Collaboration Control in Distributed Knowledge-Based Systems

ZBIGNIEW W. RAS

University of North Carolina, Department of Computer Science, Charlotte, N.C. 28223, USA

Polish Academy of Sciences, Institut of Computer Science, 01-237 Warsaw, Poland

ras@mosaic.uncc.edu

ABSTRACT

A distributed knowledge-based system (DKBS) is a collection of autonomous knowledge-based systems called agents. We assume here that these agents are capable of interacting with each other. They work together in solving queries submitted to DKBS according to their respective abilities. Each agent is represented by an information system (collection of data) and a dictionary (collection of rules). In [10], we proposed that an agent of DKBS can request from other agents rough descriptions of all unknown attribute values used in a query submitted to him. Some of these descriptions do not have to be consistent because they are created independently by different agents. The problem of repairing such descriptions (see [15]) requires additional interaction between all involved agents. Communication control between them is necessary.

Key Words: intelligent information system, cooperative query answering, communication control, multi-agent system, knowledge discovery.

1. Introduction

Traditional query processing provides exact answers to queries. It usually requires that users fully understand the database structure and content to issue a query. Due to the complexity of the database applications, incorrectly formulated queries are frequently posed and the users often receive no answers or they might need more information than they have received. In this paper a multi-agent system called a collaborative knowledge-based system (CKBS) is presented to rectify these problems. An agent in CKBS is represented by an information system (collection of data) and a dictionary (collection of rules).

Knowledge is usually separated into factual knowledge and belief. If two agents share a factual knowledge, it must be the same, unless one or both are wrong. The belief on the other hand can be contradictory and hence it need not to be consistent among agents. We assume that knowledge stored in information systems is a factual one. The knowledge stored in dictionaries, as we mentioned before, is in the form of rules and it may represent beliefs of many agents. The set of attribute values of an information system and the set of heads of rules in a dictionary form an alphabet of the initial language used by an agent. When the dictionary of an agent is updated (new rules are added or some rules are deleted), the language used by an agent will automatically changed. We assume that all dictionaries are initially empty. In CKBS, an agent learns rules describing attribute values foreign for him from other agents of the system. By learning, we mean here receiving rules, when needed,! from other agents of the system and storing them in an agent dictionary. The condition part of such rules should use, if possible, only terms of the language used by an agent who requested these rules. When the time progresses more and more rules are added to an agent dictionary which means that some attribute values (decision parts of rules) which were foreign to him are becoming a part of his vocabulary. The choice which agents should communicate first (minimum one of them should learn descriptions of new attribute values) is based on the type of user queries and on the number of common attribute values in vocabularies of agents languages. For instance if two agents $A1$ and $A2$ have to communicate with agent $A3$, the priority is given to $A1$ only if vocabularies of agents $A1$ and $A3$ have more overlapping attribute values than vocabularies of agents $A2$ and $A3$.

Another approach for handling queries foreign to an agent was proposed by Chu [3], [4], Gaasterland [7], and others. However, their approach does not handle queries built from values of locally foreign attributes. We can resolve such queries through collaboration between agents.

Whenever we interact with knowledge based system and ask a question, we expect to receive a worthwhile answer in reply (see [3],[5],[7]). Clearly, we should build our CKBS with this intention in mind. Similarly as Gaasterland [7], we assume that collaborative answer should be a correct, nonmisleading, and useful answer to a query. To be more precise, we assume that each contribution (rule created by one of the agents of CKBS) should be:

- Locally valid.
  Saying another words, an agent should only create rules that he believes to be true. Elements of dictionaries satisfy that criterion.

- Relevant to an agent which has requested that rule.

We mean here that rules should be described in a language which is local, if possible, for a receiving site.

2. Basic Definitions

In this section, we introduce the notion of an information schema, an information system, a distributed information system, a dictionary, and a query language (called local) used by an agent represented by site i.

By an information schema we mean a pair (A, V) such that A is a finite set of attributes and V is a set-theoretical union of domains of attributes from A. We assume that:

- $V = \bigcup\{V_a : a \in A\}$ is finite,
- $V_a \cap V_b = \emptyset$ for any $a, b \in A$ such that $a \neq b$.

By an information system we mean a structure $S = (X, A, V, f)$ where X is a finite set of objects, (A, V) is an information schema, and f $(f : X \times I \longrightarrow V)$ is a classification function that describes objects in terms of their attribute values. We assume that:

- $f(x, a) \in V_a$ for each $a \in A, x \in X$.

The table-representation of a classification function f $(f : X \times I \longrightarrow V)$ is naturally identified with an information system $S = (X, A, V, f)$. For instance, let us assume that $S_1 = (X_1, A_1, V_1, f_1)$ is an information system where $X_1 = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$, $A_1 = \{B, C, D, E\}$ and $V_1 = \{b_1, b_2, c_1, c_2, c_3, d_1, d_2, e_1, e_2, e_3\}$. Additionally, we assume that $V_B = \{b_1, b_2\}$, $V_C = \{c_1, c_2, c_3\}$, $V_D = \{d_1, d_2\}$ and $V_E = \{e_1, e_2, e_3\}$. Then, the function f_1 defined by Table 1 is identified with an information system S_1.

By a distributed information system [10] we mean a pair $DS = (\{S_i\}_{i \in I}, L)$ where:

- $S_i = (X_i, A_i, V_i, f_i)$ is an information system for any $i \in I$,
- L is a symmetric, binary relation on the set I,
- I is a set of sites.

Systems S_i, S_j (sites i, j) are called neighbors in a distributed information system DS if $(i, j) \in L$. The transitive closure of L in I is denoted by L^+.

Now, we introduce the notion of a dictionary D_{ki}, $(k, i) \in L^+$, which contains rules describing values of attributes from $A_k - A_i$ in terms of

X_1	B	C	D	E
$a1$	$b1$	$c1$	$d1$	$e1$
$a2$	$b1$	$c2$	$d1$	$e2$
$a3$	$b2$	$c3$	$d2$	$e1$
$a4$	$b1$	$c2$	$d1$	$e2$
$a5$	$b1$	$c3$	$d1$	$e3$
$a6$	$b2$	$c1$	$d2$	$e3$
$a7$	$b2$	$c2$	$d1$	$e2$

Table 1. Function f_1

values of attributes from $A_k \cap A_i$ (see [16]). We begin with the definition of a set T_i of $s(i)$-terms and their standard interpretation M_i in a distributed information system $DS = (\{S_j\}_{j \in I}, L)$, where $S_j = (X_j, A_j, V_j, f_j)$ and $V_j = \bigcup\{V_{ja} : a \in A_j\}$ for any $j \in I$. Elements in T_i represent queries local for a site i.

The set T_i of $s(i)$-terms is a least set such that:

- $w \in T_i$ for any $w \in V_i$, and
- if $t_1, t_2 \in T_i$, then $(t_1 + t_2), (t_1 * t_2),\ t_1 \in T_i$.

Standard interpretation M_i of $s(i)$-terms in a distributed information system $DS = (\{S_j\}_{j \in I}, L)$ is defined as follows:

- $M_i(w) = \{x \in X_i :$ if $w \in V_{ia}$ then $f_i(x, a) = w\}$ for any $w \in V_i$,
- for any $s(i)$-terms t_1, t_2, t:

$$M_i(t_1 + t_2) = M_i(t_1) \cup M_i(t_2),$$
$$M_i(t_1 * t_2) = M_i(t_1) \cap M_i(t_2),$$
$$M_i(\ t) = X_i - M_i(t)$$

By (k, i)-rule in $DS = (\{S_j\}_{j \in I}, L)$, $k \in I$, $i \in I$, we mean a triple (c, t, s) such that:

- $c \in V_k - V_i$
- t, s are $s(k)$-terms in DNF and they both belong to $T_k \cap T_i$
- $M_k(t) \subset M_k(c) \subset M_k(t + s)$

For any (k, i)-rule (c, t, s) in $DS = (\{S_j\}_{j \in I}, L)$, we say that:

- $t \Rightarrow c$ is a k-certain rule in DS and
- $t + s \Rightarrow c$ is a k-possible rule in DS.

There are several software packages for learning k-certain and k-possible rules in a system S_k. For instance, we can mention here system $LERS$ developed at the University of Kansas, $Rough - Sets - Library$ developed at Warsaw University of Technology, $DataLogic$ developed by Rough Sets Group from University of Regina or $CKB - System$ developed at UNC-Charlotte. All these systems ask a learner which attributes of the information schema are classification attributes and which attribute the learner wants to learn (such an attribute is called a decision attribute and its values are treated as foreign values for the learner). Rules generated by these systems describe values of a decision attribute in terms of values of classification attributes. An algorithm for learning k-certain and k-possible rules can be found, for instance, in a book by Grzymala-Busse [8].

A dictionary D_{ki} is defined as a set of (k, i)-rules. Its elements can be seen as rough descriptions of values of attributes from $V_k - V_i$ in terms of values of attributes from $V_k \cap V_i$.

By a distributed knowledge-based system we mean a pair
$DS = ((S_i, R_i)_{i \in I}, L)$ where:

- $(\{S_i\}_{i \in I}, L)$ is a distributed information system,
- $R_i \subset \bigcup\{D_{ki} : k \in K_i\}$ where D_{ki} is a dictionary for any $i \in I$, $k \in K_i$.

A distributed knowledge based system is i-consistent if for any two rules $(c_1, t_1, s_1), (c_2, t_2, s_2) \in R_i$, if $c_1 \neq c_2$, then $M_i((t_1 * t_2)) = \emptyset$.

A pair $DS = ((S_i, R_i, \{(A_j, V_j)\}_{j \in J_i})_{i \in I}, L)$ is called a collaborative knowledge based system ($CKBS$) of type k if $((S_i, R_i)_{i \in I}, L)$ is a distributed knowledge based system, $J_i \subset I - \{i\}$, $card(J_i) \leq k$ and (A_j, V_j) is an information schema for any $j \in I$. In addition, we assume that site i can learn descriptions of new attribute values (not in V_i) only from sites in J_i. Clearly, more sites we ask, more precise descriptions we probably get. Unfortunately, rules received by a learner from more than one site of DS, describing the same value of an attribute, are often inconsistent. To repair them, new queries have to be resolved and new information passed between sites of $CKBS$. We investigated this problem in [16].

An example of $CKBS$ with two sites is given in Fig. 1. Rules in a dictionary D_{21} have been created at $SITE2$ and rules in D_{12} at $SITE1$. Clearly, values $b1, b2, c1, c2, c3, d1, d2, e1, e2, e3$ are local for $SITE1$. Values of attributes $f1, f2, g1, g2$ are foreign for $SITE1$. With a dictionary D_{21} stored at $SITE1$ and a dictionary D_{12} stored at $SITE2$, learning between

X1	B	C	D	E
a1	b1	c1	d1	e1
a2	b1	c2	d1	e2
a3	b2	c3	d2	e1
a4	b1	c2	d1	e2
a5	b1	c3	d1	e3
a6	b2	c1	d2	e3
a7	b2	c2	d1	e2

Dictionary D21

(f1, e1, d1*e3)
(f2, d2*e3, d1*e3)
(g1, e1+c1*d1, c2*d1*e3)
(g2, d2*e3, c2*d1*e3)

SITE 1

X2	F	C	D	E	G
a1	f1	c1	d2	e1	g1
a6	f2	c1	d2	e3	g2
a8	f1	c2	d1	e3	g1
a9	f2	c1	d1	e3	g1
a10	f2	c2	d1	e3	g1
a11	f1	c2	d1	e3	g2
a12	f1	c1	d1	e3	g1

Dictionary D12

(b1, d1*e1+d1*e3, d1*e2)
(b2, d2, d1*e2)

SITE 2

Fig. 1. SITE1 and SITE2 of CKBS

these two sites is completed. If the data stored in an information system of one of the sites are changed, learning between some involved sites has to be initiated again.

3. Query Language and Its Interpretation

In this section we introduce a query language and propose its interpretation, called standard, by one of the sites of $CKBS$. We give a complete and sound set of axioms and rules for handling queries in $CKBS$.

Standard interpretation M_i, introduced in previous section, shows how to interpret $s(i)$-terms in a $CKBS$. The question of interpreting a term built from values of attributes belonging to $V = \bigcup\{V_j : j \in I\}$ remains open. Such terms are called global in $CKBS$. Their i-standard interpretation in i-consistent collaborative knowlege based system $DS = ((S_i, R_i, \{(A_j, V_j)\}_{j \in J_i})_{i \in I}, L)$, where $S_i = (X_i, A_i, V_i, f_i)$ for any $i \in I$ is proposed in this section. Elements in $V - V_i$ are called concepts for a site i.

By a query language $L(DS)$ we mean a sequence (A, T, F), where A is an alphabet, T is a set of terms and F is a set of formulas. Both terms and formulas represent queries.

The alphabet A contains:

- constants: w , where $w \in V$
- constants: $\mathbf{0}, \mathbf{1}$
- functors: $+, *, \sim$
- predicate: $=$
- connectives: $\vee, \wedge, \neg, \Rightarrow$
- auxiliary symbols: (,).

The set of terms T is a least set such that:

- if w is a constant, then w is a term
- if t_1, t_2 are terms, then $(t_1 + t_2)$, $(t_1 * t_2)$, $\sim t_1$ are terms.

Parentheses are used, if necessary, in the obvious way. As will turn out later, the order of a sum or product is immaterial.
So, we will abbreviate finite sums and products as $\sum\{t_j : j \in J\}$ and $\prod\{t_j : j \in J\}$, respectively. Intentionally, terms are names of certain features of parts being processed by $CKBS$, more complex than those expressed by constants.
The set of formulas F is a least set such that:

- if t_1, t_2 are terms, then $(t_1 = t_2)$ is a formula, and
- if α, β are formulas, then $(\alpha \vee \beta)$, $(\alpha \wedge \beta)$, $(\alpha \Rightarrow \beta)$, $(\neg\alpha)$ are formulas.

Let M_i be a standard interpretation of $s(i)$-terms in $DS = (\{S_i\}_{i \in I}, L)$. By i-standard interpretation of queries from $L(DS)$ in i-consistent collaborative knowledge based system $DS = ((S_i, R_i, \{(A_j, V_j)\}_{j \in J_i})_{i \in I}, L)$, where $S_i = (X_i, A_i, V_i, f_i)$ and $V_i = \bigcup\{V_{ia} : a \in A_i\}$, we mean the interpretation N_i such that:

- for any $w \in V_i$,

  $$N_i(w) = \{x \in X_i : \text{if } w \in V_{ia} \text{ then } f_i(x, a) = w\},$$

- if $w \in V - V_i$,

  $$N_i(w) = \{x \in X_i : (\exists t, s)[(w, t, s) \in R_i \wedge x \in M_i(t)]\},$$
  $$N_i(\sim w) = \{x \in X_i : (\exists t, s)[(w, t, s) \in R_i \wedge x \notin M_i(s)]\},$$

- $$N_i(\mathbf{0}) = N_i(\neg\mathbf{1}) = \emptyset,$$
  $$N_i(\mathbf{1}) = N_i(\neg\mathbf{0}) = X_i,$$

- for any terms t_1, t_2, t,

$$N_i(t_1 + t_2) = N_i(t_1) \cup N_i(t_2),$$
$$N_i(t_1 * t_2) = N_i(t_1) \cap N_i(t_2),$$
$$N_i(\sim (t_1 + t_2)) = N_i(\sim t_1) \cap N_i(\sim t_2),$$
$$N_i(\sim (t_1 * t_2)) = N_i(\sim t_1) \cup N_i(\sim t_2),$$
$$N_i(\sim (\sim t_1)) = N_i(t_1),$$

- for any terms t_1, t_2 and any formulas a, b

$$N_i((t_1 = t_2)) = (\text{if } N_i(t_1) = N_i(t_2) \text{ then } T \text{ else } F)$$
(T stands for True and F for False)

- $$N_i(\alpha \vee \beta) = N_i(\alpha) \vee N_i(\beta),$$
$$N_i(\alpha \wedge \beta) = N_i(\alpha) \wedge N_i(\beta),$$
$$N_i(\alpha \Rightarrow \beta) = N_i(\alpha) \Rightarrow N_i(\beta),$$
$$N_i(\neg\alpha) = \neg N_i(\alpha)$$

From the point of view of a site i, the interpretation N_i represents a pessimistic approach to query evaluation. It means that $N_i(t)$ is interpreted as a set of objects in X_i which have the property t for sure. We are not retrieving here objects which might have property t.

Let us adopt the following set A of axiom schemata for our formal theory:

- Substitutions of the axioms of distributive lattices for terms and the axioms of equality
- $\sim w * w = \mathbf{0}$ for any constant w
- $\sim v + v = \mathbf{1}$ for each $v \in V_i$
- for each $v \in V_i$,

$$\sim v = \Sigma\{w : w \in V_{ia} \wedge v \neq w \wedge v \in V_{ia}\}$$

- $v_1 * v_2 = \mathbf{0}$ if $v_1, v_2 \in V_{ia}$ for some $a \in A_i$
- for any term t,

$$\sim \mathbf{0} = \mathbf{1},$$
$$\sim \mathbf{1} = \mathbf{0},$$
$$\mathbf{1} + t = \mathbf{1},$$
$$\mathbf{1} * t = t,$$
$$\mathbf{0} * t = \mathbf{0},$$
$$t + \mathbf{0} = t,$$
$$\sim (\sim t) = t$$

- for any terms t_1, t_2,

$$\sim (t_1 + t_2) = (\sim t_1) * (\sim t_2),$$
$$\sim (t_1 * t_2) = (\sim t_1) + (\sim t_2)$$

- for any $w \in V - V_i$,

$$w = \Sigma\{t : (w, t, s) \in R_i\}$$

- for any $w \in V - V_i$,

$$\sim w = \Sigma\{t : (\sim w, t, s) \in R_i\}$$

- Substitutions of the propositional calculus axioms.

The rules of inference for our formal system are the following:

- from $(\alpha => \beta)$ and α we can deduce β for any formulas α, β
- from $t_1 = t_2$ we can deduce $t(t_1) = t(t_2)$, where $t(t_1)$ is a term containing t_1 as a subterm and $t(t_2)$ comes from $t(t_1)$ by replacing some of the occurrences of t_1 with t_2.

We write $A \vdash \alpha$ if there exists a derivation from a set A of formulas as premises to the formula α as the conclusion.

We will write $A \models \alpha$ to denote the fact that A semantically implies α, that is, for any i-standard interpretation N_i of $L(DS)$ in i-consistent cooperative knowledge-based system we have $N_i(\alpha) = T$.

THEOREM 1. For any formula α in $L(DS)$, $A \models \alpha$ iff $A \vdash \alpha$.

The above theorem, can be used to prove that for any $L(DS)$ term t there is $L(DS)$ term t_1 such that $A \vdash (t = t_1)$. This result allows us to resolve global queries in $CKBS$ eficiently since each conjunct can be easily resolved by a simple match.

4. Collaboration Control in CKBS

In this section, we introduce the notion of a communication graph of $CKBS$, DS-requests, and weakly directed subgraphs of a communication graph. Communication graphs are weighted graphs. To optimize the collaboration control between nodes of $CKBS$, we will look for weakly directed subgraphs of a communication graph. We propose an efficient algorithm for constructing an optimal weakly directed subgraph.

Let us assume that $DS = ((S_i, R_i, \{(A_j, V_j)\}_{j \in J_i})_{i \in I}, L)$ is a collaborative knowledge based system, where $S_i = (X_i, A_i, V_i, f_i)$ for any $i \in I$.

By a communication graph for DS we mean a sequence (I, L^+, r), where $r : L^+ \longrightarrow Z^+$ is a weighted function defined below (Z^+ is a set of non-negative integers):

- $r((i,j)) = 0$ if $j \notin J_i$ (it means that site i can not contact site j),
- $r((i,j)) = n$ if $j \in J_i$ and $card(A_i \cap A_j) = n$.

So, the communication graph shows the size of the overlap between alphabets of the local languages for every pair of agents (sites) of $CKBS$. If the overlap is larger, agents can understand each other better.

Let $l_1 = (i_1, j_1)$, $l_2 = (i_2, j_2) \in L^+$. We say that $\{l_1, l_2\}$ is weakly directed if $j_1 \neq j_2$. Similarly, a set $E \subseteq L^+$ is weakly directed if all its two element subsets are weakly directed. A subgraph (I, E, r) of a graph (I, L^+, r) is weakly directed if E is weakly directed. Weakly directed sets are introduced to represent a situation when an agent can not be contacted by two different agents at the same time.

Let us assume that $DS = ((S_i, R_i, \{(A_j, V_j)\}_{j \in J_i})_{i \in I}, L)$ is a collaborative knowledge-based system. By a DS-request we mean a set $R = \{(a_i, A_i) : i \in I_R\}$, where $a_i \in A_i$ and $I_R \subseteq I$. Semantically, DS-request R provides a list of attributes each agent from I_R has to learn. More precisely, an agent i has to learn the attribute a_i.

Each DS-request R defines a subgraph $G_R = (J_R, L_R, r_R)$ of a communication graph (I, L^+, r) as follows:

- $L_R = \{(i,j) \in L^+ : [(a_i, A_i) \in R \wedge r((i,j)) > 0 \wedge a_i \in A_j]\}$,
- $J_R = \{i \in I : (m_j \in I)[(i,j) \in L_R]\} \cup \{j \in I : (m_i \in I)[(i,j) \in L_R]\}$,
- $r_R : L_R \longrightarrow Z^+$ where $r_R((i,j)) = r((i,j))$.

Graph $G_R = (J_R, L_R, r_R)$ shows the visibility of implementing DS-request R in a system DS.

A weight $w(G_R)$ of a graph G_R is defined as $\Sigma\{e : e \in L_R\}$. Similarly, we define a weight of any of its subgraphs.

THEOREM 2. Let $m = card(L_R)$. There is $0(m)$ algorithm for constructing an optimal weakly directed subgraph of a graph G_R (of a maximum weight).

Proof. The graph is represented as an adjacency list structure. We travel through the list twice. In the first loop, for every node in a graph, we search for an incoming edge of maximal weight. During the second loop, we can easily check if the visited edge, let us say (a, b) is the optimal one for b. The complexity of our algorithm is $0(2 \cdot m)$. □

The weight of an optimal weakly directed subgraph of a graph G_R shows how efficient is the overal learning between all involved agents. The number is higher, the learning is more efficient.

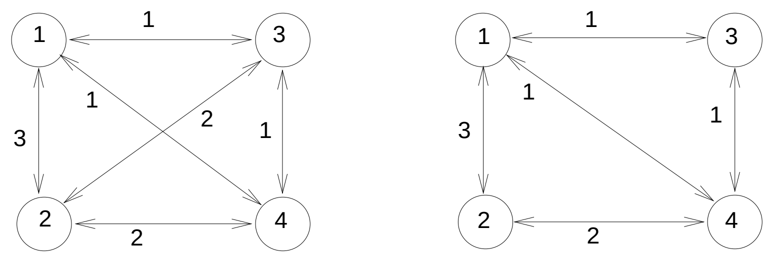

Fig. 2. Communication Graph and Graph G_R

Let us consider an example of $CKBS$ of type 3 with four sites $\{1,2,3,4\}$. So, any two sites of our system can talk to each other. Assume also that site i is represented by an information system $S_i = (X_i, A_i, V_i, f_i)$, for any $i \in \{1,2,3,4\}$. Let $A_1 = \{a,b,c,d\}$, $A_2 = \{b,c,d,e,f,g\}$, $A_3 = \{a,e,f,n,r\}$, and $A_4 = \{a,f,g,h,m\}$. So, $card(A_1 \cap A_2) = 3$, $card(A_1 \cap A_3) = card(A_1 \cap A_4) = 1$, $card(A_4 \cap A_3) = 1$, $card(A_4 \cap A_2) = 2$, and $card(A_2 \cap A_3) = 2$. The communication graph for $CKBS$ is given by Figure 2. Assume now that $R = \{(A_1, f), (A_4, e)\}$ is a DS-request. Because $f \in [A_2 \cap A_3 \cap A_4]$ and $e \in [A_2 \cap A_3]$, then $G_R = (\{1,2,3,4\}, \{(1,2),(1,3),(1,4),(4,2),(4,3)\}, r_R)$ and $w(G_R) = (3+1+1)+(1+2) = 8$. There are four weakly directed subsets of L_R, namely $L_{R1} = \{(1,2),(1,3),(1,4)\}$, $L_{R2} = \{(1,3),(1,4),(4,2)\}$, $L_{R3} = \{(1,2),(1,4),(4,3)\}$, $L_{R4} = \{(1,4),(4,2),(4,3)\}$. Since $W(G_{R1}) = 5, W(G_{R2}) = 4, W(G_{R3}) = 5, W(G_{R4}) = 4$, then there are two weakly directed optimal subgraphs of GR (see Figure 3).

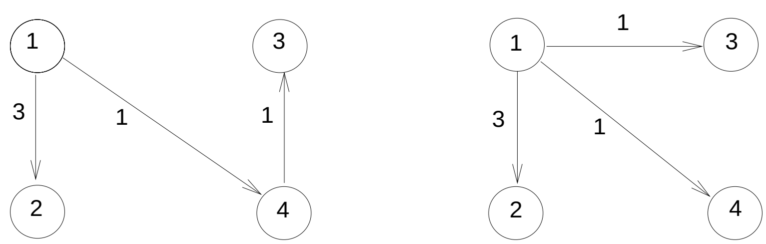

Fig. 3. Weakly Directed Optimal Subgraphs of G_R

This paper presents a methodology and theoretical foundations of a collaborative knowledge-based system (*CKBS*) which is partially implemented at UNC-Charlotte on a cluster of *SPARC*2 workstations. The query answering system of *CKBS* identifies all locally foreign attributes used in a query entering site k. Next, the query answering system sends a message to all allowable sites of k that rules approximating these foreign attributes are needed. Each allowable site, which can help k, invokes a program similar to *LERS* ([8]) which computes rules describing all these foreign attributes using terms of the language of site k. Finally, these rules are sent to site k and used by the query answering system to replace foreign values of attributes in a query by local terms.

REFERENCES

1. Bazan, J., Skowron, A., Synak, P., "Dynamic reducts as a tool for extracting laws from decisions tables", *Methodologies for Intelligent Systems, Proceedings of 8th International Symposium ISMIS'94*, Charlotte, N.C., October, 1994, Lecture Notes in Artificial Intelligence, Springer Verlag, No. 869, 1994, 346-355

2. Bosc, P., Pivert, O., "Some approaches for relational databases flexible querying", *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 1, 1992, 323-354

3. Chu, W.W., "Neighborhood and associative query answering", *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 1, 1992, 355-382

4. Chu, W.W., Chen, Q., Lee, R., "Cooperative query answering via type abstraction hierarchy", In *Cooperating Knowledge-based Systems* (ed. S.M. Deen), North Holland, 1991, 271-292

5. Cuppers, F., Demolombe, R., "Cooperative answering: a methodology to provide intelligent access to databases", *Proceedings 2nd International Conference on Expert Database Systems*, Virginia, USA, 1988

6. Deen, S.M., "A general framework for coherence in a CKBS", *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 2, 1993, 83-107

7. Gaasterland, T., Godfrey, P., Minker, J., "An overview of cooperative answering", *Journal of Intelligent Information Systems*, Kluwer Academic Publishers, Vol. 1, 1992, 123-158

8. Grzymala-Busse, J., *Managing uncertainty in expert systems*, Kluwer Academic Publishers, 1991

9. Kacprzyk, J. "On measuring the specificity of if-then rules", *International Journal of Approximate Reasoning*, Vol. 11, No. 1, 1994, 29-53

10. Maitan, J., Ras, Z.W., Zemankova, M., "Query handling and learning in a distributed intelligent system", in *Methodologies for Intelligent Systems*, 4, (ed. Z.W. Ras), North Holland, 1989, 118-127

11. Nakamura, A., Matsueda, M., "Rough logics based on incomplete knowledge systems", *Proceedings of the Third International Workshop on Rough Sets and Soft Computing*, November, 1994, San Jose State University, California, 1994, 56-63

12. Pawlak, Z., "Rough Sets - theoretical aspects of reasoning about data", Kluwer Academic Publishers, Dordrecht, 1991

13. Pawlak, Z., "Rough sets and decision tables", *Proceedings of the Fifth Symposium on Computation Theory*, Springer Verlag, Lecture Notes in Computer Science, Vol. 208, 1985, 118-127

14. Pawlak, Z., "On learning - a rough set approach", *Proceedings of the Fifth Symposium on Computation Theory*, Springer Verlag, Lecture Notes in Computer Science, Vol 208, 1985, 197-227

15. Ras, Z.W., "Fault-recovery and intelligent distributed systems", *Proceedings of ISMVL'90*, Charlotte, N.C., IEEE Computer Society Press, 1990, 372-377

16. Ras, Z.W., "Query processing in distributed information systems", in *Fundamenta Informaticae Journal*, Special Issue on Logics for Artificial Intelligence, IOS Press, Vol. XV, No. 3/4, 1991, 381-397

17. Ras, Z.W., Chilumula, N., "Answering queries by cooperative knowledge-based system", *Proceedings of the Third International Workshop on Rough Sets and Soft Computing*, November, 1994, San Jose State University, California, 1994, 586-592

18. Skowron, A., "Boolean reasoning for decision rules generation", in *Methodologies for Intelligent Systems*, Proceedings of 7th International Symposium ISMIS'93, Trondheim, Norway, June, 1993, Lecture Notes in Artificial Intelligence, Springer Verlag, No. 689, 1993, 295-305