Sustainable Production through Advanced Manufacturing, Intelligent Automation and Work Integrated Learning, J. Andersson et al. (Eds.)

doi:10.3233/ATDE240189

Knowledge Graphs for Supporting Group Decision Making in Manufacturing Industries

Ludwig MITTERMEIER [a,1], Richard SENINGTON [a], Sunith BANDARU [a], and Amos NG [a,b]

[a] *University of Skövde, School of Engineering Science*
[b] *Uppsala University, Department of Civil and Industrial Engineering*

Abstract. Group decision making is traditionally a human-centered process, where communication, synchronization and agreement are driven by the stakeholders involved. In the area of multi-objective optimization (MOO), this becomes a challenge, because MOO usually produces a large amount of trade-off solutions that need to be analyzed and discussed by the stakeholders. Moreover, for transparent group decision making, it is important that each decision maker is able to trace the entire decision process – from associated data and models to problem formulation and solution generation, as well as to the preferences and analyses of other decision makers. A graph database is capable of capturing such diverse information in the form of a knowledge graph. It can be used to store and query all dependencies and hence can support complex decision-making tasks. Further advantages are the inherent suitability for visualization and the possibilities for pattern matching, graph analytics and, if semantically enriched, to infer new connections in the graph. In this paper, we show how such a knowledge graph can be used to support more transparent and traceable decision-making activities, particularly when multiple stakeholders with differing preferences or perspectives are involved.

Keywords. knowledge graph, group decision making, multi-objective optimization

1. Introduction

In modern manufacturing industries optimization plays a vital role. It is desired on various levels of manufacturing, from business strategies over factory design to the work process execution. In many cases, however, a solution should optimize more than one objective, which leads to multi-objective optimization (MOO) where several objectives are addressed concurrently. These objectives are often conflicting, where an improvement for one objective causes a deterioration in another objective. In consequence, there is usually no single best solution for such a multi-objective optimization problem (MOOP). Instead, there is a set of solutions that outperform the remaining solutions, but perform comparably good with respect to the objectives. However, we are required to select only one of those solutions. In increasingly complex problems with more methods, simulations, and

[1]Corresponding Author: Ludwig Mittermeier, ludwig.mittermeier@his.se.

stakeholders, it becomes increasingly difficult to coordinate and reach agreement and conclusions regarding the necessary decisions. Decision support systems (DSSs) assist in the process of selecting and prioritizing solutions, and group decision support systems (GDSSs) help further by guiding the process of reaching a consensus among the stakeholders.
The methods available so far for group decision support based on MOO have a few shortcomings. They do not consider the relationship between the variable space (i.e., a parameter that can be changed) and the objective space (i.e., a parameter that should be optimized, but may not be changed directly). In [1], the first DSS to bridge variable and objective space for MOOPs was built. However, it does not address groups of decision makers. Also, many GDSSs assume a limited set of solutions for decision making [2,3]. In contrast, the set of solutions created by a modern optimization method can be very large, which requires additional exploration before an informed decision can be made [4]. Lastly, it seems each optimization problem and the decisions connected to it are treated as a single occurrence and possible connections to other problems are neglected. To overcome these limitations, we propose to use a knowledge graph to integrate data and relationships from all entities involved in the group decision making (GDM) – from physical parts and stakeholders, over methods for MOO, to the data that is used for the optimization.

2. Background

2.1. Multi-Objective Optimization

Multi-objective optimization is an optimization process with the goal of finding the best trade-off between solutions for multiple objectives. Usually, there is no single solution that is better than all others. Instead, there is a set of competing good solutions where each of the solutions has better performance with respect to some of the objectives and worse performance with respect to some other objectives. This set of solutions is called the Pareto front and the solutions as Pareto-optimal solutions or simply Pareto solutions. For complex problems, simulations are a common tool. Such simulations are usually non-deterministic, because they account for uncertainties.
As opposed to single objective optimization, an active choice depending on personal preference has to be made. There are four possible ways to address preferences in the selection process [5] as described below:

1. *No preference*: preferences are not considered in this approach, instead comparative characteristics of the candidate solutions are used to determine the final solution (e.g., knee points in a graph).
2. *A Priori*: the preferences are incorporated in the problem formulation, i.e., before any solutions are generated. The drawback is that stakeholders need to know their problem well from the beginning. Otherwise, they could miss suitable solutions that do not match their a priori preference, but might actually be even more desirable.
3. *A posteriori*: the preferences are considered after the solutions have been generated. Multi-Objective Evolutionary Algorithms (MOEAs), such as the well-known Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [6], are very commonly used for accomplishing this task. They yield a range of non-dominated solutions close to the

Pareto-front. After that, methods for Multi-Criteria Decision Making (MCDM) can help the decision maker with choosing one solution. Popular MCDM methods are analytic hierarchy process (AHP), analytic network process (ANP), elimination and choice expressing reality (ELECTRE), preference ranking organization method for enrichment evaluations (PROMETHEE), and technique for order preference by similarity to the ideal solution (TOPSIS). These methods rely on various ways of comparing alternatives and criteria. For the comparison, the preferences are invoked as weight factors, which makes finding suitable weights an important task [7]. However, MOEAs can produce a large amount of solutions, which makes assigning weights tedious. Thus, the common MCDM methods are only suitable for small sets of solutions and for larger sets, the focus of decision support moves to exploring the Pareto frontier [4].
4. *Interactively*: Stakeholders guide the optimization process toward their preferences by providing input during runtime. On the downside, this requires multiple interactions between the optimization algorithm and the stakeholders.

When compared to other methods, *a posteriori* methods have the advantage of enabling the exploration of solutions after the Pareto-front is found. In particular, the relation between variable space and objective space can influence the stakeholders' preferences. Furthermore, *a posteriori* methods are more suitable for group decision making, because they allow the group to discuss their preferences on the basis of actual solutions rather than possible solutions (*a priori*) and avoid several discussions (*an interactive process*).

2.2. Knowledge Discovery from Multi-Objective Optimization

The usual approach to gaining a good understanding of MOO solutions is through visualization. However, a simple visualization of solutions does not provide additional insights, such as the relation between variable and objective space. For that purpose, knowledge discovery methods are used. These can be split into two categories: implicit methods and explicit methods. Implicit methods aim at providing the decision makers with knowledge that is subject to further interpretation, often presented through visualizations [1]. These methods include:

- Clustering methods help to identify and highlight patterns of solutions.
- Manifold learning methods allow the representation of high-dimensional data in lower dimensions. Popular methods for this are t-SNE [8] and UMAP [9].
- Self-Organizing Maps (SOMs) [10] and Generative Topological Maps (GTMs) [11] have been used to visualize Pareto-optimal solutions in lower dimensions.
- The visualization of solutions as a network [12].
- Trend mining [13,14], which visualizes how changes in the objective space cause changes in the variable space.

While these methods make it easier to understand the relations between variables and solutions, there are also methods that aim at obtaining explicit rules for these relations. Explicit knowledge can significantly easier be stored, shared, compared and used for further analysis which are all desirable qualities for efficient decision making. Popular methods are:

- association rule mining [15] and rough set theory [16].

- Flexible Pattern Mining (FPM) [17] tries to overcome the shortcomings of previous methods like discretizing and overfitting. It is a modified version of Sequential Pattern Mining (SPM) [18] that uses the Apriori algorithm [19] to generate decision rules in the variable space that distinguish a *selected set* of solutions against an *unselected set* in the objective space. Each rule has a *selected significance* and an *unselected significance*, depicting the fraction of solutions in the respective sets that satisfy the rule. If a single rule is not enough to meet desired significance thresholds, several rules can be combined.
- Similar decision rules can be obtained with Simulation-based Innovization (SBI). Differing from the previous methods, automated innovization [20] does not yield if-then decision rules but mathematical relations between variables and the objective space that can be difficult to interpret.

In an interactive knowledge discovery setup, the FPM method has been shown to perform better than SBI [1], making it one of the most promising explicit knowledge discovery methods. Applying such a knowledge discovery method on the results of a MOOP, allows the decision makers to understand the decision problem better and to obtain decision rules according to their preferences. If there is more than one stakeholder involved, this also makes it easier for a group of decision makers to reach a consensus.

2.3. Group Decision Making

Decision making is a challenging task depending on many factors, such as the available choices, the decision maker's preferences, and the decision maker's knowledge about the consequences of the decision. Decision support systems (DSSs) primarily aim at assisting the decision maker by providing knowledge and leading the DM towards a better decision than without DSS [21]. Another desired benefit can be a better decision process itself, which could be to reach an equally good or better decision with less effort, in less time or with improved documentation [22]. While there are many possible types of DSSs, they all have four basic components [21]:

- a language system that defines messages or requests the DSS can receive,
- a presentation system that defines messages or result types the DSS can generate,
- a knowledge system that stores all knowledge available to the DSS, and
- a problem-processing system that tries to solve problems that arise during decisions making.

Group decision making is a category of decision making where at least two, usually more, stakeholders are involved in the decision process. Differences between each of the stakeholders' opinions and preferences on certain aspects of the decision problem make it substantially more difficult to find a suitable decision. Even how a decision should be reached within a group is subject to individual opinions. Historically, group decision support systems (GDSSs) were designed to assist in setting up collaborative work, where a group of stakeholders would define and clarify the possible alternatives [23]. GDSSs are not necessarily aimed at reaching a decision immediately [23], but rather to assist in finding options and improving the efficiency of the group through better communication [24].

For GDM that involves multiple criteria, MCDM methods are the central part [25]. While it is possible to extend an MCDM method with a social choice method for GDM [2] or simply aggregate the preferences of stakeholders at some level of the MCDM method, the decision making becomes generally more complex. This is because the solutions do not only need to be evaluated against the objectives but also against the preferences of all stakeholders. Additionally, the final solution is required to be a consensus of all stakeholders' opinions, and not just the top solution based on some aggregation. That means an efficient consensus-reaching process needs to have a manageable number of alternatives. As explained above, this is usually not the case for MOO solution sets. Thus, a GDSS for MOOPs should only apply MCDM methods after the solution set has been reduced. One way of doing that would be to agree on discovered decision rules first instead of deciding on a solution directly. That could yield a set of solutions small enough to proceed with traditional MCDM methods.

With more people involved in the decision-making process, it can also become more difficult to determine weights for criteria [26] and make a decision without conflicts [2]. Furthermore, decision makers might be assigned with individual weights depending on their expertise, which in turn may vary for different criteria. This could result in biased decisions, though, and so far, most methods for GDM assume equal weights [3].

During application, DSSs help to make decisions by providing the decision makers with relevant insights about the decision and fostering understanding of the problem [22]. The most important asset for that is the knowledge that the DSS has access to, which should evolve continuously and in real-time [27]. Thus, a knowledge storage that makes it easy to retrieve or add knowledge or information and that can infer knowledge on its own is desired. Using knowledge graphs for that is a promising approach [27]. Especially the possibility of connecting data to opinions and linking all influences for a decision together in one place opens opportunities for analysis that are hardly achieved by other storage types.

2.4. Knowledge Graphs

The idea of a knowledge graph is closely related to the idea of the semantic web, where all data on the internet is semantically enriched and linked [28]. In the stack of web technologies, the knowledge graph builds upon technology for ontologies, linked data and semantic networks [28]. RDFS and especially OWL are descriptions for ontologies, and SPARQL is their established query language. The less expressive RDF provides a triple-based structure for linking data and XML, in combination with IRIs are the foundations of a semantic network [28].

As for the definition of a knowledge graph, there is no commonly agreed one. There has been an effort to create a unifying definition based on a survey on current definitions of knowledge graph [29]. That definition puts the aspect of aggregating knowledge and enabling inference in its center [29]. For others, this definition is too narrow and a broader definition focused on the integration of information from various sources into a network is proposed [30]. Coming from a historical point of view about data and knowledge, a knowledge graph might even be seen as an evolving system or even a project that integrates data and knowledge in some type of graph or network [31]. Despite some differences, a commonality of these definitions is that they see the knowledge graph as a graph-like structure that incorporates information. Circling back to the web technologies, such a definition would mean that information expressed in RDF forms already a knowledge graph, which is a clear contradiction to the hierarchy in the stack of web technologies. This means conceptualized semantics, as described in ontologies, should be included or at least referred to in the network of information, if it should be a true knowledge graph. Considering these thoughts and definitions, we define a knowledge graph as a highly flexible graph-based data structure that allows to include semantics providing provenance data and metadata alongside the main data.

Already used before, the term ontology refers to an ”explicit specification of a conceptualization” [32]. It formally defines concepts and provides structure among the concepts. In other words, concepts, which can be treated as nodes, and relations between the nodes, which can be treated as edges, are defined in the ontology. The close relation to a knowledge graph is obvious, but ontologies mainly aim at providing a shared specification of knowledge and, in consequence, usually only refer to general domain knowledge and barely to instance data [28].

There are two common ways to store knowledge graphs: 1) in a graph database as labelled property graph (LPG) or 2) in a Resource Description Framework (RDF) document, including documents based on RDF such as ontologies in OWL (Web Ontology Language). The advantage of graph databases lies in the surrounding database management system that usually allows to interact easily with the stored data and can handle large amounts of data easily. For reasoning tasks, the lack of a strong semantic schema is a disadvantage. The RDF-based and, hence, schema-based documents, on the other hand, can choose from a broader range of reasoners which includes powerful reasoners for OWL. On the downside, they are not intended as databases and may have shortcomings in industrial applications. A combination of both, a graph database for large and simple data handling and an ontological schema for better reasoning capabilities, seems desirable. Some databases try to achieve this by relying on RDF graphs instead of labeled property graphs. However, their support for the representation of OWL elements and related reasoning capabilities vary, and they may not achieve the same performance as LPG databases. Either way, using a graph database in combination with an ontology will always be a trade-off solution between performance, expressivity, and reasoning capabilities. In this combined way, the knowledge graph can be used to store all data associated with the decision making and the multi-objective optimization. Furthermore, reasoning can be used to check consistency and to infer new insights from the data.

3. Supporting Group Decision Making with Knowledge Graphs: A Proposal

3.1. A Framework for Group Decision Support Using Knowledge Graphs

The proposed framework, shown in Figure 1, combines previously introduced methods to guide a group of stakeholders during multi-criteria decision making. The first step for a user would be to define the problem and select a method to solve the MOOP. As mentioned, evolutionary algorithms have been shown to be effective at solving MOOPs. They can yield a good estimate of the Pareto-optimal front of solutions, which can then be further explored by applying knowledge discovery methods. During interactive knowledge discovery, the user can become familiar with the solutions and the relations between the objective and variable spaces. This allows the user to make an informed choice when selecting the method and parameters for explicit rule extraction based on their personal preference. As for the implementation in Figure 2, the methods for MOO and knowledge discovery will mainly be executed in client applications.

The next step in the framework is to integrate the decision rules in a knowledge graph. Storing the decision rules from multiple users is important to facilitate decision making in the group at a later stage. Therefore, the rules are connected to the user who created them, as well as to the methods used for knowledge discovery and MOO. This enables traceability and supports future analysis. The knowledge graph is represented in a graph database like Neo4j and, via the interface server, connected to an ontology that stores the definitions of the concepts used in the knowledge graph (see Figure 2). Alongside the concepts, axioms that define conditions for the relations of different concepts are stored. For every update that is supposed to be applied to the knowledge graph, the ontology is invoked first to check that the update conforms to the definitions and structure specified by the ontology. Otherwise, the update would be rejected.
The decision rules that are obtained from knowledge discovery can be used for knowledge-driven optimization (KDO). Introduced in [33], there are two ways of KDO: offline KDO and online KDO. For online KDO, the decision rules are used to improve the performance of the optimization method during runtime. In contrast, offline KDO uses the decision rules from the last optimization problem to improve the problem formulation and parameter settings of the next MOO. In a similar way, past decisions can be analyzed and used to tune upcoming problems.

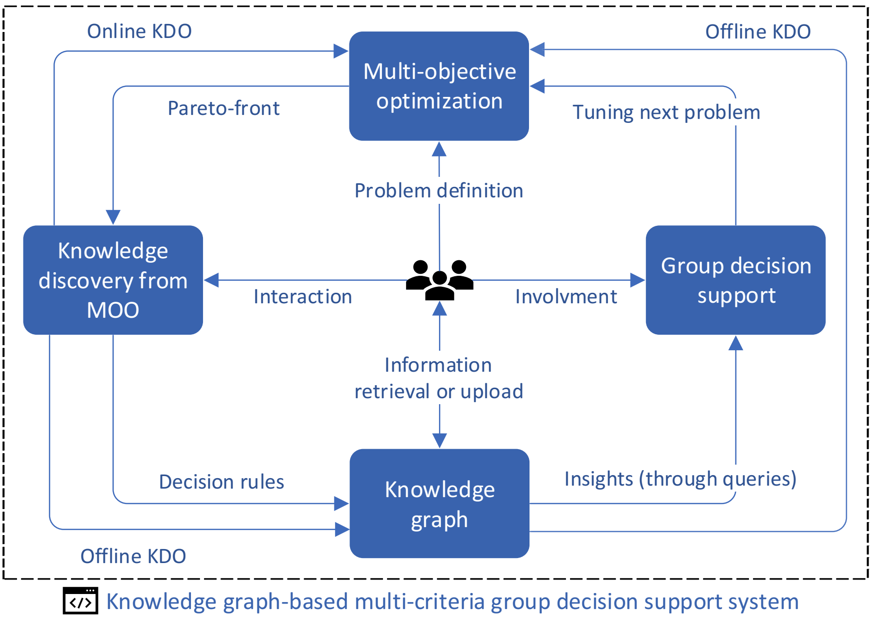

Figure 1. Information flow between the decision makers and the essential parts of the knowledge graph-based multi-criteria group decision support system.

For the group decision making to take place, all involved stakeholders should store the decision rules that depict their preferences and the methods for MOO and knowledge discovery they used in the knowledge graph. This is required to make sure that all decision makers have either agreed on certain methods to apply, a set of possible solutions prior to knowledge discovery or decision making, or are at least aware of the differences in the generation of solutions and decision rules. If the decision makers would proceed

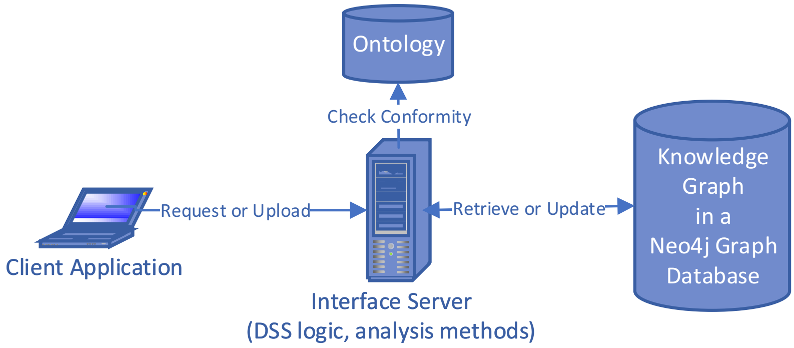

Figure 2. The architecture of the DSS, where a client application requests or uploads information to the interface server, which checks conformity with the ontology for updates and retrieves or updates parts of the knowledge graph in the graph database.

without knowing how the other stakeholders got to their decision rules, this could cause issues within communication and agreement while presumably taking longer. When all stakeholders have uploaded their preferred decision rules in the knowledge graph, the actual GDM can take place. The logic of the DSS is hosted on the interface server together with the analysis methods that can be applied to the knowledge graph (see Figure 2).

As Figure 1 shows, all the blocks are part of a knowledge graph-based multi-criteria group decision support system. Although the overall design with respect to the four subsystems defined by [21] is subject to future considerations, at the core of the multi-criteria GDSS is the knowledge graph which enables it to infer knowledge based on the information provided by the users who will in turn be provided with the insights through querying the graph. This intelligent information handling helps stakeholders find their common ground for the decisions. Identifying commonalities and differences in both the objective and the variable space among stakeholders is the key element here. Additionally, the knowledge graph can be used to find possible consensus solutions or suggest other methods to finalize the decision. An example for the latter would be a MCDM method modified for GDM and applied to a small set of selected solutions.

A general advantage of using the knowledge graph here, instead of a more rigid relational database, is the simplicity of information retrieval and updating. Not only for the decision-making process but also before and afterward it, the users can query the knowledge graph to find information they are seeking through complex pattern matching, which is much simpler with graph databases than with relational databases. They can also add additional information to enhance the understanding of the problem and support future analysis. In the same way, this can be done at any time at any place with access to the graph database servers through the web; the stakeholders can upload their decision rules at any time or place independent of their colleagues. Due to this asynchronous and decentralized participation of the stakeholders, their opinions will still be respected even if they cannot participate in the finalization of the decision.

3.2. Robot-Centered Production Cell Optimization: An Illustrative Example

As an illustrative example of the functionality of the method, we instantiate a knowledge graph in Neo4j with data from a dataset generated by multiple MOO runs on a robot-

centered production cell design, or simple robotic cell design, problem that considers cycle time, power and energy consumptions as three optimization objectives.
Using the Mimer[2] application for interactive knowledge discovery, the users can apply flexible pattern mining and define decision rulesets imposed on certain design variables in the dataset. These rules are integrated in the knowledge graph and linked to the respective method and user. The resulting knowledge graph is not very intuitive for decision support yet. Therefore, the next step is to analyze the knowledge graph in an exploratory and/or reporting manner. For exploration, clustering and pattern matching techniques can be applied to identify patterns in the definition of rulesets relating the most influencing design variables of the robotic cell design. For the reporting, targeted queries can be used to extract desired information e.g., whether there is an agreement on decision variables or what the shortest path between two stakeholders looks like. The latter can be useful to determine common ground for further discussion among stakeholders.
In the example, the connections between the stereotypical users and the variables they used in their decision rules are highlighted, as shown in Figure 3. This tells us already how the users value certain variables for the decision. For example, the user that focuses on ”Faster Balanced” values variable $velmax$ (max velocity of the robot) highly, while the user desiring ”High Throughput” does not consider this variable for the decision at all. It also shows that variable dr is considered by all four people, but the variable $veloverride$ is only considered by one person.
Once the decision is made, it will be integrated into the knowledge graph as well to serve as an example and as a reference for pattern-matching methods. In other words, a decision-making process that has worked in the past could be an important reference for an identical design problem in the future as well.

4. Conclusion and Outlook

We have shown that a knowledge graph can be a useful part of a DSS for multi-criteria group decision making. Specifically, the possibility to easily connect decision makers to their preferences as well as to find connections to other decision makers and their preferences makes it valuable. Additionally, the conformity to an ontology provides possibilities to reason over the information in the knowledge graph. Furthermore, decisions can be traced back to stakeholders, their preferences, and the methods they used for knowledge discovery and MOO. Future work may focus on the representation of preferences in the knowledge graph and how it can be improved to highlight partial agreement and disagreement between stakeholders. Exploring the possibilities of analysis and how they can help in the decision process is another interesting direction. Parallel to that, a systematic way of integrating variables into the ontology so that they can be used for reasoning can be rewarding. Finally, a goal for the implementation is a user interface that allows decision makers to create workflows all the way from defining the MOOP using offline KDO from previous decisions over applying methods for MOO and knowledge discovery to the analysis of preferences of the group of stakeholders.

[2] https://assar.his.se/mimer/html/

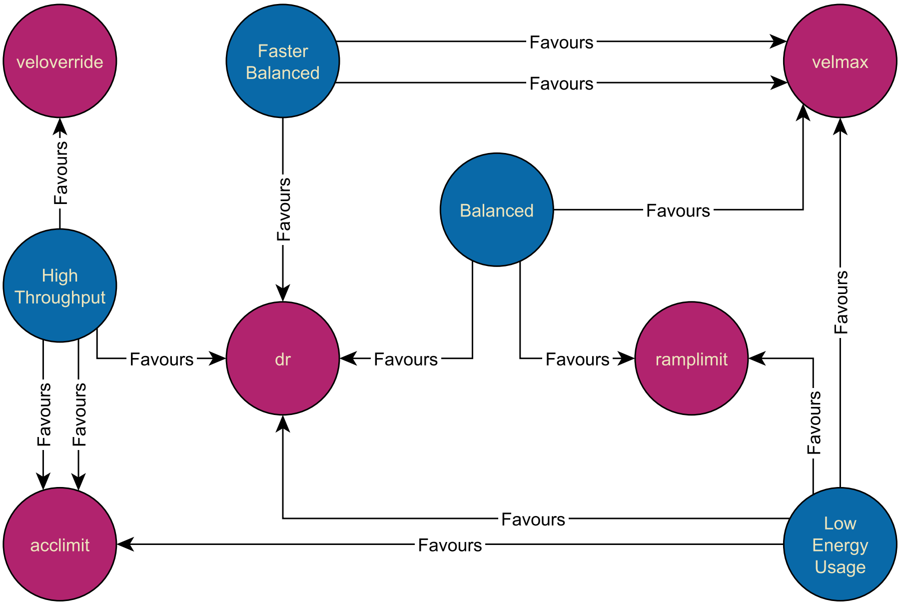

Figure 3. Simplified subgraph of the knowledge graph showing the relations of stereotypical users to the variables in their preferred decision rules.

References

[1] Smedberg H, Bandaru S. Interactive Knowledge Discovery and Knowledge Visualization for Decision Support in Multi-Objective Optimization. European Journal of Operational Research. 2023 May;306(3):1311-29. doi:10.1016/j.ejor.2022.09.008.

[2] Rosanty ES, Dahlan HM, Che Hussin AR. Multi-Criteria Decision Making for Group Decision Support System. In: 2012 International Conference on Information Retrieval & Knowledge Management; 2012. p. 105-9. doi:10.1109/InfRKM.2012.6205015.

[3] Koksalmis E, Kabak Ö. Deriving Decision Makers' Weights in Group Decision Making: An Overview of Objective Methods. Information Fusion. 2019 Sep;49:146-60. doi:10.1016/j.inffus.2018.11.009.

[4] Olson DL. Multi-Criteria Decision Support. In: Burstein F, W Holsapple C, editors. Handbook on Decision Support Systems 1: Basic Themes. International Handbooks Information System. Berlin, Heidelberg: Springer; 2008. p. 299-314. doi:10.1007/978-3-540-48713-5_15.

[5] Miettinen K. Nonlinear Multiobjective Optimization. vol. 12 of International Series in Operations Research & Management Science. Hillier FS, editor. Boston, MA: Springer US; 1998. doi:10.1007/978-1-4615-5563-6.

[6] Deb K, Pratap A, Agarwal S, Meyarivan T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 2002 Apr;6(2):182-97. doi:10.1109/4235.996017.

[7] Ayan B, Abacıoğlu S, Basilio MP. A Comprehensive Review of the Novel Weighting Methods for Multi-Criteria Decision-Making. Information. 2023 May;14(5):285. doi:10.3390/info14050285.

[8] van der Maaten L, Hinton G. Visualizing Data Using T-SNE. Journal of Machine Learning Research. 2008;9(86):2579-605.

[9] McInnes L, Healy J, Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv; 2020. doi:10.48550/arXiv.1802.03426.

[10] Kohonen T. Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics. 1982 Jan;43(1):59-69. doi:10.1007/BF00337288.

[11] Bishop CM, Svensén M, Williams CKI. GTM: The Generative Topographic Mapping. Neural Computation. 1998 Jan;10(1):215-34. doi:10.1162/089976698300017953.

[12] Trawiński K, Chica M, Pancho DP, Damas S, Cordón O. moGrams: A Network-Based Methodology for Visualizing the Set of Nondominated Solutions in Multiobjective Optimization. IEEE Transactions on Cybernetics. 2018 Feb;48(2):474-85. doi:10.1109/TCYB.2016.2642886.

[13] Bandaru S, Ng AHC. Trend Mining: A Visualization Technique to Discover Variable Trends in the Objective Space. In: Deb K, Goodman E, Coello Coello CA, Klamroth K, Miettinen K, Mostaghim S, et al., editors. Evolutionary Multi-Criterion Optimization. Lecture Notes in Computer Science. Cham: Springer International Publishing; 2019. p. 605-17. doi:10.1007/978-3-030-12598-1_48.

[14] Smedberg H, Bandaru S, Ng AHC, Deb K. Trend Mining 2.0: Automating the Discovery of Variable Trends in the Objective Space. In: 2020 IEEE Congress on Evolutionary Computation (CEC); 2020. p. 1-8. doi:10.1109/CEC48606.2020.9185892.

[15] Sugimura K, Jeong S, Obayashi S, Kimura T. Kriging-Model-Based Multi-Objective Robust Optimization and Trade-off-Rule Mining Using Association Rule with Aspiration Vector. In: 2009 IEEE Congress on Evolutionary Computation; 2009. p. 522-9. doi:10.1109/CEC.2009.4982990.

[16] Sugimura K, Obayashi S, Jeong S. Multi-Objective Optimization and Design Rule Mining for an Aerodynamically Efficient and Stable Centrifugal Impeller with a Vaned Diffuser. Engineering Optimization. 2010 Mar;42(3):271-93. doi:10.1080/03052150903171084.

[17] Bandaru S, Ng AHC, Deb K. Data Mining Methods for Knowledge Discovery in Multi-Objective Optimization: Part B - New Developments and Applications. Expert Systems with Applications. 2017 Mar;70:119-38. doi:10.1016/j.eswa.2016.10.016.

[18] Agrawal R, Srikant R. Mining Sequential Patterns. In: Proceedings of the Eleventh International Conference on Data Engineering; 1995. p. 3-14. doi:10.1109/ICDE.1995.380415.

[19] Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules in Large Databases. In: Bocca JB, Jarke M, Zaniolo C, editors. VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile. vol. 1215 of VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1994. p. 487-99.

[20] Bandaru S, Deb K. Towards Automating the Discovery of Certain Innovative Design Principles through a Clustering-Based Optimization Technique. Engineering Optimization. 2011 Sep;43(9):911-41. doi:10.1080/0305215X.2010.528410.

[21] W Holsapple C. DSS Architecture and Types. In: Burstein F, W Holsapple C, editors. Handbook on Decision Support Systems 1: Basic Themes. International Handbooks Information System. Berlin, Heidelberg: Springer; 2008. p. 163-89. doi:10.1007/978-3-540-48713-5_9.

[22] Pick RA. Benefits of Decision Support Systems. In: Burstein F, W Holsapple C, editors. Handbook on Decision Support Systems 1: Basic Themes. International Handbooks Information System. Berlin, Heidelberg: Springer; 2008. p. 719-30. doi:10.1007/978-3-540-48713-5_32.

[23] Gray P. The Nature of Group Decision Support Systems. In: Burstein F, W Holsapple C, editors. Handbook on Decision Support Systems 1: Basic Themes. International Handbooks Information System. Berlin, Heidelberg: Springer; 2008. p. 371-89. doi:10.1007/978-3-540-48713-5_19.

[24] DeSanctis G, Gallupe RB. A Foundation for the Study of Group Decision Support Systems. Management Science. 1987 May;33(5):589-609. doi:10.1287/mnsc.33.5.589.

[25] Definition of Terms and Context of the Design Study. In: Bui TX, editor. Co-oP: A Group Decision Support System for Cooperative Multiple Criteria Group Decision Making. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer; 1987. p. 7-24. doi:10.1007/3-540-18753-7_42.

[26] Odu GO. Weighting Methods for Multi-Criteria Decision Making Technique. Journal of Applied Sciences and Environmental Management. 2019 Sep;23(8):1449-57. doi:10.4314/jasem.v23i8.7.

[27] Elnagar S, Weistroffer HR. Introducing Knowledge Graphs to Decision Support Systems Design. In: Wrycza S, Maślankowski J, editors. Information Systems: Research, Development, Applications, Education. Lecture Notes in Business Information Processing. Cham: Springer International Publishing; 2019. p. 3-11. doi:10.1007/978-3-030-29608-7_1.

[28] Min W, Liu C, Xu L, Jiang S. Applications of Knowledge Graphs for Food Science and Industry. Patterns. 2022 May;3(5):100484. doi:10.1016/j.patter.2022.100484.

[29] Ehrlinger L, Wöß W. Towards a Definition of Knowledge Graphs. In: Martin M, Cuquet M, Folmer E, editors. Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16), Leipzig, Germany, September 12-15, 2016. vol. 1695/2016 of CEUR Workshop Proceedings. Leipzig, Germany: CEUR-WS.org; 2016. .

[30] Fensel D, Şimşek U, Angele K, Huaman E, Kärle E, Panasiuk O, et al. Introduction: What Is a Knowl-

edge Graph? In: Fensel D, Şimşek U, Angele K, Huaman E, Kärle E, Panasiuk O, et al., editors. Knowledge Graphs: Methodology, Tools and Selected Use Cases. Cham: Springer International Publishing; 2020. p. 1-10. doi:10.1007/978-3-030-37439-6_1.

[31] Gutierrez C, Sequeda JF. Knowledge Graphs. Communications of the ACM. 2021 Feb;64(3):96-104. doi:10.1145/3418294.

[32] Gruber TR. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition. 1993 Jun;5(2):199-220. doi:10.1006/knac.1993.1008.

[33] Bandaru S, Ng AHC, Deb K. Data Mining Methods for Knowledge Discovery in Multi-Objective Optimization: Part A - Survey. Expert Systems with Applications. 2017 Mar;70:139-59. doi:10.1016/j.eswa.2016.10.015.