Standard Analytic Activity Scenarios Optimization based on Subject Area Analysis

© Oleksandr Koval[0000-0003-0991-6405], © Valeriy Kuzminykh[0000-0002-8258-0816], © Maksym Voronko

National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine,

avkovalgm@gmail.com, vakuz0202@gmail.com, maximvoronko@gmail.com

Abstract. The optimal scenario building task solution based on typical scenarios along with the subject area ontology analysis, using the structure of ordered by action types directed graph is proposed. This approach gives the possibility to evaluate optimal scenario properties using a suitable metric. The evaluation of possible cost reduction for building more effective standard scenarios of analytical activity is suggested. The relevance of such problems solution in the sample domain of budget process analysis is shown. The algorithm of optimal scenarios search sequential refinement based on subject area ontology analysis along with ordered directed graph analysis is proposed.

Keywords: information gathering, ontology, graph analysis, budget analysis, typical scenario.

Preface

The information technologies development is characterized by several trends that require scientific comprehension and elaboration, development of new architectural solutions, including approaches to software systems design and implementation, aimed at analytical activities support. This can be explained by the current analytics systems functioning paradigm, which objectively requires increasing the intelligence of decision-making processes as well as software systems and technological components of analytical support. Considering the dynamic nature of the specific environmental requirements and the complexity of system integration tasks dictate the need for methods and tools development supporting the design of distributed information systems, based on distributed analytical activities scenarios and accounts for a large number of participants [1]. Thus, the analysis reveals such phenomena and development trends.

Today there are several methods to select best by the quality and productivity scenarios from the collection of possible or admissible scenarios, based on the factors composition and nature analysis, which affect the scenario planning process. In plan-

ning practice one can distinguish situations, differing in factors number, which is used to make decisions and define principal differences in scenarios development procedures. One of the quite difficult and urgent tasks today that require the development of effective scenarios to solve is the task of constructing and further optimizing the scenarios for collecting and analyzing various processes of organizations. Today the scenarios optimizing problem in network information collecting and analysis is one of the major tasks in the domain of big data processing [2].

This is especially true for solving the budget process problems based on the financial analysis of regional budgets. During the economic crisis, the problem of increasing the role of the analysis of the regional budgets in solving economic and social problems becomes insistent. The problems of sustainability of regional budgets gain special actuality.

The main aim of the budget financial sustainability analysis is to obtain a sustainability assessment for each subject of the analysis for a certain period and to conclude on the budget stability state for that entity. Based on these results one can make common conclusions about the state budget financial condition, make forecasts for the condition for future periods, and make decisions about steps, directed to improve the economic situation, gain state budget stability and independence.

Software development for state budget financial analysis and defining budget sustainability lets substantially reduce time, needed for analysis and facilitates the work of state institutions and authorities [3].

The overall structure of the network, which determines the ontological features of the problem of budget process analysis on a branched network, is quite complex in structure and has a significant number of the elements.

The most usual way to solve such problems is to build a scenario, based on definite ontology, which is described with the appropriate graph. Budget monitoring, as well as many other tasks in various fields, often consists of performing a series of typical actions of varying degrees of complexity. The least step, which does not need further breakdown and is usually executed by a single person, or, in case of automated execution, by a single piece of software, can be called elementary step. Unlike a project or workflow, in the case of a scenario, such a step, as well as compound ones, may have several outgoing results that can be assigned appropriate probabilities. Each step is characterized by a set of parameters that are used to optimize for a particular criterion.

It occurs plausible to use ontology to save steps hierarchy for definite activity, here for budget monitoring, in the ontology the list of connections between steps of the type “before-after”. Unfortunately, ontology is poorly suited for saving connections metrics, e.g. connection probabilities. Because of this, we need additional means to save additional data for steps pairs, connected with the relation “predecessor - successor”.

Analyst’s job begins with ontology, describing all possible steps, their hierarchy, appropriate metrics and connections of the type “before-after”, creation or development. After this, in separate software, the probabilities of transition between steps are set or updated. Then this or some other analyst use steps or step blocks to build the action graph, which as outcome gives a result with definite probability or set of results with a probability distribution.

The criteria to be used for optimizing, depends upon the task. Examples of the criteria may be threshold probability for achieving a positive result, scenario execution time or cost-minimizing, or even some metric combination, which is saved in ontology and additional storage of results probabilities.

In a more sophisticated model, some external to network factors are defined, which can alter the probability distribution among arcs or step metrics. For example, one such factor can be the course of national currency or price of energy carriers, etc. But here we will not consider this case. They can be easily implemented in future if such need occurs.

After the criteria of optimizing is selected, one can use well-defined algorithms of the shortest route search on the graph, and in complex graph to define reachability of the final aim as well as other common graph algorithms.

If needed on network obtained it is possible to build optimistic, pessimistic and optimal steps series. Such an approach makes it possible to supply analyst a set of alternative steps from the current node of the possible actions graph.

In the sample case of budget monitoring the simplified way of building the whole process may be next:

1. Building an ontology of possible actions;
2. Automatic building or updating of edges probabilities storage, based on the ontology instances as nodes;
3. Expert manually defines or updates the probability distribution of results for each ontology object, considering the results of previous analysis;
4. Possible actions in budget monitoring graph building;
5. Shortest way search criteria selection;
6. Reachability analysis of the end node from the starting one. If the node is unreachable then building of extended graph or connections correction will be needed to achieve reachability;
7. Shortest route selection using selected criteria and selection of the set of several routes with minimal lengths;
8. Probability of the positive result definition for each of the selected routes;
9. Combined graph building as support system of the analyst activity;
10. After the current state analysis it is possible to update edges probabilities according to practical results. If some new practical steps appear – the extension of ontology is made to include new steps.

If necessary the update and recalculation of the graph are made to include additional new knowledge, obtained from the practice.

There is a great number of ontology-based scenario building models in the subject area, which use different mathematical methods. Among other the scenario building models based on structured data storage in branching networks are usual enough; such is especially true for the problems of regional budget monitoring. Based on these models and their different modifications modern information-analytical and information-searching systems are built [3].

In this case, the ontology is defined as:

$$O = \{T, A, R, D\}, \tag{1}$$

where T is a set of terms, defining objects and concepts of the subject area,

A – set of concepts attributes,

R – set of relations (connections) between the terms,

D – set, holding definitions of concepts and relations.

In the graphical form, the ontology is represented as a network, the vertices of which are denoted by the concepts of the domain, and the edges denote the connections between them. Basic are hierarchical class-subclass and part-whole relationships that define the structure of the branching structure of the information storage network.

Thus, ontology represents the description of the subject area, giving the view of the concepts set with connections between them.

Descriptive and mapping techniques based on graph theory are widely used to describe ontologies for the tasks of selecting the scenarios optimal by quality or performance criteria from the set of possible or feasible scenarios. In this case, the most widespread descriptions are in the form of hierarchical graphs, usually with weight estimation and edges count.

Building scenario based on the graph analysis

Ontology consideration using a structural approach is most relevant to the task, for graph presentation of the structure allows measuring its properties with a selected metric, determining its quality and suggesting recommendations for its future improvement.

Such a structured approach makes it possible to evaluate the ontology model building effectiveness due to the graph, describing ontology search cost reduction against the usual manual method of building a scenario for the same ontology.

To estimate ontology describing graph search cost reduction one can estimate productivity using assumptions, based on some formal assessments for different types of graphs, for which it is quite simple and accurate to estimate the values selected for comparison characteristics. These estimates can be based on the difference between full and partial flow over the graph, which is a graphical representation of the analyzed ontology describing the scenario.

As such procedures basis, we can select previous information about graph flow during other queries, which partially coincide with the current that is using information obtained during previously made graph analysis. This information is stored in the corresponding knowledge base for each distinct ontology, and so for the corresponding graph. In this case, to make approach productivity estimation as a whole, without reducing the degree of generality, it is possible to consider some specific graph types used for ontology description.

As it was shown in Miller's article, ontology estimations suggest, that the number of connections for a definite concept in the fully connected graph, describing ontology, must not overcome 9 [4, 5]. Thus, we can assume that in most of the real cases the number of all ingoing and outgoing edges of the directed graph will not exceed 9.

Using this assumption, the maximal effect from the usage of previously obtained information for the way of possible scenario building, in this case, can achieve

$$E_{max} = 9n / (9n - 9(n-m)) \quad (2)$$

To estimate the productivity of the ontology model building, methods based on the shortest route search in the graph, representing the ontology, can be used.

Generally, we can write, that selecting routs to achieve target node T in the graph, using the query:

$$T = \{x \mid A(x)\}, \quad (3)$$

where x are all possible routes in the graph,

$A(x)$ – characteristic property, representing the essence of the specific query, will give the needed result.

Then

$$\forall a \exists x \forall c (c \in x \Leftrightarrow c \leq a) , \quad (4)$$

where

c – all routes in the graph, leading to the target node,

a – minimal length route to the target in the graph.

Thus there always exist minimal length route in the graph, which corresponds to target. So the result of the query also exists

$$T_{min} = min(a). \quad (5)$$

The search task of the single source shortest path (SSSP) [6] is defined based on the general ontology model graph description.

The process flow supporting the analyst work in sample domain of budget monitoring can be represented as follows:

1. Building operational OWL model in the subject area, using Protégé software. (Any other software compliant with OWL2 standard can be used as well).
2. Converting OWL model to the GraphML format using specialized converter.
3. Edit obtained graph, for the purpose of scenario creation.
4. Typical scenario corresponding to selected criteria creation.
5. The most effective current scenario search using selected criteria.

Resulting from the expert in the subject area work the corresponding activities model is created and further developed to be used in the analytic activity, fixed and stored for definite criteria and tasks as the typical scenario. The specialized software converts the ontology OWL file into the tree of the possible analyst's operations saved in the form of the GraphML file. This file represents the supporting operations list for the future analyst work, used for selection of operations subsystem, needed to build a scenario and set connections between successive steps with selected metrics of

the nodes and edges of the graph [7]. Such metrics as execution time and cost are defined for the nodes, while the probability of transition is the metric of edges. Each of the nodes can be simple or compound.

The compound node can hold a graph of possible simple or compound steps, connected with transition edges. The input logical function is stored in any node, defining the node with several incoming edges activation method. In all output edges of the node, the normalized transition probabilities to the next nodes are stored thus forming the weighted directed graph.

While practical executing the scenario the updated values of metrics and probabilities can be set in the input structure. The simple nodes hold the input metrics values. Compound node metrics are calculated from the corresponding metrics of the lower-level nodes. If the graph can have the cyclic nodes, then corresponding node metrics of such node can have several values, depending on the cycle number. The shortest route search method allows defining simple weight coefficients for nodes, as well as complex, which need metrics for calculation [9]. E.g. if the scenario nodes have time and cost parameters, then the shortest route can be calculated using the criteria of the minimal general execution time, minimal cost, or some linear combination of both parameters.

While optimizing scenario built on the graph the standard methods of shortest route search, such as the Dijkstra algorithm, are using weighted edges to be run. If we need to select the shortest route for the parameters of the nodes the most reliable way is to create supporting edges weights defined as the average of the corresponding node's parameters, placed on both ends of the edge except the beginning and final node, whose parameter values are not divided by 2. It is simple to show, that total route weight, defined on such edge metrics will be the same, as calculated on the nodes parameter values.

In this work, we shall not consider a multilevel graph because any multilevel graph can be represented with equivalent single level one by substituting the contents of the compound node instead of the node itself (Fig. 1). Such a graph will be complex enough, but a single layer.

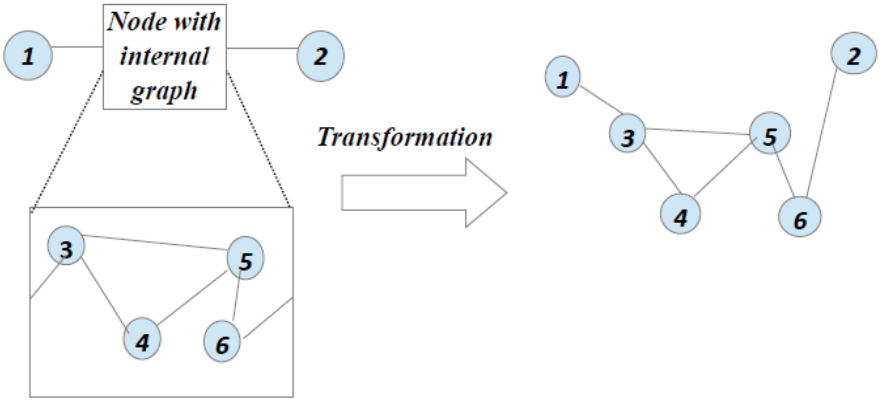

Fig. 1 Multilayer graph transformation to the single layer one.

Let us consider some sample directed weighted graph with defined start node a_s. The edges sequence from node a_s to node a_F ns called route.

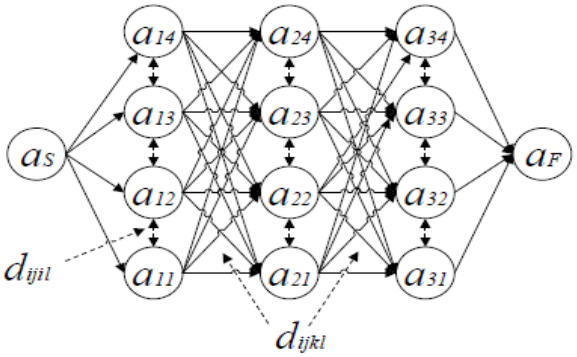

Fig. 2 The base graph structure, used to build a scenario ($n=4, m=3$).

The task is for any node v accessible from source-node a_S, to find route, having least total weight $P^*(a_S,a_F)$:

$$f^*(a_S)=f(P^*(a_S,a_F)=\min f(P(a_S, a_F)). \quad (6)$$

Such a problem can be set for both graph type, directed and undirected [9]. The problem solution satisfies the optimality principle: i.e. route in consideration is a part of the shortest route, e.g. from source a_S to finish node a_F. That means that to store the shortest path for each node, it is enough to store its last edges instead of the whole route.

Without decreasing the generality let us revue the task of enhanced route search on the graph, which defines optimized actions sequence in scenario based on the typical scenario of analytical work. Here the typical scenario is already existent and formalized scenario, which usually was received as a result of the previous activity of analysts and experts solving such problems.

Let's consider graphical representation of scenario construction on the graph (Fig. 1), which corresponds to this problem formulation with the following assumptions:

1. Every level of graph hierarchy

$$A=\{a_{ij}\} \text{ для } i=1...n \text{ та } j=1...m \quad (7)$$

corresponds to a full activity set, having analogous by quality results, but differ in activity indicators (execution time or cost).

2. Time consumed by transition between activities in one level is less then transition time from any level to the lower one.

If d_{ijkl} – is an edge between nodes a_{ij} and a_{kl}, and d_{ijil} – is an edge between nodes a_{ij} and a_{il}, then

$$d_{ijil} << d_{ijkl} \quad (8)$$

for $i,k = 1, ..., n$
$j,l = 1, ..., m$
and $i \neq k$.

3. For each graph level, representing subject area ontology there is typical scenario, the defines edges set

$$d^{T}_{ii+1.} \text{ for } i = 1, ..., n-1$$

4. Scenario can be defined as optimized (enhanced), if total evaluation of scenario edges, defining the newly formed in optimizing process new scenario edge chain d_{iji+1l} is less then total evaluation of edges chain in typical scenario $d^{T}_{ii+1.}$

The successive scenario quality indicators enhancement algorithm, representing the starting ontology of subject area using the structure representation as ordered by activity types graph, can be constructed as the following sequence of steps.

1. For each scenario level

$$i = 1, ..., n-1 \text{ and } j,l = 1, ..., m$$

value of d_{iji+1l} is compared to d^{T}_{ii+1}.

2. If values are such that

$$d_{iji+1l} < d^{T}_{ii+1}$$

then new value for current scenario is defined

$$d^{TN}_{ii+1} = d_{iji+1l} \quad (9)$$

3. New optimized scenario activities sequence from the node a_{i+1l} sequentially through all subsequent levels to level m.

4. If total sum d^{TN}_{ii+1} for the new chain is less then for d^{T}_{ii+1} of starting typical scenario, then this new scenario is accepted as typical.

5. Where the conditions of paragraph 4 are not fulfilled, the process can be repeated from the last node of the typical scenario, from which a new direction of scenario construction was begun, in another direction up till the last graph level.

Thus, when there is a chain of action better over a chosen criterion (for example, the time of receiving and processing information) compared to a typical scenario, it will be found and the typical scenario will be replaced with a new one.

As a result of running the described algorithm, we obtain a scenario that meets the given conditions, in accordance with the ontology described by the graph. The algorithm presented here shows a sequential process of detecting a sequence of arcs connecting the nodes of the graph from the top level to the bottom, taking into account the matching parameters.

When repeatedly retrieving information by close form of query, we use an existing scenario model, which significantly shortens the construction time by reduc-

ing the number of nodes under consideration. This does not exclude the possible need to revise the ontology and rebuild it.

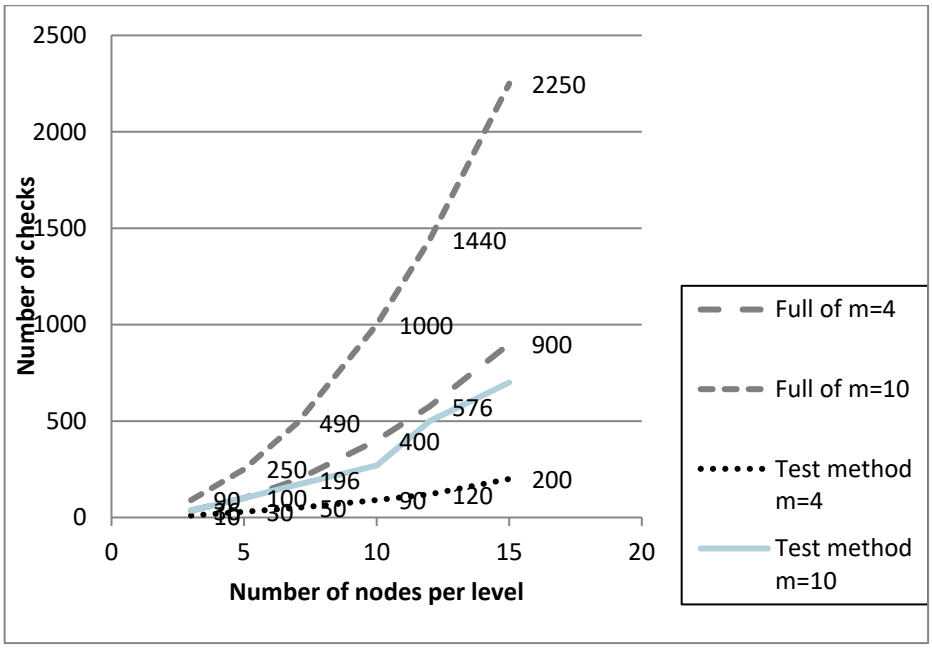

Fig. 3. The successive scenario enhancement algorithm vs full graph analysis approach.

Fig.3 shows the scenario optimizing result for test examples with averaging results for two tests. The test examples with 4 and 10 levels and nodes number 3,5,7,10 and 15 for each level where considered. The testing results show, that such approach for scenario optimizing, based on graph representation can significantly reduce the complexity and expenses of building new more effective by defined criteria scenarios. This makes it possible to significantly simplify the analytical work using typical scenarios.

Conclusions

In this article the problems of gathering flow information scenario optimization on the branched network are considered as an example of the construction and further optimization problem of scenarios built on the branched network for the budget monitoring. The structure approach in ontology analysis, as the most appropriate for the considered task, using the graph representation of the structure is suggested. The estimation of the effect of the previously obtained partial information about gathering information in the network scenario construction, described by the graph, for the further clarification of information. The formal description of multilayer hierarchical system

structure is provided. The example of ontology elements interaction structure for the problem is presented.

The complex approach based on the shortest route search on the graph and ontology model graph representation is suggested. This makes possible to use algorithms, based on graph nodes in hierarchical levels (layers) traversing.

The approach for modified search in width algorithm building is presented, which significantly decreases routes search for the gathering flow information scenario construction time in the branched network. Described in the article approach for the gathering flow information scenario optimization on the branched network was tested for the pilot system of regional budgets financial analysis project development. Algorithm suggested here makes it possible to develop the software complex, which enables sufficiently full and complete solution to the problems of scenario optimization for scenarios of search and collection of streaming information on an extensive network. One of the perspective directions in the algorithm application is to use its possibilities for information-analytic systems building. This will significantly reduce the time and improve the quality of searching the necessary streaming information on a extensive network.

References

1. Alex Guazzelli, Michael Zeller, Wen-Ching Lin and Graham Williams PMML: An Open Standard for Sharing Models: available at: https://journal.r-project.org/archive/2009-1/RJournal_2009-1_Guazzelli+et+al.pdf
2. Chernov, V.A. The Economic Theory analysis: Textbook // V.A. Chernov .- Moscow: Prospect, 2017 .- 384 p. - ISBN 978-5-392-24867-4
3. Christopher J. Manning, Prabhakar Rahavan, Heinrich Schütz. Introduction Consumer Information Search (trans. With Eng.) - M .: OOO 'Y.D. Williams' 2011 – p.504.
4. O. Koval, V. Kuzminykh, S. Otrokh and V. Kravchenko, "Optimization of Scenarios for Collecting Information Streaming Wide-Area Network," 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), Lviv, Ukraine, 2019, pp. 213-215. doi: 10.1109/AIACT.2019.8847832.
5. Miller G. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review, 1956. 63: pp.81-97.
6. Thorup, Mikkel. "Undirected Single-Source Shortest Paths with Positive Integer Weights in Linear Time." Journal of the ACM 46, no. 3 (May 1, 1999): pp.362–394
7. Moore, Edward F. "The Shortest Path Through a Maze". International Symposium on the Theory of Switching, pp.285–292, 1959.
8. Lee, C Y. "An Algorithm for Path Connections and Its Applications." IEEE Transactions on Electronic Computers 10, no. 3 (September 1961): pp. 346–365.