Multicriteria Analysis of Ontologically Represented Information

Katarzyna Wasielewska[*], Maria Ganzha[†], Marcin Paprzycki[**], Costin Bădică[‡], Mirjana Ivanovic[§] and Ivan Lirkov[¶]

[*]*System Research Institute Polish Academy of Sciences, Warsaw, Poland*
[†]*System Research Institute Polish Academy of Sciences, Warsaw, Poland and University of Gdansk, Gdansk, Poland*
[**]*System Research Institute Polish Academy of Sciences, Warsaw, Poland and Warsaw Management Academy, Warsaw, Poland*
[‡]*Department of Computer Science, University of Craiova, Craiova, Romania*
[§]*Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad, Novi Sad, Serbia, mira@dmi.uns.ac.rs*
[¶]*Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev, bl. 25A, 1113 Sofia, Bulgaria, ivan@parallel.bas.bg*

Abstract.

Our current work concerns the development of a decision support system for the software selection problem. The main idea is to utilize expert knowledge to help the user in selecting the best software / method / computational resource to solve a computational problem. Obviously, this involves multicriterial decision making and the key open question is: which method to choose. The context of the work is provided by the Agents in Grid (AiG) project, where the software selection (and thus multicriterial analysis) is to be realized when *all* information concerning the problem, the hardware and the software is ontologically represented. Initially, we have considered the Analytical Hierarchy Process (AHP), which is well suited for the hierarchical data structures (e.g. such that have been formulated in terms of ontologies). However, due to its well-known shortcomings, we have decided to extend our search for the multicriterial analysis method best suited for the problem in question. In this paper we report results of our search, which involved: (i) TOPSIS (Technique for Order Preference by Similarity to Ideal Solution), (ii) PROMETHEE, and (iii) GRIP (Generalized Regression with Intensities of Preference). We also briefly argue why other methods have not been considered as valuable candidates.

Keywords: semantic data processing, domain knowledge, user support
PACS: <Replace this text with PACS numbers; choose from this list: `http://www.aip..org/pacs/index.html`>

INTRODUCTION

Multicriterial analysis has been applied to many practical problems and has long history of fruitful research. However, with an increasing interest in semantic technologies, a new question has to be answered. Is it possible to apply existing multicriterial methods in a scenario when data used in decision making is formally represented as an ontology. Here, the method of choice – if one can be found – should take *full advantage* of the way that the data is represented. The context (and use case scenarios) for our work is provided by the Agents in Grid (AiG; [1, 2, 3]) project, which aims at the development of an intelligent agent-semantic meta-level middleware for the Grid. The AiG system is based on the following assumptions:

- Grid is treated as an open environment that can be freely joined by "resources",
- all "users" and resources are represented by software agents,
- agents work in teams managed by team manager,
- users contract Grid resources to execute computational jobs,
- users have knowledge about the problem and may have various levels of knowledge concerning available solution methods, they may also have somewhat limited knowledge about software libraries, tools and computer hardware that should be used to solve the problem efficiently,

- all information (knowledge, message content) in the system is ontologically represented and semantically processed (for detailed description of ontologies developed for the AiG project, see [4, 5, 6]).

In this paper we focus our attention on the following use case scenario. Let us assume that the user would like to submit a job to be executed in the Grid, while using specific resources; e.g. available RAM should be greater than 1024 MB because of the matrix size. The problem to be solved is: to find the smallest eigenvalue for a real symmetric matrix. Additionally, she wants to execute the job as soon as possible; regardless of the costs. The preferred payment mechanism is pay as you go, and contract should consider penalties in case of a delay in job execution. Obviously, mentioned conditions can be divided into those describing the job (problem) itself, and those describing "business terms" of job commissioning.

Note that, very often (i) users are not willing to spend time to learn how to use Grid middleware, and (ii) once they learn how to use a specific method / library / code to solve their problem(s), they are not inclined to change their habits ([7]). Specifically, they tend to solve the problem with the same method using the same resource configuration, which they mastered, despite the fact that there may be more suitable algorithm and / or computational resource available. This being the case, our aim is to help the user (potentially without "Grid-knowledge") to solve their problems efficiently. Specifically: (a) the system should utilize expert knowledge to support user in selecting the best approach to solve a given problem; and (b) it should help user to select best terms of collaboration with resource provider (see, also, [8, 9]).

In the job execution scenario, in the AiG system, the following steps can be distinguished:

1. User enters requirements regarding the job and resources to be used, i.e. according to his best knowledge he defines the problem, input data properties, configuration of needed resources.
2. **User's agent searches repository with expert knowledge and selects pertinent expert opinions, which are then used in multicriterial analysis to suggest "the best" configuration of resources to be used.**
3. In case of discrepancy between the original request and the system suggestion, user accepts or rejects modified requirements.
4. User's agent requests a list of registered teams that have a resource with the required configuration.
5. User enters requirements regarding the contract e.g. deadline, price.
6. User's agent negotiates with selected teams leaders.
7. **User's agent selects optimal contract offer (as a result of multicriterial analysis).**
8. If contract negotiations are successful, job is sent to and executed by the selected team.

It should be stressed that in steps 2 and 7 the system utilizes MCDA to analyze ontologically represented information. Note that contract negotiations (step 7) may involve multicriterial analysis on "both sides," i.e. can be used also by the team leaders, but we omit this aspect of the process in the current contribution.

Ontology in the AiG System

As mentioned above, all knowledge in the system is represented as an ontology, while communication protocols utilize messages with ontological content ([4, 6]). Typically, ontology is defined as a formal specification of shared conceptualization ([10]). It defines common terms and rules that are used to describe a given domain of knowledge. Ontology is formally specified using the following constructs:

- concepts describing domain elements,
- properties describing concepts,
- restrictions on values that properties can take,
- relations between concepts,
- axioms and assertions.

During the development of the AiG system, three ontologies were designed to provide concepts necessary to describe: (i) resources and Grid structure (AiG Grid Ontology), (ii) contract details (AiG Conditions Ontology), and (iii) content of exchanged messages (AiG Messaging Ontology) [4]. Furthermore, during development of the user decision support, additional expert domain knowledge ontology has been formulated (AiG Expert Ontology) [9].

The main goal of the expert ontology is to provide concepts necessary to capture three aspects of the computational domain: (i) problems to be solved, (ii) algorithms to solve them, (iii) objects that these algorithms operate on. Additionally, classes *DomainExpert* and *ExpertOpinion* where introduced to represent experts knowledge (recommendations) allowing matching of problems, algorithms and resource specification. Therefore, the *ExpertOpinion* class has property *hasRecommendedResource*, which points to a resource that is most suitable for solving a specific problem (according to the expert). Obviously, resources originate from the AiG ontology. Class *JobProfile* allows to define restrictions on the combination of problem, algorithm and input data properties, and (optionally) resource configuration(s). The AiG Expert Ontology formalizes the domain knowledge description. Additionally, the domain of computational linear algebra was modeled as a proof-of-concept ontology. Detailed descriptions of developed ontologies can be found in [4, 5, 6].

The structure of the paper is the following. In the next section we describe where (and how) in AiG multicriterial analysis is used, and what characteristics should the required method have. Then, we provide brief characteristics of methods that seem to be most appropriate in our use case, and therefore, were selected for further evaluation. Afterwards, we present evaluation of selected methods with respect to several criteria. In conclusion, we look at the methods in broader context by presenting pros and cons of all analyzed methods and, additionally, methods that were excluded at earlier stages of the evaluation.

NEED FOR USER SUPPORT

One of key challenges of designing user support within the AiG system is the selection of a suitable MCDA technique. The MCDA is performed by the user's agent in two steps of the job execution scenario: (i) selecting pertinent ontologically demarcated expert opinions to support user in precise and correct problem formulation, (ii) selecting best contract offer describing terms of collaboration with resource provider. Both, expert opinions and contract offers, are represented as ontology instances, described with a set of possible properties that have hierarchical structure e.g. for the resource configuration there are properties describing CPU, memory, operating system; while for the contract there are properties describing pricing, payment mechanisms, job execution timeline, etc. Therefore, multicriterial decision analysis is required to analyze alternatives with respect to multiple criteria – numerical and textual.

MCDA to support user in precise and correct problem formulation

In this case the MCDA is used to select expert opinions combining possible methods to solve computational problem indicated by the user, and recommended resource configuration that should be used to execute the job. Specifically, in this scenario user has a problem (and a corresponding computational task / job) that she wants to execute on a resource available on the Grid. Note that, she may have a limited knowledge about available (and recommended methods) and how problem's input data properties determine method selection. Therefore, the system should verify user's problem / job formulation, and make appropriate recommendations based on expert opinions stored in expert knowledge repository. The following steps describe in more detail the problem formulation and user support:

1. Experts input into the system their opinions (recommendations) that define relations between problems, input data properties, methods for solving problems and required resource configuration. Available expert opinions (existing as "independent entities") constitute expert domain knowledge repository (expert ontology). Trustworthiness of opinions is expressed with weights that can be assigned to experts. It can be assumed that the number of expert opinions available in the system will increase over time. Additionally, experts should have the possibility to assign weights to properties that are used to define opinions e.g. for a given problem, when assessing CPU, number of cores is much more important that clock speed, but at the same time all CPU properties are less important than memory properties.

2. Grid user, who wants to execute a specific computational task, in the first step, defines a job profile (i.e. a problem, and if possible input data properties and a method with which the problem should be solved), and, if the user has appropriate level of knowledge, configuration of wanted resource(s).

3. Based on the job profile that the user specified, user's agent selects from the repository (containing expert knowledge) opinions that match the profile.
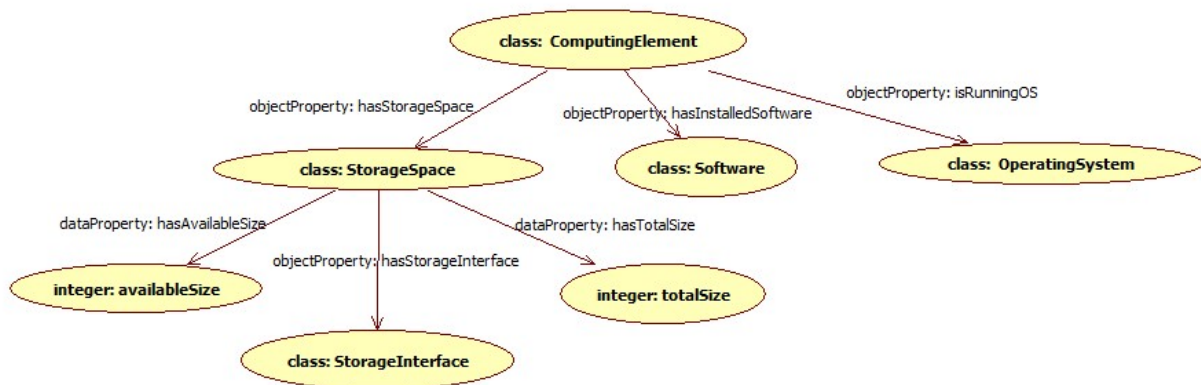
FIGURE 1. Sample resource description structure

4. If there are no expert opinions matching the job profile, user will not receive any recommendations and system will use requirements that were initially specified by the user.

5. If there are expert opinions matching job profile, multicriterial analysis is conducted. The system considers selected opinions to choose "representative" opinion for a given problem. Assessment is performed without experts interaction (automatically), however the system represents expert opinions "from a given expert perspective." The selected opinion is confronted with the original job profile.

6. As a result, two situations are possible:
   - user underspecified what he wants to do, and the system can help him to make requirements more specific; e.g. user provided problem, input data properties, but he does not know which method (library) is to be used to solve this problem,
   - user specified requirements that are not optimal (from the point of view of available expert opinions); in this situation the system should propose another approach to solve the problem e.g suggest other algorithm or resource configuration.

Ontological representation of expert opinions

As mentioned before, expert opinions are ontologically demarcated, which allows to represent them as trees (edges correspond to properties, while nodes are instances and data objects). As a result, the MCDA methods analyze alternatives that can be represented as a hierarchical structure.

Figure 1 (simplified) was generated from an ontology with resource description (nodes correspond to given data types for properties – criteria – instances or simple data types). Sample resource is described with properties (and assessed based on these properties): *hasInstalledSoftware*, *hasStorageSpace* (with subcriteria *hasAvailableSize*, *has-TotalSize*, *hasStorageInterface*), *isRunningOS*. The tree structure of the decision problem allows to group elementary criteria such as *hasAvailableSize*, *hasTotalSize*, *hasStorageInterface* into more general concepts that correspond to e.g. *StorageSpace*, *CPU* on a meta-level. Leaves (representing values that elementary criteria take) are evaluated when the concept represented on the tree is assessed. On the other hand, weights can be assigned also to nodes on higher levels, e.g. all properties describing *CPU* may be more important than all properties describing *StorageSpace*.

Note that, in the ontology, there are defined much more properties that can be used to describe a resource. As a result there may exist more complex tree structures. For a given resource, a set of properties used to describe it may vary; i.e. trees generated for different resources do not need to have the same structure of the description, however the structure is determined by the ontology (that can be extended).

Figures 2 and 3 show ontological description (represented on a tree) of sample expert opinions (in nodes there are sample values that criteria can take). It can be noticed that resource configuration is included in the expert opinion. In the expert knowledge repository, each recommendation of a given expert is stored as a separate opinion.
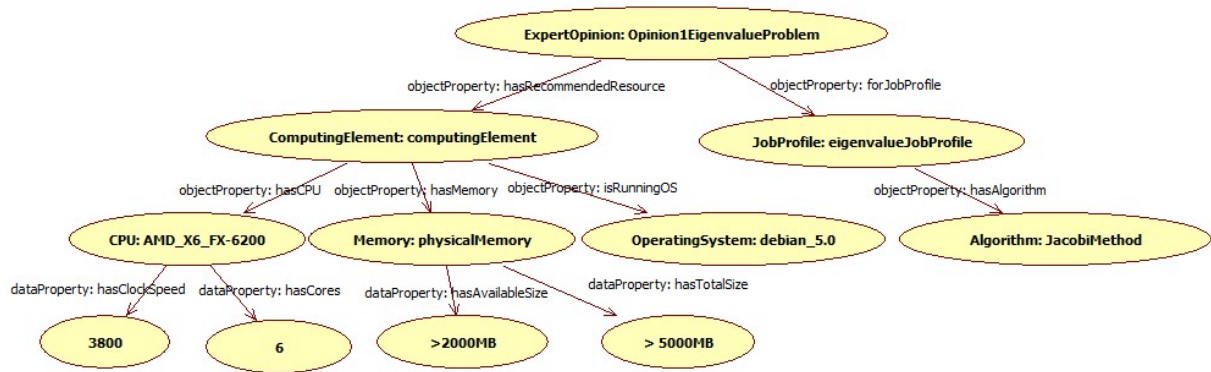
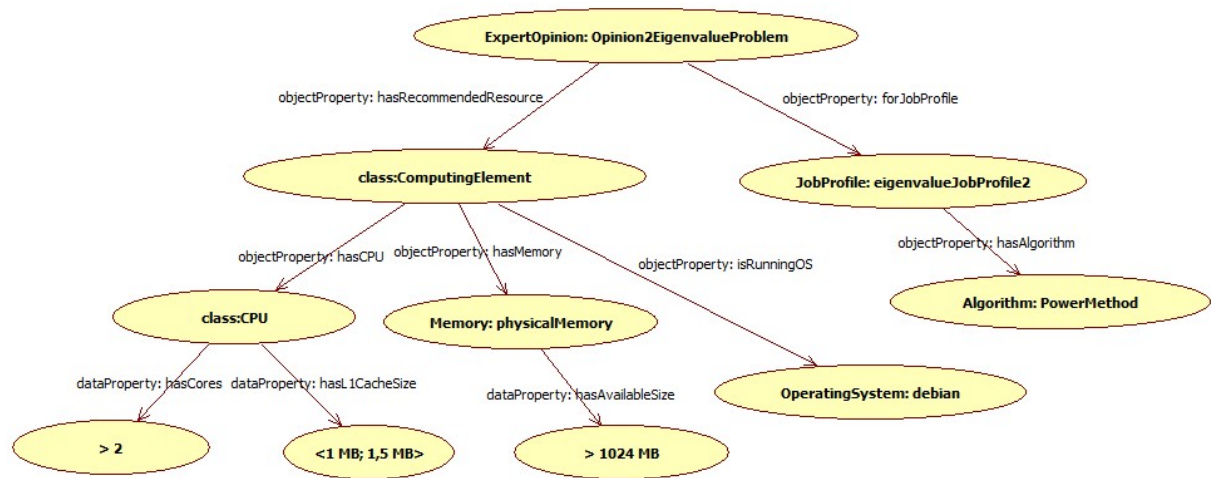**FIGURE 2.** Sample expert opinion structure 1

FIGURE 3. Sample expert opinion structure 2

As in the case of the resource description tree, structures corresponding to expert opinions can vary due to different set of properties selected from the ontology to describe the resource and the input data properties. The MCDA method, selected to "combine" expert opinion, should handle situation when opinions are based on different properties existing in the ontology. Furthermore, these properties may differ from those existing in the original problem formulation provided by the user.

Expert that defines opinions can use properties from the ontology that take values of type:

- numeric e.g. *hasClockSpeed*, *hasCores* for *CPU*, *hasTotalSize* for *Memory*
- text e.g. *hasVersion*, *hasModel*
- class instance e.g. *hasStorageInterface* for *StorageSpace*, *hasArchitecture* for *CPU*, *hasInstalledSoftware*

The ontology for the grid structure description, at the moment, has about 20 data properties and about 20 object properties defined. It can be assumed that user / expert would use much fewer properties than available to characterize a resource and a problem, however more than was included in the above provided examples. Additionally user / expert may give a set of acceptable values that the property may take e.g. (i) for numerical value – available RAM size between 1 GB and 2 GB, (ii) for class instance values – file system ext2, ext3, reiserfs. As a result, apart from giving specific value, it is also possible to use intervals and discrete sets as values (represented as class expressions).

MCDA to support user in contract selection

In this step one decision maker (software agent representing the user) chooses an optimal contract offer from all offers that have been received (or informs that there was no acceptable offer). Evaluation of offers is based on requirements (corresponding to the criteria of assessment; for details about contract structure see [4]) that were provided by the user. Contract inquiry defines requirements on selected contract properties e.g. *peak time price* > 100, 10 < *deadline penalty* <100, *job execution timeline* = 3 Apr 2014 etc. In received offers, these properties should have specified values e.g. *peak time price* = 120, *deadline penalty* = 50. Additionally, user should be able to specify his priorities regarding requirements. Note that this case is a simplified version of the MCDA involved in the first phase of the job execution scenario (analogical structure of the decision problem and alternatives; expert domain knowledge is not utilized), and therefore, it will not be further elaborated in this paper.

General remarks

Naturally, the same MCDA method can be used in phase one and two. However, in the scope of conducted research, it is planned to evaluate the possibility of using different methods in each phase. In conclusion, preferable MCDA methods should have the following characteristics:

- usage of knowledge coming from many experts that can have different weights assigned i.e. student's opinion is not equivalently significant as professor's opinion. Additionally, the same expert may have assigned different weights depending on the domain problem e.g. expert in the field of properties of eigenvalues for complex matrices, may not be equally proficient in solving linear equations with almost block diagonal matrices,
- usage of many criteria that have weights assigned, originating from different sources, e.g from user's preferences that the price is more important than the execution timeline,
- hierarchical structures of alternatives vary from case to case (i.e. different trees corresponding to different ranges of used properties from the ontology), however they all are based on the common ontology,
- very important quality of the MCDA is a mechanism to handle hierarchical criteria structure in an easy and intuitive way (tree is generated from graph that represents an ontology),
- in the first phase, as a result of the MCDA application, the system should indicate (if possible) one expert opinion (alternative) that will be confronted with user's requirements.

METHODS SELECTED FOR EVALUATION

Let us now focus our attention on selecting the MCDA method(s) suitable for the scenarios described above. In [11], authors formulate five key questions guiding the choice of the MCDA method in a given decision context. While, in the text, its authors provide sample decision contexts (e.g. responses to a tender), we would like to consider these questions in the context of our use case scenarios. Specifically, this analysis should justify why we have chosen methods described in following sections as the most suitable to our scenario. Authors of [11] say: *"suppose that the analyst has arrived at the stage of reflection where she is about to choose the most suitable multicriteria method to be used within the decision process. This method should be seen as a tool for going deeper into the decision problem, for exploring various possibilities, interpreting them, debating and arguing, rather than a tool able to make the decision.".* At the current state of research we want not only to apply MCDA method, but to choose methods that return a result as valuable as possible. The proposed technique is used to justify our choice of methods.

The MCDA technique that we started with, was the Analytical Hierarchy Process (AHP; [12, 13, 14]), and partial results of our research were described in [8, 15]. The main reason was that hierarchical application of pairwise comparisons matched very well with the structure of ontologically formalized knowledge. However, given the fact that AHP works ONLY in case when comparison matrices are consistent , we have decided to consider also other methods. The aim is to seek another technique that also works well with ontologically represented information and matches our use case scenarios. Moreover, the applied MCDA should have characteristics described in previous section.

During initial analysis of available methods, we selected those that seem to be appropriate for AiG scenario, and will be analyzed from the point of view of the methodology introduced in [11]. Besides AHP (which we will now analyze on "equal footing" with other methods), considered approaches include: TOPSIS ([16, 17]), PROMETHEE

([18, 19, 20]), and GRIP ([21, 22]) combined with ontological matchmaking. Let us start with a brief description of the four methods.

Analytical Hierarchy Process

The Analytical Hierarchy Process is a multicriterial analysis known since the 70s. It was designed for complex problems with multiple, antagonistic and subjectively assessed criteria (with hierarchical structure) supporting analysis of finite number of alternatives by multiple experts. It also provides an opportunity to check consistency of user's preferences. AHP can be used in both MCDA phases i.e. to select "representative" expert opinion from domain knowledge to help user in problem formulation, and to select an optimal contract offer. Since the detailed description of how AHP was applied to support user in problem formulation and contract selection was given in [8, 15] we will skip these details.

Technique for Order Preference by Similarity to Ideal Solution; TOPSIS

The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS; [16, 17]) is a method based on selection of a positive ideal alternative (PIS) and a negative ideal alternative (NIS) (i.e. the best and the worst "offer"). Next, the goal is to select an alternative that is the "closest" to the PIS and the "furthest" from the NIS. In the AiG system PIS for the problem formulation phase would be an expert-defined alternative (from the point of view of an expert, her opinion is positive ideal), while in case of contract selection it would be an offer with the best attribute values among received offers (artificially constructed positive ideal). At the same time, NIS in the case of problem formulation needs to be specified by an expert, while in the case of contract selection the NIS would be the proposed contract with the worst attribute values (artificially constructed negative ideal). With PIS and NIS constructed in this way, all alternatives would be evaluated.

As a method for measuring distances in TOPSIS, an ontological matchmaking proposed in [23, 24] could be applied. This method was developed specifically to calculate distances between ontologically described alternatives (instances) where ontology is modeled on the graph. Distances are calculated in two phases:

1. distance between concepts in the conceptual model – optionally defined by experts,
2. semantic distance between instances – scaling of expert-defined distances with values received from the user.

Let us denote:
S_i^+ – distance between $i-th$ alternative and PIS
S_i^- – distance between $i-th$ alternative and NIS
$C_i^* = \frac{S_i^-}{S_i^+ + S_i^-}$ – distance from ideal alternative

In our case, TOPSIS for group decision making proceeds as follows:

1. From the point of view of every $k-th$ expert select PIS V^{k+} and NIS V^{k-} (the evaluation is performed by the system, without experts interaction, however assessment is done on behalf of an expert basing on opinions that she has entered)
2. From the point of view of every expert opinion, calculate the distance between $i-th$ alternative and PIS S_i^{k+}, and NIS S_i^{k-} – using ontological matchmaking algorithm
3. For a group, calculate aggregated distance between $i-th$ alternative and PIS $\overline{S_i^+} = S_i^{1+} \otimes ... \otimes S_i^{K+}$ and NIS $\overline{S_i^-} = S_i^{1-} \otimes ... \otimes S_i^{K-}$; e.g. by using arithmetic or geometric average (or any other suitable norm)
4. For a group, calculate the distance between $i-th$ alternative and ideal alternative $\overline{C_i^*} = \frac{\overline{S_i^-}}{\overline{S_i^+} + \overline{S_i^-}}$
5. Select the alternative with the maximum $\overline{C_i^*}$

As one can see, when TOPSIS is selected to be used in the AiG, the starting point is to store expert opinions and distances between concepts in the ontology-based conceptual model. This information is provided by experts, who

interact with the system only at this stage. Later, the system performs MCDA TOPSIS analysis, additionally using semantic distances between instances provided by the user (to express her preferences).

PROMETHEE

PROMETHEE ([18, 19, 20]) is a group decision support method from class of outranking methods. The method proceeds in the following steps:

1. Selection of experts (decision makers) $E = \{e_r\}_{r=1,...,R}$
2. Criteria selection $C_1, \ldots, C_M$ (common and individual for a given expert)
3. Alternatives evaluation by each decision maker
    - criteria priorities calculation
      $w_1^r, w_2^r, ..., w_M^r, \sum_{i=1}^{M} w_i^r = 1$
    - assignment of preference functions to criteria $P_i(a,b), i = 1,...,M$ – pairwise comparison of alternatives $a, b \in A$ with respect to criteria i, where A is set of alternatives
    - determination of individual order of alternatives
      $\pi^r(a,b) = \sum_{i=1}^{M} P_i(a,b) w_i^r$
      $\phi^{+r}(a) = \sum_{x \in A} \pi^r(a,x)$ – strength of alternative a
      $\phi^{-r}(a) = \sum_{x \in A} \pi^r(x,a)$ – weakness of alternative a
      $\phi^r(a) = \phi^{+r}(a) - \phi^{-r}(a)$ – *net flow* of alternative a
4. Determination of global solution
    - experts priorities
      $\omega_1, \omega_2, ..., \omega_r, ..., \omega_R, \sum_{r=1}^{R} \omega_r = 1$
    - determination of global order of alternatives
      $\Phi^G(a) = \sum_{r=1}^{R} \phi^r(a) \omega_r$

Note that, in the standard PROMETHEE method, there is no hierarchical structure of criteria considered, which can be a great disadvantage in the case of the AiG system. The ontology structure can be intuitively transformed into a hierarchy, and with the complexity of ontological description it would be difficult to analyze the problem without any decomposition. Fortunately, authors of [25] propose an extension to PROMETHEE that handles the hierarchy of criteria. In such case PROMETHEE compares couple of alternatives with respect to criteria and subcriteria from a hierarchy in order to construct an outranking relation.

Generalized Regression with Intensities of Preference; GRIP

The Generalized Regression with Intensities of Preference (GRIP, [21, 22]) method is used for ranking a finite set of alternatives on multiple criteria. GRIP builds a set of additive value functions compatible with preference information composed of a partial preorder and of intensities of preference on a subset of alternatives. One of the key advantages of GRIP is that the preference information does not have to be complete, i.e. in our case user/expert may specify information corresponding to her degree of knowledge. The preference information is provided indirectly, in the form of holistic comparisons between reference alternatives on specific criteria. Note that, providing indirect preference requires less cognitive efforts from the decision maker. Based on such information, the preference model is built and applied to the set of all alternatives, in order to rank them in necessary or possible preference relation. Additionally, the GRIP extends methods based on ordinal regression with preference information in the form of comparisons of intensities of preference established between a pair of reference alternatives, e.g. *a* is preferred to *b* as much as *c* is preferred to *d*. The way in which GRIP handles group decision making is described in [26].

It is worthy noting, that a comparison between GRIP and AHP is discussed in [21].

RESPONDING TO QUESTIONS GUIDING THE CHOICE OF THE MCDA

The following subsections correspond to questions proposed by authors of [27]. In each subsection we assess methods described in previous section with respect to subsequent questions guiding the choice of the MCDA.

Taking into account the context of the decision process, what type(s) of results the method is expected to bring, so as to allow elaboration of relevant answers to questions asked by the decision maker?

This is the initial and most crucial issue allowing to significantly short-list available techniques. Obviously, there may be more than one method that gives result of the desired type. Consequently, the answer to this question is just a "preliminary" consideration, before continuing to the five following questions. Authors distinguish five main types of result (for references to the specific methods mentioned below, see [27]): (i) a numerical value is assigned to each potential action (e.g. MAUT, AHP, SMART, TOPSIS), (ii) the set of alternatives is ranked (without associating a numerical value to each of them) as a complete or partial weak order (e.g. ELECTRE III, IV, PROMETHEE I, II, ROR, GRIP); alternatives have to be known a'priori and not evaluated as they arrive, (iii) a subset of alternatives, as small as possible, is selected in view of a final choice of one or, at first, few alternatives (e.g. ELECTRE I, PROMETHEE V); alternatives have to be known a priori, (iv) each alternative is assigned to one or several categories, given that the set of categories has been defined a priori (e.g. Dominance-based Rough Set Approach, UTADIS, PREFDIS), (v) a subset of potential alternatives enjoying some remarkable properties is provided to serve as a base in the following stage of the decision aiding process; used when there is a very large number of alternatives often in interactive multiobjective optimization.

In the case of supporting user in problem formulation, the welcomed results are (i) and (ii). To select the most representative expert opinion among available and matching job profile, user's agents needs to construct an order either by assigning score / utility value or by ranking without associating numerical value. Considering methods selected for further evaluation the first case is realized by e.g. AHP and TOPSIS, the second case is realized by GRIP and PROMETHEE.

Do the original performance scales have all required properties for a rightful application of the considered method?

Some methods can have problems with handling evaluation put on a verbal scale. In our case the method should handle both numerical and verbal scales, e.g. price is assigned a numerical value, payment mechanism is selected from list of available values, however, job execution timeline is a datetime interval. In AiG the Grid resource can be characterized by properties of various types, and the selected MCDA technique should be able to handle them by providing means to transform a multidimensional scale into a one dimensional scale.

Let's focus our attention on the following methods: AHP, TOPSIS, PROMETHEE, GRIP, that are considered for the AiG scenario. For example in the case of AHP, the original scales are transformed to a scale accepted by the method (proposed by Saaty). In other methods there is also a requirement to transform original property values into numerical values, and the transformation mechanism should be designed by the user. For the AHP method, there are various possible scales, and choice of each of them is associated with some criticism. On the other hand, no guidelines regarding transformations into numerical means that the analyst that is to do it, must have a good understanding of the method. TOPSIS and GRIP are methods among four considered that do not require any transformation. In case of TOPSIS, it is because of the ontological matchmaking that is applied. Determining distance between ontological instance can be performed using the original performance scale. In case of GRIP decision maker provides her preferences in the form of holistic judgements (direct comparison between values).

Is it simple or hard (even impossible) to get preference information that the method requires?

Usually, the MCDA requires to acquire preferences from the decision maker in order to construct a preference model that the method exploits to get required results. This procedure may take different forms e.g. ordering of criteria, ordering of alternatives, pairwise comparisons, assignment to categories etc. The key issue is for the interaction protocol to be comprehensive, and at the same time it should ensure the traceability of the impact that preference information has on the result.

In the four methods under evaluation the most easily comprehensible method for acquiring preferences are provided by GRIP and AHP. In the GRIP, the preference model is constructed by an analysis of decision maker actions i.e. decision maker decides, which criteria value, in which alternative (from the set of reference alternatives) suits him better, and a set of additive value functions is constructed and applied to all alternatives. This approach is most user-friendly since there is no need to learn and understand the MCDA. Additionally, the constructed decision model is traceable when one wants to analyze why given results were obtained.

The other method that provides quite intuitive procedure for defining preferences is the AHP. In order to calculate criteria priorities, user fills comparison matrices where criteria from each level of hierarchical problem structure are compared in pairs. Assessment uses scale proposed by Saaty, in which numerical values are assigned to verbal statements e.g. equally important, is more important than, is strongly more important than, etc. Using values from the verbal scale and comparing pairs of criteria, instead of ranking the full set of criteria in terms of importance, can be regarded as user friendly. However, when problem structure is large, and when there are many criteria on the same level, then the number of comparisons that decision maker needs to make can be cumbersome. Fortunately, there are methods to automatically fill parts of the matrices. The advantage of the AHP is the possibility to check consistency of preferences, which is crucial when information is entered by user.

In the PROMETHEE method, preference information is obtained from the user as weights assigned to criteria. Unfortunately, there is no predefined method to calculate weights, therefore user is required to have a good understanding of the method, in order to choose appropriate values.

Finally, TOPSIS with ontological matchmaking gives the possibility to assign relevance between concepts on the conceptual and individual levels. Relevance, and inverse distance is equivalent to weights (priorities) assigned to the criteria. The procedure, however, is not user-friendly. The ontology needs to be visualized and analyzed as a graph, which can be problematic (especially, for large ontologies). Moreover, there is no indication which method should be used to determine distances, and what scale is acceptable.

Should the part of imprecision, uncertainty or indetermination in the definition of performances be taken into account, and if so, in what way?

When selecting the MCDA, one should also consider the issue of ambiguity in evaluating the performance of alternatives for a given criteria. It may happen that criteria are ill-determined or imprecise, which may result in inconsistencies in indirect preference information given by the decision maker's decision examples. In such cases methods should provide means to handle imperfect knowledge.

In the AiG scenario decision problem structure and criteria are determined by the ontology. It may be assumed that criteria are well-defined and described. All evaluated MCDA use ontology, therefore, there is no need to consider imprecision, uncertainty or indetermination.

Is the compensation of bad performances on some criteria by good ones on other criteria acceptable?

In the case of the AiG, compensation of bad performances on some criteria is acceptable but not required. Therefore, answer to this question will not shorten the list of possible MCDA techniques.

Is it necessary to take into account some forms of interaction among criteria?

Most available methods do not take into account any interactions among criteria. As a result, a set of criteria is usually designed in a way that excludes any interaction. If this is impossible, authors of [27] suggest to perform analysis of the form of interactions and choose method that may handle them. Fortunately, in case of the AiG system, there should be no interaction among criteria that correspond to independent properties from the ontology. Therefore, any of the methods selected for evaluation can be applied.

In the rest of the section four selected methods are presented a little bit more detail. For each method pros and cons are given. Left column for each method contains pros for this particular method while cons are given in the right column.

Summarizing pros and cons of the four methods

AHP pros & cons

- it is a method for multiple criteria (qualitative and quantitative) and multiple decision makers (experts)
- method result is a numerical value assigned to each alternative – alternative with the highest score is selected
- original performance scales are relatively suitable to be considered in this method – verbal scale that allows assessment of qualitative and quantitative criteria, user friendly approach
- it is relatively straight-forward how to get preference information required by the method
- in case of many decision makers aggregation is possible when defining preferences and/or alternatives assessment
- hierarchical structure of decision problem – intuitive problem decomposition and transformation from ontology graph
- consistency check for users and experts input

- there has been a lot of discussion about possible methods of transformation of verbal scale into numerical scale and their implications
- rank reversal problem,
- criticism of "scientific" foundations – method always gives result, but "value" of the result may be arguable

TOPSIS pros & cons

- method result is a numerical value assigned to each alternative – alternative with the highest score is selected
- original performance scales are suitable to be considered in this method – no transformation is required due to ontological matchmaking
- global PIS and NIS can be determined from aggregated experts alternatives (expert opinions)
- ontological matchmaking method is specifically designed for ontologically represented information
- no need to transform performance scale into numerical values

- there is a question how experts should input NIS into the system since it is not intuitive to specify the worst alternative with such a complex description
- it is hard to get preference information – experts should define distances for conceptual model to establish weights for properties describing instance representing expert opinion – however, it may be problematic to analyze ontology graph in case of complex ontology structure
- a common range of possible distances has to be defined for experts to assign in conceptual model (common unit), however authors of the method do not suggest what values can be assigned
- no method for checking consistency

- method result is the ranked set of alternatives – alternative ranked first is selected
- outranking is defined at the level of pairwise comparisons between each pair of alternatives
- outranking is more interactive between decision maker and model
- method for group decision making
- there exists an extension that allows to consider hierarchical structure of criteria (MCHP)

- ontological representation of information is not an advantage
- to consider hierarchical structure of criteria we need to apply extension to the method
- original performance scales have to be transformed into numerical values to be considered by the method
- there is no predefined method to assign weights to criteria, user has to have good understanding of the method

GRIP pros & cons

- method result is the ranked set of alternatives – alternative ranked first is selected
- intuitive method to provide preferences (holistic pairwise comparisons) – partial preorder on the set of reference actions
- original preference scales are suitable for the considered method
- decision maker provides only preferences that he is sufficiently certain
- as a result we obtain necessary/possible relations on the set of all alternatives
- support for many decision makers

- ontological problem structure has to be transformed into structure that is accepted as input by GRIP

OTHER POSSIBLE APPROACHES

Discussion presented above makes it easy to see that the problem of selecting the MCDA method that matches ontologically represented information and handles the considered scenario with many experts, turned out to be nontrivial. Methods under evaluation should not only choose "the best" alternative but also:

- help to improve requirement specification process,
- handle choice between many alternatives that match user criteria.

To complete the picture, let us now briefly summarize other groups of methods (or methods) that were investigated by us, before we decided to focus on AHP, TOPSIS, PROMETHEE and GRIP, and evaluate them with questions from [11]. In the rest of the section three selected groups of methods (or methods) are presented and for each pros and cons are given. Left column contains pros for this particular group of methods (or method) while cons are given in the right column.

MAUT

The family of MAUT methods aggregate different criteria into a function that has to be maximized. Weights assigned to criteria express their relative importance, however scores are required to be from one dimensionless scale. As discussed in [28], MAUT allows complete compensation between criteria, i.e. the gain on one criterion can compensate the loss on another.

- most broadly accepted model of rational choice
- handles multi-criteria analysis
- simple in terms of understanding meaning of information that users have to provide
- utility functions are derived independently of alternatives' properties i.e. alternatives do not influence the decision

- method to a minimal degree (or not at all) indicates how utility value for an alternative with respect to a given criteria has to be calculated
- the mathematical structure of the utility function gets much more complex in case of dependent criteria; moreover, in case of AiG the number and complexity of criteria can lead to problems with determining correct utility function
- criteria weights have to be provided as input either as direct allocation or using other procedure – problematic in case of complex criteria structure
- utility functions are based on the decision maker's expertise and preferences – difficult to construct for complex problem with tangible and intangible criteria

The main reasons for not selecting MAUT methods in AiG are: (i) construction of utility function would be non-trivial with problem structure in AiG (many multi-dimensional criteria), (ii) no user-friendly method for assignment of weights in case of hierarchical criteria structure.

Linear additive models

Linear additive models are simple MCDA model that require assumption about independence of criteria. In the linear additive model alternative's values on many criteria can be combined into one overall value. This is done by multiplying the score on each criterion by the weight of that criterion, and then adding all those weighted scores together. Following we present pros & cons of linear additive models on the example of SMART method (Simple Multi-Attribute Rating Technique; [29]) and its modifications SMARTS (SMART with Swing; [30]) and SMARTER (SMART Exploiting Ranks; [30]).

SMART, SMARTS, SMARTER pros & cons

- straightforward and intuitive
- SMART and SMARTS provide guidance with weight assignment and criteria scores (quantifying qualitative values), however there hierarchical structure of criteria is not considered
- SMARTER is an extension of SMART that is less restrictive in information input requirements – restrictions on weights and scores in form of linear inequalities

- because of appealing simplicity it is often misused e.g. critical is scaling of performance values for criteria and assignment of weights; failure in correct assignment of values can lead to method that appears well-founded but in reality does not represent proper understanding of a problem
- the assessment of value functions and swing weights can be difficult especially with large quantity of attributes
- in the case of the AiG, extension to group decision making is not straight forward, especially an issue of scaling and weighting (how to choose mathematically sound)

The main reasons for not selecting linear additive models in AiG are: (i) assessment of value functions would be non-trivial with problem structure in AiG (many multi-dimensional criteria), (ii) swing weights in case of hierarchical structure with many criteria can be difficult.

Outranking methods

The concept of outranking was discussed in [31, 32]. Outranking methods are based on concordance and non-discordance conditions. Concordance means that alternative A_i outranks alternative A_j if on most citeria A_i performs better than A_j, while non-discordance means that its worse performance is still acceptable.

ELECTRE

- shortlisting of alternatives based on the concept of dominance
- the assessment of criteria weights is given as input – there is a question how to assign proper values (well-justified values are crucial to construct concordance and discordance matrices)
- in basic form there is no hierarchy of criteria, however MCHP extension can be used (as in PROMETHEE)

- ontological representation of information is not an advantage
- to consider hierarchical structure of criteria we need to apply extension to the method

ELECTRE method could be also used in AiG. However, we decided to consider only one method from the outranking methods i.e. PROMETHEE.

CONCLUDING REMARKS

To conclude, we have considered multiple methods and analyzed them in the context of AiG scenerio. We compared their features, and after initial evaluation we selected four methods (AHP, TOPSIS, PROMETHEE, GRIP) that seemed to match our requirements. In this paper these methods were evaluated with respect to questions suggested in [11]. The evaluation confirms that AHP, TOPSIS, PROMETHEE and GRIP constitute good choices. First of all, their features match our requirements (can be used with ontologies and support multiple criteria with many decision makers), they are all well-founded and boadly used, and moreover each of the method represents a different approach to perform MCDA. Finally, to complete the paper, we also present pros and cons of groups of methods (or methods) that we discarded.

ACKNOWLEDGMENTS

REFERENCES

1. M.Dominiak, W.Kuranowski, M.Gawinecki, M.Ganzha, and M.Paprzycki, "Utilizing agent teams in grid resource management—preliminary considerations," in *Proc. of the IEEE J. V. Atanasoff Conference*, IEEE CS Press, Los Alamitos, CA, 2006, pp. 46–51.
2. W.Kuranowski, M.Ganzha, M.Gawinecki, M.Paprzycki, I.Lirkov, and S.Margenov, *International Journal of Computational Intelligence Research* **4**, 9–16 (2008).
3. K.Wasielewska, M.Drozdowicz, M.Ganzha, M.Paprzycki, N.Attaui, D.Petcu, C.Badica, R.Olejnik, and I.Lirkov, "Negotiations in an Agent-based Grid Resource Brokering Systems," in *Trends in Parallel, Distributed, Grid and Cloud Computing for Engineering*, edited by P. Ivanyi, and B. Topping, Saxe-Coburg Publications, Stirlingshire, UK, 2011.
4. M.Drozdowicz, K.Wasielewska, M.Ganzha, M.Paprzycki, N.Attaui, I.Lirkov, R.Olejnik, D.Petcu, and C.Badica, "Ontology for Contract Negotiations in Agent-based Grid Resource Management System," in *Trends in Parallel, Distributed, Grid and Cloud Computing for Engineering*, edited by P.Ivanyi, and B.H.V.Topping, Saxe-Coburg Publications, Stirlingshire, UK, 2011.
5. K.Wasielewska, M.Drozdowicz, P.Szmeja, M.Ganzha, M.Paprzycki, I.Lirkov, D.Petcu, and C.Badica, "Agents in Grid System—Design and Implementation," in *Large Scale Scientific Computing*, Springer Germany, 2012, vol. 7116 of *LNCS*, pp. 654–661.

6. P.Szmeja, K.Wasielewska, M.Ganzha, M.Drozdowicz, M.Paprzycki, S.Fridanova, and I.Lirkov, "Reengineering and extending the Agents in Grid Ontology," in *Large Scale Scientific Computing*, LNCS, Springer Germany, 2013.

7. K.Łysik, M.Ganzha, K.Wasielewska, M.Paprzycki, J.Brennan, V.Holmes, and I.Kureshi, "Combining AiG Agents with Unicore grid for improvement of user support," in *Proceedings of the First International Symposium on Computing and Networking – Across Practical Development and Theoretical Research*, 2013 submitted for publication.

8. K.Wasielewska, and M.Ganzha, "Using analytic hierarchy process approach in ontological multicriterial decision making - Preliminary considerations," in *AIP Conference Proceedings*, American Institute of Physics, 2012, vol. 1487, pp. 95–103.

9. K.Wasielewska, M.Ganzha, M.Paprzycki, and I.Lirkov, "Developing ontological model of computational linear algebra Ű preliminary considerations," in *AIP Conference Proceedings*, American Institute of Physics, 2013, vol. 1561, pp. 133–144.

10. Ontologia w ujeciu T. Grubera, `http://www.inzynieriawiedzy.pl/ontologie/definicje` (2013), [Online; accessed 07-November-2013].

11. B.Roy, and R.Slowinski, *EURO Journal on Decision Processes* **1**, 69–97 (2013), ISSN 2193-9438.

12. T. Saaty, *The Analytic Hierarchy Process*, RWS Publications, Pittsburg, 1990.

13. T.L.Saaty, *International Journal of Services Sciences* **1**, 83–98 (2008).

14. T.L.Saaty, *European Journal of Operational Research* **48**, 9–26 (1990).

15. K.Wasielewska, M.Ganzha, M.Paprzycki, P.Szmeja, M.Drozdowicz, I.Lirkov, and C.Badica, *Information Technology And Control* (2014 submitted for publication).

16. H.-S. Shih, H.-J. Shyur, and E. S. Lee, *Mathematical and Computer Modelling* **45**, 801 – 813 (2007), ISSN 0895-7177.

17. C.L.Hwang, and K.Yoon, *Multiple attribut decision making : Methods and applications*, Springer-Verlag, 1981.

18. B.Mareschal, J.P.Brans, and C.Macharis, The GDSS PROMETHEE procedure: a PROMETHEE-GAIA based procedure for group decision support, ULB Institutional Repository 2013/9373, ULB – Universite Libre de Bruxelles (1998).

19. J.P.Brans, and Ph.Vincke, *Management Science* **31**, 647–656 (1985).

20. J.P.Brans, Ph.Vincke, and B.Mareschal, *European Journal of Operational Research* **24**, 228–238 (1986).

21. J.R.Figueira, S.Greco, and R.Slowinski, *European Journal of Operational Research* pp. 460–486 (2009).

22. S.Greco, R.Slowinski, J.R.Figueira, and V.Mousseau, "Robust Ordinal Regression," in *Trends in Multiple Criteria Decision Analysis*, edited by M. Ehrgott, J. R. Figueira, and S. Greco, Springer US, 2010, vol. 142 of *International Series in Operations Research & Management Science*, pp. 241–283, ISBN 978-1-4419-5903-4.

23. S.K.Rhee, J.Lee, M. Park, M.Szymczak, G.Frackowiak, M.Ganzha, and M.Paprzycki, *Fundam. Inform.* **96**, 395–418 (2009).

24. M.Szymczak, G.Frackowiak, M.Ganzha, M.Paprzycki, S.K.Rhee, J.Lee, Y.T.Sohn, J.K.Kim, Y. Han, and M. Park, "Ontological matchmaking in a Duty Trip Support application in a virtual organization," in *Proceedings of the International Multiconference on Computer Science and Information Technology, IMCSIT 2008, Wisla, Poland, 20-22 October 2008*, 2008, pp. 243–250.

25. S.Corrente, S.Greco, and R.Slowinski, *Omega* **41**, 820–846 (2013).

26. M.Kadzinski, S.Greco, and R.Slowinski, *Group Decision and Negotiation* **22**, 429–462 (2013), ISSN 0926-2644.

27. B.Roy, and R.Slowinski, *EURO Journal on Decision Processes* **1**, 69–97 (2013), ISSN 2193-9438.

28. R. Keeney, and H. Reiffa, *Decisions with Multiple Objectives: Performances and Value Trade-Offs*, Wiley, New York, 1976.

29. W.Edwards, *IEEE Transactions on Systems, Man and Cybernetics* **7**, 326–340 (1977).

30. W.Edwards, and F.Hutton Barron, *Organizational Behavior and Human Decision Processes* **60**, 306–325 (1994).

31. B.Roy, *RAIRO - Operations Research - Recherche Opérationnelle* **2**, 57–75 (1968).

32. P.Vincke, *Multi-criteria Decision-Aid*, John Wiley, Chichester, 1992.