Solutions Outlining on the Set of Structured Technological Problems with Imposed Constraints

Vasyl Sheketa, Roman Vovk, Volodymyr Pikh, Yulia Romanyshyn, Kostiantyn Kravtsiv, Liudmyla Poteriailo, Volodymyr Protsiuk and Mykola Pasyeka

Ivano-Frankivsk National Technical University of Oil and Gas, Karpatska 15 street, Ivano-Frankivsk, 76019, Ukraine

Abstract

A method of estimating technological parameters is introduced, which allows to represent sets of preferences and express their influence on the process of satisfaction and violation of constraints by solving technological problems. At the final stage, a set of basic technological parameters is allocated, which allow to fully describe the technological process of drilling, by constructing systems of constraints and their ranking by relevance, which makes it possible to analyze an abnormal situation as a case of violation of technological parameters with imposed sets, systems and hierarchies of constraints. A formal structure consisting of a set of variables (technological parameters), a set of domains (confidence intervals) and a set of constraints is introduced, which allows to describe the technological process of drilling oil and gas wells in terms of formal-logical constructions of representation and satisfaction of constraints possible states.

Keywords

Constraints, comparators, preferences, weights, problems solving, reasoning, decisions-making, intelligent decisions-making support.

1. Introduction

In general case technological process of oil and gas wells drilling, is a very complex and dynamic process, the full formalization of which does not give the expected results in terms of completeness and correctness. An effective method of constructing solutions to the technological problems based on constraints [1,2] in drilling of oil and gas wells is the use of logical programming techniques in constraints [3–5]. This implementation will consist of several parts: the first part will contain the definition of all variables of the technological problem with their domains. Accordingly, the domains of the variables will be reduced due to the constraints that will be set in the next steps. Therefore, the search method in the solution space will be described by entering a label for a set of variables or by introducing an enumeration for value generation processes for individual variable domains. In this case, the search tree will be described based on the heuristic of the ordering of values and variables, which is applied before the assignment of values by calling constraint propagation procedures[6,7].

MoMLeT+DS 2021: 3rd International Workshop on Modern Machine Learning Technologies and Data Science, June 5, 2021, Lviv-Shatsk, Ukraine
EMAIL: vasylsheketa@gmail.com (V. Sheketa); r.vovk@nung.edu.ua (R.Vovk); pixclg@gmail.com (V. Pikh); yulromanyshyn@gmail.com (Y. Romanyshyn); kostya.kravtsiv@gmail.com (K. Kravtsiv); milapoteriailo@gmail.com (L. Poteriailo); v.v.protsiuk@gmail.com (V. Protsiuk); pms.mykola@gmail.com (M. Pasyeka)
ORCID: 0000-0002-1318-4895 (V. Sheketa); 0000-0003-0681-4534 (R.Vovk); 0000-0001-9420-5522 (V. Pikh); 0000-0001-7231-8040 (Y. Romanyshyn); 0000-0002-7916-3659 (K. Kravtsiv); 0000-0002-0501-5928 (L. Poteriailo); 0000-0003-0055-2780 (V. Protsiuk) ; 0000-0002-3058-6650 (M. Pasyeka)

The labeling procedure can also be used to find solutions to an optimization problem with an entered objective function. The method of binarization of constraints [8] when used as a strategy for solving technological problems based on constraints, is expected to increase the space for solutions, so the efficiency of the solution search procedure, in this case, will decrease. To eliminate this shortcoming in this case, it is advisable to use special algorithms for propagating constraints to solve selected subproblems of variables described on a certain subset by introducing global constraints [9,10]. Modeling the problem through the corresponding global constraints is one of the main ideas of logical programming in constraints in terms of computational efficiency. In such an application, global constraints will express a condition that must be met. In particular, when using global constraints for constrained problems, it will be advisable to introduce additional specifiers.

In the general case, such hierarchies is built on a set of constraint labels with an additional order relationship imposed by global constraint labels in the middle of each level of the hierarchy. Because hierarchical labels can be thought of as variable labels in a broader sense than fuzzy labels, hierarchy-level labels allow so far to more fully represent the semantics of labels as a whole. It should be taken into account a more complete solution strategy obtained in this case in comparison with fuzzy labels, which allows to show the correspondence between individual sets of labels and technological problems based on constraints and technological problems based on fuzzy constraints in general [11-13].

The question of constructing heuristics of ordering variables in technological problems on the basis of constraints with the choice of those variables that are most "critical" in terms of their substitution, i.e. in considering the most "critical" sets of variables with the most important preferences, *remains unexplored.*

Thus, the purpose of this research is to synthesize solutions, which can be considered as a search heuristic that processes the search tree simultaneously. It can also be interpreted as a method of narrowing a problem that restricts the whole set of variables, which narrows to such a level that constraints the space of possible labels to such an extent that it will contain only tuples of solutions.

2. Solutions refinement for technological problems

The use of multilevel intelligent technologies allows to optimize the drilling process of oil and gas wells by performing the necessary reconfigurations of equipment and applying methods of control of the drilling process through solving technological problems and preventing emergencies **[2]**.

Next figure presents the structuring of the drilling process in terms of available control and automation functions.

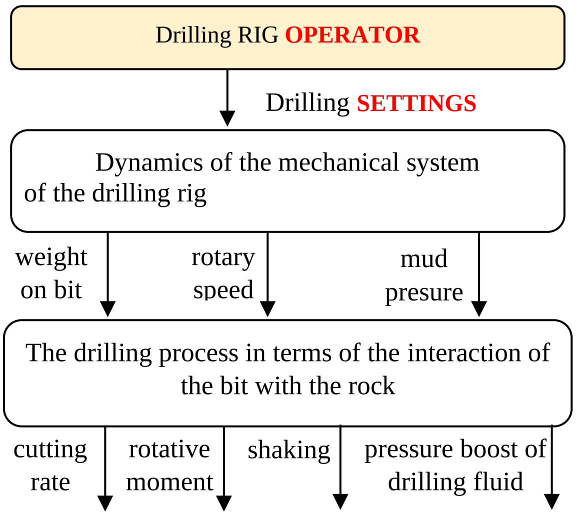

Figure 1: Modeling of the overall structure of the drilling process as of an object of automation in terms of controlled parameters

Thus, under the *technological problem* **TP** we do understand a certain essence of the subject area of *oil and gas wells drilling* that contains a problem that requires a solution that belongs to the space of possible solutions to the technological context and does consist in values assignment to the controlled technological parameters.

2.1. Comparative analysis of the comparators

So far, the classes of *technological problems in drilling* can be reduced to the corresponding classes of classical search problems based on constraints and there will be obviously a correspondence between the method of constructing of an solution to a technological problem and the way of solving classical search problems, in particular methods of working with solution space that is formed by the set of possible assignments. Since for the intelligent system is more important the process of finding a solution during which, by analyzing the violated and satisfied constraints, the system selects and applies some reasonable strategy, it is advisable to analyze the process of finding of all solutions, selecting of candidate solutions, and the process of finding of the best optimal solution accordingly to the specified criteria. The method of narrowing of the technological problem solution space reduces the size of the domains and possibly increases the constraints number. Other way the tightening of the constraints helps to narrow the search space in its relevant stages as well. Accordingly, the method of narrowing of the technological problem can be applied at any stage of the search. So far, there can be constructed some number of strategies for combining search routines with narrowing of the technological problem in different ways, which are effective under certain conditions presented in the form of constraints insofar. In the case of technological problems, "*domain best*" comparators perform assignments that cannot be implemented by local comparators. Assignments can be incomparable up to a certain level, and at the next one plus level the success functions of constraints can be compared separately. Thus, the formalization of technological problems on the basis of constraints for comparators "*lexicographically best*" can be done in the form of lexicographic technological problems on the basis of soft constraints. After introducing of weights for the comparator "*best on the sum of weights*" there can be introduced the form of technological problems on the basis of soft constraints with **CF** (certainty factors) and for "*locally best*" comparators – the form of local technological problems based on *crispy* soft constraints, respectively. In particular, it can be accepted for the subject of study that the comparators "*lexicographically better*" and "*best in terms of weights*" are so far equivalent to search problems based on weighted constraints.

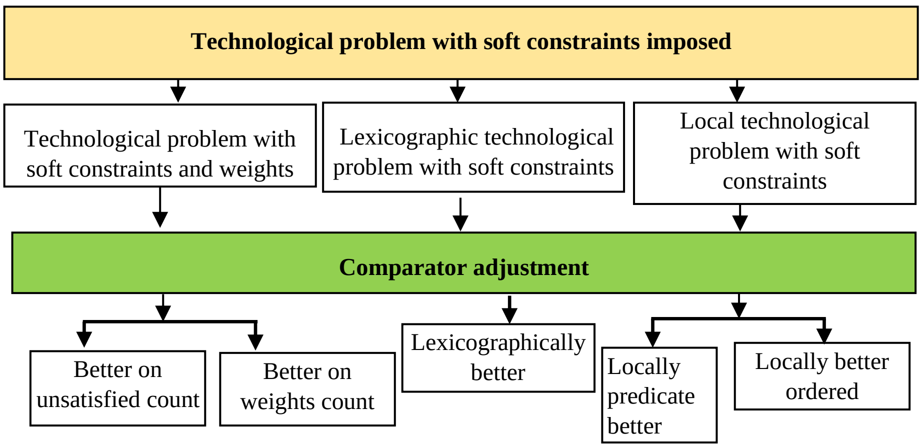

Figure 2: Classification of comparators on the level of constraints hierarchy

Such kind of equivalence is based on some well defined polynomial transformation. Hierarchy of constraints with local comparators does belong to some separate class of problems, because the formal meta structural sets introduced in them are only partially ordered. So far, every class of local technological problems based on soft constraints can be transformed into a class of search problems based on constraints with weights, by applying of relevant *refinement procedure* with polynomial characteristics. However, the construction of the inverse refinement procedure will be impossible due to the nature of partial ordering of the set of assignments for the hierarchy of constraints. For all identified classes of technological problems, it is possible to construct a relationship between these classes and classical search problems based on constraints and weights. The related figures do present the relationship between the basic classes of technological problems on the basis of constraints in relation to their properties. From the practical implementation reasons variable labels can be combined in the process of calculating global constraint labels, as well as in the process of determining the preference for choosing a solution based on the minimum optimization routine or by introducing of *objective function*.

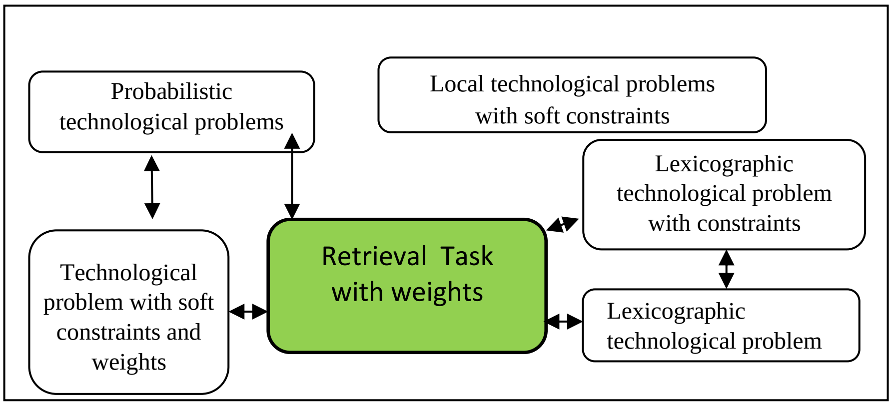

Figure 3: Relationships between basic classes of technological problems based on constraints

An alternative approach to work with labels, compared to the fuzzy approach, can be the way to interpret them too as a kind of entities based on constraints.

Classes of simple technological problems based on soft constraints with weights and lexicographic technological problems based on soft constraints can be considered as classes of *search problems with evaluations* based on the construction of some relationship between classes of technological problems based on soft constraints and search problems with evaluations. However, such a kind of matching routine would be not complete insofar, cause the search tasks with evaluations do require a complete ordering of the evaluated values of the controlled technological parameters [2]. So far , such a property would not be satisfiable for a class of local technological problems based on soft constraints, so it is not possible to define a class of search problems with evaluations for the hierarchy of constraints with any of locally based comparators.

Technological problems based on soft weight constraints can be considered as the first optional level for hierarchy of constraints with the comparator "*best in the sum of weights*". There should be a simple refinement with finite characteristics from the initial hierarchy of constraints with the comparator "*best by the sum of weights*" in the direction of the classical search problem with weights. Lexicographic technological problems based on soft constraints will correspond as well to the first optional level for hierarchy of constraints with the comparator "*lexicographically better*".

A local *technological constraints* based on soft constraints can be transformed into a problem that is equivalent to a *constraint-based search problem* with weights by some finite refinement. In this case, the optimal tuple of a local technological parameters based on soft constraints not only maximizes the success function of an individual constraint at each level, but also minimizes the sum of the weights of all the constraints that were violated. Thus, the local technological problem based on soft constraints cannot be specified as a clarification of a lexicographic technological problem based on soft constraints because of incomparable elements of their meta structures. Insofar a local technological problem based on soft constraints must also be incomparable with the class of technological problems based on classical search problems routines with weights for which the set of formal meta structure is completely ordered. In classical search problems, the process for finding of optimal solution is considered as an optimization problem. Consider the formal representation of such a process for the introduced classes of technological problems on the basis of constraints, the formal structure of which is presented in Figure 4. There are selected the main types of constraints that will be used to formalize technological problems:

1) constraints with weights $c^{weigh.}$;

2) constraints with probability coefficients $c^{probl.}$;
3) constraints with possibility coefficients $c^{posbl.}$;
4) constraints with estimated values $c^{ev.}$;
5) constraints with preferences $c^{pref.}$;
6) fuzzy constraints $c^{lv.}$ (constraint with a linguistic label) - a linguistic label characterizes the linguistic meaning of one of the characteristics of the constraint, such as validity .

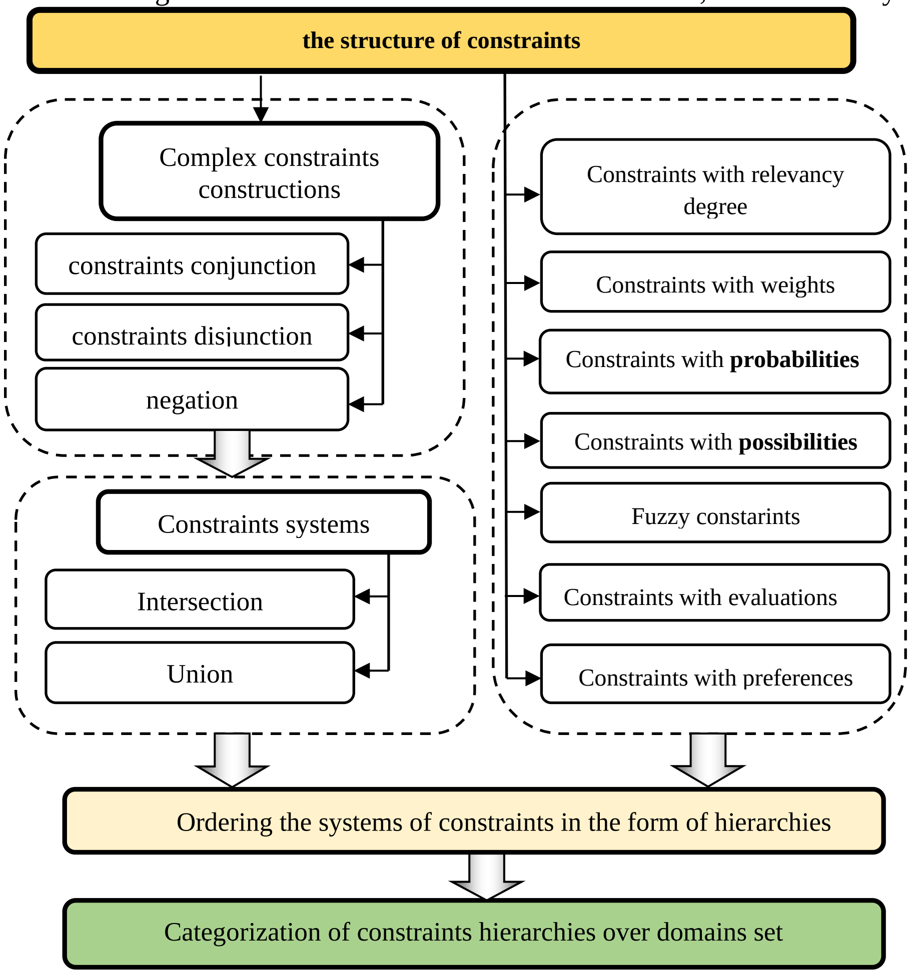

Figure 4: Classification and structuring of constraints for technological problems

To operate with constraints in the formal representations of technological problems, there is need for introducing of their following characteristics:

1. *The degree of relevance* (validity, $rd - relevancy\ \ degree$) – characterizes the degree of completeness of the descriptions made by constraint for the selected technological problem. This characteristic is considered as static profiling scheme.

2. *The degree of satisfaction* ($sd - satisfaction\ degree$) - dynamic characteristics of the run time stage. Value $sd = 1$ corresponds to the level of full satisfaction of the constraints and value $sd = 0$ corresponds to complete violation of the constraint. These values are marginal so far and as a rule, the degree of satisfaction(violation) will receive values from the range [0; 1]. It is also possible to consider the constraints in the terms of the probability of its satisfaction and of the probability of its relevance, which will be disclosed in the following definition.

3. *The weight of the constraint* is characterized by the weight value $(cw - constraint\ weight)$.

The system can operate both with individual constraints (with weights or estimated values, respectively), with systems of constraints **CS** *(constraints system)* and constraint hierarchies **CH** *(constraints hierarchy)* with a given number of levels. The hierarchy distinguishes between mandatory levels (constraints at this level must be met) and optional (satisfaction of constraints is preferential). Constraints may be satisfied in whole or in part with a degree of satisfaction *sd*. Complex constraints can be built on a set of introduced constraints on the basis of combining and intersecting of existing sets and systems of constraints, as well as on the basis of conjunction, disjunction and negation of individual constraints. Accordingly to the scope (*activity level*) constraints would be divided into *local*, *domain* and *global* ones.

The *degree of relevance* of the constraint c to *technological problem* TP will be considered as an measure of the relation of constraint c to *technological problem* TP in terms of completeness of its description. The degree of relevance will be indicated by $rd, rd \in [0;1]$. Degree of relevance $rd = 1$ means the absolute relevance of the constraint to the technological problem, and the degree of relevance $rd = 0$ means the absolute irrelevance of given constraint to this concrete problem.

2.2. Formal substantiation of the solutions

For each technological problem TP_i from the set $\{TP_i\}_{i=1..n}$, described by its own set of constraints $ConstrSet_i = \{c_i^j\}_{j=1..m}$, $m, n \in N$, the set of constraints with the introduced degree of relevance will look like $\{ConstrSet_i = \{c_i^j : rd^j\}_{j=1..m}\}_{i=1..n}$. The *weighting factor* of the constraint will be considered as an measure of the consistency for the description of the technological problem posed by this constraint. Let's denote the weighting of the constraint in $cw, cw \in [0;1]$. The value of the weighting factor $cw = 1$ means the absolute completeness of the description by constraint of the technological problem, and the value of the weighting factor $cw = 0$ means the complete absence of such an description in the constraint. The weighting factor of the assignment will be considered as an characteristic of the importance of the assignment of an certain value to the variable. The most convenient way to represent the weight values of assignments is to use them as variable labels itself. The set of constraints with the introduced weights will look like:

$$\{ConstrSet_i = \{c_i^j : cw^j\}_{j=1..m}\}_{i=1..n} . \tag{1}$$

The *probability coefficient* of constraint will denote the measure of the probability of satisfaction or violation of the constraint by technological problem solving routine. Let's denote the probabilistic coefficient for constraint as $cpr, cpr \in [0;1]$. The value of the probability coefficient $cpr = 1$ denotes the absolute certainty of satisfaction (violation) of the constraint by solving of technological problem, and the value of the probability coefficient $cpr = 0$ denotes the absolute impossibility of satisfaction (violation) of the constraint by solving the technological problem. The set of constraints with the probability coefficients will look like:

$$\{ConstrSet_i = \{c_i^j : cpr^j\}_{j=1..m}\}_{i=1..n} . \tag{2}$$

If we do denote as $cpr^{sat.}$ – the probability of satisfying of the constraint, then $cpr^{viol.}$ will define the probability that the constraint will be violated, it is obvious that $cpr^{sat.} = 1 - cpr^{viol.}$ will have place.

The *possibility coefficient* of constraint will be considered as an possibility measure of satisfaction or violation of the constraint in solving technological problem. Let's denote the possibility coefficient for constraint as $cps, cps \in [0;1]$. The value of the possibility coefficient $cps = 1$ means the absolute degree of realization of satisfaction (violation) of the constraint in solving a technological problem, and the value of the possibility coefficient $cps = 0$ means the absolute impossibility of satisfying (violating) of the constraint in solving the problem. The set of constraints with the possibility coefficients will look like:

$$\{ConstrSet_i = \{c_i^j : cps^j\}_{j=1..m}\}_{i=1..n} . \tag{3}$$

Constraints with evaluation – characterized by a description based on the evaluation value $evc \in [0;1]$, which is a subjective assessment of the significance of the constraint indicated by decision making routine. Constraints $ConstrSet_i$ with the estimated values will look like:

$$\{ConstrSet_i = \{c_i^j : evc^j\}_{j=1..m}\}_{i=1..n} . \tag{4}$$

Constraints with preferences – characterized by a description based on the coefficient of preference pfc, $pfc \in [0;1]$, which is a subjective assessment of the importance (significance) of the constraint indicated by subject domain expert. Constraints $ConstrSet_i$ with the introduced preferences will look like:

$$\{ ConstrSet_i = \{ c_i^j : pfc^j \}_{j=1..m} \}_{i=1..n} . \tag{5}$$

Fuzzy constraints are characterized by a description based on linguistic meanings, i.e. values such as "most likely", "in most cases", "almost never", "almost always", "always", "very often", "often", "average", "rarely", “very rarely”, “never”, “unknown ”, etc. The set of constraints with the linguistic values will look like

$$\{ ConstrSet_i = \{ c_i^j : lv^j \}_{j=1..m} \}_{i=1..n} . \tag{6}$$

Thus, at each level, evaluations can be performed based on the assumption that the set of constraints with the lowest index will be mandatory and all of its constraints will be satisfied, which will ultimately allow building a solution at the level of the overall structure of the technological problem.

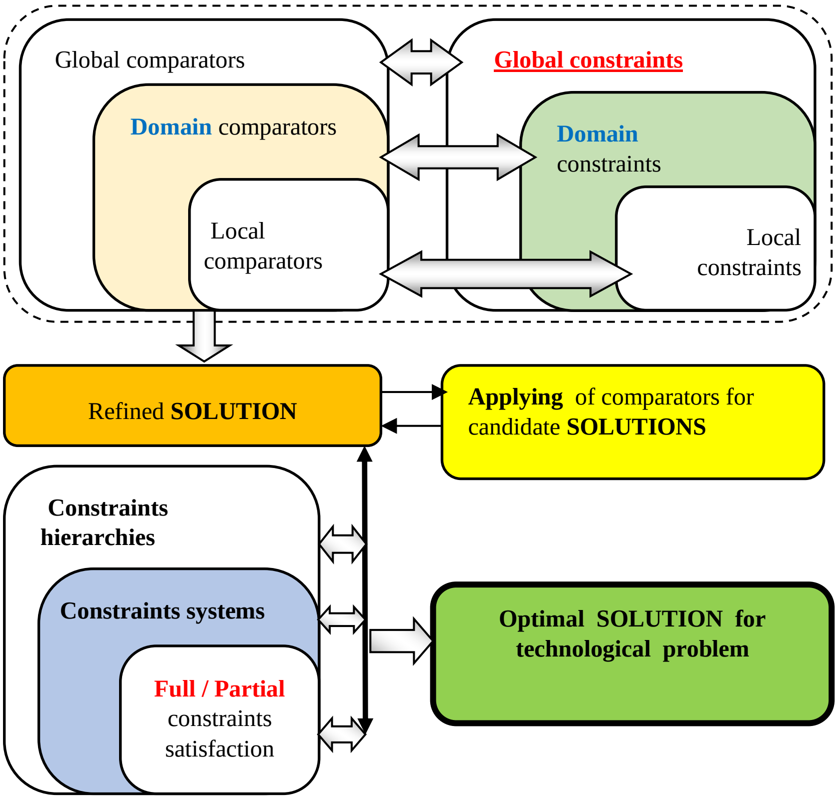

Figure 5: Structuring of *optimal solution refinement routine* for technological problem

The objective function OF (objective function) is considered to be some function given on the ordered set W^{set} over a set of variables V. There is assumed that on the set W^{set} some ordering has been introduced $\leq W^{set}$. So far, elements of the set W^{set} can be considered as coefficients of preferences for over imposed constraints set. Thus, the process of finding of the optimal solution to a technological problem **TP** can be considered as a process of satisfaction (violation) of the superimposed set in form of a system or hierarchy of constraints $ConstrSet \quad ConstrSyst, ConstrHrch$ with the introduced *objective function* OF. In the process of finding of the optimal solution **Sol**$^{opt.}$ for technological problem **TP** assignment ψ we will consider more acceptable (preferential) in relation to assignment ψ^1 when the value of the objective function for it is more than for the assignment ψ^1, that is $OF(\psi) \geq OF(\psi^1)$. We will consider such an assignment as the optimal solution for the technological problem on the basis of constraints, which is the most acceptable (preferential) one from all possible.

To be able to evaluate assignments ψ at constraint system levels $ConstrSyst_i$ and constraint sets level $ConstrSet_i$ in particular, it is obviously necessary to move to the level of multisets. For each assignment ψ and given levels of the hierarchy $ConstrSet_i = \{ c_1,\ldots,c_{n_k} \}$ for the hierarchy of constraints $ConstrHrch$ value $OF(ConstrSet_i \psi)$ will correspond to the multiset

$$MultiSet^{lex} = \{ cw(c_1)OF(c_1\psi),\ldots,cw(c_{n_k})OF(c_{n_k}\psi) \}, \quad (7)$$

that is $OF(ConstrSet_i \psi) \in N$, which take place for every $OF(c\psi), cw(c) \in N$.

This will mean that a very possible way to *compare technological problems* will be the method "*lexicographically better*", which will add elements to the multiset $OF(ConstrSet_i \psi)$ either generate a weighting of the violated constraint or assign “1” for each satisfied constraint.

Thus, the use of the *constraint success function* as well as the comparators itself to implement the constraint weights at each of the levels $i = 1..k_{max}$ of the hierarchy would be an expectedly effective tool.

Assignment ψ^1 can be considered as "*better ordered*" than another assignment ψ^2 in relation to the hierarchy of constraints, if for each of the constraints of the levels $1..k-1$, success after application ψ^1 is equal to the success after application ψ^2:

$$\left(\forall c \in \bigcup_{i=1}^{k-1} ConstrSyst_i \right) \vdash \; OF(c\psi^1) = OF(c\psi^2) \;, \quad (8)$$

and at the level k the success of the constraints can be compared using their weights $cw(c)$:

$$\forall c \in ConstrSyst_k \models \; OF([c : cw]\psi^1) =_{cw(c)} OF([c : cw]\psi^2) \;. \quad (9)$$

Let’s consider the hierarchy of constraints for some technological problem with weights:

$$ConstrHrch^{weight} = \bigcup_{i=1}^{k_{max}} ConstrSyst_i = \bigcup_{i=1}^{k_{max}} \{ c_{i_1} : cw(c_{i_1}),\ldots,c_{i_{n_i}} : cw(c_{i_{n_i}}) \}_{n \in N} \;. \quad (10)$$

The way of ordering for the formal structure $(ConstrHrch^{weight}, W^{set}, \leq_{W^{set}})$ allows to outline the relationship between comparators of the type "*better ordered*" and "*best locally*". Let’s $ConstrHrch^{weight} = \bigcup_{i=1}^{k_{max}} ConstrSyst_i$ be an hierarchy of constraints with weights and $cw : ConstrHrch^{weight} \to W^{set}$ - weight function. Refinement of the hierarchy $ConstrHrch^{weight} / cw$ let’s present in form

$$\bigcup_{i=1}^{k_{max}} ConstrSyst_i / cw, ConstrSyst_i / cw = \bigcup_{l=1}^{n_i} ConstrSet_{i_l} \;. \quad (11)$$

If the statement $ConstrSyst_l / cw = ConstrSyst_l$ takes place, then values $ConstrSet_{il}$ would be set for $\forall i \in 1\ldots n$, $\forall l \in 1\ldots n_i$ according to the formula:

$$\begin{aligned} &(\forall c \in ConstrSyst_i, \forall c^l \in ConstrSyst_i : \\ &(c \in ConstrSet_{il}, c^l \in ConstrSet_{il_1}, l_1 \in 1\ldots n_i, . \\ &l < l_1) \Leftrightarrow (cw(c^1) <_W cw(c))) \end{aligned} \quad (12)$$

Because the level $ConstrSyst_l$ is mandatory and the weights have the same interpretation, it can be assumed that:

$$\forall c_1, c_2 \in ConstrSyst_l \models \; cw(c_1) = cw(c_2) \;. \quad (13)$$

Thus, we get the equality

$$ConstrSyst_l / cw = ConstrSyst_l = ConstrSet_{il} \;. \quad (14)$$

Also, since the level $ConstrSyst$ is required, then

$$\mathbf{Sol}(\mathbf{CH}^{weight}) \models c, \forall c \in \mathbf{CS} \;. \quad (15)$$

On the other hand, the refinement of the hierarchy can be seen as some new hierarchy in which the level $ConstrSet_{i_2 l_2}$ is more important than the level $ConstrSet_{i_1 l_1}$. Let's the hierarchy be given as $ConstrHrch^{weight}$, weight function cw and two assignment ψ^1 i ψ^2. Then here we have that if the formal structure $(\psi^1, \psi^2, ConstrHrch^{weight})$ is "*better ordered*" then we can expect that formal structure $(\psi^1, \psi^2, ConstrHrch^{weight} / cw)$ would be "*locally better*". If the assignment ψ^1 is a "*better ordered*" solution for the hierarchy $ConstrHrch^{weight}$ with weight function cw, then ψ^1 can be considered as the "*locally best*" solution for refining the initial hierarchy $ConstrHrch^{weight} / cw$. So far every "*better ordered*" solution ψ for the hierarchy $ConstrHrch^{weight}$ would be accordingly *locally preferred*.

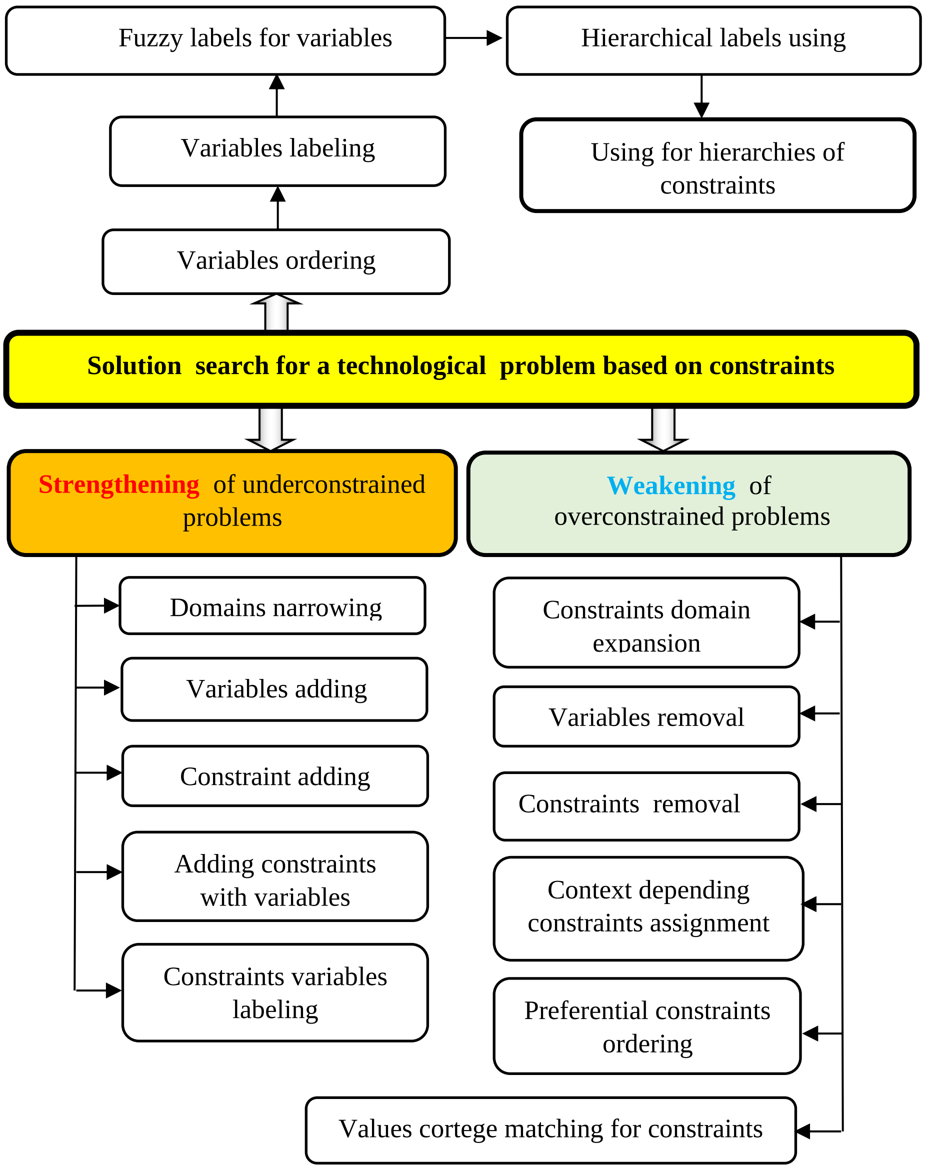

Figure 6: Solution control for technological problem

Accordingly the whole set of constraints imposed on the technological problem can be divided into two subsets: a subset of absolute constraints and a subset of relative (*preferential*) constraints. On the set of evaluations for assignments $OF(\Psi_V)$ in case if some assignment ψ does violates one of the absolute constraints, it will accordingly be excluded from the process of finding the *optimal solution*. At the same time, the violation of relative preferential constraints does not exclude the current assignment, which violates them, but on the contrary allows to evaluate the solution (assignment) in terms of its acceptability and accordingly to compare assignments accordingly to their acceptability.

Finally two technological problems $TP=(V,D,C)$ i $TP'=(V',D',C')$ can be considered as equivalent when they have the same *set of variables* and the same *set of solutions*:

$$TP(V,D,C)\sim TP(V',D',C')\models\left[V\equiv V'\right]\cap\left[Sol^{set}(V,D,C)\Leftrightarrow Sol'^{set}(V',D',C')\right]. \qquad (16)$$

Technological problem $TP'=(V',D',C')$ can be considered as *narrowed* in relation to the initial technological problem $TP=(V,D,C)$ if problems TP and TP' are equivalent; the domain of each variable D_i' is a subset of the corresponding domain D_i, $D_i'\subset D_i$; set of constraints C' does *more strictly* constrain the set of all possible variables assignments as the initial set C.

Since every of imposed constraint can to be understood finally as some subset of all possible assignments, the narrowing of the *constraint satisfaction problem* can be understood as the removal from the constraint of some assignments that do not participate in any of the relevant solution tuples.

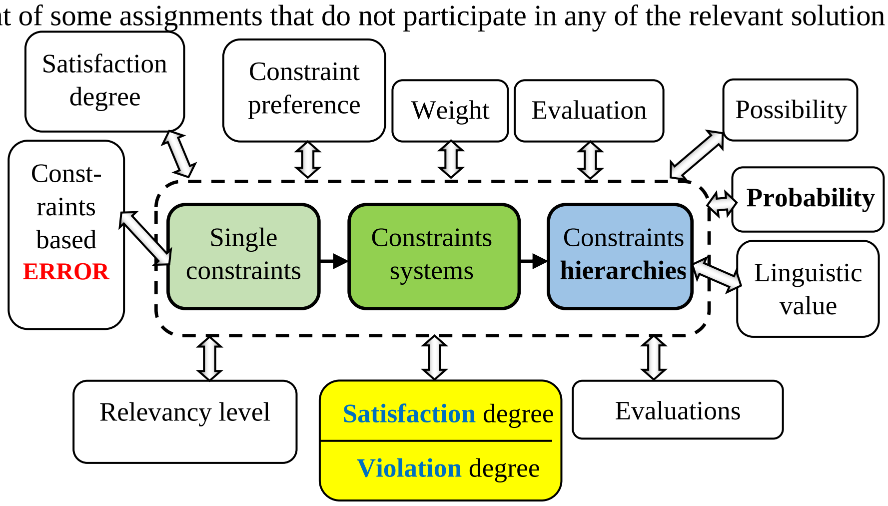

Figure 7: The structure of levels and attributes of constraints

Excessive assignments in the constraint will be marked such a constraint as one that is not a projection of any of the possible solution tuples. That is

$$\textbf{if } (s_1,...s_{n_i})\in Sol_i, i\in N \textbf{ then } \nexists\psi\in\Psi, p\downarrow_{\psi}^{Sol_i}\vdash p\in\Psi. \qquad (17)$$

An redundant domain value is a value that is not part of any of the solution tuples*:*

$$redundant(d_{i_1}\in D_j)\models d_i\notin\forall Sol_i, i,i_1,j\in N. \qquad (18)$$

Assignments and values that are interpreted as "*redundant*" can be removed from the problem.

3. Discussion. Assumptions and limitations of the research.

If the domain of any variable or of any constraint can be reduced to an empty set of constraints, it can be concluded that the problem has no solutions in the general case. Accordingly, narrowing of the problem will reduce the number of potential solutions and, consequently, the problem will be simpler in terms of finding possible solutions. Such methods are based on assigning values to certain variables with subsequent verification of assignments for compatibility with the constraints imposed on the technological problem. The choice of such values should not be random, but should be compared with the assignments that were made before. With the primary substitution of more preferential variables, the task of assigning them more acceptable values becomes simpler. Variable preferences (labels) allow to express the user's preferences along with his expectations about the complexity of assigning relevant values. The application of this technique eliminates the difficulties that arise when solving superimposed problems or problems with a large solution space. The most complex variables are the source for the constraint propagation procedure, and their initial initialization can substantially narrow the solution space. For a given constraint system with input labels for variables, the ordering of variables is calculated based on global variable labels. This way of ordering variables belongs to the class of static ordering. The final ordering calculation depends on the choice of annotation triplets. When choosing a substitute for an annotation triplet, it is necessary to consider all the properties of a given problem. If it is desirable to clearly distinguish between single variables, then the best solution is to consider assignments at each level. If the assignment does violates certain constraints, the next value for this variable is selected, if it exists. If there is no relevant value for a variable that does not violate any of the constraints, a step back is performed and the variable to which the value was assigned before the current variable would be reassigned as well. This process continues until a solution is found, or until all combinations of variable assignments are proved to be erroneous. In this case, it can be concluded that the technological problem is inconsistent due to imposed constraints.

4. Conclusions

Formal methods of choosing a solution from the set of all assignments that will reflect the possible semantics of annotation of variables and assignment of variables to labels are presented. Labels will determine how variables are arranged within process problems based on constraints with preferences as in the case of optimization problems. The task of displaying labels on a set of technological problems with imposed constraints and displaying in the case of a hierarchy of constraints, allows to specify the process of finding solutions to superimposed problems, based on the specifications of classical formal structures. An assessment of possible domain comparators for technological problems arising in the drilling process allows to form a structure for building a solution for the introduced hierarchy of constraints. A partial assignment is extended by including of new variables until a solution is found, or until all partial assignments would be checked. The main idea of the approach is to collect the sets of all partial assignments that do not violate the constraint for growing set of variables. *The future research should* ensure the correctness of the approach, when all partial assignments that violate certain constraints at a certain step of the routine would be removed, and to ensure the completeness of the approach, when the set of all partial assignments that do not violate any of the constraints must be controlled.

5. References

[1] M.Chesanovskyy, K.Kravtsiv, V.Protsiuk, L.Poteriailo, Software outlines for decisions making support in oil and gas engineering, Scientific Papers of Silesian University of Technology: Organization and Management Series , Issue No. 151., pp.81-99.

[2] V.Sheketa, M.Chesanovskyy, M.Pasyeka, V.Pikh, Y.Romanyshyn, V.Bandura, "Formal aspects of case-based data modelling for intelligent drilling control", in Proc. of the 2019 10th IEEE Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), Volume 2, September 18-21, 2019,Metz, France, pp. 907– 911.

[3] A. Köksal, V. Kuncak, P. Suter, Constraints as control , ACM SIGPLAN Notices, 2012, dl.acm.org

[4] L.D.Raedt, A.Passerini, S.Teso, Learning constraints from examples. In: AAAI(2018)

[5] Y.A. Liu, S.D. Stoller, Knowledge of uncertain worlds: programming with logical constraints, Journal of Logic and Computation, 2021 , academic.oup.com

[6] C. Bessiere, Constraint propagation, Foundations of Artificial Intelligence, 2006, Elsevier

[7] C. Schulte, P.J. Stuckey, Efficient constraint propagation engines, ACM Transactions on Programming Languages , 2008, dl.acm.org

[8] F. Bacchus, X. Chen, P. Van Beek, T.Walsh, Binary vs. non-binary constraints. AI,2002 , Elsevier.

[9] J.C. Régin, Global constraints: A survey. Hybrid optimization, Springer, 2011.

[10] W.J. van Hoeve, I. Katriel, Global constraints : Foundations of Artificial Intelligence, 2006, Elsevier.

[11] Z. Ruttkay, "Fuzzy constraint satisfaction" , Proceedings of 1994 IEEE 3rd International Fuzzy , 1994 - ieeexplore.ieee.org

[12] L.A. Zadeh, Fuzzy logic = computing with words - Computing with Words in Information/Intelligent, 1999 - Springer

[13] R.Dechter, Constraint Processing, Morgan Kaufmann Publishers, 2003.