A Mathematical Foundation for Improved Reduct Generation in Information Systems

Janusz A. Starzyk , Senior *Member, IEEE*, Dale E. Nelson, *Member, IEEE*, and Kirk Sturtz

Abstract— *When data sets are analyzed, statistical pattern recognition is often used to find the information hidden in the data. Another approach to information discovery is data mining. Data mining is concerned with finding previously undiscovered relationships in data sets. Rough set theory provides a theoretical basis from which to find these undiscovered relationships. We define a new theoretical concept,* ***strong compressibility****, and present the mathematical foundation for an efficient algorithm, the* ***Expansion Algorithm****, for generation of all reducts of an information system. The process of finding reducts has been proven to be NP-hard. Using the elimination method, problems of size 13 could be solved in reasonable times. Using our Expansion Algorithm, the size of problems that can be solved has grown to 40. Further, by using the strong compressibility property in the Expansion Algorithm, additional savings of up to 50% can be achieved. This paper presents this algorithm and the simulation results obtained from randomly generated information systems.*

Keywords— **Rough sets, information systems, data mining.**

I. Introduction

Information systems are used in data mining and intelligent decision support for pattern recognition and neural network training to represent the knowledge that can be extracted from the database. These systems were investigated by several researchers [3], [6], [7], [10] using rough set theory. In recent years rough set theory and its applications have captured a lot of attention from AI researchers and developers of intelligent systems. Rough sets parallel fuzzy sets in their domain of applications, usefulness to capture the imprecise knowledge, and mathematical formalism based on set theory. Among the various problems addressed by researchers who study rough sets are knowledge representation and reduction, dependencies in knowledge bases, imprecise knowledge, and software implementation of decision support systems.

Rough set theory and its strict mathematical formalism is appealing to many researchers working in logic and deductive reasoning [2], [6], [9]. Specialized logic tools were developed to deal with approximate reasoning. A basic problem for many practical applications of the rough sets is an efficient selection of the set of attributes (features) necessary for the classification of objects in the considered universe (signal space). This problem, known as the knowledge reduction problem, was treated in [4], [1], and an algorithmic approach based on the discernibility matrix and expansion and reduction of discernibility function was developed. The hardest problem to solve algorithmically in a data reduction system is the problem of generating all reducts of a given information system. It was shown in [7] that generation of all reducts is NP-hard.

This paper complements the work by [7] for generation of reducts with the notion of strong equivalence of the attributes of the information system and attribute expansion. Applying the concepts of strongly compressible and attribute expansion, we demonstrate a significant reduction in computational complexity of the reduct generation procedure. Improvement stems from breaking a complex reduct generation problem into a number of simpler problems.

Manuscript received ________; revised _________.

J. Starzyk is with School of Electrical Engineering and Computer Sciences, Ohio University, Athens, OH 45701 (e-mail starzyk@bobcat.ohiou.edu)

D. Nelson is with the Air Force Research Laboratory, Sensors Directorate, WPAFB, OH 45433-7001 (e-mail nelsonde@sensors.wpafb.af.mil)

K. Sturtz is with Veridian Inc., Dayton, OH, 45431 (e-mail ksturtz@mbvlab.wpafb.af.mil)

II. INFORMATION SYSTEMS AND REDUCTS

In this section we review basic definitions of rough set theory related to selection of the set of attributes for the purpose of classifying a given set of objects. Discernibility function is formally defined and an alternative characterization of reducts is given which is easier to manipulate for algorithmic purposes. For a full description of rough set theory and related terms see [5].

Consider the information system (U, A), where $U = \{x_1,...,x_n\}$ is a nonempty finite set called the **universe**, and $A = \{a_1,...,a_m\}$ is a nonempty set. The elements of A, called **attributes**, are functions

$$a_i : U \to V_i$$

where V_i is called the value set of a_i. The **discernibility matrix** of A is the $n \times n$ matrix with i, j^{th} entry

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j)\}.$$

Let $B \subseteq A$, and let $P(A)$ be the power set of A. The **Boolean-valued function** χ_B is

$$\begin{aligned} \chi_B &: P(A) \to \{0,1\} \\ &: C \mapsto \begin{cases} 1 & \text{when } B \cap C \neq \varnothing \\ 0 & \text{when } B \cap C = \varnothing \end{cases} \end{aligned}$$

Let $S_\chi = \{\chi_B : B \in P(A)\}$. Define the binary operator $\wedge$, called conjunction, by

$$\begin{aligned} \wedge &: S_\chi \times S_\chi \to S_\chi \\ &: (\chi_B, \chi_C) \mapsto \chi_B \wedge \chi_C \end{aligned}$$

where

$$\begin{aligned} \chi_B \wedge \chi_C &: P(A) \to \{0,1\} \\ &: D \mapsto \chi_B(D)\chi_C(D) \end{aligned}$$

It is not difficult to prove $(S_\chi, \wedge)$ forms a commutative monoid, with the identity element being χ_A. The associativity property

$$(\chi_B \wedge \chi_C) \wedge \chi_D = \chi_B \wedge (\chi_C \wedge \chi_D)$$

allows the parenthesis to be dropped without any possibility of confusion; moreover we can now define $\wedge$ for any finite collection of functions $\{\chi_{B_i}\}_{i=1}^{p}$ by recursion

$$\underset{i=1,\dots,p}{\wedge} \chi_{B_i} = \left(\underset{i=1,\dots,p-1}{\wedge} \chi_{B_i} \right) \wedge \chi_{B_p}$$

The **discernibility function** of the information system is

$$\begin{aligned} f_A &: P(A) \to \{0,1\} \\ &: C \mapsto \left(\underset{\substack{1 \le i,j \le n \\ \chi_{c_{ij}} \neq \overline{0}}}{\wedge} \chi_{c_{ij}} \right)(C) \end{aligned}$$

where "$\overline{0}$" is the constant function

$$\begin{aligned} \overline{0} &: P(A) \to \{0,1\} \\ &: C \mapsto 0 \end{aligned}$$

If f_A is an empty conjunction we define f_A to be the constant zero function. This is an uninteresting case and we assume throughout that f_A is not an empty conjunction.

The condition $\chi_{c_{ij}} \neq \overline{0}$ used in the definition of the discernibility function is equivalent to the condition that $c_{ij} \neq \varnothing$ since

$$\chi_{c_{ij}} \neq \overline{0} \Leftrightarrow \chi_{c_{ij}}(A) = 1 \Leftrightarrow c_{ij} \cap A \neq \varnothing \Leftrightarrow \exists a_k \in c_{ij} \Leftrightarrow c_{ij} \neq \varnothing$$

Using the fact the discernibility matrix is symmetric and that $c_{ii} = \varnothing$ we obtain the discernibility function simplifies to $f_A = \bigwedge_{\substack{1 \le i < j \le n \\ c_{ij} \neq \varnothing}} \chi_{c_{ij}}$. An immediate consequence of the definition is

Proposition 1. $f_A(A) = 1$.

Proof. Suppose the contrary, $f_A(A) = 0$. Then

$$\begin{array}{l} (\exists \chi_{c_{ij}})(\chi_{c_{ij}}(A) = 0) \\ \quad \Downarrow \\ A \cap c_{ij} = \varnothing \\ \quad \Downarrow \\ c_{ij} = \varnothing \end{array}$$

contradicting the definition of f_A as a conjunction of functions with a nonempty indexing set. ▯

Let $B \subseteq A$. The ***B-indiscernability*** relation is

$$Ind(B) = \{(x, y) \in U \times U : (\forall a \in B)(a(x) = a(y))\}$$

The ***B-discernibility*** relation is the complement of $Ind(B)$ in $U \times U$,

$$Dis(B) = U \times U - Ind(B).$$

The following lemma is an immediate consequence of the definition. It is used repeatedly in the propositions to follow.

Lemma 2. *Let* $B \subseteq A$. Then

$$Dis(B) = Dis\left(\bigcup_{a \in B} \{a\}\right) = \bigcup_{a \in B} Dis(\{a\}).$$

Consequently, if $a, b \in B$ and $Dis(\{a\}) = Dis(\{b\})$ then

$$Dis(B) = Dis(B - \{a\}) = Dis(B - \{b\}).$$ ▯

Essential for the information system are the reducts that describe knowledge represented in this system. A set $B \subseteq A$ is a **discern** in *A* if $Ind(B) = Ind(A)$. A discern is called a **reduct** if $(\forall a \in B)$ $Ind(B - \{a\}) \supset Ind(B)$, where "$\supset$" denotes a proper subset relation. The set of all reducts of *A* is denoted $Red(A)$. The reduct generation procedure developed in [8] is based on the expansion of the discernibility function into a disjunction of its prime implicants by applying the absorption or multiplication laws. It is our intention to improve the computational efficiency of the reduct generation procedure.

The following proposition provides an alternative characterization of a reduct in terms of the discernibility function. This characterization is more convenient for purposes of the reduct generation algorithm to follow.

Proposition 3. *B* is a reduct in *A* iff

(1) $f_A(B) = 1$, and

(2) $C \subset B \Rightarrow f_A(C) = 0$.

Proof. Suppose *B* is a reduct in *A*. We first show $f_A(B) = 1$. By Proposition 1, $f_A(A) = \bigwedge_{\substack{1 \le i < j \le n \\ c_{ij} \neq \varnothing}} \chi_{c_{ij}}(A) = 1$ which implies $(\forall i, j)(c_{ij} \neq \varnothing)\chi_{c_{ij}}(A) = 1$. So by definition of $\chi_{\cdot}$, $(\forall i, j)(c_{ij} \neq \varnothing)(c_{ij} \cap A \neq \varnothing)$ which implies $(\forall i, j)(c_{ij} \neq \varnothing)(\exists a \in A)(a(x_i) \neq a(x_j))$ which in turn implies $(\forall i, j)(c_{ij} \neq \varnothing)((x_i, x_j) \in Dis(A))$. Using the hypothesis $Ind(B) = Ind(A)$ we obtain $(\forall i, j)(c_{ij} \neq \varnothing)((x_i, x_j) \in Dis(B))$ which implies $(\forall i, j)(c_{ij} \neq \varnothing)(\exists a \in B)(a(x_i) \neq a(x_j))$. Hence $(\forall i, j)(c_{ij} \neq \varnothing)(c_{ij} \cap B \neq \varnothing)$ from which we obtain $(\forall i, j)(c_{ij} \neq \varnothing)(\chi_{c_{ij}}(B) = 1)$. This implies $f_A(B) = \bigwedge_{\substack{1 \le i < j \le n \\ c_{ij} \neq \varnothing}} \chi_{c_{ij}}(B) = 1$. We now prove $(\forall a \in B)\ \ Ind(B-\{a\}) \supset Ind(B)$ implies $C \subset B \Rightarrow f_A(C)=0$. Let $C \subset B$ so $(\exists a \in B)(a \notin C)$. The hypothesis implies $Ind(C) \supset Ind(B)$ so

(i) $(\exists (x_i, x_j))(\forall b \in C)(b(x_i)=b(x_j))$ while

(ii) $a(x_i) \neq a(x_j)$.

Condition (i) implies $c_{ij} \cap C = \varnothing$ which in turn implies $\chi_{c_{ij}}(C)=0$.
Condition (ii) $a(x_i) \neq a(x_j)$ implies $c_{ij} \neq \varnothing$ so $\chi_{c_{ij}}$ is a factor of f_A. Combining these two conditions we obtain $f_A(C)=0$.
Conversely, suppose the two conditions (i) $f_A(B)=1$, and (ii) $C \subset B \Rightarrow f_A(C)=0$ hold.

We first prove, using only the condition (i) $f_A(B)=1$, that $Ind(B)=Ind(A)$ by showing $Ind(B) \subseteq Ind(A)$ and $Ind(A) \subseteq Ind(B)$. Let $(x_1,x_2) \in Ind(B)$. Then $(\forall a \in B)\, a(x_i)=a(x_j)$. Now either $c_{ij} = \varnothing$ or $c_{ij} \neq \varnothing$. The latter condition cannot hold true. For suppose $c_{ij} \neq \varnothing$. Then since, by hypothesis, $f_A(B)=1$ we obtain that $\chi_{c_{ij}}(B)=1$ which implies $c_{ij} \cap B \neq \varnothing$. Hence $(\exists a \in B)(a(x_i) \neq a(x_j))$ contradicting our hypothesis $(x_1,x_2) \in Ind(B)$. Thus we must conclude $c_{ij} = \varnothing$ which implies $(x_1,x_2) \in Ind(A)$. Hence $Ind(B) \subseteq Ind(A)$. The condition $Ind(A) \subseteq Ind(B)$ is immediate since $B \subseteq A$. Hence $Ind(B)=Ind(A)$.

Now we show $(\forall a \in B)\ \ Ind(B-\{a\}) \supset Ind(B)$ follows from the two given conditions of the proposition. Let $a \in B$ and $C = B-\{a\} \subset B$. By the second condition $C \subset B \Rightarrow f_A(C)=0$ so $(\exists \chi_{c_{ij}})\chi_{c_{ij}}(C)=0$ which implies $c_{ij} \cap C = \varnothing$. Hence $(\forall a \in C)\ \ a(x_i)=a(x_j)$ so $(x_i,x_j) \in Ind(C)$. But by the first condition, $f_A(B)=1$ which implies $\chi_{c_{ij}}(B)=1$, which in turn implies $c_{ij} \cap B \neq \varnothing$. Thus $(\exists b \in B)(b(x_i) \neq b(x_j))$ - since $c_{ij} \cap C = \varnothing$ we conclude $b=a$. Because $(x_i,x_j) \in Ind(C)$ and $(x_i,x_j) \notin Ind(B)$ we obtain $Ind(C) = Ind(B-\{a\}) \supset Ind(B)$. ∎

III. REDUCTION OF KNOWLEDGE

Not all knowledge presented in the information system is necessary to describe it. Reduction of knowledge in the information system (which results in generation of reducts) is analogous to mathematical independence of vectors in linear algebra. Reduction of knowledge will be based on the expansion and simplification of the discernibility function. Basic tools for this simplification are the absorption and expansion laws discussed in this section. These tools will be used to produce a specific form of the discernibility function defined here as a simple form.

As before, let $S_\chi = \{\chi_B : B \in P(A)\}$. Define the binary operator $\vee$, called disjunction, by

$$\begin{aligned} \vee &: S_\chi \times S_\chi \to S_\chi \\ &: (\chi_B, \chi_C) \mapsto \chi_B \vee \chi_C \end{aligned}$$

where

$$\begin{aligned} \chi_B \vee \chi_C &: P(A) \to \{0,1\} \\ &: D \mapsto \begin{cases} 1 & if\ \chi_B(D)=1\, or\ \chi_C(D)=1 \\ 0 & if\ \chi_B(D)=0\, and\ \chi_C(D)=0 \end{cases} \end{aligned}$$

It is easy to prove that the operator $\vee$ satisfies associativity, commutativity, and distributes with respect to conjunction,

$$\chi_B \wedge (\chi_{C_1} \vee \ldots \vee \chi_{C_k}) = (\chi_B \wedge \chi_{C_1}) \vee \ldots \vee (\chi_B \wedge \chi_{C_k}) \ .$$

Likewise it distributes with respect to disjunction,

$$\chi_B \vee (\chi_{C_1} \wedge \ldots \wedge \chi_{C_k}) = (\chi_B \vee \chi_{C_1}) \wedge \ldots \wedge (\chi_B \vee \chi_{C_k}) .$$

These last two properties are called the **distribution laws**. One consequence of these laws is given in the following proposition which provides the fundamental tools used the reduct generation algorithm to follow.

Proposition 4. (a) (**Absorption Law**) Let $B \subseteq A$. Suppose $\varnothing \neq C \subseteq D \subseteq A$. If $\chi_C(B)=1$ then $\chi_D(B)=1$.

(b) (**Factorization Law**) Let $a \in A$ and suppose $\chi_{C_i}(\{a\}) = 1 \; for \; i = 1,...,k$. Then for $B \subseteq A$

$$(\chi_{C_1} \wedge ... \wedge \chi_{C_k})(B) = 1 \qquad iff \qquad (\chi_{\{a\}} \vee (\chi_{C_1-\{a\}} \wedge ... \wedge \chi_{C_k-\{a\}}))(B) = 1.$$

Proof.

(a) $\chi_C(B) = 1 \Rightarrow C \cap B \neq \varnothing \Rightarrow D \cap B \neq \varnothing \Rightarrow \chi_D(B) = 1$.

(b) Let $B \subseteq A$. Then

$$\chi_{C_1}(B) \wedge ... \wedge \chi_{C_k}(B) = 1$$
$$\Updownarrow$$
$$\chi_{C_i}(B) = 1 \; for \; i = 1,...,k$$
$$\Updownarrow$$
$$\text{Case I. } a \in B \Rightarrow \chi_{\{a\}}(B) = 1$$
$$\text{Case II. } a \notin B \Rightarrow \chi_{C_i-\{a\}}(B) = 1 \; for \; i = 1,...,k$$
$$\text{since } \chi_{C_i}(B) = 1 \; for \; i = 1,...,k$$
$$\Updownarrow$$
$$\chi_{\{a\}}(B) \vee \left(\chi_{C_1-\{a\}}(B) \wedge ... \wedge \chi_{C_k-\{a\}}(B)\right) = 1$$

∎

Let $B \subseteq A$ and suppose $\varnothing \neq C \subseteq D \subseteq A$. In ascertaining the validity of $f_A(B) = 1$ the absorption law implies it suffices to check the validity of $\chi_C(B) = 1$ to ascertain in addition the validity of $\chi_D(B) = 1$. In this manner the absorption law is exploited in reduct algorithms.

Using commutativity of the conjunction operator to rearrange factors if necessary, and the distribution law, the factorization law gives

Corollary 5. (**Expansion Law**) Suppose $f_A = \chi_{C_1} \wedge ... \wedge \chi_{C_k} \wedge \chi_{C_{k+1}} \wedge ... \wedge \chi_{C_s}$. Let $a \in A$ and suppose $\chi_{C_i}(\{a\}) = 1 \; for \; i = 1,...,k$, and $\chi_{C_i}(\{a\}) = 0 \; for \; i = k+1,...,s$. Then

$$\begin{aligned} f_A &= (\chi_{C_1} \wedge ... \wedge \chi_{C_k}) \wedge (\chi_{C_{k+1}} \wedge ... \wedge \chi_{C_s}) \\ &= ((\chi_{\{a\}} \vee (\chi_{C_1-\{a\}} \wedge ... \wedge \chi_{C_k-\{a\}})) \wedge (\chi_{C_{k+1}} \wedge ... \wedge \chi_{C_s})) \\ &= (\chi_{\{a\}} \wedge (\chi_{C_{k+1}} \wedge ... \wedge \chi_{C_s})) \vee ((\chi_{C_1-\{a\}} \wedge ... \wedge \chi_{C_t-\{a\}}) \wedge (\chi_{C_{k+1}} \wedge ... \wedge \chi_{C_s})) \end{aligned}$$

∎

Letting

$$f_1 = \chi_{\{a\}} \wedge (\chi_{C_{k+1}} \wedge ... \wedge \chi_{C_s})$$
$$f_2 = (\chi_{C_1-\{a\}} \wedge ... \wedge \chi_{C_t-\{a\}}) \wedge (\chi_{C_{k+1}} \wedge ... \wedge \chi_{C_s})$$

the conclusion of Corollary 5 reads $f_A = f_1 \vee f_2$, where both f_1 and f_2 are conjunctions of the Boolean-valued functions χ. Since each $\chi_B \in S_\chi$ is a function $\chi_B : P(A) \to \{0,1\}$, both f_1 and f_2 are functions $f_i : P(A) \to \{0,1\}$. This suggest the following definitions:

A **cover** of a discernibility function f_A is a family of Boolean-valued functions $\{f_1,...,f_k\}$ satisfying $f_A = f_1 \vee ... \vee f_k$ where each $f_i : P(A) \to \{0,1\}$ is a conjunction of Boolean-valued functions.

Let $\{f_1,...,f_k\}$ be a cover of a discernibility function f_A. The **reduct of** f_i, denoted $Red(f_i)$, consist of all subsets $B \subseteq A$ such that

(1) $f_i(B) = 1$, and

(2) $C \subset B \Rightarrow f_i(C) = 0$.

A **simple cover** of a discernibility function f_A is a cover of f_A such that if $B \in Red(f_i)$ then $B \notin Red(f_j) \quad \forall j, \quad j \neq i$. A consequence of these definitions and Proposition 3 is

Proposition 6. Let $\{f_1,...,f_k\}$ be a simple cover of a discernibility function f_A. If $B \in Red(f_i)$ then B is a discern in A, $Ind(B) = Ind(A)$.

Proof. Suppose $B \in Red(f_i)$. Then $f_i(B) = 1 \Rightarrow f_A(B) = 1$. The proof of Proposition 3, as noted, shows the condition $f_A(B) = 1$ is equivalent to $Ind(B) = Ind(A)$. The condition $B \subseteq A$ follows by the definition of $Red(f_i)$. ▯

Definition. A simple cover $\{f_1,...,f_k\}$ is called a **simple form** of the discernibility function if for each $f_i = \chi_{C_1} \wedge ... \wedge \chi_{C_{k_i}}$ the indexing sets are pairwise disjoint, $C_i \cap C_j = \varnothing \;\; for \; i \neq j$.

The expansion law, as stated in Corollary 5, shows that given any conjunction $f_A = \chi_{C_1} \wedge ... \wedge \chi_{C_s}$ with two or more the indexing sets having an element (attribute) $a \in A$ in common we can factor f_A as $f_A = f_1 \vee f_2$. In turn, if either f_1 or f_2 have two or more indexing sets in their respective conjunctions with an element in common, they can be factored to obtain

$$\begin{aligned} f_A &= f_1 \vee f_2 \\ &= (f_{11} \vee f_{12}) \vee (f_{21} \vee f_{22}) \end{aligned}$$

This process can be repeated on each disjunction $f_i = \chi_{C_1} \wedge ... \wedge \chi_{C_t}$ of f_A in turn. If all the indexing sets C_i in a given Boolean-valued function f_i are distinct $C_i \cap C_j = \varnothing \;\; for \; i \neq j$ then the corresponding function is in simple form and is no longer expanded. This process must terminate by the finiteness of A. We have proven

Proposition 7. Any discernibility function f_A can systematically be put into simple form by repeatedly applying the expansion law.

The reduct generation algorithm, given in Section V, is based upon Proposition 7. For computational efficiency, an expansion proceeds with respect to an attribute a which belongs to the largest number of indexing sets C_i not less than two. The absorption law provides computational savings, as does the concept of strong equivalence given next.

The Boolean-valued functions χ_B are completely determined by the indexing set B. As such, it is unnecessary, for computational purposes, to carry along the "χ" notation. In the following example, illustrating Proposition 7, we first solve the problem formally, and then solve it dropping the explicit χ function notation.

Example 1. Consider the following Boolean-valued function f_A:

$$f_A = \chi_{C_1} \wedge \chi_{C_2} \wedge \chi_{C_3} \wedge \chi_{C_4}$$

where

$$C_1 = \{a,b,c\},\; C_2 = \{b,d\},\; C_3 = \{a,d\},\; C_4 = \{e\}$$

Using the expansion law to expand f_p with respect to a gives

$$f_A = (\chi_{D_1} \wedge \chi_{D_3} \wedge \chi_{D_4}) \vee (\chi_{D_3} \wedge \chi_{D_2} \wedge \chi_{D_4} \wedge \chi_{D_5})$$

where

$$D_1 = \{a\},\; D_2 = \{b,d\},\; D_3 = \{b,c\},\; D_4 = \{e\},\; D_5 = \{d\}.$$

By using the absorption law f_A can be further reduced to

$$f_A = (\chi_{D_1} \wedge \chi_{D_3} \wedge \chi_{D_4}) \vee (\chi_{D_3} \wedge \chi_{D_4} \wedge \chi_{D_5})$$

The last equation is a simple form of f_A. The discerns of *A* are $\{a,b,e\}, \{a,d,e\}, \{b,d,e\},$ and $\{c,d,e\}$. By definition, the minimal elements of the set of all discerns are reducts. Hence these four sets are the reducts of *A*.

The same problem without the "χ" notation:

$$f_A = C_1 \wedge C_2 \wedge C_3 \wedge C_4 .$$

where

$$C_1 = \{a,b,c\},\ C_2 = \{b,d\},\ C_3 = \{a,d\},\ C_4 = \{e\}$$

Expanding with respect to the attribute a gives $f_A = (D_1 \wedge D_2 \wedge D_4) \vee (D_3 \wedge D_2 \wedge D_4 \wedge D_5)$ where $D_1 = \{a\},\ D_2 = \{b,d\},\ D_3 = \{b,c\},\ D_4 = \{e\},\ D_5 = \{d\}$. By using the absorption law f_A can be further reduced to $f_A = (D_1 \wedge D_2 \wedge D_4) \vee (D_3 \wedge D_4 \wedge D_5)$ whereby the reducts can be readily determined. The latter approach is a more efficient representation and we employ it computationally.

IV. STRONG COMPRESSIBILITY

Efficiency of the reduct generation depends on the simplification of the discernibility function. Since reduct generation is an NP-hard problem, savings in both the memory and computational effort yields a more practical algorithm applicable for complex knowledge systems. The concept of strong compressibility, defined in this section, can be applied together with the other tools of reduct generation algorithm and reduces its computational cost.

The property of discernibility can be used to define an equivalence relation on *A* as

$$a \sim_{Dis} b \Leftrightarrow Dis(\{a\}) = Dis(\{b\}) .$$

The equivalence class of an attribute $a \in A$ is the subset

$$[a] = \{b \in A : Dis(\{a\}) = Dis(\{b\})\} .$$

Let $B \subseteq [a]$. By Lemma 2

$$Dis(B) = Dis\left(\bigcup_{b \in B} \{b\}\right) = \bigcup_{b \in B} Dis(\{b\}) = Dis(\{a\}) ,$$

or equivalently,

$$(\forall b \in B) \quad Dis(B) = Dis(\{b\}) .$$

Let $B \subseteq A$. If *there exists* an attribute $a \in A$ such that $B \subseteq [a]$ then we say *B* is **strongly compressible**. An immediate consequence of the above definitions is:

Proposition 8. If a subset of attributes $B \subseteq A$ is strongly compressible then any two singleton sets, whose elements are attributes in *B,* are equivalent

$$(\forall a, b \in B) \quad Dis(\{a\}) = Dis(\{b\}) . \qquad \square$$

The knowledge connected with a strongly compressible subset is redundant within the knowledge base in the sense that a single attribute from this set provides the same characterization of objects as does the whole set. Consequently the number of attributes (the amount of knowledge) required to distinguish between all the objects occurring in the considered universe may be reduced.

Proposition 9. Let *B* be a strongly compressible subset of *A*. Suppose $a, b \in B$. Then

$$\{a,b\} \not\subseteq C \in Red(A) .$$

Proof. Suppose the contrary. Let *C* be a reduct of *A* containing both attributes $a, b \in B$, which implies by the definition of strongly compressible that $Dis(\{a\}) = Dis(\{b\})$. Now apply Lemma 2 to conclude $Ind(C - \{a\}) = Ind(C) = Ind(A)$ contradicting the hypothesis that *C* is a reduct. $\square$

Based on this result, *at most* a single attribute from any strongly compressible subset can occur in any reduct of A. This concept can be extended to the components $\{f_1,...,f_p\}$ of a simple cover of f_A as shown in

Proposition 10. Let $\{f_1,...,f_p\}$ be a simple cover of f_A. Let B be strongly compressible. Let $\{a,b\} \in B$. Then

$$\{a,b\} \not\subseteq C \in Red(f_i) .$$

Proof. Suppose the contrary. Let C be a reduct of f_i such that $\{a,b\} \subseteq C \in Red(f_i)$. Since $\{a,b\} \in B$ and B is strongly compressible we obtain $Dis(\{a\}) = Dis(\{b\})$. Now apply Lemma 2 to $Ind(C-\{a\}) = Ind(C) = Ind(f_i)$ contradicting the hypothesis that C is a reduct of f_i. ∎

The equivalence relations determined by discernibility, $a \sim_{Dis} b \Leftrightarrow Dis(\{a\}) = Dis(\{b\})$, are of considerable power but of limited use since the equivalence classes are often singleton sets. However a closely related equivalence relation, obtained by replacing the definition of the relation in terms of the discernibility function by the $\chi_{\cdot}$ functions is of considerable computational use in the calculation of reducts.

Let $\{f_1,...,f_p\}$ be a simple cover of f_A. For each *i=1,...,p*, define a relation on A by

$$a \sim_{f_i} b \Leftrightarrow \chi_{C_j}(\{a\}) = \chi_{C_j}(\{b\}) \;\; for\ j = 1,...,k_i$$

where $f_i = \chi_{C_1} \wedge ... \wedge \chi_{C_{k_i}}$.

It is readily verified that for each index i, the above condition determines an equivalence relation on A. Denote the equivalence class of a under the equivalence relation determined by f_i by $[a]_i$.

Note: By the definition of the $\chi_{\cdot}$ functions it is equivalent to say

$$a \sim_{f_i} b \Leftrightarrow (C_j \cap \{a\} = \varnothing \Leftrightarrow C_j \cap \{b\} = \varnothing)\ for\ j = 1,...,k_i .$$

In words: the attribute a is equivalent to the attribute b iff the two attributes are simultaneously either present or absent in each of the indexing sets of each conjunct. Let us introduce some terminology.

Let $\{f_1,...,f_p\}$ be a simple form of f_A. Let $B \subseteq A$. If *there exist* an attribute $a \in A$ such that $B \subseteq [a]_i$ then we say B is a **local strongly compressible** subset. When we need to emphasize the particular index i, we say B is a local strongly compressible subset of f_i. A local strongly complressible subset is characterized by

Proposition 11. A subset $B \subseteq A$ is a local strongly compressible subset of f_i if and only if

$$b \in B \Rightarrow \begin{cases} b \in C_j & j = 1,\ldots,k \\ or & \\ b \notin C_j & j = 1,\ldots,k \end{cases}$$

where $f_i = \chi_{C_1} \wedge ... \wedge \chi_{C_{k_i}}$.

Proof. This follows from the above note and the fact that $\sim_{f_i}$ is an equivalence relation. ∎

The utility of a local strongly compressible subset is stated in

Proposition 12. Let $\{f_1,...,f_p\}$ be a simple form of f_A. Let B be a local strongly compressible subset of f_i. Suppose $\{a,b\} \in B$. Then $\{a,b\} \not\subseteq C \in Red(f_i)$.

Proof. Suppose the contrary, $a,b\in C$. Let *C* be a reduct of f_i. Since $\chi_{D_j}(\{a\})=\chi_{D_j}(\{b\})\quad j=1,...,k_i$ we obtain, since $a,b\in C$, that $f_i(C-\{a\})=f_i(C)$ which implies $Ind(C-\{a\})=Ind(C)=Ind(f_i)$ contradicting the hypothesis that *C* is a reduct of f_i. ∎

Note: If $a\sim_{f_i} b$ then $\chi_{C_j}(\{a\})=\chi_{C_j}(\{b\})\qquad j=1,...,k$. Thus we can replace each element of $[a]_i$ occurring in each C_j by a single representative, so

$$(\chi_{C_1}\wedge...\wedge\chi_{C_{k_i}})(B)=1\quad\Leftrightarrow\quad(\chi_{\hat{C}_1}\wedge...\wedge\chi_{\hat{C}_{k_i}})(B)=1$$

where $\hat{C}_j$ is C_j after substituting with representatives.

We now show the relationship between strong compressible and local strongly compressible.
Proposition 13. Let. A subset $B\subseteq A$ is strongly compressible iff it is local strongly compressible,

$$Dis(\{a\})=Dis(\{b\})\Leftrightarrow\chi_{C_{ij}}(\{a\})=\chi_{C_{ij}}(\{b\})\qquad\forall C_{ij}\neq\varnothing$$

Proof. We prove the contrapositive statement.

$$\begin{gathered}Dis(\{a\})\neq Dis(\{b\})\\ \Updownarrow\\ \exists(x_i,x_j)\in Dis(\{a\})\quad and\quad(x_i,x_j)\notin Dis(\{b\})\quad(or\ vice-versa)\\ \Updownarrow\\ a(x_i)\neq a(x_j)\quad and\quad b(x_i)\neq b(x_j)\\ \Updownarrow\\ a\in C_{ij}\quad and\quad b\notin C_{ij}\\ \Updownarrow\\ \chi_{C_{ij}}(\{a\})=1\quad and\quad\chi_{C_{ij}}(\{b\})=0\end{gathered}$$

∎

The knowledge connected with a strongly compressible subset is redundant within the knowledge base in the sense that a single attribute from this set provides the same characterization of objects as does the whole set. Consequently the number of attributes (the amount of knowledge) required to distinguish between all the objects occurring in the considered universe may be reduced.

V. REDUCT GENERATION

All the results presented in the preceding sections are exploited in the algorithm to follow. At each stage, elements of a local strongly compressible subset will be replaced by a single attribute from the subset. Moreover, each repetition of step 3 maintains a simple cover form of the discernibility function under the expansion law.

Reduct Generation Algorithm

Given: $f_A=f_1=\bigwedge_{\substack{1\le i\le j\le n\\ C_{ij}\neq\varnothing}}\chi_{C_{ij}}$ which is an initial simple cover of f_A.

Step 1. In each component f_i of the simple cover, apply the absorption law to eliminate all conjuncts χ_D where there exist a conjunct χ_C such that $C\subseteq D$.

Step 2. Replace each local strongly compressible subset of attributes in each simple cover component f_i by a single attribute that represents this class. A local strongly compressible subset is identified in each component f_i if the corresponding set of attributes is simultaneously either present or absent in each indexing subset of its conjuncts.

Step 3. In each component f_i of the simple cover select an attribute $a \in A$ which belongs to the largest number of indexing sets C_i, numbering at least two, and apply the expansion law. Write the resulting form as a disjunction $f_i = f_{i1} \vee f_{i2}$.

Step 4. Repeat steps 1 through 3 until f_A is in a simple form.

Step 5. For each component f_i of the resulting simple form, substitute all local strongly compressible classes for their corresponding attributes, i.e., replace each function χ_C by $\chi_{\hat{C}}$ where $\hat{C} = \cup\{[a]_i : a \in C\}$.

Step 6. Calculate the reducts $Red(f_i)$.

Step 7. Determine the minimal elements, with respect to the inclusion relation, of the set $\bigcup_{i=1}^{p} Red(f_i)$, where $f_A = f_1 \vee ... \vee f_p$. These minimal elements are the elements of $Red(A)$.

Example 2. To illustrate the reduct generation algorithm consider the discernibility function (without the explicit χ notation)

$$f_A = \{a,b,c,f\} \wedge \{b,d\} \wedge \{a,d,e,f\} \wedge \{b,c,d\} \wedge \{b,d,e\} \wedge \{d,e\}$$

1. Since $\{b,d\} \subset \{b,d,e\}$ and $\{b,d\} \subset \{b,c,d\}$ we use the absorption law to eliminate conjuncts 4 and 5 and get an equivalent discernibility function:

$$f_A = \{a,b,c,f\} \wedge \{b,d\} \wedge \{a,d,e,f\} \wedge \{d,e\}$$

2. $\{a,f\}$ is a strongly compressible class so we can represent it by a single attribute *g* which yields:

$$f_A = \{g,b,c\} \wedge \{b,d\} \wedge \{g,d,e\} \wedge \{d,e\}$$

3. The remaining function attribute d is the most frequent so we apply the expansion law with respect to this attribute to obtain

$$\begin{aligned} f_A &= f_1 \vee f_2 \\ &= (\{d\} \wedge \{g,b,c\}) \vee (\{g,b,c\} \wedge \{b\} \wedge \{g,e\} \wedge \{e\}) \\ &= (\{d\}) \wedge \{g,b,c\}) \vee (\{b\} \wedge \{e\}) \end{aligned}$$

where the simplification in the last step resulted from the absorption law.

4. All functions f_i are in simple form.

5. Substituting all strongly compressible classes for their equivalent attributes we get

$$\begin{aligned} f_A &= f_1 \vee f_2 \\ &= (\{d\} \wedge \{a,f,b,c\}) \vee (\{b\} \wedge \{e\}) \end{aligned}$$

6. Reducts which correspond to the simple cover functions are

$$\begin{aligned} Red(f_1) &= \{\{a,d\},\{d,f\},\{b,d\},\{c,d\}\} \\ Red(f_2) &= \{\{b,e\}\} \end{aligned}$$

7. The reducts of *A* are obtained by determining the minimal elements of the set

$$\bigcup_{i=1}^{2} Red(f_i) = \{\{a,d\},\{d,f\},\{b,d\},\{c,d\},\{b,e\}\}$$

from which we conclude

$$Red(A) = \{\{a,d\},\{d,f\},\{b,d\},\{c,d\},\{b,e\}\}.$$

(The reducts of *A* are obtained by "throwing away" supersets in $\bigcup_{i=1}^{p} Red(f_i)$; in this example there are no supersets.)

It is essential that the algorithm has the desired property of determining all the reducts. This is shown in
Proposition 12. The algorithm determines all the reducts of *A*,

$$Red(A) \subseteq \bigcup_{i=1}^{p} Red(f_i)$$

Proof. There are two main points to prove:
(1). The algorithm converges (doesn't loop indefinitely), and
(2). $Red(A) \subseteq \bigcup_{i=1}^{p} Red(f_i)$.

Part (1) follows from Proposition 8. To prove part (2), note that the form $f_A = f_1 \vee ... \vee f_k$ follows from applying the expansion law repeatedly in step 3. Then

$$\begin{array}{l}
B \in Red(A) \\
\quad \Downarrow \text{Prop. 3} \\
\text{(a) } f_A(B) = 1, \text{and} \\
\text{(b) } C \subset B \Rightarrow f_A(C) = 0 \\
\quad \Downarrow \\
\text{(a) } f_1(B) \vee ... \vee f_k(B) = 1, \text{and} \\
\text{(b) } C \subset B \Rightarrow f_1(C) \vee ... \vee f_k(C) = 0 \\
\quad \Downarrow \\
\text{(a) } (\exists l) f_l(B) = 1, \text{and} \\
\text{(b) } C \subset B \Rightarrow f_l(C) = 0 \\
\quad \Downarrow \\
(\exists l)\ B \in Red(f_l) \\
\quad \Downarrow \\
B \in \bigcup_{i=1}^{p} Red(f_i)
\end{array}$$

□

VI. Results of Computer Simulation

Simulations were run using MATLAB 5.2 on test data generated randomly. A random number generator provided uniformly distributed numbers to represent each attribute of each record. These values were multiplied by 8 and then the fractional part was truncated. This resulted in integer attribute values between 0 and 8. The number of attributes varied from 10 to 40 in increments of 5 and the number of records varied from 10 to 40 in increments of 5. All simulations were accomplished using a dual Pentium Pro 200 MHz computer using 256MB of memory. Figure 1 illustrates how the run times increase with problem size using the Expansion Algorithm. Note the abscissa is $\log_{10}$ of the run time. The curves shown are for 10, 15, 20, 25, 30, 35, and 40 attributes. Note that the computational time is growing exponentially.

Figure 2 shows the difference in time to run the problems using the Elimination Method and the new Distribution Algorithm. The graph only shows the results for 10 and 15 attributes. This is because when the problem size was larger than this, the elimination method required so much time that results could not be obtained without the simulation running for many days! Note that the time expressed is the $\log_{10}$ of the time.

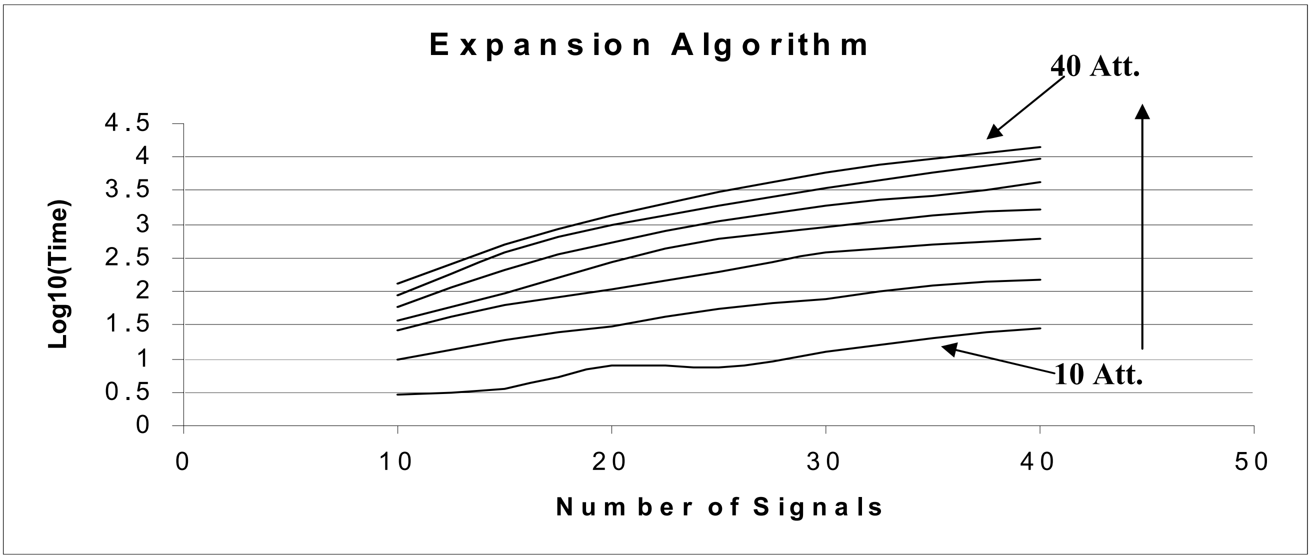


Figure 1. Expansion Algorithm Run Times

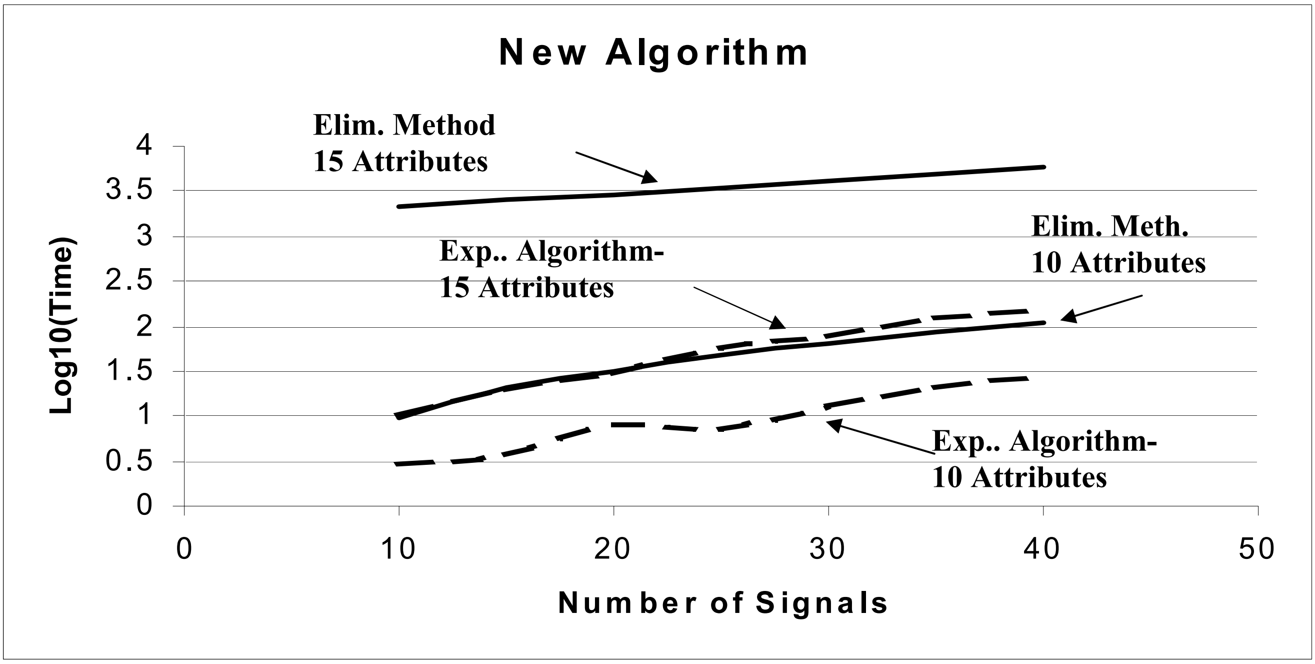


Figure 2. Time Savings of the Expansion Algorithm vs. the Elimination Method

Figure 3 shows the run times for the Expansion Algorithm with and without strong equivalence. Incorporating strong equivalence into the Expansion Algorithm does cost computational time. However, as seen in Figure 3, the time savings can be significant (as much as 50%) when strong equivalence is present.

Figure 4 illustrates the specific run times with and without strong compressibility. Graphed in this fashion it is easy to see the reduction in run time due to exploiting strong compressibility. The examples run did not have strongly compressible subsets present for more than 25 signals with 40 attributes and for more than 30 signals for 30 attributes.

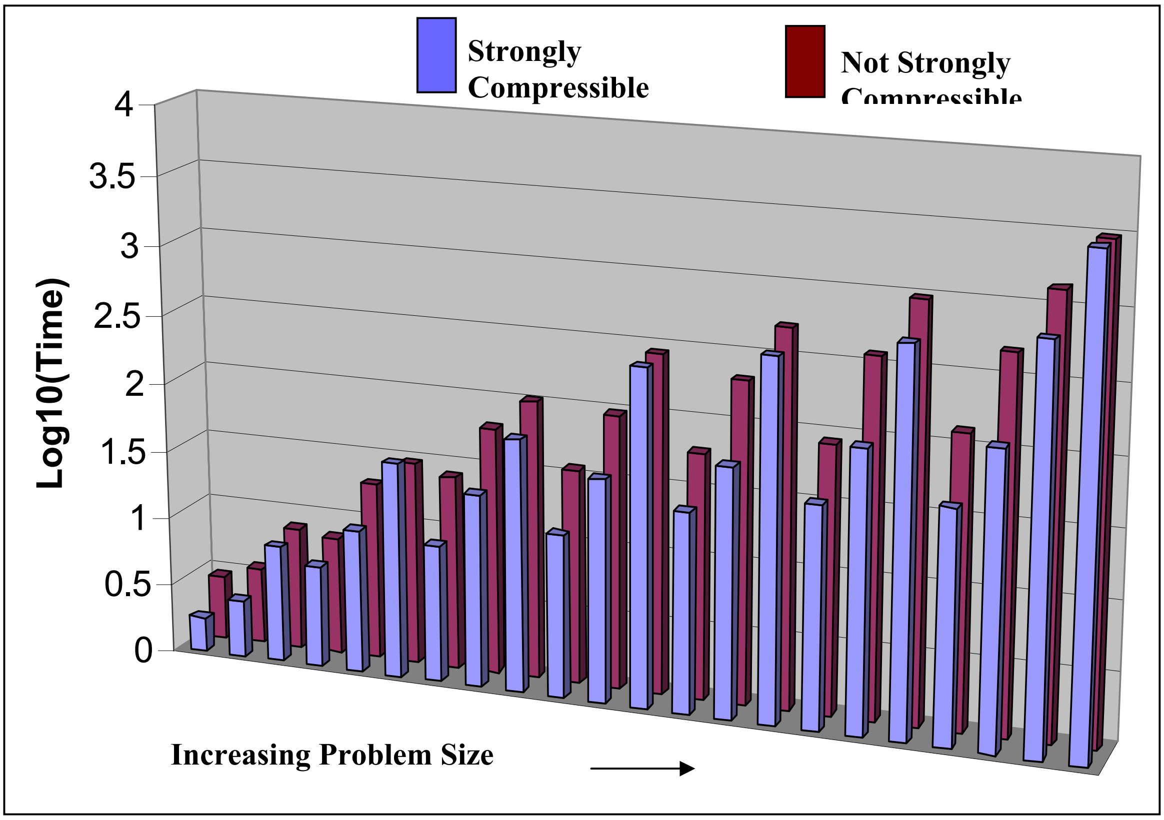


Figure 3. Time Savings with Strong Compressibility.

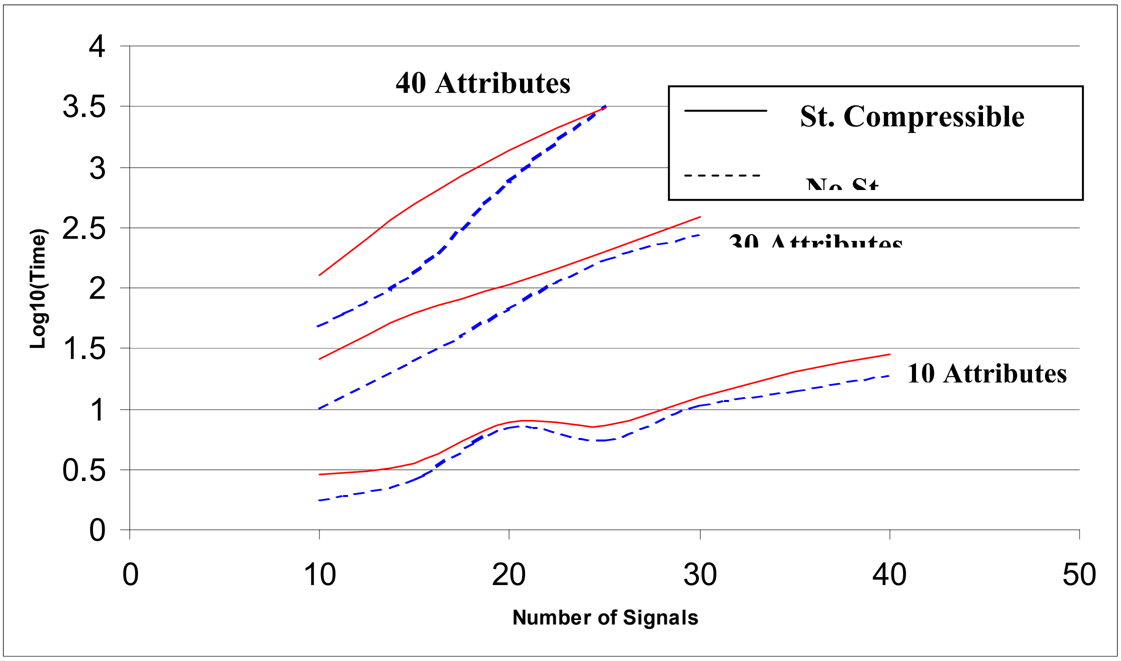


Figure 4. Benefit of Strong Compressibility

VII. Conclusions

We have shown that the use of the Expansion Algorithm allows the generation of all reducts in a much less time than the elimination method. Further, this algorithm is ideal for implementation on multiprocessor computers. Using this algorithm, larger problems should be able to be addressed.

The addition of strong equivalence to the Expansion Algorithm further reduces computation time when strong compressibility is present. In the simulations we ran, strong compressibility was not always present and thus the run times were increased. It is possible that in real world problems, where structure is present, strong compressibility will manifest itself more frequently. Therefore, the computational time should be reduced in most cases.

This work presented a study of knowledge reduction in information systems using properties of the discernibility function and proposed an improved version of a reduct generation algorithm. Further simplification of reduct generation may be achieved using graph theory and hierarchical partitioning. This topic is under investigation and will be presented in another paper.

VIII. References

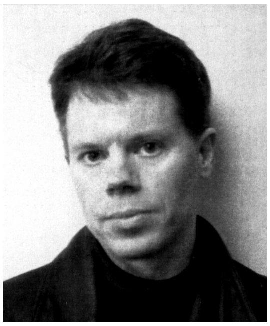
1. W. Buszkowski and E. Orlowska, "On the logic of database dependencies", *Bull. Polish Sci. Math.*, Vol 34, pp345-354, 1986.
2. A. Nakamura and G. Jian-Miang, "A modal logic for similarity-based data analysis", Hiroshima Univ. Technical. Report., 1988.
3. Z. Pawlak, "Information systems - theoretical foundations", *Information Systems*, Vol. 6, pp.205-218, 1981.
4. Z. Pawlak, "On rough dependency of attributes in information systems", *Bull. Polish Acad. Sci. Tech*. Vol. 33, pp.481-485, 1985.
5. Z. Pawlak, Rough Sets - Theoretical Aspects of Reasoning About Data, *Kluwer Academic Publ*., 1991.
6. Rasiowa and A. Skowron, "Approximation logic", *Mathematical Methods Specification and Synthesis of Software Systems*, Akademie-Verlag, Berlin, Band 31, pp. 123-139,1986.
7. A. Skowron and J. Stepaniuk, "Towards an approximation theory of discrete problems: Part I", *Fundamenta Informaticae* 15(2), pp.187-208, 1991.
8. A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems", *Fundamenta Informaticae* 15(2), pp.331-362, 1991.
9. D. Vakarelov, "Modal logic of knowledge representation systems", *Lecture Notes on Computer Science*, Springer Verlag, 363, pp.257-277, 1989.
10. W. Ziarko, "Acquisition of design knowledge from examples", *Math Comput. Modeling* Vol. 10, pp. 551-554.

Janusz A. Starzyk received the M.S. degree in applied mathematics and Ph.D. degree in electrical engineering from Warsaw University of Technology, Warsaw, Poland, in 1971 and 1976 respectively. From 1977 to 1981 he was an Assistant Professor at the Institute of Electronics Fundamentals, Warsaw University of Technology. From 1981 to 1983 he was a Post-Doctorate Fellow and research engineer at McMaster University, Hamilton, Canada. He is currently a professor of EECS at Ohio University, Athens, Ohio.

He has been a consultant to National Institute of Standards and Technology, AT&T Bell Laboratories, Sverdrup Technology, and Magnetek Corporation. He was a visiting professor at the University of Florence, Italy. He was a visiting researcher at Redstone Arsenal and the Air Force Research Laboratory. Current research interests iclude neural networks, rough sets, VLSI/VHDL design, computer-aided design of analog MOS circuits, and mixed signal testing.

Dale E. Nelson received the B.S. degree in Aerospace Engineering from Tri-State University, Angola, Indiana in 1969. He received the M.S. degree in Computer Science from Wright State University in 1981. His is currently a candidate for the Ph.D. degree from Ohio University, Athens, Ohio. He holds a certificate as a Certified Computer Programmer/Systems from the Institute for Certification of Computer Professionals.

He began his career in 1969 as a structural fatigue engineer at Cessna Aircraft, Wichita, Kansas. In 1972 he accepted a position as a preliminary design engineer at the Air Force Flight Dynamics Laboratory. He served at the Aeronautical Systems Division database administrator from 1986 to 1989. He is currently Chief or the Target Recognition Branch of the Air Force Research Laboratory. Current research interests include automatic target recognition, rough sets, neural networks, and learning theory.



Kirk E. Sturtz earned his Baccalaureate in Mechanical Engineering at the University of Toledo, his Masters in Aerospace Engineering from Iowa State University, Ames Iowa, and his Doctorate in Systems Science and Mathematics from Washington University, St. Louis, Missouri. His is employed at Veridian, Incorporated working at the Air Force Research Laboratory, Wright Patterson Air Force Base, Ohio. His research interests include invariant theory and category theory.