2014 47th Hawaii International Conference on System Science

A Multiple Criteria Decision Analysis (MCDA) Software Selection Framework

Yan Li
Virginia Commonwealth University
School of Business
liy26@vcu.edu

Manoj A. Thomas
Virginia Commonwealth University
School of Business
mthomas@vcu.edu

Abstract

With the gaining popularity of multiple criteria decision analysis (MCDA) among researchers and practitioners, a variety of software that implements sophisticated MCDA methods and techniques is now available. To address the issue of the missing methodological approach in MCDA software selection, especially the mismatch between the decision making situation (DMS) structuring and the preference structuring employed by the tools, we propose a framework to enable the decision maker to choose relevant MCDA software. This is the first attempt to formally model the MCDA software selection process. A comprehensive set of MCDA methods meta-data for software selection is identified and demonstrated using the example of a specific DMS.

1. Introduction

Classic decision making process models try to optimize a single objective function over a set of feasible solutions. However, it is often the case that the decision is related to a plurality of viewpoints (or decision criteria), many of which may be conflicting and need to be given due consideration. As a result, the decision often tends to be satisfactory rather than optimal. In order to explicitly take into consideration the pros and cons of the decision criteria, Multiple Criteria Decision Analysis (MCDA) has gained significant attention from researchers and practitioners (Figueira*, et al.*, 2004). The MCDA approach consists of four non-linear recursive steps: (1) decision problem structuring, (2) preferences articulation, (3) alternative evaluation aggregation, and (4) solution recommendations (Cho, 2003; Guitouni & Martel, 1998).

Many MCDA methods are proposed in the literature to solve MCDA problem, most of which require a considerable amount of computation. Correspondingly, MCDA software packages have been developed to implement these sophisticated methods and techniques. MCDA software covers various stages of the decision making process ranging from problem exploration and structuring to discovering the DM's preferences and the preferred compromise solution. The wide variety of MCDA software posits additional challenges for business users and researchers in choosing the right software to suit their decision making situations (DMS). In practice, the choice is often motivated by the familiarity and affinity with the tool and the MCDA method(s) implemented within. Thus, the DMS for the software selection is conditioned by the MCDA methodology employed in the tool. This can result in a mismatch between the process of structuring the actual DMS and the structuring and modeling techniques employed by the tool. For example, in a specific DMS, the DM may prefer direct rating for preference elucidation. However, the tool selected by the DM may only provide pairwise analysis. The mismatch may compel the DM to adapt the DMS to the undesirable method.

Currently, there exists no methodological approach for selecting the appropriate MCDA software based on the unique nature of the DMS. Selecting an improper MCDA tool for a specific application not only leads to wasted resources and time, but also the opportunity cost of responding to spurious results. To address this limitation, we propose a framework for MCDA software selection. Furthermore, we identify a comprehensive set of MCDA methods metadata for software selection based on the DMS. To the best of our knowledge, this is the first attempt to model a general framework for software selection in the MCDA domain.

2. Background

2.1 MCDA Methods Overview

978-1-4799-2504-9/14 $31.00 © 2014 IEEE
DOI 10.1109/HICSS.2014.141

In order to understand the fundamentals of MCDA process, three basic concepts need to be defined as follows (S. Greco, 2004):

(1) *Alternatives (or potential actions):* denote a set of potential actions that are worth some consideration for a given decision process.

(2) *Criteria (or family of criteria):* the first step in MCDA modeling is to build *n* criteria (n>1) so that each potential action according to a point of view can be evaluated and compared against. Different types of scales can be used in evaluating a criterion against a potential action based on the views of each DM. The types of scales are characteristics of input information from the DM, which are generally expressed as ordinal, cardinal, and mixed (see Appendix A for description).

(3) *Decision Problematic*: refers to the way in which a DMS is formulated. Roy (1985) suggested four possible decision problematic: the description problematic, the choice problematic, the ranking problematic, and the sorting problematic.

MCDA methods can be categorized into two groups: multiple criteria design methods and multicriteria evaluation methods (Cho, 2003). The first group of methods is designed to solve multiobjective optimization (MOO) problems, in which a finite number (for integer problems) or infinite number (for continuous problems) of alternatives are i*mplicitly* (assumed to exist but is otherwise unknown) known. The alternatives are defined by a finite number of constraints (criteria) that can be expressed in the form of linear or nonlinear mathematical objective functions. The MOO methods are sometimes referred as continuous MCDA methods. Hwang and Masud (1979) provide a systematic classification of MOO methods into four categories based on preference articulation: (1) no preference articulation, (2) a priori preference articulation, (3) a posteriori preference articulation, and (4) interactive method.

The second group of methods is designed to solve multiple attribute decision analysis (MADA) problems, in which a finite number of *explicitly* known alternatives are characterized by a set of multiple attributes. The MADA methods are sometimes referred as discrete MCDA methods. Vincke (1989) characterized MADA methods into three categories: (1) the multiattribute utility theory (MAUT) methods, (2) the outranking methods, and (3) the interactive methods. Guitouni and Martel (1998) categorize MADA methods into three similar categories: (1) the single synthesizing criterion approach, (2) the outranking synthesizing approach, and (3) the interactive local judgments with trial-and-error approach. The characterization is based on the multicriteria aggregation procedures (MCAP), which are considered as the heart of MADA methods (Guitouni, et al., 1998). Table 1 lists a set of known MADA methods that are implemented in MCDA software.

Table 1: MADA Methods

MCAP Type	**MADA Methods**
Elementary Methods	Weighted Sum (Churchman & Ackoff, 1954)
Single Synthesizing Criterion	*MAUT* (Keeney & Raiffa, 1976), *MAVT* (Multi attribute value theory) (Edwards, 1977), *TOPSIS* (Hwang & Youn, 1981), *SMART* (Von Winterfelt & Edwards, 1986), *UTA* (UTilités Additives) (Jacquet-Lagrèze & Siskos, 1982), *AHP* (Analytic Hierarchy Process) (Saaty, 1980), *PAPRKA* (Hansen & Ombler, 2008), *DELTA* (Danielson & Ekenberg, 1998), *Bayesian analysis* (Newman, 1971), *DRSA* (S. Greco*, et al.*, 1999; Salvatore Greco*, et al.*, 2001)
Outranking Synthesizing Approach	*ELECTRE* family (B. Roy, 1973), *PROMETHEE* (J. P. Brans, 1982) and *PROMETHEE-GAIA* (J.-P. Brans & Mareschal, 1994), NAIDE (Munda, 1995)
Interactive Approach	MACBETH (Bana e Costa & Vansnick, 1999), Cognitive mapping

The diversity of MCDA methods posits a crucial drawback in the MCDA software selection. It is impossible to decide whether one method is better than the other method in a specific problem situation (French, 1993). Guitouni and Martel (1998) proposed seven tentative guidelines to help choose appropriate MCDA methods applicable to a particular DMS. The guidelines provide a first step to characterize the MCDA methods. However, it falls short in providing a methodological approach to map the DMS structure to these characteristics. The authors also acknowledge the need to integrate the method selection process into a decision support system (DSS) so that recommendations of applicable methods can be provided to the DM. The framework proposed in this paper enables the mapping between the DMS and the MCDA methods. It provides a methodological approach that can be integrated in a DSS.

2.2 MCDA Software

Decision analysis software can assist DMs at various stages of structuring and solving decision problems. These stages include problem exploration and formulation, decomposition, and preference and tradeoff judgments. Weistroffer et al. (2005) surveyed both commercial and academic MCDA software solutions based on seven different problem types. Even though the survey provides a comprehensive set of available MCDA software, it does not provide a general guideline on how to select MCDA software based on a specific DMS. Many of software packages from that survey have since been discontinued or not currently supported. Weistroffer and Li (2013) provide an updated review of the state of MCDA software. It is structured around several decision considerations when searching for appropriate available software. They provide an initial pool of candidate MCDA software packages. Poles et al. (2008) reviewed MOO software available since 1999, focusing on the tools and features that that the MOO software should contain. Both reviews identify some criteria related to MCDA software selection, though neither provides a formalized approach for MCDA software selection.

From an alternate viewpoint, an Analytic Hierarchy Process (AHP) based software application (Seixedo & Tereso, 2010) was constructed for selecting MCDA software. In this study, MCDA tools are presented similar to the approach used by Weistroffer et al. (2005), and the criteria are specified similar to the simulation software selection criteria (Banks, 1991). One limitation of this study is that the MCDA-specific selection criteria are not considered in software selection (for example, the MCDA methods implemented). Secondly, the software selection criteria evaluation is limited to AHP, a single synthesizing criterion approach, which is only one type of MCAP (Table 1). If a DM prefers the outranking synthesizing approach or interactive approach to model the MCAP, he or she might not want to use AHP when evaluating the MCDA software. Thus, a more general MCDA software selection framework is needed that can articulate the decision preferences from the DM and map the preferences in the selection result evaluation.

2.3 Software Selection

The growing availability of off-the-shelf software packages require specific considerations in software selection, especially in how to fit the customer's requirements to the software selection process. An improper selection of software package can be costly and may adversely affect business processes. Literature in software selection mainly focuses on software selection methodology, software selection criteria, and/or software evaluation techniques (Jadhav & Sonar, 2009). Different types of software have been considered in the software selection literature, such as accounting software systems (Adhikari*, et al.*, 2004), simulation software (Cochran & Chen, 2005), ERP systems (Wei*, et al.*, 2005), COTS (Commercial off-the-shelf) products (Leung & Leung, 2002), data mining software (Collier*, et al.*, 1999), etc. However, very little research has been done in the area of MCDA software selection.

Table 2: Generic Software Selection Methodology

Step	Description
1	Initial investigation of the available software
2	Short listing of candidate packages
3	Eliminating software that do not have required features
4	Use an evaluation technique to evaluate the remaining software
5	Pilot testing the tool in an appropriate environment by obtaining trial copy
6	Negotiating a contract
7	Purchasing and implementing

During the software selection process, methodological approaches are needed to demonstrate software features, issues, and processes that should be taken into consideration. Jadhav, et al. (2009) proposed a 7-step generic selection methodology based on a comprehensive review of software selection literature (refer Table 2). However, the methodology should only serve as a guideline and not intended to be followed without any deviation. It should be adapted based on the requirements of the individual DM or the organization (Patel & Hlupic, 2002).

Software evaluation itself falls into the domain of MCDA, where the DM makes preference decisions over the available alternatives (candidate software packages) based on a set of selection criteria. AHP and weighted sum are two popular MCDA methods applied in the evaluation of software packages (Jadhav, et al., 2009). However, as discussed earlier, the software evaluation technique should also map the DM's decision preferences. To best of our knowledge, the preference modeling has not been considered in the MCDA software selection and evaluation literature. A framework for MCDA software selection should not be limited to a specific MCDA technique in the evaluation process. Instead, it should provide recommendations on relevant techniques and let the DM choose the one that fits the decision situation.

Related literature provides a hierarchical list of software selection criteria. Selection criteria are compared with different types of requirements from the DMs, such as managerial, political, and software quality requirements (Franch & Carvallo, 2003). While managerial and political requirements are often subjective and unique to the individual organization, quality requirements can be standardized. The ISO and ISO/IEC standards related to software quality include the families of 9126 and 14598, within which ISO/IEC 9126-1 specifically defines a quality model for software evaluation. The 9126-1 quality model defines general software characteristics and the different sub-characteristics. Each sub-characteristic can be further decomposed into measurable software attributes. For a given software domain, a structured quality model can provide a taxonomy of software evaluation criteria and metrics for computing their values (Franch, et al., 2003). The software evaluation quality model can be integrated into a DSS to evaluate the candidate software as described in the software evaluation methodology (step 4 in Table 2).

3. MCDA Software Selection Framework

In this section, we propose a general framework (Figure 1) for MCDA software selection based on the DM's specific decision context. The framework only includes steps that can be automated or semi-automated in the aforementioned general software selection methodology (step 1 through 4 in Table 2). Our framework consists of three stages: (1) building a MCDA software knowledge base, which would provide an initial pool of MCDA software packages; (2) shortlisting candidate software packages by structuring and modeling a specific DMS from the DM and selecting only software packages that implement the appropriate MCDA methods; (3) providing software recommendations by using the DM preferred MCAP evaluation techniques. In the following section, we provide a detailed discussion of each stage.

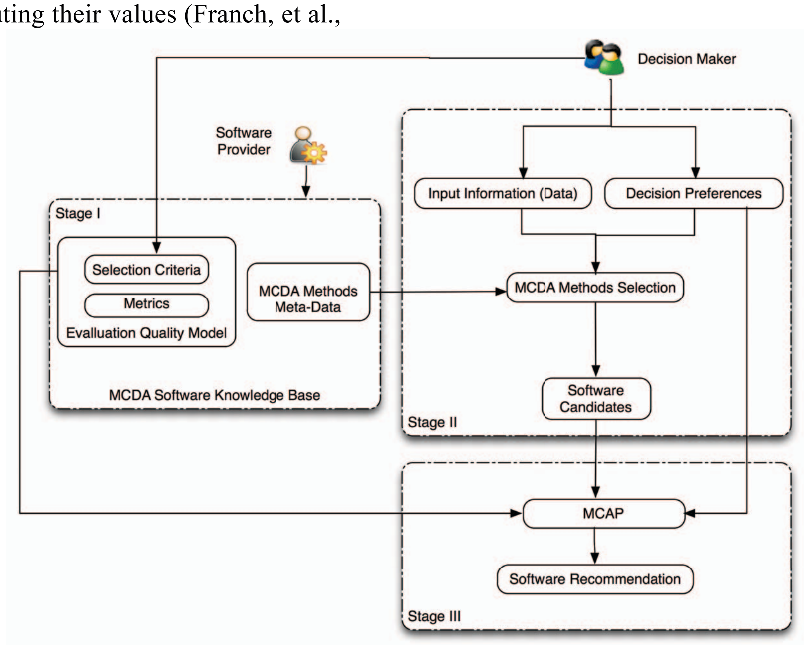

Figure 1: MCDA Software Selection Framework

3.1 *Stage I*: Building MCDA Software Knowledge Base

The first stage of software selection requires the creation of an initial pool of software packages (step 1 in Table 2). A knowledge base of available MCDA software packages and their meta-data is currently unavailable. The MCDA software survey (Weistroffer & Li, 2013) provides a baseline for building such a knowledge base. The MCDA software knowledge base constitutes of two parts: a MCDA software meta-data model and a software evaluation quality model. In this study, we assume that a MCDA-specific software evaluation quality model is available, through which the MCDA software selection criteria and their matrices are captured and stored in the knowledge base. The software selection criteria can be either vendor supplied that can be objectively assessed (e.g., platforms support, license cost, etc.) or DM selected that needs to be subjectively scored (e.g., ease of use, adaptability, completeness, etc.)

A wiki for MCDA software is created, and a meta-data schema for the MCDA software is stored in XML format. The XML schema aggregates the software selection criteria characteristics and sub-characteristics defined by the ISO/IEC 9126-1, and the meta-data for MCDA methods (e.g., information input, elucidation mode, decision problematic, and alternative aggregations evaluation, etc.) Figure 2 shows a snippet of MCDA methods meta-data stored in XML format.

```xml
<MCDAMethod>
  <AHP>
    <Input>
      <inputDataScale>cardinal</inputDataScale>
      <criteria>true</criteria>
      <alternative>explicit</alternative>
    </Input>
    <PreferenceModeling>
      <elucidationMode>pairwise comparison</elucidationMode>
      <momentsOfElucidation>a priori</momentsOfElucidation>
      <preferenceStructure>preference
      </preferenceStructure>
      <preferenceStructure>incomparability</preferenceStructure>
      <altrnativeOrdering>totalPreorder</altrnativeOrdering>
      <decisionProblematic>choice</decisionProblematic>
      <decisionProblematic>ranking</decisionProblematic>
    </PreferenceModeling>
    <Aggregation>
      <alternativeEvaluation>partiallyCompensatory</alternativeEvaluation>
    </Aggregation>
  </AHP>
  <ELECTRE_I>
  </ELECTRE_I>
</MCDAMethod>
```

Figure 2: MCDA Method (AHP) Meta-data

Once the XML schema is defined and standardized, all required meta-data for software selection can be populated by the software provider and stored in the knowledge base. Since new MCDA methods and tools are frequently introduced and others may be outdated, it is impossible for a single entity to maintain and update the MCDA software knowledge base. In future, the knowledge base can be maintained as a collaborative effort from the MCDM community. A web-based environment is suitable for such a collaborative effort. XML-based schema representation and knowledge storage makes it easy to build a web-application to add new software meta-data or query the knowledge base, as well as for the implementation of MCDA software selection framework on the web.

3.2 *Stage II*: Shortlist Candidate Software based on a specific DMS

In order to shortlist a set of candidate software packages from the software pool, the first step is to explicitly model the DMS based on the DM's preferences and determine the appropriate MCDA methods based on these preferences. Hence, a systematic approach to elicit the preferences from the DM to determine appropriate MCDA methods is needed. Guitouni et al. (1998) presented a set of guidelines to help choose appropriate MCDA methods. Drawing upon these guidelines, stage II comprises of three steps (described below) to elicit the preferences from the DM, which can be mapped to the appropriate MCDA methods.

***Step II.a: Input Data*:** The input capability of a MCDA method concerns the information accepted (i.e. cardinal or nominal, ambiguous or unambiguous, uncertain or certain), the criteria (i.e., true criteria, semi-criteria, pre-criteria, pseudo criteria), and the alternatives (i.e. implicit or explicit). For example, if the DM defines the input data as ordinal, then the utility-based methods such as MAVT, SMART & AHP are not applicable. If the alternatives are expressed as implicit known, the DMS is a multi-criteria design problem and hence only MOO software packages can be included.

***Step II.b: Decision Preferences*:** This step elicits the DM's decision preferences towards the MCAP, which include preference modeling and aggregation evaluation. The elicitation process is driven by the guidelines and choices summarized in Appendix A. Regardless of the output of *step II.a*, there are four different preference elucidation modes: (1) tradeoffs, (2) direct rating, (3) lotteries, and (4) pairwise comparison. The DM can select one or more of the preferred elucidation modes, which also feeds into *Stage III.* Furthermore, there are three types of the moments of elucidation: a priori, progressive, or a posterior. All existing MADA methods have a priori moments of elucidation. If the output of *Step II.a* indicates a multi-criteria design problem, four preference articulation categories will be presented (see previous discussion on MOO methods classification): (1) no preference, (2) a priori preference, (3) a posteriori preference, and (4) interactive method. If the output of of *Step II.a* indicates a MADA problem, five preference structure will be presented: (1) indifference, (2) preference, (3) weak preference, (4) incomparability, and (5) outranking. The DM can select one or more of the preferred structure. In addition, there are four different decision problematic: (1) description, (2) choice, (3) sorting, and (4) ranking, where the DM can select one or more based on the DMS. Finally, the alternative aggregation evaluation includes three different types of compensation logic: compensatory, non-compensatory, and partially compensatory.

Step II.c: MCDA methods selection: The output of *Step II.a and Step II.b* are mapped to the MCDA methods metadata in *Stage I.* The MOO software packages are characterized by the preference articulation modes. If the output of *Step II.a* refers to a multi-criteria design DMS, then the next step will be to map the preference Elucidation mode from *Step II.b* with the preference articulation mode in MOO software metadata, and retrieve the matching software. If the output of *Step II.a* refers to a MADA DMS, the captured DM's decision preference (e.g., the case example described in the next section) will be mapped to the MCDA-method metadata model in the knowledge base, and a shortlist of MCDA software that implement the applicable method(s) will be provided.

3.3 *Stage III*: Software Recommendation

Once the candidate software packages are generated from the knowledge base, the DM can review the vendor supplied MCDA software selection criteria and their measures. Software that do not have the required objective features can be eliminated (step 3 in Table 2). The MCDA software knowledge base integrated in the DSS can then prompt the DM to determine the subset of criteria that require further evaluations. For example, some academic researchers may consider the software cost as a more influential factor, and less concerned with the efficiency. Once the set of software evaluation criteria is selected, the DM can then proceed to further investigate the software.

4. Case Study

In this section, we present a case study of how our framework can augment real estate websites for finding a home to buy, rent or sell. Numerous websites such as zillow.com, realtor.com, etc. provide buyers and sellers with relevant information pertaining to houses such as listing price, description, payment estimates, pricing history, tax history, and neighborhoods. Finding houses or rentals are a classic case of MCDA problem, where the comprehensive assessment of houses is conducted against a set of selection criteria. Currently, no websites provide any form of MCDA support for buyers. A website that provides MCDA support would effectively attract more potential customers.

The first step in providing such a support is to investigate existing MCDA solutions that can be embedded into the company's website. The company may use our framework to carry out this initial investigation. The decision maker from the company will first provide the characteristics and the decision preferences that are listed in Appendix A.

For example, in the real estate website case, the input data scales are cardinal (such as listing price, number of bedrooms, etc.), the criteria have absolute discriminating power, and it is a MADA problem given that there are finite number of candidate houses to be compared. A preferred alternative comparison mode (preference elucidation mode) in this case is to compare houses in pairs, which is elucidated a priori. The DM's preference between alternatives is modeled to be either strictly preferred or indifference. That means for a given criterion (such as house price), where "a" and "b" are respective price values for two alternatives (houses A and B); if $a > b$, then house A is strictly preferred; if $a < b$, then house B is strictly preferred; and if $a=b$, the house A and B are considered indifference. The decision problematic is modeled as either a ranking or choice problem, and a total preorder of the alternatives is preferred. The preferred alternative aggregation evaluation mode is partial compensatory, which means there is some compensation accepted between the different criteria. For example, if a house has a good open floor plan, it may compensate for the smaller square footage. However, the number of bedrooms is a criterion that may be not compensated at all. Figure 3 shows the preference modeling with the DM input screenshot.

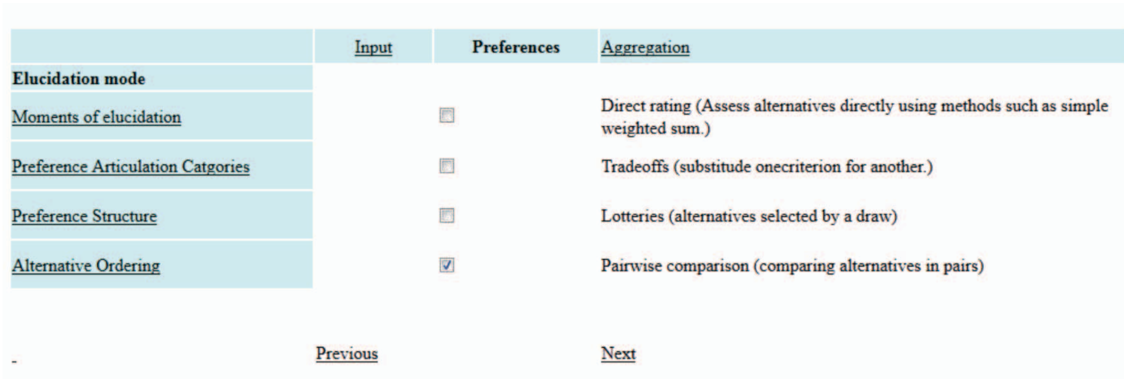

Figure 3: Preference Modeling Screenshot

The mapping of the decision preferences with the MCDA software meta-data model identifies a relevant MCDA method – AHP, which provides a systematic procedure to model MCDA problems in multilevel hierarchical structures and derives ratio scales from pairwise comparison of the hierarchical elements.

The mapping also provides a list of candidate software packages that implements AHP, as shown in Figure 4. A short description of each software is available and the DM can drilldown to review vendor provided evaluation criteria. The DM can review the software packages and exclude those that do not have the required features. For example, the DM can browse through the candidate software description (Figure 4) and recognize that *Priority Map* is an GIS-integrated application and *Triptych* is an excel-based application, both which are not desired in the real estate web application DMS. Thus, these two software will not be included in the software evaluation. For those software that require further investigation, the DM can obtain evaluation copies and use the set of software-specific selection criteria to evaluate the software. It is out of the scope of this paper to provide details steps of how to provide MCAP support in the MCDA software evaluation process. However, the evaluation technique used in the evaluation (step 4 in table 2) should match the result of decision preference modeling in *step II*.

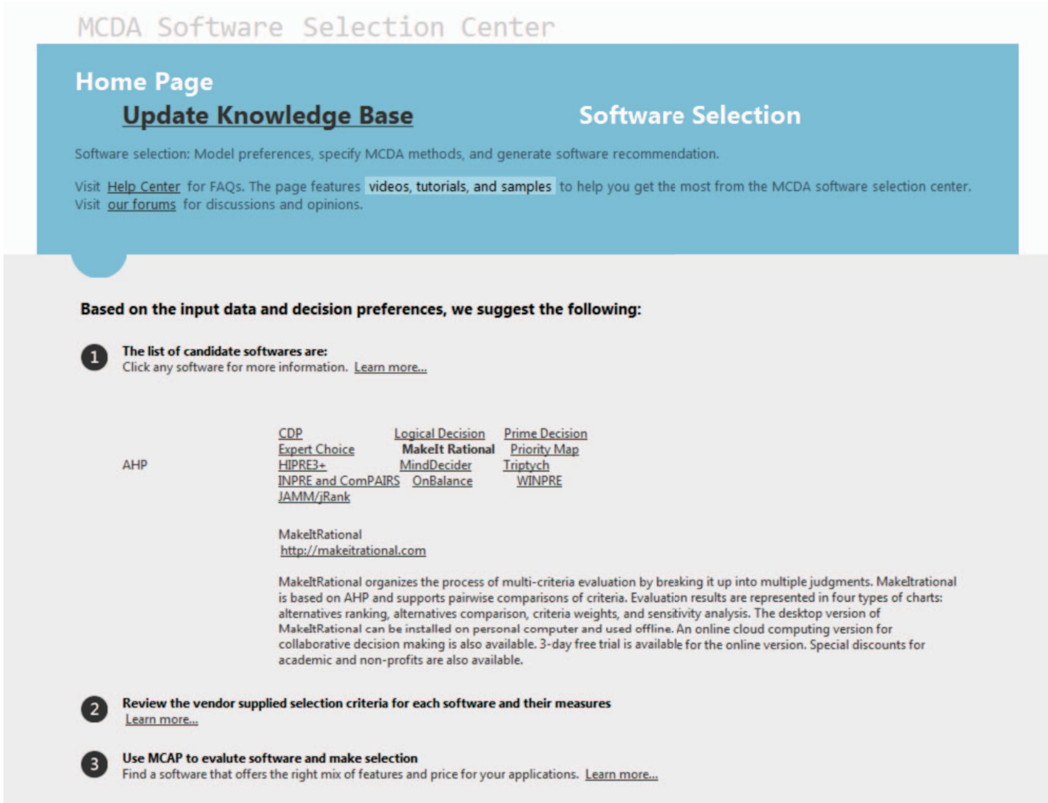

Figure 4: Screenshot of Candidates (Output of *Stage II*)

5. Conclusion and Future Directions

Increases in computing power have been at the heart of substantial growth in applications of MCDA. Several research streams have surveyed the state of art of the MCDA software (Poles, *et al.*, 2008; Seixedo & Tereso, 2010; Weistroffer, *et al.*, 2005), though none of them provide guidelines for MCDA software selection. The contributions of this research are two-fold. First, the framework proposed is the first attempt to address the need for decision support in the MCDA software selection process, which can be integrated with a DSS. It can also be integrated into the general software selection methodology to provide more diverse MCDA capabilities, which is currently limited to either simple weighted sum or AHP approach. Second, a variety of sophisticated MCDM methods proposed in the literature have mostly been implemented only on an *ad hoc* basis to solve a specific problem situation (Weistroffer, et al., 2005). To best of our knowledge, this research is the first methodological approach to provide decision support in selecting appropriate MCDA methods based on a specific DMS.

Currently, a web-based MCDA software selection center is under development. The software selection center provides two web frontends. The first is for updating MCDA software knowledge base, which requires a collaborative effort from the academic and commercial software vendors in the MCDA community. The second caters to DMs who are interested in integrating MCDA solutions into their applications. Future research directions will address evaluating the MCDA software selection center. Furthermore, a MCDA-specific software quality model needs to be developed and integrated to provide a comprehensive decision support capability in the MCDA software selection center.

6. References

Adhikari, A., Lebow, M. I., & Zhang, H. (2004). Firm characteristics and selection of international accounting software. *Journal of International Accounting, Auditing and Taxation, 13*(1), 53-69.

Bana e Costa, C. A., & Vansnick, J.-C. (1999). The MACBETH approach: Basic ideas, software, and an application. In N. Meskens & M. Roubens

(Eds.), *Advances in Decision Analysis* (pp. 131-157). Dordrecht, Netherlands: Kluwer Academic Publishers.

Banks, J. (1991). *Selecting simulation software*. Paper presented at the Proceedings of the 23rd conference on Winter simulation.

Brans, J.-P., & Mareschal, B. (1994). The PROMCALC & GAIA decision support system for multicriteria decision aid. *Decision Support Systems, 12*(4–5), 297-310.

Brans, J. P. (1982). *L'ingénierie de la décision: élaboration d'instruments d'aide à la décision. La méthode PROMETHEE.* In Proceedings of the L'aide à la décision: Nature, Instruments et Perspectives d'Avenir, Québec, Canada.

Buckshaw, D. (2010). Decision Analysis Software: 10th biennial survey. *OS/MS Today, 37*(5).

Cho, K. T. (2003). Multicriteria decision methods: An attempt to evaluate and unify. *Mathematical and Computer Modelling, 37*(9–10), 1099-1119.

Churchman, C. W., & Ackoff, R. L. (1954). An approximate measure of value. *Operations Research, 2*(2), 172-187.

Cochran, J. K., & Chen, H.-N. (2005). Fuzzy multi-criteria selection of object-oriented simulation software for production system analysis. *Computers & operations research, 32*(1), 153-168.

Collier, K., Carey, B., Sautter, D., & Marjaniemi, C. (1999). *A methodology for evaluating and selecting data mining software.* In Proceedings of the System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on.

Danielson, M., & Ekenberg, L. (1998). A framework for analysing decisions under risk. *European Journal of Operational Research, 104*(3), 474-484.

Edwards, W. (1977). How to use multiattribute utility measurement for social decision making. *IEEE Transactions on Systems, Man and Cybernetics, 7*(5), 326-340.

Figueira, J., Greco, S., & Ehrgott, M. (2004). *Multiple criteria decision analysis: state of the art surveys* (Vol. 78). New York: Springer.

Franch, X., & Carvallo, J. P. (2003). Using quality models in software package selection. *IEEE Software, 20*(1), 34-41.

Greco, S. (2004). *Multiple criteria decision analysis: state of the art surveys* (Vol. 78): Springer.

Greco, S., Matarazzo, B., & Slowinski, R. (1999). *The use of rough sets and fuzzy sets in MCDM.* Dordrecht, Boston: Kluwer Academic Publishers.

Greco, S., Matarazzo, B., & Slowinski, R. (2001). Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research, 129*(1), 1-47.

Guitouni, A., & Martel, J.-M. (1998). Tentative guidelines to help choosing an appropriate MCDA method. *European Journal of Operational Research, 109*(2), 501-521.

Hansen, P., & Ombler, F. (2008). A new method for scoring additive multi-attribute value models using pairwise rankings of alternatives. *Journal of Multi-Criteria Decision Analysis, 15*(3-4), 87-107.

Hwang, C. L., & Masud, A. S. (1979). *Multiple objective decision making, methods and applications*. Berlin/New York: Springer-Verlag.

Hwang, C. L., & Youn, K. (1981). *Multiple Attribute Decision Making - Methods and Application: A State of the Art Survey*. New York: Springer.

Jacquet-Lagrèze, E., & Siskos, J. (1982). Assessing a set of additive utility functions for multicriteria decision-making: The UTA method. *European Journal of Operational Research 10*(2), 151-164.

Jadhav, A. S., & Sonar, R. M. (2009). Evaluating and selecting software packages: A review. *Information and Software Technology, 51*(3), 555-563.

Keeney, R. L., & Raiffa, H. (1976). *Decisions with Multiple Objectives*. New York: John Wiley and Sons.

Leung, K. R., & Leung, H. K. (2002). On the efficiency of domain-based COTS product selection method. *Information and Software Technology, 44*(12), 703-715.

Munda, G. (1995). *Multicriteria Evaluation in a Fuzzy Environment*. Heidelberg: Physica-Verlag.

Newman, J. W., (1971). (1971). *Management Applications of Decision Theory*. New York: Harper & Row.

Patel, N., & Hlupic, V. (2002). *A methodology for the selection of knowledge management (KM) tools.* In Proceedings of the 24th International Conference on Information Technology Interfaces (ITI), Cavtat, Croatia.

Poles, S., Vassileva, M., & Sasaki, D. (2008). Multiobjective Optimization Software. In J. Branke, K. Deb, K. Miettinen & R. Slowinski (Eds.), *Multiobjective Optimization* (Vol. 5252, pp. 329-348). Berlin / Heidelberg: Springer.

Roy, B. (1973). How outranking relation helps multiple criteria decision making. In J. Cochrane & M. Zeleny (Eds.), *Topics in Multiple Criteria Decision Making* (Vol. 1973, pp. 179-201): University of South Carolina Press.

Roy, B. (1985). *Méthodologie multicritere d'aide ala décision* Paris: Économica.
Saaty, T. L. (1980). *The Analytic Hierarchy Process*. New York: McGraw-Hill.
Seixedo, C., & Tereso, A. P. (2010). *A multicriteria decision aid software application for selecting MCDA software using AHP*. Paper presented at the 2nd International Conference on Engineering Optimization.
Vincke, P. (1989). *L'aide Multicritère a la Décision*: Editions de l'Université e de Bruxelles.
Von Winterfelt, D., & Edwards, W. (1986). *Decision Analysis and Behavioral Research*. Cambridge Cambridgeshire and New York: Cambridge University Press.
Wei, C.-C., Chien, C.-F., & Wang, M.-J. J. (2005). An AHP-based approach to ERP system selection. *International Journal of Production Economics, 96*(1), 47-62.
Weistroffer, H., & Li, Y. (2013). Multiple Criteria Decision Support Software *Multiple Criteria Decision Analysis: State of the Art Surveys - Forthcoming*. New York: Springer
Weistroffer, H., Smith, C., & Narula, S. (2005). Multiple Criteria Decision Support Software *Multiple Criteria Decision Analysis: State of the Art Surveys* (Vol. 78, pp. 989-1009): Springer New York.

Appendix A

Steps	**Guideline**	**Choice**	**Description**	***Case study example***
Input (decision problem structuring)	Input Data Scales	Ordinal	The gap between two degrees does not have a clear meaning	Cardinal (price, number of bedrooms, etc.)
		Cardinal	The ratio between two degrees can receive a meaning	
		Mixed	Both ordinal and cardinal	
	Criteria*	True criteria	Either indifference or strict preferences	True Criteria
		Pre-criteria	Either strict or weak preference, no indifference	
		Pseudo criteria	A gradation of preference	
	Alternatives	Implicit	MOO problem	MADA (finite number of candidate houses)
		Explicit	MADA problem	
Preference Modeling	Elucidation Mode	Direct Rating	Directly assess the alternatives	Pairwise Comparison (The houses are compared in pairs)
		Tradeoffs	One criterion can substitute another	
		Lotteries	Assess by a draw	
		Pairwise comparison	Assess alternatives in pairs	
	Moments of Elucidation	A Priori, Progressive, A posterior,	The preference is elucidated either a priori, or progressively, or a posterior.	A Priori (for all MADA methods)
	Preference Articulation Categories (MOO)	No, A priori, A posterior, Interactive methods	Either the preference is not articulated, or it is articulated a priori, a posterior, or interactively.	Not Applicable
	Preference Structure**	Indifference (a ***I*** b)	A is indifference to alternative B.	***(P, I)***
		Preference (a ***P*** b)	A is strictly preferred to B.	
		Weak preference (a ***Q*** b)	Hesitation between the indifference and preference.	
		Incomparability (a **R** b)	Hesitation between "A is preferred to B" and "B is	

			preferred to A".	
		Outranking (a ***S*** b)	$\boldsymbol{S} = (\boldsymbol{P} \cup \boldsymbol{Q} \cup \boldsymbol{I})$	
	Alternative Ordering*** (Binary relations between alternatives)	Partial	Binary relation has reflexivity, transitivity, and antisymmetry.	Total Preorder (All houses are comparable, and there are no uncertain cases)
		Weak	Binary relation has irreflexivity, asymmetry, transitivity, and transitivity of incomparability.	
		Semi	A special case of partial ordering with alternatives can be incomparable if their scores are within a given margin of error.	
		Total Preorder	A strong case of partial ordering with totality.	
	Decision Problematic	Description, Choice, Sorting Ranking	The DMS is formulated by description, choice, sorting, or ranking.	Ranking or Choice (the houses would be ranked in the result)
Aggregation	Alternative Aggregation Evaluation	Compensatory	Absolute compensation.	Partially compensatory (There are some compensation accepted between the different criteria)
		Non-compensatory	No compensation is accepted.	
		Partially compensatory	Some kind of compensation is accepted.	

* Please refer to reference (Roy 1985) for detailed definition.
** A and B are two alternatives, and "a" and "b" are their respective values for the criterion considered.
*** Please refer to classic mathematic *order theory* for detailed definition.