Telecommunications and Radio Engineering, 75 (7):621-630 (2016)

SOLVING OF SAT – PROBLEMS OF ARTIFICIAL INTELLIGENCE WITH THE HELP OF LOCAL ELIMINATION ALGORITHMS

M.A. Miroshnik, R.I. Tsekhmistro[♦]***, & A.I. Demichev***

Ukrainian State University of Railway Transport
7, Feierbakh Sq., 61050, Kharkiv, Ukraine

[♦]Address all correspondence to R.I. Tsekhmistro E-mail: 1970ts@rambler.ru

Using elimination algorithms are promising for solving of the problem of propositional calculus testing. These problems are widely used in the practical checking of digital electron circuits of modern telecommunications. Based on the fuzzy logic, acyclic graph these algorithms provides a way for typical presentations of SAT-problems.

KEY WORDS: *resolver, resolution, graph of interrelationships, directed acyclic graph, local algorithm*

1. INTRODUCTION

Approaches and algorithms of an artificial intelligence (AI) allows solving multiple application problems like theory of scheduling [1], problems related to designing expert systems and decision support systems [2], proving of theorems, and problems of electronic schemes testing and images processing.

The constraint satisfaction problem is one of the important problems of AI [3]. Most of the interesting problems of AI are NP-difficult and their solution may require in the worst scenario possible an enumeration of an exponential number of solutions. Most practical problems contain a huge number of variables and/or constraints that creates difficulties while trying to solve these problems with the help of the present-day resolvers.

The graph decomposition methods describing the AI problems, are promising decomposition approaches with using the sparse graphs structure [4]. The interest to which has grown during recently. It is explained by the results obtained by Arnborg et al. [5] proving that a number of NP-difficult problems set in the monadic second order logic can be solved for a polynomial time with the help of the dynamic programming methods on the graphs describing the structure of the problem with a constrained tree width.

The class of local elimination algorithms (LEA) for information computing [6] including local decomposition algorithms [1,4], non-serial dynamic programming (NSDP) algorithms [6,7]), segmented elimination algorithms [8], and tree decomposition methods [5] are referred to the graph decomposition approaches.

Application of LEA to solving the problems of discrete optimization (DO) is considered in [6]. Achievements of the graph decomposition schemes allowing coupling with solutions to NP- difficult problems with the help of dynamic programming algorithms [8] caused the interest in application of these methods in the spheres other than optimization. LEA can also be used for solving of non-optimization problems, which can be divided into sub-problems with using the obtained solutions to smaller sub-problems while solving the large ones. The interrelationship of NSDP with local algorithms ([1,4,6]) is stipulated by the fact that same as local algorithms NSDP solves problems by converting local information into the global one. The elimination game [8] represents a graph interpretation, which is one of the common features of the above methods.

2. SETTING OF THE PROBLEM

Intelligent systems of processing and representing the knowledge are presently developed towards the integration of symbol and image representations of scientific knowledge. In combination with well-developed hardware and software multimedia tools this trend in engineering of science has a great practical importance especially at the stage of conversion of paper documents into their electronic equivalents. In this situation there occur some problems related to the form of representation of knowledge, configuration of user interfaces, recognition of the input information and providing for high speed of its search and processing.

Due to a significant increase of the volumes of information systems (in local access networks, electronic libraries, e-catalogs etc.) and limited capabilities for navigation and search for the information the problems of developing new approaches and improving efficiency of the existing methods for search of the information acquire greater importance. Investigations in these spheres have been performed actively during recently both in Ukraine and abroad. One can determine the following main directions of the investigations concerned:

- extraction of informational objects from the documents, determination of their characteristics (statistical, linguistic, semantic);
- development of the semantic structure of documents;
- topical analysis and topical search of information in the document repository;
- topical classification, clustering and filtering of the documents.

While solving the above problems there are applied elements of the theory of information systems, tools from the theories of graphs and fuzzy mathematics, and the theory of decision making.

Based on the foregoing the problem of designing artificial intelligence systems using the elements of fuzzy logic appears to be extremely important. In this paper it is suggested one of the approaches to solving the problem under consideration.

In order to solve a number of sparse discrete problems of artificial intelligence computing and subsequent use of local information (i.e., the information about the elements in the neighborhood) while solving such problems is possible with the help of local elimination algorithms for computing of the information allowing performance of computation of global information with the help of local calculations.

The class of local elimination algorithms for computing of information and their use while solving the constraint satisfaction problems is considered below.

3. CONSTRAINT SATISFACTION PROBLEMS

The constraint satisfaction problems (CSP) [11] are widely used for solving of multiple practically important AI problems like scheduling, designing of electronic schemes, and decision support. This paper will consider the CSP problems with discrete variables.

Flowing ratios are used at setting the constraints.

Definition. For the data in the set of variables $X = \{x_1, \ldots, x_n\}$ and corresponding to them domains of the values $D_1, \ldots, D_n$ **the ratio** R on the set of variables shall mean any subset of the Cartesian product of domains of their values. The set of variables, upon which it is determined the ratio R, is called the **scope** of the ratio and designated as $scope(R)$.

The ratios can be set with the help of the tables describing the acceptable combinations of values of the variables.

Definition. The CSP is defined by the set of discrete variables $x_1, \ldots, x_n$, for each of the variables it is set an area of definition or the domain $D_j = \left\{d_j^{(1)}, \ldots, d_j^{(p_j)}\right\} \left(j = 1, \ldots, n\right)$, and by the set of constraints. **A constraint** means the pair (R,S), where R is the ratio determined upon the scope S. The solution to CSP is represented by assigning values to all the variables, which assigning satisfies all the constraints. **The objective** of solving a CSP can be in finding on solution to the problem or all of the solutions.

The following operations are defined to be performed with the ratios**: crossing** $R_1 \cap R_2$ means the ratio including all the arrays of values of the variables, which are available simultaneously in both ratios R_1 and R_2; **combination** $R_1 \cup R_2$ means the ratio including all the arrays of values of the variables, which are available either in R_1 or in R_2, or in both ratios; **difference** R_1 - R_2 means the ratio including the arrays of values of the variables, which are available in R_1, but not included into R_2; **projection** $\Pi_Y\left(R\right)$ of the ratio R upon the set of variables $Y \subseteq scope\left(R\right)$ means the ratio including all the arrays of values of the variables, which are available in Y only;

Combination of the ratios R_S with the scope S and R_T with the scope T $R_S \bowtie R_T$ means the ratio including their common variables in S and T. The scope of the obtained ratio is $S \cup T$. Combination of two ratios having the same scopes is equivalent to crossing of such ratios.

Examples of the constraint satisfaction problems. The problem of graph coloring, the problem of scheduling and the SAT-problem are considered among the most important examples of CPS.

The problem of scheduling. It is given a set $V = \{v_i\}$ of the learning courses in the university. There are known the time intervals T_i, during which the relevant course is taught $\{v_i\}$. It is required to distribute the courses among the classrooms in a way that two courses were not taught in the same time and in the same classroom. This problem can be reduced to searching for a proper coloring of the graph $G = (V, E)$, where $(v_i, v_j) \in E \Leftrightarrow T_i \cap T_j \neq$ Ø. In this case each classroom is matched with its own 'color'.

SAT-problem. The SAT (satisfiability) problem (the problem of checking validity of the propositional calculus formula) has an important applicable value, whereas the applications are lying in the sphere of electronic schemes testing, designing of computers and image analysis [12]. The SAT-problem is in determining whether the given propositional calculus formula is true at any value of the literals. A solution to the SAT-problem implies the **interpretation**, i.e., such assignment of trustworthiness values (1 or 0) to the literals of the formula, at which the given formula becomes truth.

4. LOCAL ELIMINATION ALGORITHMS FOR COMPUTING THE INFORMATION

The works on practical implementation of fuzzy controllers and regulators and creation of intelligent control systems on their basis, along with implementation of expert systems with a fuzzy logic into industrial and non-industrial spheres are actively performed in the developed countries of the world. Over 400 practical applications of fuzzy controllers and control systems are known by now. The experts are of the option that about 70% of all the developments related to intelligent systems will be based on fuzzy logic in the nearest future. Despite various architectural solutions and related thereto different rates of fast-action of the developed and being developed in present software and hardware tools used for processing fuzzy knowledge, all of them are united by orientation towards one of the possible modifications of the fuzzy logic output algorithms, and namely – of the compositional output. This algorithm is efficiently used in the systems of fuzzy control over dynamic objects, which operate according to the principle of the controller. At that, a vast class of the systems, which are based on decision-making and situational control, remains absolutely non-covered by it. The hard- and software computer appliance FuzEx – FuzCop [1,2] can be used for designing and programming of fuzzy processors used in such systems, as well as in the systems applied for controlling dynamic objects.

FuzEx represents an integrated computer appliance for designing systems, which are based on fuzzy knowledge using either a booster or its program emulator to provide for an efficient fuzzy logic output. The booster of fuzzy logic output based on the FuzCop fuzzy processor is designed for executing hardware support of fuzzy logic intelligent systems operating both on the basis of situational fuzzy logic output and the compositional fuzzy logic output.

FuzEx contains five basic components: the vocabulary editor, the editor of productions, the implementer of the systems based on fuzzy knowledge, the library of standard modules and system tools for booster support. The vocabulary editor serves for description of linguistic variables related to the domain of the subject. The editor of productions allows creation and modification of the knowledge database rules. The implementer is intended for setting the requirements to the system being designed and for assembling of the executable modules. The library of standard modules includes an array of procedures for input-output of information and the fuzzy logic output. System tools for booster support are represented by the program shell of the input language compiler, the booster microprograms downloader and the booster emulator. The input language compiler is used for translation of the program written in the input language, which is close to the high-level languages, into the microprograms of the fuzzy booster. The microprograms downloader serves for sending (downloading) of the obtained code into eigen memory of the booster. The emulator is intended for performance of the sequence of commands in the input language without using the booster.

The FuzShell system software tools included into the FuzEx computer appliance allow downloading data and internal microcommands into eigen memory of the fuzzy controller, or reading them; initializing the fuzzy processor contained in the structure of the controller for processing of fuzzy data; performing tuning of the processor to one or another type of logic output; exercising distribution of internal memory of the booster depending upon the number of attributes at estimation of the state-of-the-art and fuzzy productions. The FuzShell program shell possesses the tools for multi-window editing of both basic texts of the programs and the microprograms generated by the compiler. There exists an opportunity of tuning input-output ports of the booster and reading of the memory cells. Input of the state-of-the-art can be executed by means of data transmission on the basis of the DDE standard from another application (program), which is reading the information from the input sensors, with subsequent downloading of the obtained situation into the controller for further processing thereof.

As compared to analogs, FuzEx-FuzCop possesses a number of advantages, like the possibility of programming both the processor and the controllers; and availability of the graphic editor of the functions of belonging and fuzzy productions. A user-friendly interface is favorably distinguishing FuzEx among other program media. It allows setting linguistic variables in a natural manner; filling of the knowledge database; determining stages of the logic output sequence; reading the contents of internal memory cells of the fuzzy processor etc. Availability of command buttons permitting activation of some most frequently used commands relieves the user form the necessity of searching within the menu system of one or another command.

The FuzEx-FuzCop hard- and software computer appliance can be used while developing the decision-making systems. At studying complex objects obtaining (or computing) of full information about the object in general is not always affordable, therefore, an interest is in obtaining information about the object considering it on a part-by-part basis, i.e., locally. Local algorithms for computing of information suggested by Yu.I. Zhuravlev are described in [1].

The local algorithm (LA) is enumerating the elements in the order, which is set by the ordering algorithm A_π , using, at that, local information about the elements in the neighborhood [1,2] of the given element. The algorithm, which assigns the markers, performs computing of the function φ , the value of which at every step of the algorithm will determine the type of the marker assigned at the step concerned. The function φ that generates the algorithm is the function from two arguments, the first of which runs through the set of elements and the other – the set of neighborhoods. LA decompositions [4] of DO problems have specific features of their own concluded in the fact that they do not compute the predicates, but using the Bellman optimality principle, compute optimal partial solutions to the sub-problems corresponding to the blocks of the DO problem.

Computing and using of local information properly (i.e., the information about the elements in the element neighborhood) while solving the problems is an important particularity of local algorithms. For that reason, local elimination algorithms (LEA) of informational computing [6] allow performing computing of global information with the help of local computations.

The CSP structure is set by the constraints, and it can be set by both the system of neighborhoods of variables of the problem (the graph of interrelationships of the variables) and by the order of enumeration of these variables with the help of LEA [6], and by various derivate structures – the block ones [6,10], and the block-tree ones [5], which are set by the structurally condensed graphs. In a condensed graph the vertices represent subsets of variables of the problem.

LEA is computing the information about local elements of the CSP structure set by the structural graph by recording local information about such elements in the form of new dependences, which are added to the problem, with further elimination of the enumerated elements and used constraints. The algorithm scheme of LEA represents a directed acyclic graph (DAG), the vertices of which are represented by local sub-problems correspondent to the neighborhoods of the elements, while the arcs express the informational dependence of the sub-problems upon each other.

The LEA procedure consists of two parts:

- **the direct part** includes separation of local elements and their neighborhoods in a current structural graph along with corresponding to them sub-problems; computing and memorizing of local information in the form of new constraints added to the problem, elimination of the enumerated elements and used constraints, and obtaining the criterion value (whether CSP is satisfiable);

- **the reverse part** comprises finding a global solution to the initial CSP based on the available tables with local solutions in order to provide for attaining of the criterion (satisfiability of the CSP).

The direct part of LEA analyzes the neighborhood $Nb(x)$ of the current element x in the structural graph of the problem, applies to the given element the elimination operator comprising solving of the CSP sub-problem corresponding to the neighborhood $Nb(x)$ of the same element in the current structural graph, and computes a local information about the x in the form of the new constraint $(R,\{x\}\cup Nb(x))$ containing local solutions in the form of acceptable arrays of variables of the kind $R(x,Nb(x))$.

Then it is built the projection $R' = \Pi_{Nb(x)} R(x,Nb(x))$; the constraint $(R',Nb(x))$ is added to the system of constraints. Subsequently, the element x is eliminated together with the used constraints, and from the elements of its neighborhood $Nb(x)$ it is created a clique in the structural graph (the clique corresponds to the constraint $(R',Nb(x))$). Creation of the cliques modifies the structural graph and neighborhoods of the elements.

The reverse part of LEA reconstructs the solution of the entire CSP on the basis of the saved tables with local solutions $R(x,Nb(x))$.

In order to solve the CSP described by the variables $x_1,\ldots,x_n$ and the system of constraints $(R_1,S_1),\ldots,(R_m,S_m)$, where R_i is the ratio, $S_i = scope(R_i), i = 1,\ldots,m,$ and at the set ordering A_π of the variables the LEA has the following representation:

1. Selection of the next in line element x (a variable or a group of variables) according to the ordering A_π. Formulation of the sub-problem of the CS problem corresponding to the neighborhood $Nb(x)$ of the element x in the current structural graph by forming up a new constraint $(R,\{x\}\cup Nb(x))$ with the scope $(x,Nb(x))$, solving of the sub-problem concerned with memorizing in the table of all the solutions to the above constraint.
2. Projection of the obtained constraint upon the set of elements of the sub-problem corresponding to the neighborhood $Nb(x)$ of the element x. As the result, it is obtained a new constraint, which is added to the constraints of CSP. If a constraint with the same array of variables is already available, their crossing has to be found. If the crossing is empty, then CSP is unsatisfiable and has no acceptable solutions.
3. Elimination of the element x together with the related constraints.
4. Proceeding until there are no unsolved constraints.

Let us consider in more detail particularities of LEA realization while solving CS problems in the case when the structural graph is the **graph of interrelationships** of the variables, which is also called **the graph of constraints** in the reference literature.

In the graph of interrelationships of CSP the vertices correspond to the variables of CSP, at that, two vertices are connected by an edge, if relevant variables are available in one and the same constraint (i.e., within one and the same scope of a ratio). We consider the variable x_i and its neighborhood $U(x_i)$, $Nb(x_i)$ in the current graph of inter relationships G : $U(x_i)=\left\{R_p \middle| x_i \in scope(R_p)\right\}$, $Nb(x_i)=\left\{x_j \middle| \exists p : x_i, x_j \in scope(R_p)\right\}$. Let $R_{i_1},\ldots,R_{i_{m_i}}$ are the ratios with the scopes $S_{i_1},\ldots,S_{i_{m_i}}$, the indices of which are included into $U(x_i)$, at that, their scopes contain x_i. Solving of the CS sub-problem, which is set by the constraints $R_{i_1},\ldots,R_{i_{m_i}}$ and the relevant variables, where as $x_i \in scope\left(R_{i_p}, p=1,\ldots,m_i\right)$, and subsequent elimination of x_i can be described in the following manner. We define the scope of the new constraint as $S^{(i)}=\bigcup_{r=1}^{m_i} S_{i_r}$ - $\{x_i\}\times$ $R^{(i)}=\Pi_{S^{(i)}}\ {}_{r=1}^{m_i} R_{i_r}$ .and the new ratio. Then it is found the crossing with the previously existing ratio having the same scope $S^{(i)}$: $R^{(i)}=\mathrm{R}_{S^{(i)}} \cap R^{(i)}$. At elimination of the variable x_i the graph of interrelationships $G=(V,E)$ varies according to the elimination game algorithm [14]:

$$V \leftarrow V\text{-}\ \{x_i\};\ E \leftarrow E \bigcup \left\{(x_k,x_r)\middle| x_k,x_r \in Nb(x_i)\right\} \setminus \bigcup \left\{(x_i,x_j)\in E\right\}.$$

5. SAT-PROBLEM AND ITS GRAPH REPRESENTATION

Let us consider an example of solving the SAT-problem, which is defined in a conjunctive normal form consisting of 7 seven elementary disjunctions $C_1, C_2, C_3, C_4, C_5, C_6, C_7$, which are called disjunctives: $(x_2 \vee x_6)\wedge(x_5 \vee \neg x_6)\wedge(\neg x_1 \vee x_5)\wedge(\neg x_2 \vee \neg x_5)\wedge(x_1 \vee x_3 \vee x_4)\wedge(x_2 \vee \neg x_4)\wedge(x_1 \vee x_2)$.

The structure of the formula can be prescribes by the graph of interrelationships, i.e., an acyclic graph, the vertices of which correspond to the variables-literals, whereas the edge is joining two vertices, if the relevant variables are included into one and the same disjunctive of the formula (see Fig. 1).

The elimination operator in this case is the **resolution**, which draws on the basis of two disjunctives $(\alpha \vee Q)$ and $(\beta \vee \neg Q)$ the disjunctive $(\alpha \vee \beta)$ called the **resolvent**, in which the literal Q is eliminated. The elimination operator (in this case – the resolution) generates new disjunctives, which are corresponded by new edges in the graph of interrelationships. We determine the order of enumeration of the neighborhoods [2] based on application of the Minimal Degree heuristics:

$A_\pi = \{6, 5, 2, 4, 3, 1\}$ (Table 1). Application of LEA allows determining the solution to the given SAT-problem:

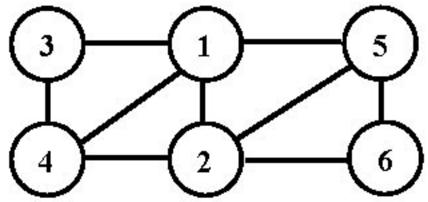

FIG. 1: Graph of interrelationships for the SAT-problem

TABLE 1: Order of enumeration of neighborhoods

x_1	x_2	x_3	x_4	x_5	x_6
0	1	0	1	0	0
0	1	1	1	0	0
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	1	1

Local elimination algorithms can also be used for solving the problems of discrete optimization, which find practical applications in many sectors of the telecommunication trends. Traffic optimization and routing in telecommunication networks represents one of the above-mentioned trends.

However, most of the problems of discrete optimization are difficult and their solving might require creation of an exponential-size tree for searching for solutions. Most practical problems of optimization contain a great number of variables or constraints that creates difficulties. In this respect, a good perspective for practical application [9] is provided by local elimination algorithms, which possess acceptable values of convergence and stability, in combination with randomizing algorithms having good independence and homogeneity parameters.

Elimination algorithms are also promising in the problems of artificial intelligence like the problems of calculation of probabilities in an expert system using the Bayes rule.

6. CONCLUSION

The local elimination algorithm for information computing represents a perspective approach providing for the possibility of solving application sparse constraint satisfaction problems including SAT- and graph coloring problems. Creation of efficient local elimination algorithm schemes at solving various constraint satisfaction problems possessing special structure, with the help of using different kinds of structural graphs is referred to promising trends of further development.

REFERENCES

1. Miroshnik, M.A., (2015), Development of information security in distributed computer systems and networks, *Information and management systems for railway transport*. 1:18-25 (in Russian).
2. Miroshnik, M.A., (2015), Development of intellectual diagnostic infrastructure in distributed computer systems, *Information and management systems for railway transport*. 2:25-36 (in Russian).
3. Miroshnik, M.A., (2015), Development of intellectual diagnostic infrastructure in distributed computer systems, *Information and management systems for railway transport*. 3:3-9 (in Russian).
4. Miroshnik, M.A., (2015), Design of computer systems with a built-in intellectual diagnostic infrastructure, *Radiotekhnika*, 180:64–67 (in Russian).
5. Miroshnik, M.A., Kotukh, V.G., and Selevko, S.N., (2013), Application of software complex for query processing in the database management system with a view of dispatching problem solving in Grid systems, *Telecommunications and Radio Engineering*. 72(10):875-891.
6. Miroshnik, M.A., (2013), Synthesis of distributed computer environments based on computer networks, *Data processing systems*. **7**(114):86-89 (in Russian).
7. Miroshnik, M.A. and Kovalenko, M.A., (2013), Uses of programmable logic integrated circuits for implementations of data encryption standard and its experimental linear cryptanalysis, *Information and management systems for railway transport*, 6:36-45 (in Russian).
8. Miroshnik, M.A., Kotukh, V.G., and German, E.Ye., (2015), Design of computer systems with a built-in intellectual diagnostic infrastructure, *Radiotekhnika,* 180:64–67 (in Russian).
9. Tsekhmistro, R.I., Kirichenko, L.O., Krug, O.Y., and Storozhenko, A.W., (2009), Comparative Analysis of Pseudorandom Number Generation in the Up-To-Date Wireless Data Communication, *Telecommunications and Radio Engineering*. **70**(4):325-333**.**