Nature Inspired Population-Based Heuristics for Rough Set Reduction

Hongbo Liu[1], Ajith Abraham[2], and Yanheng Li[1]

[1] School of Computer Science and Engineering, Dalian Maritime University, 116026 Dalian, China
lhb@dlut.edu.cn
http://hongboliu.torrent.googlepages.com

[2] Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Trondheim, Norway
ajith.abraham@ieee.org
http://www.softcomputing.net

Summary. Finding reducts is one of the key problems in the increasing applications of rough set theory, which is also one of the bottlenecks of the rough set methodology. The population-based reduction approaches are attractive to find multiple reducts in the decision systems. In this chapter, we introduce two nature inspired population-based computational optimization techniques, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) for rough set reduction. Particle Swarm Optimization (PSO) is particularly attractive for the challenging problem as a new heuristic algorithm. The approach discover the best feature combinations in an efficient way to observe the change of positive region as the particles proceed throughout the search space. We evaluated the performance of the two algorithms using some benchmark datasets and the corresponding computational experiments are discussed. Empirical results indicate that both methods are ideal for all the considered problems and particle swarm optimization technique outperformed the genetic algorithm approach by obtaining more number of reducts for the datasets. We also illustrate a real world application in fMRI data analysis, which is helpful for cognition research.

1 Introduction

Rough set theory [1, 2, 3] provides a mathematical tool that can be used for both feature selection and knowledge discovery. It helps us to find out the minimal attribute sets called '*reducts*' to classify objects without deterioration of classification quality and induce minimal length decision rules inherent in a given information system. The idea of reducts has encouraged many researchers in studying the effectiveness of rough set theory in a number of real world domains, including medicine, pharmacology, control systems, fault-diagnosis, text categorization, social sciences, switching circuits, economic/financial prediction, image processing, and so on [4, 5, 6, 7, 8, 9, 10].

Usually real world objects are the corresponding tuple in some decision tables. They store a huge quantity of data, which is hard to manage from a computational point of view. Finding reducts in a large information system is still an

A. Abraham, R. Falcón, and R. Bello (Eds.): Rough Set Theory, SCI 174, pp. 261–278.
springerlink.com

NP-hard problem [11]. The high complexity of this problem has motivated investigators to apply various approximation techniques to find near-optimal solutions. Many approaches have been proposed for finding reducts, e.g., discernibility matrices, dynamic reducts, and others [12, 13]. The heuristic algorithm is a better choice. Hu et al. [14] proposed a heuristic algorithm using discernibility matrix. The approach provided a weighting mechanism to rank attributes. Zhong and Dong [15] presented a wrapper approach using rough sets theory with greedy heuristics for feature subset selection. The aim of feature subset selection is to find out a minimum set of relevant attributes that describe the dataset as well as the original all attributes do. So finding reduct is similar to feature selection. Zhong's algorithm employed the number of consistent instances as heuristics. Banerjee et al. [16] presented various attempts of using Genetic Algorithms in order to obtain reducts. Although several variants of reduct algorithms are reported in the literature, at the moment, there is no accredited best heuristic reduct algorithm. So far, it is still an open research area in rough sets theory.

Conventional approaches for knowledge discovery always try to find a good reduct or to select a set of features [17]. In the knowledge discovery applications, only the good reduct can be applied to represent knowledge, which is called a single body of knowledge. In fact, many information systems in the real world have multiple reducts, and each reduct can be applied to generate a single body of knowledge. Therefore, multi-knowledge based on multiple reducts has the potential to improve knowledge representation and decision accuracy [18]. However, it would be exceedingly time-consuming to find multiple reducts in an instance information system with larger numbers of attributes and instances. In most of strategies, different reducts are obtained by changing the order of condition attributes and calculating the significance of different condition attribute combinations against decision attribute(s). It is a complex multi-restart processing about condition attribute increasing or decreasing in quantity. Population-based search approaches are of great benefits in the multiple reduction problems, because different individual trends to be encoded to different reduct. So it is attractive to find multiple reducts in the decision systems.

Particle swarm algorithm is inspired by social behavior patterns of organisms that live and interact within large groups. In particular, it incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the Swarm Intelligence (SI) paradigm has emerged [19]. The swarm intelligent model helps to find optimal regions of complex search spaces through interaction of individuals in a population of particles [20, 21, 22]. As an algorithm, its main strength is its fast convergence, which compares favorably with many other global optimization algorithms [23, 24]. It has exhibited good performance across a wide range of applications [25, 26, 27, 28, 29]. The particle swarm algorithm is particularly attractive for feature selection as there seems to be no heuristic that can guide search to the optimal minimal feature subset. Additionally, it can be the case that particles discover the best feature combinations as they proceed throughout the search space.

The main focus of this chapter is to introduce how particle swarm optimization algorithm may be applied for the difficult problem of finding multiple reducts. The rest of the chapter is organized as follows. Some related terms and theorems on rough set theory are explained briefly in Sect. 2. The proposed approach based on particle swarm algorithm is presented in Sect. 3. In Sect. 4, experiment results and discussions are provided in detail. In Sect. 5, we illustrate an application in fMRI data analysis. Finally conclusions are made in Sect. 6.

2 Rough Set Reduction

The basic concepts of rough set theory and its philosophy are presented and illustrated with examples in [1, 2, 3, 15, 30, 31, 17]. Here, we illustrate only the relevant basic ideas of rough sets that are relevant to the present work.

In rough set theory, an information system is denoted in 4-tuple by $S = (U, A, V, f)$, where U is the universe of discourse, a non-empty finite set of N objects $\{x_1, x_2, \cdots, x_N\}$. A is a non-empty finite set of attributes such that $a : U \to V_a$ for every $a \in A$ (V_a is the value set of the attribute a).

$$V = \bigcup_{a \in A} V_a$$

$f : U \times A \to V$ is the total decision function (also called the information function) such that $f(x, a) \in V_a$ for every $a \in A$, $x \in U$. The information system can also be defined as a decision table by $S = (U, C, D, V, f)$. For the decision table, C and D are two subsets of attributes. $A = \{C \cup D\}$, $C \cap D = \emptyset$, where C is the set of input features and D is the set of class indices. They are also called condition and decision attributes, respectively.

Let $a \in C \cup D$, $P \subseteq C \cup D$. A binary relation $IND(P)$, called an equivalence (indiscernibility) relation, is defined as follows:

$$IND(P) = \{(x, y) \in U \times U \quad | \quad \forall a \in P, f(x, a) = f(y, a)\} \tag{1}$$

The equivalence relation $IND(P)$ partitions the set U into disjoint subsets. Let $U/IND(P)$ denote the family of all equivalence classes of the relation $IND(P)$. For simplicity of notation, U/P will be written instead of $U/IND(P)$. Such a partition of the universe is denoted by $U/P = \{P_1, P_2, \cdots, P_i, \cdots\}$, where P_i is an equivalence class of P, which is denoted $[x_i]_P$. Equivalence classes U/C and U/D will be called condition and decision classes, respectively.

Lower Approximation: Given a decision table $T = (U, C, D, V, f)$. Let $R \subseteq C \cup D$, $X \subseteq U$ and $U/R = \{R_1, R_2, \cdots, R_i, \cdots\}$. The R-lower approximation set of X is the set of all elements of U which can be with certainty classified as elements of X, assuming knowledge R. It can be presented formally as

$$APR_R^-(X) = \bigcup \{R_i \quad | \quad R_i \in U/R, R_i \subseteq X\} \tag{2}$$

Positive Region: Given a decision table $T = (U, C, D, V, f)$. Let $B \subseteq C$, $U/D = \{D_1, D_2, \cdots, D_i, \cdots\}$ and $U/B = \{B_1, B_2, \cdots, B_i, \cdots\}$. The B-positive region of D is the set of all objects from the universe U which can be classified with certainty to classes of U/D employing features from B, i.e.,

$$POS_B(D) = \bigcup_{D_i \in U/D} APR_B^-(D_i) \tag{3}$$

Positive Region: Given a decision table $T = (U, C, D, V, f)$. Let $B \subseteq C$, $U/D = \{D_1, D_2, \cdots, D_i, \cdots\}$ and $U/B = \{B_1, B_2, \cdots, B_i, \cdots\}$. The B-positive region of D is the set of all objects from the universe U which can be classified with certainty to classes of U/D employing features from B, i.e.,

$$POS_B(D) = \bigcup_{D_i \in U/D} B_-(D_i) \tag{4}$$

Reduct: Given a decision table $T = (U, C, D, V, f)$. The attribute $a \in B \subseteq C$ is $D - dispensable$ in B, if $POS_B(D) = POS_{(B-\{a\})}(D)$; otherwise the attribute a is $D - indispensable$ in B. If all attributes $a \in B$ are $D - indispensable$ in B, then B will be called $D - independent$. A subset of attributes $B \subseteq C$ is a $D - reduct$ of C, iff $POS_B(D) = POS_C(D)$ and B is $D - independent$. It means that a reduct is the minimal subset of attributes that enables the same classification of elements of the universe as the whole set of attributes. In other words, attributes that do not belong to a reduct are superfluous with regard to classification of elements of the universe. Usually, there are many reducts in an instance information system. Let 2^A represent all possible attribute subsets $\{\{a_1\}, \cdots, \{a_{|A|}\}, \{a_1, a_2\}, \cdots, \{a_1, \cdots, a_{|A|}\}\}$. Let RED represent the set of reducts, i.e.,

$$RED = \{B \quad | \quad POS_B(D) = POS_C(D), POS_{(B-\{a\})}(D) < POS_B(D)\} \tag{5}$$

Multi-knowledge: Given a decision table $T = (U, C, D, V, f)$. Let RED represent the set of reducts, Let φ is a mapping from the condition space to the decision space. Then multi-knowledge can be defined as follows:

$$\Psi = \{\varphi_B \quad | \quad B \in RED\} \tag{6}$$

Reduced Positive Universe and *Reduced Positive Region*: Given a decision table $T = (U, C, D, V, f)$. Let $U/C = \{[u_1^{'}]_C, [u_2^{'}]_C, \cdots, [u_m^{'}]_C\}$, Reduced Positive Universe $U^{'}$ can be written as:

$$U^{'} = \{u_1^{'}, u_2^{'}, \cdots, u_m^{'}\}. \tag{7}$$

and

$$POS_C(D) = [u_{i_1}^{'}]_C \cup [u_{i_2}^{'}]_C \cup \cdots \cup [u_{i_t}^{'}]_C. \tag{8}$$

Where $\forall u_{i_s}^{'} \in U^{'}$ and $|[u_{i_s}^{'}]_C / D| = 1 (s = 1, 2, \cdots, t)$. Reduced positive universe can be written as:

$$U_{pos}^{'} = \{u_{i_1}^{'}, u_{i_2}^{'}, \cdots, u_{i_t}^{'}\}. \tag{9}$$

and $\forall B \subseteq C$, reduced positive region

$$POS'_B(D) = \bigcup_{X \in U'/B \wedge X \subseteq U'_{pos} \wedge |X/D|=1} X \tag{10}$$

where $|X/D|$ represents the cardinality of the set X/D. $\forall B \subseteq C$, $POS_B(D) = POS_C(D)$ if $POS^{'}_B = U^{'}_{pos}$ [31]. It is to be noted that $U^{'}$ is the reduced universe, which usually would reduce significantly the scale of datasets. It provides a more efficient method to observe the change of positive region when we search the reducts. We didn't have to calculate U/C, U/D, U/B, $POS_C(D)$, $POS_B(D)$ and then compare $POS_B(D)$ with $POS_C(D)$ to determine whether they are equal to each other or not. We only calculate U/C, $U^{'}$, $U^{'}_{pos}$, $POS^{'}_B$ and then compare $POS^{'}_B$ with $U^{'}_{pos}$.

3 Nature Inspired Heuristics for Reduction

Combinatorial optimization problems are important in many real life applications and recently, the area has attracted much research with the advances in nature inspired heuristics and multi-agent systems.

3.1 Particle Swarm Optimization for Reduction

Given a decision table $T = (U, C, D, V, f)$, the set of condition attributes, C, consist of m attributes. We set up a search space of m dimension for the reduction problem. Accordingly, each particle's position is represented as a binary bit string of length m. Each dimension of the particle's position maps one condition attribute. The domain for each dimension is limited to 0 or 1. The value '1' means the corresponding attribute is selected while '0' not selected. Each position can be "decoded" to a potential reduction solution, an subset of C. The particle's position is a series of priority levels of the attributes. The sequence of the attribute will not be changed during the iteration. But after updating the velocity and position of the particles, the particle's position may appear real values such as 0.4, etc. It is meaningless for the reduction. Therefore, we introduce a discrete particle swarm optimization for this combinatorial problem.

During the search procedure, each individual is evaluated using the fitness. According to the definition of rough set reduct, the reduction solution must ensure the decision ability is the same as the primary decision table and the number of attributes in the feasible solution is kept as low as possible. In our algorithm, we first evaluate whether the potential reduction solution satisfies $POS^{'}_E = U^{'}_{pos}$ or not (E is the subset of attributes represented by the potential reduction solution). If it is a feasible solution, we calculate the number of '1' in it. The solution with the lowest number of '1' would be selected. For the particle swarm, the lower number of '1' in its position, the better the fitness of the individual is. $POS^{'}_E = U^{'}_{pos}$ is used as the criterion of the solution validity.

As a summary, the particle swarm model consists of a swarm of particles, which are initialized with a population of random candidate solutions. They move iteratively through the d-dimension problem space to search the new solutions, where the fitness f can be measured by calculating the number of condition attributes in the potential reduction solution. Each particle has a position represented by a position-vector $\mathbf{p}_i$ (i is the index of the particle), and a velocity represented by a velocity-vector $\mathbf{v}_i$. Each particle remembers its own best position so far in a vector $\mathbf{p}_i^{\#}$, and its j-th dimensional value is $p_{ij}^{\#}$. The best position-vector among the swarm so far is then stored in a vector $\mathbf{p}^*$, and its j-th dimensional value is p_j^*. When the particle moves in a state space restricted to zero and one on each dimension, the change of probability with time steps is defined as follows:

$$P(p_{ij}(t)=1) = f(p_{ij}(t-1), v_{ij}(t-1), p_{ij}^{\#}(t-1), p_j^*(t-1)). \tag{11}$$

where the probability function is

$$sig(v_{ij}(t)) = \frac{1}{1+e^{-v_{ij}(t)}}. \tag{12}$$

At each time step, each particle updates its velocity and moves to a new position according to Eqs.(13) and (14):

$$v_{ij}(t) = wv_{ij}(t-1)+\phi_1 r_1(p_{ij}^{\#}(t-1)-p_{ij}(t-1))+\phi_2 r_2(p_j^*(t-1)-p_{ij}(t-1)). \tag{13}$$

$$p_{ij}(t) = \begin{cases} 1 & \text{if } \rho < sig(v_{ij}(t)); \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Algorithm 1. A Rough Set Reduct Algorithm Based on Particle Swarm Optimization

Calculate $U^{'}$, $U_{pos}^{'}$ using Eqs.(7) and (9)
Initialize the size of the particle swarm n, and other parameters
Initialize the positions and the velocities for all the particles randomly
while the stop criterion is not met **do**
    $t \leftarrow t+1$
    Calculate the fitness value of each particle
    if $POS_E^{'} \neq U_{pos}^{'}$ **then**
        the fitness is punished as the total number of the condition attributes
    else
        the fitness is the number of '1' in the position
    end if
    $\mathbf{p}^* = argmin_{i=1}^{n}(f(\mathbf{p}^*(t-1)), f(\mathbf{p}_1(t)), f(\mathbf{p}_2(t)), \cdots, f(\mathbf{p}_i(t)), \cdots, f(\mathbf{p}_n(t)))$
    for i= 1 **to** n **do**
        $\mathbf{p}_i^{\#}(t) = argmin_{i=1}^{n}(f(\mathbf{p}_i^{\#}(t-1)), f(\mathbf{p}_i(t))$
        for $j = 1$ **to** d **do**
            Update the j-th dimension value of $\mathbf{p}_i$ and $\mathbf{v}_i$
            according to Eqs.(13) and (14)
        end for
    end for
end while

Where ϕ_1 is a positive constant, called as coefficient of the self-recognition component, ϕ_2 is a positive constant, called as coefficient of the social component. r_1 and r_2 are the random numbers in the interval [0,1]. The variable w is called as the inertia factor, which value is typically setup to vary linearly from 1 to near 0 during the iterated processing. ρ is random number in the closed interval [0, 1]. From Eq.(13), a particle decides where to move next, considering its current state, its own experience, which is the memory of its best past position, and the experience of its most successful particle in the swarm. The pseudo-code for the particle swarm search method is illustrated in Algorithm 1.

3.2 Genetic Algorithms for Reduction

In nature, evolution is mostly determined by natural selection, where individuals that are better are more likely to survive and propagate their genetic material. The encoding of genetic information (genome) is done in a way that admits asexual reproduction, which results in offspring's that are genetically identical to the parent. Sexual reproduction allows some exchange and re-ordering of chromosomes, producing offspring that contain a combination of information from each parent. This is the recombination operation, which is often referred to as crossover because of the way strands of chromosomes crossover during the exchange. Diversity in the population is achieved by mutation. A typical evolutionary (genetic) algorithm procedure takes the following steps: A population of candidate solutions (for the optimization task to be solved) is initialized. New solutions are created by applying genetic operators (mutation and/or crossover).

Algorithm 2. A Rough Set Reduct Algorithm Based on Genetic Algorithm

Calculate U', U'_{pos} using Eqs.(7) and (9)
Initialize the population randomly, and other parameters
while the stop criterion is not met **do**
    Evaluate the fitness of each individual in the population
    if $POS'_E \neq U'_{pos}$ **then**
        the fitness is punished as the total number of the condition attributes
    else
        the fitness is the number of '1' in the position
    end if
    Select best-ranking individuals to reproduce
    Breed new generation through crossover operator and give birth to offspring
    Breed new generation through mutation operator and give birth to offspring
    Evaluate the individual fitnesses of the offspring
    if $POS'_E \neq U'_{pos}$ **then**
        the fitness is punished as the total number of the condition attributes
    else
        the fitness is the number of '1' in the position
    end if
    Replace worst ranked part of population with offspring
end while

The fitness (how good the solutions are) of the resulting solutions are evaluated and suitable selection strategy is then applied to determine which solutions will be maintained into the next generation. The procedure is then iterated [38]. The pseudo-code for the genetic algorithm search method is illustrated in Algorithm 2.

4 Experiments Using Some Benchmark Problems

For all experiments, Genetic algorithm (GA) was used to compare the performance with PSO. The two algorithms share many similarities [33, 34]. Both methods are valid and efficient methods in numeric programming and have been employed in various fields due to their strong convergence properties. Specific parameter settings for the algorithms are described in Table 1, where D is the dimension of the position, i.e., the number of condition attributes. Besides the first small scale rough set reduction problem shown in Table 2, the maximum number of iterations is $(int)(0.1 * recnum + 10 * (nfields - 1))$ in each trial, where $recnum$ is the number of records/rows and $nfields - 1$ is the number of condition attributes. Each experiment (for each algorithm) was repeated 3 times with different random seeds. If the standard deviation is larger than 20%, the times of trials were set to larger, 10 or 20.

To analyze the effectiveness and performance of the considered algorithms, first we tested a small scale rough set reduction problem shown in Table 2. In the experiment, the maximum number of iterations was fixed as 10. Each experiment was repeated 3 times with different random seeds. The results (the

Table 1. Parameter settings for the algorithms

Algorithm	*ParameterName*	*Value*
GA		
	size of the population	$(int)(10 + 2 * sqrt(D))$
	Probability of crossover	0.8
	Probability of mutation	0.08
PSO		
	Swarm size	$(int)(10 + 2 * sqrt(D))$
	Self coefficient ϕ_1	1.49
	Social coefficient ϕ_2	1.49
	Inertia weight w	$0.9 \rightarrow 0.1$
	Clamping Coefficient ρ	0.5

Table 2. A decision table

Instance	c_1	c_2	c_3	c_4	d
x_1	1	1	1	1	0
x_2	2	2	2	1	1
x_3	1	1	1	1	0
x_4	2	3	2	3	0
x_5	2	2	2	1	1
x_6	3	1	2	1	0
x_7	1	2	3	2	2
x_8	2	3	1	2	3
x_9	3	1	2	1	1
x_{10}	1	2	3	2	2
x_{11}	3	1	2	1	1
x_{12}	2	3	1	2	3
x_{13}	4	3	4	2	1
x_{14}	1	2	3	2	3
x_{15}	4	3	4	2	2

number of reduced attributes) for 3 GA runs were all 2. The results of 3 PSO runs were also all 2. The optimal result is supposed to be 2. But the reduction result for 3 GA runs is $\{2, 3\}$ while the reduction results for 3 PSO runs are $\{1, 4\}$ and $\{2, 3\}$. Table 3 depicts the reducts for Table 2. Figure 1 shows the performance of the algorithms for Table 2. For the small scale rough set reduction problem, GA has faster convergence than PSO. There seems like a conflict between the instances 13 and 15. It depends on conflict analysis and how to explain the obtained knowledge, which is beyond the scope of this chapter.

Further we consider the datasets in Table 4 from AFS[1], AiLab[2] and UCI[3]. Figures 2, 3, 4 and 5 illustrate the performance of the algorithms for lung-cancer, lymphography and mofn-3-7-10 datasets, respectively. For lung-cancer dataset, the results (the number of reduced attributes) for 3 GA runs were 10: {1, 3, 9, 12, 33, 41, 44, 47, 54, 56} (The number before the colon is the number of condition attributes, the numbers in brackets are attribute index, which represents a reduction solution). The results of 3 PSO runs were 9: { 3, 8, 9,

[1] http://sra.itc.it/research/afs/
[2] http://www.ailab.si/orange/datasets.asp
[3] http://www.datalab.uci.edu/data/mldb-sgi/data/

Table 3. A reduction of the data in Table 2

Reduct	*Instance*	c_1	c_2	c_3	c_4	d
$\{1,4\}$						
	x_1	1			1	0
	x_2	2			1	1
	x_4	2			3	0
	x_6	3			1	0
	x_7	1			2	2
	x_8	2			2	3
	x_9	3			1	1
	x_{13}	4			2	1
	x_{14}	1			2	3
	x_{15}	4			2	2
$\{2,3\}$						
	x_1		1	1		0
	x_2		2	2		1
	x_4		3	2		0
	x_6		1	2		0
	x_7		2	3		2
	x_8		3	1		3
	x_9		1	2		1
	x_{13}		3	4		1
	x_{14}		2	3		3
	x_{15}		3	4		2

12, 15, 35, 47, 54, 55}, 10: {2, 3, 12, 19, 25, 27, 30, 32, 40, 56}, 8: {11, 14, 24, 30, 42, 44, 45, 50}. For zoo dataset, the results of 3 GA runs all were 5: {3, 4, 6, 9, 13}, the results of 3 PSO runs were 5: {3, 6, 8, 13, 16, }, 5: {4, 6, 8, 12, 13}, 5: {3, 4, 6, 8, 13}. For lymphography dataset, the results of 3 GA runs all were 7: {2, 6, 10, 13, 14, 17, 18}, the results of 3 PSO runs were 6: {2, 13, 14, 15, 16, 18}, 7: {1, 2, 13, 14, 15, 17, 18}, 7: {2, 10, 12, 13, 14, 15, 18}. For mofn-3-7-10 dataset, the results of 3 GA runs all were 7: {3, 4, 5, 6, 7, 8, 9} and

Table 4. Data sets used in the experiments

Dataset	*Size*	*ConditionAttributes*	*Class*	*GA*	*PSO*
lung-cancer	32	56	3	10	8
zoo	101	16	7	5	5
corral	128	6	2	4	4
lymphography	148	18	4	7	6
hayes-roth	160	4	3	3	3
shuttle-landing-control	253	6	2	6	6
monks	432	6	2	3	3
xd6-test	512	9	2	9	9
balance-scale	625	4	3	4	4
breast-cancer-wisconsin	683	9	2	4	4
mofn-3-7-10	1024	10	2	7	7
parity5+5	1024	10	2	5	5

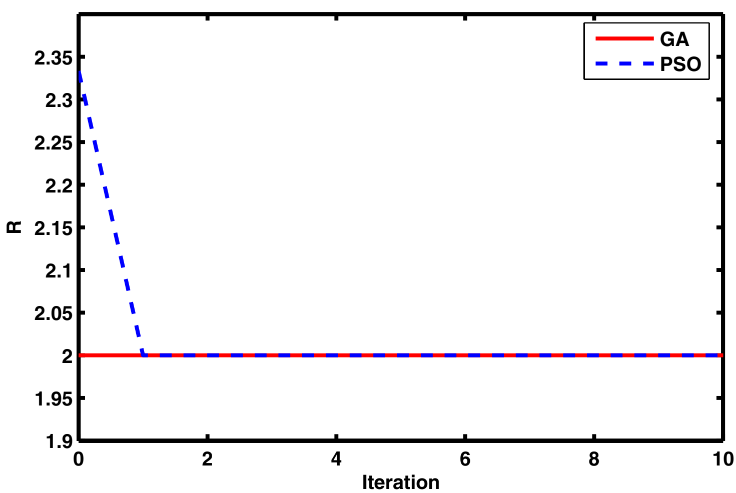

Fig. 1. Performance of rough set reduction for the data in Table 2

the results of 3 PSO runs were 7: {3, 4, 5, 6, 7, 8, 9}. Other results are shown in Table 4, in which only the best objective results are listed. PSO usually obtained better results than GA, specially for the large scale problems. Although GA and

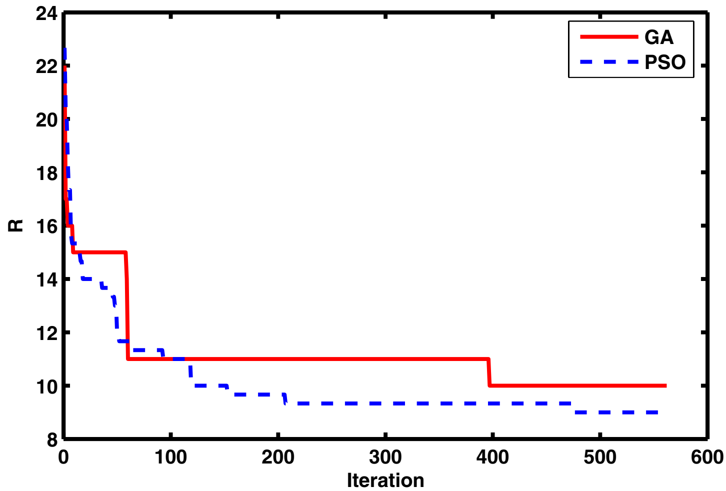

Fig. 2. Performance of rough set reduction for lung-cancer dataset

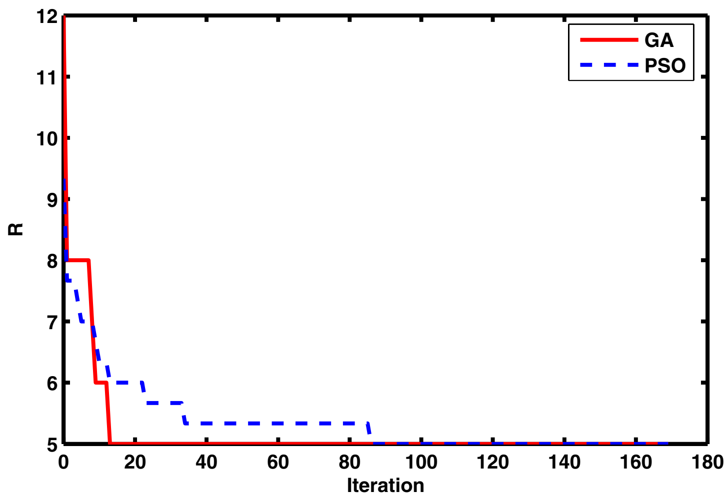

Fig. 3. Performance of rough set reduction for zoo dataset

PSO achieved the same results, PSO usually requires only very few iterations, as illustrated in Fig. 4. It indicates that PSO have a better convergence than GA for the larger scale rough set reduction problem, although PSO is worst for some small scale rough set reduction problems. It is to be noted that PSO usually can obtain more candidate solutions for the reduction problems.

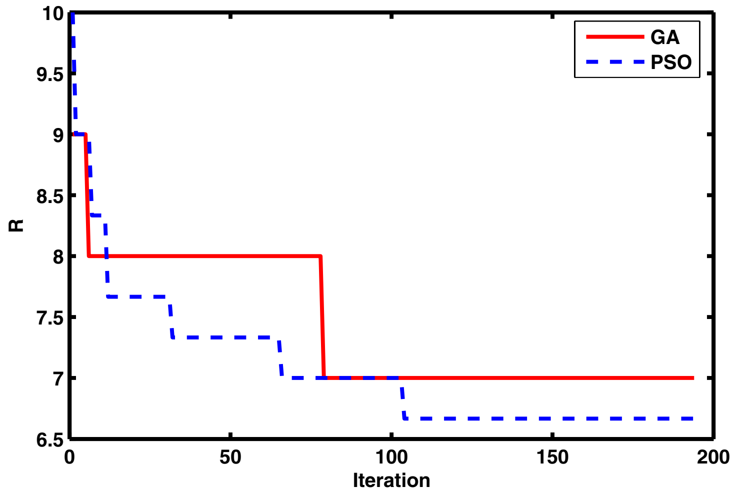

Fig. 4. Performance of rough set reduction for lymphography dataset

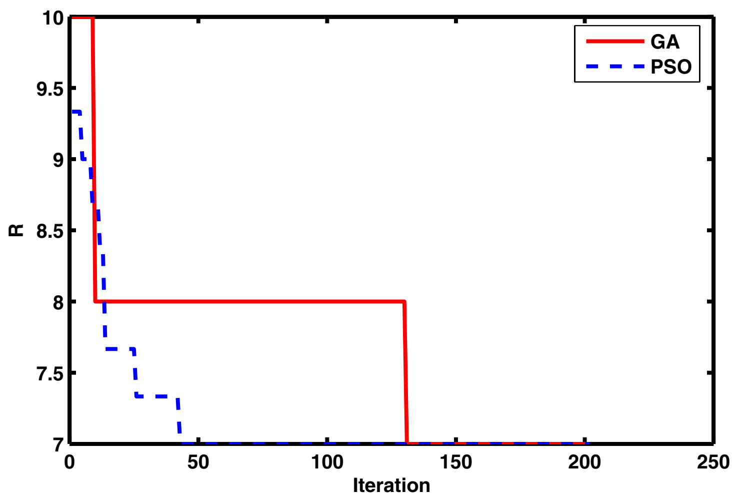

Fig. 5. Performance of rough set reduction for mofn-3-7-10 dataset

5 Application in fMRI Data Analysis

Functional Magnetic Resonance Imaging (fMRI) is one of the most important tools for Neuroinformatics, which combines neuroscience and informatics science and computational science to develop approaches needed to understand human brain [35]. The study of human brain function has received a tremendous boost in recent years due to the advent of the new brain imaging technique.

With the development of the new technology, a mass of fMRI data is collected ceaselessly. These datasets implicate very important information, which need to be extracted and translated to intelligible knowledge. Recently most of the research are focused on the activation features on the Region of Interest (ROI) through statistical analysis for single experiment or using only a few data. Neuroscientists or psychologists provide explanation for the experimental results, which depends strongly on their accumulative experience and subjective tendency. What is more, it is difficult to deal with slightly large datasets. So it is exigent to develop some computational intelligence methods to analyze them effectively and objectively. Rough set theory provides a novel approach to reduct the fMRI data and extract meaningful knowledge. There are usually many reducts in the information system, which can be applied to generate multi-knowledge. The rough set approach consists of several steps leading towards the final goal of generating rules [36].

The main steps of the rough set approach are: (1)mapping of the information from the original database into the decision system format; (2) completion of data; (3) discretization of data; (4) computation of reducts from data; (5) derivation of rules from reducts; (6) filtering of rules. One of most important task is the data reduction process.

Algorithm 3. Feature selection & extraction algorithm for fMRI data

Step 1. Find out the most active voxels in several regions of brain under the t-test of basic models in SPM99 and save their coordinates
Step 2. Scan fMRI image and search the voxels according to the coordinates saved
Step 3. Respectively average all voxels in the spherical region whose center is corresponding saved voxel and whose radius is a predefined constant. These results of a single image are formed one feature vector
Step 4. If the image isn't the last one, go to Step 2, otherwise, end

A typical normalized image contains more than 500,000 voxels, so it is impossible that feature vector can contain so immense voxels. We transform datasets from MNI template to Talairach coordinate system. Then we can use the region information in Talairach as features to reduce the dimensionality of the images. We used a SPM99 software package[4] and in-house programs for image processing, including corrections for head motion, normalization and global fMRI signal shift [37]. A simplified workflow is illustrated in Fig. 6. Feature selection & extraction algorithm for fMRI data is described in Algorithm 3. The location for feature selection & extraction is shown in Fig. 7.

We analyzed the fMRI data from three cognition experiments: Tongue movement experiment, Associating Chinese verb experiment, and Looking at or silent reading Chinese word experiment. They are involved in 9 tasks: 0 - Control task; 1 - Tongue movement; 2 - Associating verb from single noun; 3 - Associating verb

[4] http://www.fil.ion.ucl.ac.uk/spm/

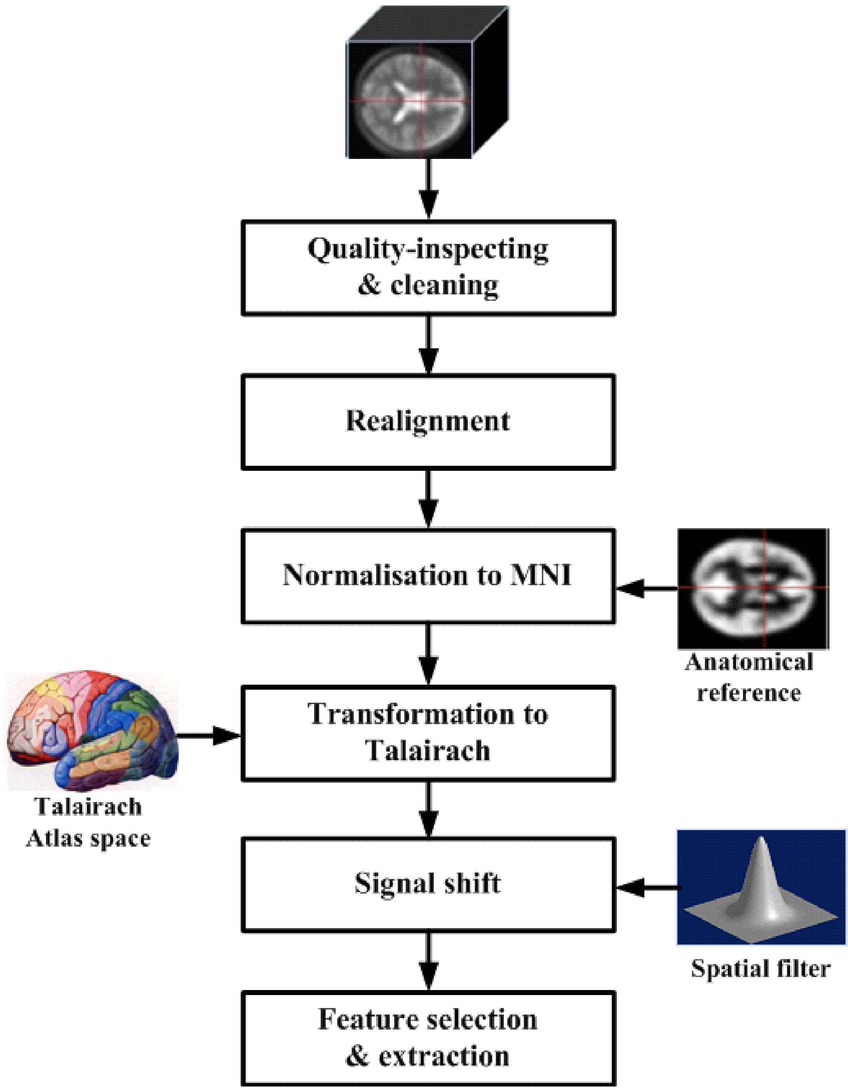

Fig. 6. Pre-processing workflow for fMRI data

from single non-noun; 4 - Making verb before single word; 5 - Looking at number; 6 - Silent reading Number; 7 - Looking at Chinese word; 8 - Silent reading Chinese word. Some of rules are described as follows:

Rule1: if M1=2, SMA=2, Broca=2 then Task=1;
Rule2: if BAs { 7,19,20,40,44,45 } =3, BSC=2 then Task=2;
Rule3: if BAs { 10,11,13,44,45 } =3, BSC=1 then Task=3;

Fig. 7. Developed software interface for feature selection and extraction

Rule4: if BAs { 7,19,40 } =3, BSC=3 then Task=4;
Rule5: if SMA=2, Broca=3 then Task=6;
Rule6: if SMA=2, Broca=2, Wernike=3 then Task=8.

6 Conclusions

In this Chapter, we introduced the problem of finding optimal reducts using particle swarm optimization and genetic algorithm approaches. The considered approaches discovered the good feature combinations in an efficient way to observe the change of positive region as the particles proceed throughout the search space. Population-based search approaches are of great benefits in the multiple reduction problems, because different individual trends to be encoded to different reduct. Empirical results indicate that PSO usually required shorter time to obtain better results than GA, specially for large scale problems, although its stability need to be improved in further research. PSO have a better convergence than GA for the larger scale rough set reduction problem, although PSO is worst for some small scale rough set reduction problems. PSO also can obtain more candidate solutions for the reduction problems. The population-based algorithms could be ideal approaches for solving the reduction problem. We also illustrated an application in fMRI data analysis. Although the correctness of the rules need neuroscientists to analyze and verify further, the approach is helpful for cognition research.

Acknowledgements

This work was partly supported by NSFC (60373095) and DLMU (DLMU-ZL-200709).

References

1. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
2. Pawlak, Z.: Rough Sets: Present State and The Future. Foundations of Computing and Decision Sciences 18, 157–166 (1993)
3. Pawlak, Z.: Rough Sets and Intelligent Data Analysis. Information Sciences 147, 1–12 (2002)
4. Kusiak, A.: Rough Set Theory: A Data Mining Tool for Semiconductor Manufacturing. IEEE Transactions on Electronics Packaging Manufacturing 24, 44–50 (2001)
5. Shang, C., Shen, Q.: Rough Feature Selection for Neural Network Based Image Classification. International Journal of Image and Graphics 2, 541–555 (2002)
6. Francis, E.H., Tay, S.L.: Economic And Financial Prediction Using Rough Sets Model. European Journal of Operational Research 141, 641–659 (2002)
7. Świniarski, R.W., Skowron, A.: Rough Set Methods in Feature Selection and Recognition. Pattern Recognition Letters 24, 833–849 (2003)
8. Beaubouef, T., Ladner, R., Petry, F.: Rough Set Spatial Data Modeling for Data Mining. International Journal of Intelligent Systems 19, 567–584 (2004)
9. Shen, L., Francis, E.H.: Tay Applying Rough Sets to Market Timing Decisions. Decision Support Systems 37, 583–597 (2004)
10. Gupta, K.M., Moore, P.G., Aha, D.W., Pal, S.K.: Rough set feature selection methods for case-based categorization of text documents. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 792–798. Springer, Heidelberg (2005)
11. Boussouf, M.: A Hybrid Approach to Feature Selection. In: Żytkow, J.M. (ed.) PKDD 1998. LNCS, vol. 1510. Springer, Heidelberg (1998)
12. Skowron, A., Rauszer, C.: The Discernibility Matrices and Functions in Information Systems. In: Świniarski, R.W. (ed.) Handbook of Applications and Advances of the Rough Set Theory, pp. 331–362. Kluwer Academic Publishers, Dordrecht (1992)
13. Zhang, J., Wang, J., Li, D., He, H., Sun, J.: A new heuristic reduct algorithm base on rough sets theory. In: Dong, G., Tang, C., Wang, W. (eds.) WAIM 2003. LNCS, vol. 2762, pp. 247–253. Springer, Heidelberg (2003)
14. Hu, K., Diao, L., Shi, C.: A heuristic optimal reduct algorithm. In: Leung, K.-S., Chan, L., Meng, H. (eds.) IDEAL 2000. LNCS, vol. 1983, pp. 139–144. Springer, Heidelberg (2000)
15. Zhong, N., Dong, J.: Using Rough Sets with Heuristics for Feature Selection. Journal of Intelligent Information Systems 16, 199–214 (2001)
16. Banerjee, M., Mitra, S., Anand, A.: Feature Selection Using Rough Sets. Studies in Computational Intelligence 16, 3–20 (2006)
17. Wu, Q., Bell, D.: Multi-knowledge Extraction and Application. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS, vol. 2639, pp. 574–575. Springer, Heidelberg (2003)
18. Wu, Q.: Multiknowledge and Computing Network Model for Decision Making and Localisation of Robots. Thesis, University of Ulster (2005)
19. Kennedy, J., Eberhart, R.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
20. Clerc, M., Kennedy, J.: The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation 6, 58–73 (2002)

21. Clerc, M.: Particle Swarm Optimization. ISTE Publishing Company, London (2006)
22. Liu, H., Abraham, A., Clerc, M.: Chaotic Dynamic Characteristics in Swarm Intelligence. Applied Soft Computing Journal 7, 1019–1026 (2007)
23. Parsopoulos, K.E., Vrahatis, M.N.: Recent approaches to global optimization problems through particle swarm optimization. Natural Computing 1, 235–306 (2002)
24. Abraham, A., Guo, H., Liu, H.: Swarm Intelligence: Foundations, Perspectives and Applications. In: Nedjah, N., Mourelle, L. (eds.) Swarm Intelligent Systems. Studies in Computational Intelligence, pp. 3–25. Springer, Heidelberg (2006)
25. Salman, A., Ahmad, I., Al-Madani, S.: Particle Swarm Optimization for Task Assignment Problem. Microprocessors and Microsystems 26, 363–371 (2002)
26. Sousa, T., Silva, A., Neves, A.: Particle Swarm Based Data Mining Algorithms for Classification Tasks. Parallel Computing 30, 767–783 (2004)
27. Liu, B., Wang, L., Jin, Y., Tang, F., Huang, D.: Improved Particle Swarm Optimization Combined With Chaos. Chaos, Solitons and Fractals 25, 1261–1271 (2005)
28. Schute, J.F., Groenwold, A.A.: A Study of Global Optimization Using Particle Swarms. Journal of Global Optimization 3, 108–193 (2005)
29. Liu, H., Abraham, A., Clerc, M.: An Hybrid Fuzzy Variable Neighborhood Particle Swarm Optimization Algorithm for Solving Quadratic Assignment Problems. Journal of Universal Computer Science 13(7), 1032–1054 (2007)
30. Wang, G.: Rough Reduction in Algebra View and Information View. International Journal of Intelligent Systems 18, 679–688 (2003)
31. Xu, Z., Liu, Z., Yang, B., Song, W.: A Quick Attibute Reduction Algorithm with Complexity of Max$(O(|C||U|), O(|C|^2|U/C|))$. Chinese Journal of Computers 29, 391–399 (2006)
32. Wu, Q.X., Bell, D.A.: Multi-Knowledge Extraction and Application. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS, vol. 2639, pp. 274–279. Springer, Heidelberg (2003)
33. Cantú-Paz, E.: Efficient and Accurate Parallel Genetic Algorithms. Kluwer Academic publishers, Netherland (2000)
34. Abraham, A.: Evolutionary Computation. In: Sydenham, P., Thorn, R. (eds.) Handbook for Measurement Systems Design, pp. 920–931. John Wiley and Sons Ltd., London (2005)
35. Arbib, M.A., Grethe, J.S. (eds.): Computing the Brain: A Guide to Neuroinformatics. Academic Press, London (2001)
36. Røed, G.: Knowledge Extraction from Process Data: A rough Set Approach to Data Mining on Time Series. Knowledge Extraction from Process data. Thesis, Norwegian University of Science and Technology (1999)
37. Ji, Y., Liu, H., Wang, X., Tang, Y.: Congitive States Classification from fMRI Data Using Support Vector Machines. In: Proceedings of The Third International Conference on Machine Learning and Cybernetics, pp. 2920–2923. IEEE Computer Society Press, Los Alamitos (2004)
38. Goldberg, D.E.: Genetic Algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Corporation, Inc., Reading (1989)