifications of the universe versus different numbers of cores.

Table 3 lists the computational time of the serial algorithm on singleton core environment and that of the parallel algorithm on 2, ..., 8 cores environments, respectively. In Table 3, t_1 is the running time of serial algorithm and $t_2, \ldots, t_8$ are those of 2, ..., 8 cores environments, respectively. Fig. 5 reflects the computational time on each of these experimental data sets from 1 core to 8 cores, respectively.

From Table 3 and Fig. 5, we can see that the running time of two algorithms rise with the increasing number of objects. The running

Table 2
The classification of the universe.

Cores	Universes	Cores	Universes
1	1	5	5
2	2	6	6
3	3	7	7
4	2	8	4

Table 3
A running time comparison for parallel and serial algorithms.

Objects	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
1498	5.2915	3.0140	2.2947	2.1287	1.9017	1.5217	1.4600	1.4349
2996	12.9692	7.0296	4.9915	3.9149	3.4805	3.1262	3.0383	2.9405
4494	25.6166	13.4203	9.3078	7.1568	6.2303	5.6137	5.3204	4.8185
5992	42.4644	21.9638	16.0972	11.6866	9.7594	8.8031	7.9850	7.7203
7490	63.0405	32.4722	22.4753	17.1812	14.3662	12.6245	11.5866	10.4434
8988	88.6007	45.5488	31.2137	23.8798	19.6764	17.4432	15.8289	14.8131
10,486	118.0680	60.5104	41.5749	31.9134	26.1754	23.1782	20.7651	19.8283
11,984	156.3259	82.2001	56.6601	43.3076	36.1922	31.2945	28.4966	26.9449
13,482	195.2052	102.2506	70.5394	54.3222	44.6829	38.9572	35.3462	32.7888
14,980	238.2680	130.3852	85.0007	66.3397	54.4273	48.4099	44.3813	42.8852

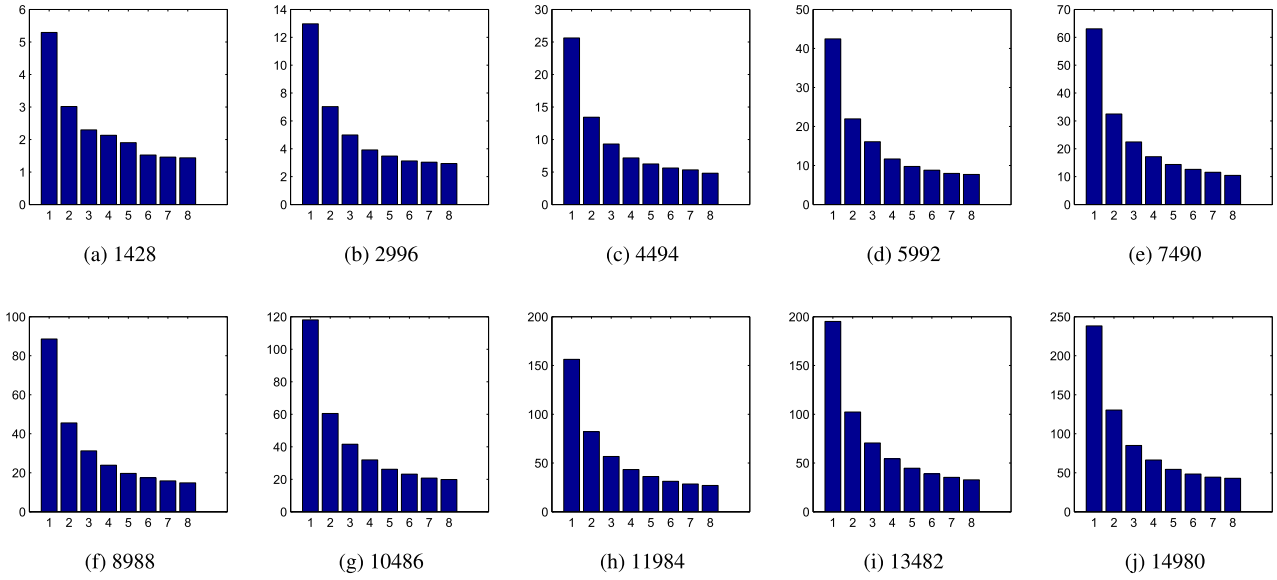

Fig. 5. The scheme of parallel computing approximations in DRSA.

Table 4
The speedup ratio between the parallel algorithm and the serial algorithm under different running environments.

Objects	α_2	α_3	α_4	α_5	α_6	α_7	α_8
1498	1.7556	2.3060	2.4858	2.7825	3.4774	3.6243	3.6877
2996	1.8449	2.5983	3.3128	3.7262	4.1486	4.2686	4.4105
4494	1.9088	2.7522	3.5793	4.1116	4.5632	4.8148	5.3163
5992	1.9334	2.6380	3.6336	4.3511	4.8238	5.3180	5.5004
7490	1.9414	2.8049	3.6692	4.3881	4.9935	5.4408	6.0364
8988	1.9452	2.8385	3.7103	4.5029	5.0794	5.5974	5.9812
10,486	1.9512	2.8399	3.6996	4.5106	5.0939	5.6859	5.9545
11,984	1.9018	2.7590	3.6097	4.3193	4.9953	5.4858	5.8017
13,482	1.9091	2.7673	3.5935	4.3687	5.0108	5.5227	5.9534
14,980	1.8274	2.8031	3.5916	4.3777	4.9219	5.3687	5.5559

time of the serial algorithm on singleton core environment is less than those of the parallel algorithm on multi-core environments. The more cores are employed in the experiment, the more time is reduced. By Table 3, we can compute the speedup ratio between the parallel algorithm and the serial algorithm under different running environments. The results are depicted in Table 4, where $\alpha_2 = t_1/t_2, \ldots, \alpha_8 = t_1/t_8$.

From Table 4, we can see that $\alpha_2 < \alpha_3 < \cdots < \alpha_8$ on these ten experimental data sets.

7. Conclusions and future works

In this paper, we have discussed the importance of accelerating computation of approximations in DRSA and proposed an approach for computing approximations in parallel as well as the corresponding parallel algorithm. By a numerical illustration and some experimental evaluations, the following conclusions can be drawn.

- It is feasible to use the parallel approach for computing approximations in DRSA.
- The parallel algorithm can reduce the computational time. For the same data set, the more cores the processor has, the more time the parallel algorithm reduces.

In our future work, we will improve this algorithm for running it on the distributed computing environment further. We also will study rule extraction in parallel from preference-ordered information systems based on the strategies of decomposition and composition of approximations in DRSA.

Acknowledgements

This work is supported by the National Science Foundation of China (Nos. 61175047, 71201133) and NSAF (No. U1230117).

Please cite this article in press as: S. Li et al., Parallel computing of approximations in dominance-based rough sets approach, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.05.003

References

[1] B. Barney, Introduction to Parallel Computing, 2014. <https://computing.llnl.gov/tutorials/parallel_comp/>.
[2] H. Chen, T. Li, D. Ruan, J. Lin, C. Hu, A rough-set-based incremental approach for updating approximations under dynamic maintenance environments, IEEE Trans. Knowl. Data Eng. 25 (2013) 274–284.
[3] Y. Cheng, The incremental method for fast computing the rough fuzzy approximations, Data Knowl. Eng. 70 (2011) 84–100.
[4] S. Greco, B. Matarazzo, R. Slowinski, Rough sets theory for multicriteria decision analysis, Eur. J. Oper. Res. 129 (2001) 1–47.
[5] F. Hu, G. Wang, H. Huang, Y. Wu, Incremental attribute reduction based on elementary sets, in: RSFDGrC2005, 3641, LNAI, pp. 185–193.
[6] M. Kryszkiewicz, Rough set approach to incomplete information system, Inform. Sci. 112 (1998) 39–49.
[7] S. Li, T. Li, A parallel matrix-based approach for computing approximations in dominance-based rough sets approach, in: RSKT 2014, 8818, LNCS, pp. 173–183.
[8] S. Li, T. Li, Incremental update of approximations in dominance-based rough sets approach under the variation of attribute values, Inform. Sci. 294 (2015) 348–361.
[9] S. Li, T. Li, H. Chen, A parallel approach for computing approximations of dominance-based rough sets approach, in: R.M. de Moraes, E.E. Kerre, L. dos Santos Machado, J. Lu (Eds.), Proceedings of the 11th International FLINS Conference Decision Making and Soft Computing, World Scientific, 2014, pp. 580–585.
[10] S. Li, T. Li, D. Liu, Dynamic maintenance of approximations in dominance-based rough set approach under the variation of the object set, Int. J. Intell. Syst. 28 (2013) 729–751.
[11] S. Li, T. Li, D. Liu, Incremental updating approximations in dominance-based rough sets approach under the variation of the attribute set, Knowl.-Based Syst. 40 (2013) 17–26.
[12] C. Luo, T. Li, H. Chen, D. Liu, Incremental approaches for updating approximations in set-valued ordered information systems, Knowl.-Based Syst. 50 (2013) 218–233.
[13] Z. Pawlak, Rough sets, Int. J. Comput. Inform. Sci. 11 (1982) 341–356.
[14] Z. Pawlak, Rough set approach to knowledge-based decision support, Eur. J. Oper. Res. 99 (1997) 48–57.
[15] Z. Pawlak, A. Skowron, Rough sets: some extensions, Inform. Sci. 177 (2007) 28–40.
[16] Z. Pawlak, R. Slowinski, Rough set approach to multi-attribute decision analysis, Eur. J. Oper. Res. 72 (1994) 443–459.
[17] J. Qian, D.Q. Miao, Z.H. Zhang, X.D. Yue, Parallel attribute reduction algorithms using MapReduce, Inform. Sci. 279 (2014) 671–690.
[18] Y. Qian, J. Liang, F. Wang, Positive-approximation based accelerated algorithm to feature selection from incomplete decision tables, Chin. J. Comput. 34 (2011) 435–442.
[19] Y. Qian, J. Liang, Y. Yao, C. Dang, MGRS: a multi-granulation rough set, Inform. Sci. 180 (2010) 949–970.
[20] L. Shan, W. Ziarko, Data-based acquisition and incremental modification of classification rules, Comput. Intell. 11 (1995) 357–370.
[21] L. Shen, H. Loh, Applying rough sets to market timing decisions, Decis. Support Syst. 37 (2004) 583–597.
[22] A. Skowron, J. Stepaniuk, Information granule decomposition, Fundam. Inform. 47 (2001) 337–350.
[23] S. Tsumoto, Automated extraction of hierarchical decision rules from clinical databases using rough set model, Expert Syst. Appl. 24 (2003) 189–197.
[24] Y. Yao, S. Wong, A decision theoretic framework for approximating concepts, Int. J. Man Mach. Stud. 37 (6) (1992) 793–809.
[25] J. Zhang, T. Li, H. Chen, Composite rough sets, in: AICI2012, 7530, LNCS, pp. 150–159.
[26] J. Zhang, T. Li, H. Chen, Composite rough sets for dynamic data mining, Inform. Sci. 257 (2014) 81–100.
[27] J. Zhang, T. Li, D. Ruan, Neighborhood rough sets for dynamic data mining, Int. J. Intell. Syst. 27 (2012) 317–342.
[28] J. Zhang, T. Li, D. Ruan, A parallel method for computing rough set approximations, Inform. Sci. 194 (2012) 209–223.
[29] J. Zhang, T. Li, D. Ruan, Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems, Int. J. Approx. Reason. 53 (2012) 620–635.
[30] J. Zhang, J. Wong, T. Li, Y. Pan, A comparison of parallel large-scale knowledge acquisition using rough set theory on different MapReduce runtime systems, Int. J. Approx. Reason. 55 (2014) 896–907.
[31] J. Zhang, Y. Zhu, Y. Pan, T. Li, A parallel implementation of computing composite rough set approximations on GPUs, in: RSKT2013, pp. 240–250.
[32] W. Ziarko, Variable precision rough set model, J. Comput. Syst. Sci. 46 (1993) 39–59.

Please cite this article in press as: S. Li et al., Parallel computing of approximations in dominance-based rough sets approach, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.05.003