



Finding Optimal Rough Set Reduct with A* Search Algorithm

Abhimanyu Bar, Anil Kumar and P.S.V.S Sai Prasad

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

March 1, 2021

Finding Optimal Rough Set Reduct with A^* Search Algorithm

Abhimanyu Bar¹, Anil Kumar², and P. S. V. S. Sai Prasad³

School of Computer and Information Sciences, University of Hyderabad,
Hyderabad-500046

¹abhi16@uohyd.ac.in, ²anilhcu@uohyd.ac.in, ³saics@uohyd.ernet.in

Abstract. Feature subset selection or reduct computation is a prominent domain for the classical rough set theory, which can preserve the most predictive features of a decision system. A given decision system has several reducts. Computation of all possible reducts was achieved through the computing prime implicants of the discernibility function. Currently, an optimal reduct based on any optimality criteria can only be achieved post-generation of all possible reducts. Indeed, it is an NP-hard problem. Several researchers have extended the alternative aspects with search strategies such as Genetic Algorithm, Ant Colony Optimization, Simulated Annealing, etc., for obtaining near-optimal reducts. In this paper, we propose an admissible and consistent heuristic for computing the optimal reduct having least number of induced equivalence classes or granules. A^* RSOR reduct computation algorithm is developed using the proposed consistent heuristic in A^* search. The proposed approach is validated both theoretically and experimentally. The comparative results establish the relevance of the proposed optimality criterion as the achieved optimal reduct has obtained significantly better accuracies with different classifiers.

Keywords: Rough Sets · Attribute Reduction · Feature Subset Selection · Optimal Reduct · A^* Search · Consistent Heuristic

1 Introduction

The task of feature subset selection (FS) is a necessary preprocessing step for building learning models that increases predictive accuracy and model comprehensibility. Finding informative features is the most challenging task under inconsistent and imprecise information. The classical rough set theory (RST), proposed by Pawlak [14], is an essential mathematical method for dealing imprecise information without additional information about data. RST is only applicable to categorical decision systems [14] thus, it requires a prior discretization for numerical decision systems.

A given decision system has many reducts, and even a single reduct has the adequate capability for inducing a reliable classification model. The generalizability of classifier performance induced by the reduct varies from one reduct to others, and there is no guarantee that the preferred single reduct can impose the best performance. Hence, researchers are interested in computing the

best (optimal) reduct out of all possible reducts. Several heuristics approaches based on dependency measure, and discernibility matrix for single reduct computation are proposed in literature [8,15,17]. Even though these approaches are computationally efficient with polynomial time complexity, they can't assure the computation of optimal reduct.

In 1992, A. Skowron et al. [17] introduced boolean reasoning based approach for all reduct computation using crisp discernibility matrix. An optimal reduct can thus be selected by introducing the optimality criterion on all possible reducts. At present, this is the only way to obtain an optimal reduct. Though, in RST computing minimal/all reducts is an NP-hard problem [17]. Subsequently, several aspects has been investigated using the evolutionary algorithm such as genetic algorithm (GA)[19], ant colony optimization (ACO) [9], simulated annealing [11], particle swarm optimization (PSO) [18] etc., for near-optimal reduct computation. In this context, J. Wroblewski [19] proposed three approaches by combining GA with the greedy heuristics to generate minimal reduct. Jensen et al. [9] adopted a stochastic approach based on ACO to create a near-optimal reduct. While Wang et al. [18] proposed a reduct computation approach through PSO, Chen et al. [2] incorporated fish swarm optimization with rough sets for finding the reduct. Jensen et al.[11] proposed a mechanism for feature selection that combined the simulated annealing algorithm with the rough set theory.

However, most of the existing optimal/near-optimal reduct computation approaches are formulated with the optimality criteria of reduct having a minimum number of features (shortest length reduct). The dependency measures, such as gamma measure [8,20], conditional information entropy measure [10], discernibility based measure [1,17] etc., are used in reduct computation algorithms favouring those attributes with larger domain cardinality as the resultant finer granular space achieves better dependency measure value and can result in reduct with the smallest size. There is a correspondence between reduct and rule induction. Rules are induced from the granules(equivalence classes) obtained using reduct attributes [4]. In a finer granular space, the cardinality of rules will be larger, and the strength of the rule will be smaller, which can affect the generalizability of reduct induced classifiers. In contrast, coarser granular space will generate a smaller size rule set with higher strength and can induce better classifiers. Hence, in this work the optimality criterion is taken as minimizing the cardinality of equivalence classes or granules induced by reduct attributes. Based on this criterion, the resulting optimal reduct will produce coarsest granular space.

In this work, a consistent heuristic is proposed based on the considered optimality criterion. An optimal reduct computation algorithm (A^*RSOR) is developed by using the proposed consistent heuristic in A^* search. The resulting approach acquires a significance as an optimal reduct can be computed without generation of all possible reducts. The significance of A^*RSOR is validated both theoretically as well as through experimental analysis.

The remaining part of this paper is structured in the following order. The theoretical background of the classical rough sets along with the preliminaries of relative dependency measure and A^* search strategy are discussed in section 2.

In section 3, the detailed theoretical explanation of the proposed optimal rough set reduct computation algorithm A^*RSOR is explored. Section 4 covers the comparative experimental evaluation and analysis of results. Lastly, in Section 5, the paper ends with the remarks and the future possibility of the proposed method.

2 Theoretical Background

2.1 Rough set theory

The decision system is represented as $DS = (\mathbb{U}, \mathbb{C} \cup \mathbb{D})$, where \mathbb{U} represents the non-empty finite set of objects called universe, \mathbb{C} is the set of conditional attributes such that $a : \mathbb{U} \rightarrow V_a, \forall a \in \mathbb{C}$ where V_a is the set of domain values with respect to 'a'. \mathbb{D} is a set of decision attributes or response variables, usually $\mathbb{D} = \{d\}$ containing a single decision attribute. For any arbitrary subset $\mathbb{B} \subseteq \mathbb{C}$, there exist an associated equivalence relation called indiscernibility relation $IND(\mathbb{B})$ defined as:

$$IND(\mathbb{B}) = \{(u_i, u_j) \in \mathbb{U}^2 \mid \forall a \in \mathbb{B}, a(u_i) = a(u_j)\} \quad (1)$$

The collection of all equivalence classes (granules) of $IND(\mathbb{B})$ is represented as $\mathbb{U}/IND(\mathbb{B})$ or \mathbb{U}/\mathbb{B} , and is defined as: $\mathbb{U}/IND(\mathbb{B}) = \otimes \{a \in \mathbb{B} : \mathbb{U}/IND(\{a\})\}$, where \otimes is the refinement operator. In the rest of the paper granules of $IND(\mathbb{B})$ are represented as \mathbb{U}/\mathbb{B} .

The set of granules \mathbb{U}/\mathbb{B} constitute the granular space through which rough set based approximations are defined. Let $X \subseteq \mathbb{U}$ be the concept to be approximated, then the \mathbb{B} -lower and \mathbb{B} -upper approximations of X are computed as follows:

$$\underline{\mathbb{B}}X = \{u \in \mathbb{U} \mid [u]_{\mathbb{B}} \subseteq X\} \text{ and } \overline{\mathbb{B}}X = \{u \in \mathbb{U} \mid [u]_{\mathbb{B}} \cap X \neq \emptyset\} \quad (2)$$

where $[u]_{\mathbb{B}}$ is the equivalence class of 'u'.

The positive region $POS_{\mathbb{B}}(\mathbb{D})$ is the collection of all objects that are certainly belongs to the concept \mathbb{D} , defined as:

$$POS_{\mathbb{B}}(\mathbb{D}) = \bigcup_{X \in \mathbb{U}/\mathbb{D}} \underline{\mathbb{B}}X \quad (3)$$

Consistent Decision System: A decision system DS is said to be a consistent decision system (CDS), if and only if $POS_{\mathbb{C}}(\mathbb{D}) = \mathbb{U}$. In case the given decision system is inconsistent, then the generalized decision operator has given in [5] is applied to convert DS into CDS . Hence in the rest of the paper, DS is assumed to be consistent.

2.2 Relative Dependency Measure

Han. et al. [6] defined a dependency measure named as relative dependency. Let $\mathbb{B} \subseteq \mathbb{C}$, then the degree of relative attribute dependency, denoted as $\kappa_{\mathbb{B}}(\mathbb{D})$, of \mathbb{D} over \mathbb{B} , is defined as:

$$\kappa_{\mathbb{B}}(\mathbb{D}) = \frac{|\mathbb{U}/\mathbb{B}|}{|\mathbb{U}/(\mathbb{B} \cup \mathbb{D})|} \quad (4)$$

\mathbb{B} is a reduct of CDS , if only if, $\kappa_{\mathbb{B}}(\mathbb{D}) = \kappa_{\mathbb{C}}(\mathbb{D}) = 1$ and $\forall Q \subset \mathbb{B}, \kappa_Q(\mathbb{D}) \neq \kappa_{\mathbb{C}}(\mathbb{D})$. Here $|S|$ for any set S , represents the cardinality of S .

2.3 A* Search Algorithm

In several domains, the solution to a problem can be formulated as a state-space search, represented as a graph, which contains a source node and several possible goal nodes. The solution is given in the form of a path or state according to the nature of the problem. A* is a popular search strategy [7], introduced as part of Shakey Project [13]. This is a complete and optimal search algorithm, which was formulated by combining the Dijkstra's Shortest Path and Best-First Search mechanisms. Here along with the path cost, there is a prediction heuristic cost for reaching the goal node from a particular state 'n' is associated. Hence, the total cost to reach the goal state can be estimated as:

$$f(n) = g(n) + h(n) \quad (5)$$

where $g(n)$ denotes the path cost from the source node to the current node 'n' and $h(n)$ is the prediction heuristic which estimates the cost to reach the goal from node 'n'. Any node with $h(n)$ as zero is a candidate goal node.

Generally, A* uses the openlist, which is defined as a priority queue to conduct the repeated selection of the least cost nodes to explore and closelist defined as the collection of explored nodes. The process is started by placing the source node into the *openlist*. At each iteration, the node with the lowest $f(n)$ value is removed from the *openlist* queue and placed in *closelist* queue, and their corresponding successor nodes with updated f values are added to the *openlist* queue. If the node selected for exploration from openlist has the heuristic values of zero, then A* algorithm stops and returns the solution associated with the node.

A* search is optimal, if $h(n)$ is admissible and consistent. A heuristic is admissible, if the value of $h(n)$ never overestimate the actual cost. The heuristic is consistent, if $h(n) \leq c(n, n') + h(n')$ provided that n' is the child node of n in the search space graph and $c(n, n')$ denotes the corresponding edge cost. In general, every consistent heuristic is admissible [7].

3 A*RSOR Search Algorithm

The purpose of the A*RSOR method is to find the reduct with the coarsest granular space for inducing least number of rules. Let $RED(CDS)$ represent set of all possible reducts for CDS , then an optimal reduct \mathbb{B}^* with the coarsest granular space must satisfy the property:

$$|\mathbb{U}/\mathbb{B}^*| = \min_{\mathbb{B} \in RED(CDS)} |\mathbb{U}/\mathbb{B}| \quad (6)$$

The computation of reduct is a search problem in the search space of the power set of \mathbb{C} . The traditional SBE and SFS control strategies are the example of hill climbing approaches in this search space. The goal state corresponds to a reduct in $RED(CDS)$. In formulating A* reduct computation algorithm, a heuristic function is needed for evaluating the cost of reaching any goal state. We have devised a heuristic function for states in the search space of reduct computation in the next subsection.

3.1 Partition Refinement Heuristic(PR-heuristic)

The Kappa measure $\kappa_{\mathbb{B}}(\mathbb{D})$ is a measure of the purity of granular space induced by \mathbb{B} . A granule $g \in \mathbb{U}/\mathbb{B}$ is pure if ‘ g ’ contains objects of single decision class. A granular space is said to be a pure granular space if all of its granules are pure. For a pure granule, there is no further refinement takes place with the inclusion of decision attribute. As every reduct of CDS induces a pure granular space, we have $\kappa_{\mathbb{B}}(\mathbb{D}) = 1, \forall \mathbb{B} \in RED(CDS)$ as $|\mathbb{U}/\mathbb{B}| = |\mathbb{U}/(\mathbb{B} \cup \mathbb{D})|$.

If an attribute collection \mathbb{B} is almost pure, then many of the granules in \mathbb{U}/\mathbb{B} are pure, that is $|\mathbb{U}/(\mathbb{B} \cup \mathbb{D})|$ will be slightly higher than $|\mathbb{U}/\mathbb{B}|$ as only the remaining granules will participate in the refinement resulting in $\kappa_{\mathbb{B}}(\mathbb{D})$ near to one. If \mathbb{B} induces almost impure granular space, then $|\mathbb{U}/(\mathbb{B} \cup \mathbb{D})|$ is much higher than $|\mathbb{U}/\mathbb{B}|$ resulting in $\kappa_{\mathbb{B}}(\mathbb{D})$ which is near to zero. Hence, the number of splits additionally occurring through the refinement of $IND(\mathbb{B})$ into $IND(\mathbb{B} \cup \mathbb{D})$ estimates the furtherness of the current attribute set \mathbb{B} in becoming reduct. This motivated us to formulate a heuristic named as partition refinement heuristic (PR-Heuristic) to estimate the cost attributes set \mathbb{B} in becoming the reduct. For $\mathbb{B} \subseteq \mathbb{C}$, PR-heuristic h_{PR} is given by

$$h_{PR}(\mathbb{B}) = |\mathbb{U}/(\mathbb{B} \cup \mathbb{D})| - |\mathbb{U}/\mathbb{B}| \quad (7)$$

Here $h_{PR}(\mathbb{B})$ is the number of split (refinements) in \mathbb{U}/\mathbb{B} occurring with inclusion of \mathbb{D} . In the reduct computation search space, let the child node of \mathbb{B} is $\mathbb{B}' = \mathbb{B} \cup \{a\}$ for any $a \in \mathbb{C} - \mathbb{B}$. The cost of refinement from \mathbb{B} to \mathbb{B}' is $c(\mathbb{B}, \mathbb{B}') = |\mathbb{U}/\mathbb{B}'| - |\mathbb{U}/\mathbb{B}|$. h_{PR} is said to be consistent heuristic, if and only if $h_{PR}(\mathbb{B}) \leq h_{PR}(\mathbb{B}') + c(\mathbb{B}, \mathbb{B}')$. Theorem-1 gives the proof for establishing that h_{PR} is a consistent heuristic.

Theorem 1. *The partition refinement heuristic h_{PR} for state space of reduct computation in CDS is a consistent heuristic.*

Proof. Let h_{PR} denote the PR-heuristic. Let $\mathbb{B} \subseteq \mathbb{C}$ denotes a node being explored in A^* algorithm, and for any $a \in \mathbb{C} - \mathbb{B}$, a new node $\mathbb{B}' = \mathbb{B} \cup \{a\}$ is generated. Using Eqn.7, it follows as:

$$h_{PR}(\mathbb{B}) = |\mathbb{U}/(\mathbb{B} \cup \mathbb{D})| - |\mathbb{U}/\mathbb{B}| \text{ and } h_{PR}(\mathbb{B}') = |\mathbb{U}/(\mathbb{B} \cup \{a\} \cup \mathbb{D})| - |\mathbb{U}/(\mathbb{B} \cup \{a\})|$$

The edge cost $c(\mathbb{B}, \mathbb{B}') = |\mathbb{U}/(\mathbb{B} \cup \{a\})| - |\mathbb{U}/\mathbb{B}|$, then consider,

$$h_{PR}(\mathbb{B}) + c(\mathbb{B}, \mathbb{B}') = |\mathbb{U}/(\mathbb{B} \cup \{a\} \cup \mathbb{D})| - |\mathbb{U}/\mathbb{B}| \geq |\mathbb{U}/(\mathbb{B} \cup \mathbb{D})| - |\mathbb{U}/\mathbb{B}| = h_{PR}(\mathbb{B})$$

(Since $IND(\mathbb{B} \cup \{a\} \cup \mathbb{D})$ is a refinement of $IND(\mathbb{B} \cup \mathbb{D})$). Hence, it is proved that h_{PR} is a consistent heuristic.

3.2 A* Algorithm with PR-Heuristic

An A^* search based algorithm is formulated using PR-heuristic as A^* rough set optimal reduct (A^*RSOR) algorithm, which is presented in algorithm 1. For a set of attributes $\mathbb{B} \subseteq \mathbb{C}$, path cost $g(\mathbb{B})$ represents cardinality of granular space induced by \mathbb{B} . i.e., $g(\mathbb{B}) = |\mathbb{U}/\mathbb{B}|$. Hence, the total cost $f(\mathbb{B})$ becomes $f(\mathbb{B}) = g(\mathbb{B}) + h_{PR}(\mathbb{B}) = |\mathbb{U}/(\mathbb{B} \cup \mathbb{D})|$.

In A^*RSOR algorithm, openlist represents a priority queue of frontier nodes in the increasing order of ‘ f ’ values. Initially, openlist is inserted with nodes corresponding to individual attributes of \mathbb{C} . In each iteration, the least ‘ f ’ value node CN is removed from openlist and inserted in closelist. If $h_{PR}(CN) = 0$ then we have identified the optimal reduct and the attribute set in CN is return as optimal reduct. Otherwise, a child node is generated for each attribute addition into CN , which is not already included in CN . The resulting node CS is inserted in the openlist, if and only if it is not in either the openlist or the closelist. In case CS corresponds to attribute set which is a superset to a candidate reduct node in the openlist (a node with $h_{PR} = 0$) then, it is not included in the openlist as it results in the superset of reduct. This verification is represented as a function $SuperReductCheck(CS, openlist)$.

Algorithm 1: A^* Rough Sets Optimal Reduct (A^*RSOR) Algorithm

Input : CDS: Consistent decision system, C: Set of conditional attributes, openlist: Priority queue over f , closelist: List of explored states.

Output: Optimal Reduct.

```

1  openlist =  $\phi$  ;
2  closelist =  $\phi$  ;
3  for every  $x$  in  $C$  do
4  |    $N$  = Create a node corresponding to  $\{x\}$ 
5  |   Insert(openlist,  $N$ )
6  end
7  while openlist is not empty do
8  |    $CN$  = openList(1);
9  |   remove(openlist,  $CN$ );
10 |   Insert(closelist,  $CN$ );
11 |   if  $h_{PR}(CN) == 0$  then
12 |   |   Opt_Reduct =  $CN$ ;
13 |   |   Return(Opt_Reduct);
14 |   end
15 |    $Cr = C - CN$  ;
16 |   for every  $r$  in  $Cr$  do
17 |   |    $CS = CN \cup \{r\}$ ;
18 |   |   if  $CS$  is in closelist then
19 |   |   |   Continue;
20 |   |   end
21 |   |   if  $CS$  is in openlist then
22 |   |   |   Continue;
23 |   |   else
24 |   |   |   if SuperReductCheck( $CS$ , openlist) == TRUE;
25 |   |   |   then
26 |   |   |   |   Continue;
27 |   |   |   end
28 |   |   end
29 |   |    $N$  = Create a node  $CS$ ;
30 |   |   Insert(openlist,  $N$ );
31 |   end
32 end
```

4 Empirical Results and Observations

The proposed algorithm A^*RSOR is implemented in Matlab-2017a environment, and comparative experiments are conducted in the system with the following configuration: 3.40 GHz \times 4-Intel(R) Core i5-7500 processor, 8GB DDR4 RAM, Ubuntu-16.04.1 LTS 64-bit operating system. Eight categorical benchmark datasets are used from UCI-machine learning repository [3], as shown

Table 1. Additionally, the Wine, Sahart and Zoo datasets are discretized using “mdlp” discretization method [12] to transform into the categorical datasets. The verification of optimal reduct computation by A^*RSOR through ranking experiment and 10-fold cross-validation based induced classifier performance analysis is conducted. Comparative experimental studies are conducted with simulated annealing based near-optimal reduct computation algorithm $SimRSAR$ [11] and hill-climbing search based greedy reduct computational algorithm $IQRAIG$ [15].

Table 1. The description of experimental datasets

Sl.No.	Data Sets	No.of Objects	No.of Variables	No.of Classes
1	Austra	690	15	2
2	Breastcancer	699	10	2
3	Diab	768	9	2
4	Heart	294	14	2
5	Lymphography	148	19	7
6	Sahart	462	9	2
7	Wine	178	14	3
8	Zoo	101	16	7

4.1 Ranking Experiments

The correctness of the proposed method and implementation is verified in ranking experiment by checking the satisfiability of optimality criteria in the computed reduct. Towards this objective for each dataset, all reducts are computed using rough set exploration system ($RSES$) [16] and are ranked in the increasing order of optimality criteria ($f(\mathbb{R}) = |\mathbb{U}/IND(\mathbb{R})|$). The ranks obtained by reducts from A^*RSOR , $SimRSAR$, and $IQRAIG$ approaches are reported along with obtained ‘ f ’ values in Table 2. Table 2 also reported reduct length obtained by the respective algorithms. Total number reducts, obtained by $RSES$ tool, are reported under the column $|RSESReducts|$.

Table 2. Results of ranking experiment with total granular space and the corresponding length of the optimal reduct.

Datsets	$ RSESReducts $	$SimRSAR$ Reduct			$IQRAIG$ Reduct			A^*RSOR Reduct		
		Rank	$f(R)$	Length	Rank	$f(R)$	Length	Rank	$f(R)$	Length
<i>Austra*</i>	44	43	689	3	43	689	3	1	674	4
Breastcancer	20	4	299	4	5	308	4	1	288	4
<i>Diab*</i>	28	20	768	3	20	768	3	1	764	3
<i>Heart*</i>	20	18	290	6	1	278	7	1	278	7
Lymphography	17	5	145	11	5	145	11	1	143	12
Sahart	27	5	460	3	1	458	4	1	458	4
<i>Wine*</i>	81	80	178	3	67	176	3	1	152	4
<i>Zoo*</i>	33	18	27	5	4	22	5	1	20	5

Analysis of Results: The results demonstrate that the implemented A^*RSOR algorithm achieves the optimal reduct in all the datasets by obtaining rank one. The compared approaches $SimRSAR$, and $IQRAIG$ have varying rankings across the datasets. A significant difference in ranking order is observed in Austra, Diab, Heart, Wine, and Zoo datasets and are marked by * symbol. Out of

these only in Austra and Wine datasets, there is a significant variation in $f(\mathbb{R})$ values.

4.2 Ten-fold experiments

In this section, we conducted the 10-fold cross-validation experiments on given benchmark datasets for assessing the relevance of reduct in inducing different classifiers (Naive Bayes (NB), CART and Random Forest (RF)). In all the classifiers, default options are used while treating all attributes as categorical. Table 3 depicts the mean and standard deviation of reduct length and computational time (in seconds) resulted in ten-fold experiments. Table 4 demonstrates the resulting mean and standard deviation ($\mu \pm \sigma$) of classification accuracies obtained in ten-fold experiments. In table 4, the column *ALLAttributes* refers to results from un-reduced training data.

Table 3. Reduct length and computational time in ten-fold cross-validation experiment

Datsets	Reduct Length			Time		
	SimRSAR	<i>IQRAIG</i>	<i>A*RSOR</i>	SimRSAR	<i>IQRAIG</i>	<i>A*RSOR</i>
Austra	3 \pm 0	3 \pm 0	5.60 \pm 1.07	157.05 \pm 7.79	0.11 \pm 0.01	234.31 \pm 10.13
Breastcancer	4 \pm 0	4 \pm 0	4 \pm 0	7.14 \pm 0.31	0.03 \pm 0.00	3.69 \pm 0.17
Diab	2.80 \pm 0.42	2.80 \pm 0.42	3 \pm 0	250.24 \pm 24.64	0.06 \pm 0.007	3.60 \pm 0.19
Heart	5.90 \pm 0.56	6.70 \pm 0.67	6.80 \pm 0.42	4.74 \pm 0.27	0.06 \pm 0.01	64.02 \pm 3.11
Lymphography	10.30 \pm 0.82	10.80 \pm 0.63	10.70 \pm 0.82	2.41 \pm 0.12	0.11 \pm 0.01	1.72 \pm 0.05
Sahart	3 \pm 0	4 \pm 0	4 \pm 0	36.58 \pm 1.93	0.07 \pm 0.009	3.28 \pm 0.14
Wine	2.70 \pm 0.48	2.70 \pm 0.48	5.30 \pm 0.82	9.34 \pm 0.97	0.02 \pm 0.007	13.42 \pm 0.83
Zoo	4.90 \pm 0.31	5.30 \pm 0.67	5.10 \pm 0.73	0.31 \pm 0.02	0.02 \pm 0.005	26.09 \pm 7.39

Student t-test is conducted on classification accuracies for analysis of results between the proposed algorithm *A*RSOR* with *IQRAIG*, *SimRSAR* and *ALLAttributes*, and the results of the t-test are depicted in Table 4 along with classification accuracies. Table 4 entries show the four significance levels indicated as (+/-)*, (+/-)**, (+/-)***, # as per p-value in the t-test. The significance level for experimental results based on t-test are divided as statistically significant and are indicated as * ($p - value \leq 0.05$), statistically highly significant as ** ($p - value \leq 0.01$), statistically extremely significant as *** ($p - value \leq 0.001$) and no statistical difference indicated as #. The prefix of ‘+’ denotes the *A*RSOR* has performed better than the compared algorithm and ‘-’ denotes that it has underperformed.

Analysis of Results: In Table 3, *IQRAIG* algorithm has obtained much lesser computational time than *A*RSOR* and *SimRSAR* algorithms. This is in correlation with the theoretically lesser complexity of a greedy hill-climbing search algorithm in comparison to multiple subspace search algorithms of *A*RSOR* and *SimRSAR*. The practical time complexity of *A*RSOR* search algorithm depends on the depth at which the optimal reduct is formed. For instance, in Diabetes and Sahart datasets, the proposed *A*RSOR* algorithm incurred only 3.6, 3.2 seconds on average, whereas *SimRSAR* incurred 250, 36.5 seconds. In

contrast, *SimRSAR* obtained significantly lesser computational time in Austra, Heart, wine, and Zoo datasets. It is to be noted that out of these three algorithms, the worst cases time complexity of A^*RSOR is exponential while the other two algorithms are having polynomial time complexity. Both *SimRSAR* and *IQRAIG* are formulated towards obtaining shorter length reduct. Hence, the reduct lengths of A^*RSOR are slightly higher or equal than other algorithms.

Table 4. Ten-fold cross-validation results for classification

Datasets	<i>AllAttributes</i>			<i>SimRSAR</i>			<i>IQRAIG</i>			A^*RSOR		
	NB	CART	RF	NB	CART	RF	NB	CART	RF	NB	CART	RF
Austra	62.89 ± 6.66 +*	81.30 ± 3.08 #	80 ± 4.77 #	58.40 ± 5.79 +***	55.36 ± 8.79 +***	57.10 ± 8.20 +***	56.37 ± 6.08 +***	51.88 ± 4.96 +***	53.33 ± 6.17 +***	69.85 ± 5.90	78.84 ± 8.002	77.39 ± 7.61
Breastcancer	97.13 ± 1.67 #	93.40 ± 3.09 #	96.84 ± 1.91 #	96.39 ± 2.30 #	93.23 ± 3.78 #	95.86 ± 2.64 #	96.29 ± 2.22 #	93.27 ± 3.34 #	95.25 ± 3.63 #	96.58 ± 1.61	93.41 ± 2.98	95.44 ± 2.15
Diab	65.41 ± .60 #	59.23 ± 6.38 #	64.22 ± 4.86 #	64.48 ± 6.67 #	61.22 ± 5.20 #	62.53 ± 6.85 #	64.24 ± 5.08 #	62.01 ± 5.66 #	61.76 ± 4.99 #	62.80 ± 5.14	58.07 ± 6.90	60.18 ± 4.67
Heart	82.14 ± 8.60 #	79.64 ± 9.68 #	81.10 ± 9.27 #	77.05 ± 9.36 #	73.90 ± 11.88 #	75.67 ± 9.85 #	80.33 ± 7.28 #	74.94 ± 11.04 #	79.38 ± 10.20 #	81.10 ± 8.21	77.31 ± 10.99	79.72 ± 10.25
Lymphography	38.83 ± 14.47 #	35.51 ± 12.49 #	43.11 ± 13.06 #	39.61 ± 13.74 #	35.12 ± 10.16 #	37.72 ± 9.70 #	36.75 ± 13.90 #	33.37 ± 12.92 #	38.37 ± 12.63 #	39.80 ± 12.19	30.84 ± 16.08	36.29 ± 12.44
Sahart	57.38 ± 7.68 #	52.21 ± 7.51 #	55.41 ± 6.50 #	53.03 ± 3.72 +*	50.45 ± 7.29 #	52.61 ± 9.14 #	54.52 ± 6.65 #	54.13 ± 5.64 #	53.48 ± 7.53 #	60.39 ± 8.13	52.83 ± 5.41	55 ± 7.80
Wine	59.74 ± 9.23 +***	57.97 ± 11.63 +***	94.30 ± 4.83 **	55.22 ± 14.41 +***	53.45 ± 13.36 +***	70.94 ± 17.48 +*	52.87 ± 11.01 +***	54.04 ± 12.88 +***	67.03 ± 17.55 +***	88.02 ± 6.01	83.10 ± 8.52	83.88 ± 7.96
Zoo	91.00 ± 11.00 #	87.18 ± 14.86 #	94.18 ± 8.11 #	88.00 ± 10.32 #	89.09 ± 5.70 #	93 ± 6.74 #	84.09 ± 9.72 #	89.09 ± 8.77 #	95 ± 8.49 #	86.18 ± 14.22	88.18 ± 9.03	96.18 ± 6.53

In NB, CART and RF classifiers, A^*RSOR obtained statistically similar in classification accuracies with compared algorithms in Breastcancer, Diab, Lymphography and Zoo datasets. In Sahart dataset, A^*RSOR approach performed statistically significant than *SimRSAR* in NB classifiers. In Austra and Wine datasets, A^*RSOR algorithm achieved statistically extremely significant accuracies than compared algorithms using all three classifiers. Also, in Wine dataset, A^*RSOR algorithm performed extremely significant than *AllAttributes* in both NB & CART classifiers. Likewise, in Austra dataset, A^*RSOR algorithm performed statistically significant accuracy than *AllAttributes* using NB classifier. Results also demonstrate the additional advantages for RF in achieving better accuracies with *AllAttributes*, as RF is an ensemble classifier with bagging over attribute space. It is further noted that A^*RSOR achieves statistically similar results with *AllAttributes* in RF for all datasets except Wine dataset.

The datasets in which A^*RSOR obtained better accuracies i.e., Austra and Wine, are also the datasets in which *IQRAIG*, *SimRSAR* algorithm obtained higher rank reducts along with with significantly higher $f(R)$ value. This establishes that the chosen optimality criterion of the coarser granular space is relevant in obtaining reducts with greater potential in building better classification models.

5 Conclusion

Many approaches of rough set based reduct algorithm are aiming towards computing shortest length reduct both optimally and near optimally. In this work, the need for an alternative optimal criterion for reduct computation is identified, and A^*RSOR algorithm is developed for the computation of optimal reduct using A^* search. Partition refinement heuristic is introduced and proved to be a consistent heuristic. Comparative experimental results validated the utility

of proposed optimality criteria. In the future, scalable algorithms for proposed *A*RSOR* will be developed for enhancing the applicability to large scale decision systems.

References

1. Chen, J., Lin, Y., Li, J., Ma, Z., Tan, A.: A rough set method for the minimum vertex cover problem of graphs. *Applied Soft Computing* **42**, 360–367 (2016)
2. Chen, Y., Zhu, Q., Xu, H.: Finding rough set reducts with fish swarm algorithm. *Knowledge-Based Systems* **81**, 22–29 (2015)
3. Dua, D., Karra Taniskidou, E.: UCI machine learning repository (2017), <http://archive.ics.uci.edu/ml>
4. Grzymala-Busse, J.W.: *Rule Induction*, pp. 249–265. Springer US, Boston, MA (2010)
5. H., A.E., Hassanien, A.E., Suraj, Z., Slezak, D., Lingras, P.: *Rough Computing: Theories, Technologies and Applications*. IGI Global, Hershey, PA, USA (2007)
6. Han, J., Hu, X., Lin, T.Y.: Feature subset selection based on relative dependency between attributes. In: *International Conference on RSCTC2004*. pp. 176–185. Springer (2004)
7. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* **4**(2), 100–107 (1968)
8. Jensen, R.: Rough set-based feature selection: A review. In: *Rough computing: theories, technologies and applications*, pp. 70–107. IGI Global (2008)
9. Jensen, R., Shen, Q.: Finding rough set reducts with ant colony optimization. In: *Proceedings of UKCI-2003*. vol. 1, pp. 15–22 (2003)
10. Jensen, R., Shen, Q.: Fuzzy-rough attribute reduction with application to web categorization. *Fuzzy sets and systems* **141**(3), 469–485 (2004)
11. Jensen, R., Shen, Q.: Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. *IEEE Transactions on knowledge and data engineering* **16**(12), 1457–1471 (2004)
12. Kim, H.J.: mdlp: Discretization using the minimum description length principle (2010), <https://rdrr.io/cran/discretization/>
13. Kuipers, B., A. Feigenbaum, E., E. Hart, P., J. Nilsson, N.: Shakey: From conception to history. *Ai Magazine* **38**, 88–103 (03 2017)
14. Pawalk, Z.: *Rough sets: theoretical aspects of reasoning about data*. Dordrecht: Kluwer Academic Publishers (1991)
15. Sai Prasad, P.S.V.S., Rao, C.R.: Extensions to iquickreduct. In: *MIWAI. Lecture Notes in Computer Science*, vol. 7080, pp. 351–362. Springer (2011)
16. Skowron, A., et al.: Rough set exploration system (rses). Warsaw: Warsaw University (2005), <https://www.mimuw.edu.pl/szczuka/rses/>
17. Skowron, A., Rauszer, C.: *The Discernibility Matrices and Functions in Information Systems*, pp. 331–362. Springer Netherlands, Dordrecht (1992)
18. Wang, X., Yang, J., Teng, X., Xia, W., Jensen, R.: Feature selection based on rough sets and pso. *Pattern recognition letters* **28**(4), 459–471 (2007)
19. Wroblewski, J.: Finding minimal reducts using genetic algorithms. In: *Proceedings of the second annual join conference on information science*. vol. 2, pp. 186–189 (1995)
20. Zheng, K., Hu, J., Zhan, Z., Ma, J., Qi, J.: An enhancement for heuristic attribute reduction algorithm in rough set. *Expert Systems with Applications* **41**(15), 6748–6754 (2014)