Mathematical and Computational Applications

MDPI

Article

An Interactive Recommendation System for Decision Making Based on the Characterization of Cognitive Tasks

Teodoro Macias-Escobar [1,2,*,†], Laura Cruz-Reyes [3,†], César Medina-Trejo [3,†], Claudia Gómez-Santillán [3,†], Nelson Rangel-Valdez [4,†] and Héctor Fraire-Huacuja [3,†]

[1] Department of Computing Science, Tijuana Institute of Technology, Av Castillo de Chapultepec 562, Tomas Aquino, Tijuana 22414, Mexico
[2] Departamento de Ingeniería Informática, Universidad de Cádiz, 11519 Puerto Real, Spain
[3] Graduate Program Division, Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Cd. Madero 89440, Mexico; lauracruzreyes@itcm.edu.mx (L.C.-R.); cesarmedinatrejo@gmail.com (C.M.-T.); Claudia.gomez@itcm.edu.mx (C.G.-S.); automatas2002@yahoo.com.mx (H.F.-H.)
[4] CONACyT Research Fellow at Graduate Program Division, Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Cd. Madero 89440, Mexico; nelson.rangel@itcm.edu.mx
* Correspondence: teodoro_macias@hotmail.com or teodoro.maciasescobar@alum.uca.es
† These authors contributed equally to this work.

check for updates

Citation: Macias-Escobar, T.; Cruz-Reyes, L.; Medina-Trejo, C.; Gómez-Santillán, C.; Rangel-Valdez, N.; Fraire-Huacuja, H. An Interactive Recommendation System for Decision Making Based on the Characterization of Cognitive Tasks. *Math. Comput. Appl.* **2021**, *26*, 35. https://doi.org/10.3390/mca26020035

Academic Editors: Marcela Quiroz, Juan Gabriel Ruiz, Luis Gerardo de la Fraga and Oliver Schütze

Received: 28 February 2021
Accepted: 20 April 2021
Published: 21 April 2021

Abstract: The decision-making process can be complex and underestimated, where mismanagement could lead to poor results and excessive spending. This situation appears in highly complex multi-criteria problems such as the project portfolio selection (PPS) problem. Therefore, a recommender system becomes crucial to guide the solution search process. To our knowledge, most recommender systems that use argumentation theory are not proposed for multi-criteria optimization problems. Besides, most of the current recommender systems focused on PPS problems do not attempt to justify their recommendations. This work studies the characterization of cognitive tasks involved in the decision-aiding process to propose a framework for the Decision Aid Interactive Recommender System (DAIRS). The proposed system focuses on a user-system interaction that guides the search towards the best solution considering a decision-maker's preferences. The developed framework uses argumentation theory supported by argumentation schemes, dialogue games, proof standards, and two state transition diagrams (STD) to generate and explain its recommendations to the user. This work presents a prototype of DAIRS to evaluate the user experience on multiple real-life case simulations through a usability measurement. The prototype and both STDs received a satisfying score and mostly overall acceptance by the test users.

Keywords: decision making process; cognitive tasks; recommender system; project portfolio selection problem; usability evaluation

1. Introduction

The decision-making process consists of selecting the best solution among a set of possible alternatives, considering difficult and complicated decisions [1]. Finding efficient strategies or techniques to aid this process is challenging due to the complexity of the problems.

In decision-making processes, such as the solution of optimization problems, the decision-maker (DM) is the person or group whose preferences are decisive for choosing an adequate solution to problems with multiple objectives (which are sometimes in conflict) and multiple efficient solutions [2]. The DM is the one who makes the final decision and chooses the solution that seems more appropriate from the preferences previously established.

There is a recent growing interest in using various techniques to incorporate the DM's preferences within a methodology, heuristic, or meta-heuristic to solve an optimization problem [3]. Among the different preference incorporation techniques available, using a weight vector that defines the importance of each objective is one of the most commonly used and accepted approaches.

The project portfolio selection (PPS) problem is a challenging optimization problem that presents several conditions to consider. First, these problems are usually multi-objective, searching for the best possible outcome for each objective. However, these objectives usually face conflicts between them based on the constraints that the problem sets. Second, the number of constraints that a PPS problem presents can make the decision-making process difficult since many possible solutions within the solution search space may not be feasible.

Usually, PPS problems define a limited number of resources to be distributed to improve each of the objectives while considering a maximum and minimum threshold of said resources for each of the elements defined in the constraints, limiting each objective's gain. Under these circumstances, it is most likely that it will not be possible to determine an optimal single solution, but instead, a set of optimal solutions that define a balance between the objectives of the problem and the DM's preferences, identified by using different strategies [4]. Therefore, it is crucial to select the most suitable solution that reflects the preferences of the DM. Multi-criteria decision analysis (MCDA) methods are among the most widely used tools for solving PPS problems because of their capacity to handle complex problems with multiple objectives (usually in conflict) to satisfy [5].

A practical methodology to solve PPS problems is the decision support system (DSS), which allows the DM to analyze a PPS problem under the current set of preferences and facilitate the decision-making process. However, choosing the best solution is a complex task because of the problem's subjective nature and the DM's preferences, which could be specific to a person or group and might change during the solution process. An interactive DSS allows the DM to show the best solutions based on the current preferences and receive new information from the DM and update its search to adapt to changes. As the name infers, this system can establish a user-system interaction during the solution process.

This paper proposes the Decision Aid Interactive Recommender System (DAIRS), a multi-criteria DSS (MCDSS) framework that considers integrating cognitive tasks to the user-system interaction. DAIRS is able to perform several tasks aiming to aid the DM during the decision-making process, such as evaluating alternatives, interacting with the DM, and recommending a solution while presenting arguments to justify this selection. The most relevant and novel feature DAIRS provides is that it not only is able to obtain information from the DM and adapt it to present an appropriate recommendation. This proposal can also present new information to DM or defend its current recommendation. In other words, DAIRS establishes a dialogue game with the user instead of only being a system that receives information.

This paper addresses the characterization of cognitive tasks involved in the decision support process and its integration in recommender systems to develop more robust DSS. These systems should allow the precise analysis of possible solutions, provide solutions that optimize the results, and at the same time, satisfy the preferences established by a DM. DAIRS includes on its MCDSS framework different MCDA methods supported by argumentation theory in the form of argumentation schemes and proof standards.

This proposal intends to present a recommender system that is able to provide a bidirectional interaction. Both the user and system provide and obtain new information based on the knowledge obtained during a dialogue. For this purpose, this work uses concepts related to argumentation theory, which allow both participants (user and system) to establish a well-structured dialogue.

DAIRS uses a bidirectional interaction under the assumption that the user will satisfactorily carry out a decision-making process even without extensive knowledge of the problem. DAIRS provides information to the DM through the dialogue game, seeking to enhance and accelerate learning about the problem to aid in selecting a suitable solution.

This work seeks to meet three main objectives. First, develop a recommender system, called DAIRS, which suggests a solution to a multi-objective optimization problem (MOP), precisely a PPS problem, with a deep interaction between the decision-maker (DM) and the system. Second, this work seeks to simplify the DM interaction with the proposed recommender system. Lastly, DAIRS endeavors to achieve high-level satisfaction of a DM. For the last objective, this proposal seeks to validate the developed recommender system, evaluating the effects of using argumentation theory and a bidirectional dialogue concerning several properties related to the usability of an MCDSS.

The main contributions of this work, proposed to meet the above objectives, can be summarized in three elements, whose originality is shown in Section 2 :

1. The development of an interactive MCDSS framework prototype, called DAIRS, supported by argumentation theory to perform a study of the effects of the proposed state transition diagrams (STDs), which regulates the flow of interaction with real users when solving a real-life optimization problem. For this paper, this work focuses on PPS problems.
2. The incorporation of argumentation schemes (reasoning patterns) and proof standards (MCDA methods to compare solutions) in an interactive MCDSS to evaluate and analyze their effects on the decision-making process.
3. The proposal, design, and implementation of two STDs, which determine the evolution of a dialogue game established between the DM and DAIRS.

The remaining part of this paper is structured as follows: Section 2 shows a brief review of works related to the proposal in this paper. Section 3 presents the necessary concepts on recommender systems employed in this work. Section 4 describes the proposed methodology and the developed prototype. Section 5 presents the experimental design and the results and analysis regarding the proposed prototype's performance when used to solve a test case study, which simulates a real-life scenario of a PPS problem. Finally, Section 6 addresses conclusions regarding the usability and effectiveness of the proposal and possible future work.

2. Related Work

This section reviews some of the most relevant works related to three topics in particular: (i) DSS frameworks used to solve PPS problems, (ii) interactive systems used for optimization problems, and (iii) proposals that use the characterization of cognitive tools to improve the interaction between the user and the recommender system.

2.1. DSS Frameworks Used to Solve PPS Problems

There are multiple DSS proposed focused on solving PPS problems. These works consider different strategies to incorporate the preferences established by the DM and select the most appropriate solution based on the current preference set or a preference weight vector.

While the proposal presented on this paper focuses on the development of a system that performs a project portfolio selection reflecting the DM's preferences in the best possible form and is able to interact with the user by entering in a dialogue, this bidirectional interaction feature, to the best of the authors' knowledge, has not been considered for solving PPS problems. It is important to understand some of the most relevant approaches to solve this problem.

Chu et al. [6] presents one of the first DSS proposed to solve PPS problems. Their DSS presents an approach based on a cost/benefit model for research and development (R&D) project management. Their work considers monetary and time cost, as well as the probability of success of each project to determine the optimal sequence of R&D projects to execute. The impact of each element when performing a selection of a solution is defined by a pair of weight variables that allowed to define how relevant is for the user to save money or time. More recently, Hummel et al. proposed a DSS framework based on the *Measuring*

Attractiveness by a Categorical Based Evaluation Technique (MACBETH) approach [7] to solve R&D project portfolio management problems interactively [8].

Archer and Ghasemzadeh [9] propose a framework to design decision support systems to solve PPS problems. In their work, they attempt to simplify the PPS process in three main phases: strategic consideration, individual project evaluation, and portfolio selection. At each phase the users are free to select the techniques that they find the most suitable. This framework is used to develop a DSS named Project Analysis and Selection System (PASS) [10] which is able to perform tasks such as data entry, pre-screening, project evaluation, screening and optimization models without the involvement of the DM. PASS is used to solve successfully solve a single-objective PSS problem.

DSS frameworks have proven to be suitable alternatives for solving multi-objective PPS problems. Hu et al. [11] proposes a multi-criteria DSS (MCDSS) framework to solve PPS problems implementing the Lean and Six Sigma concepts [12] and considering the cost and benefit of each project. Their framework considers flexible weight vectors that can be modified during the solution process and the output is a Pareto optimal portfolio set which allows the DM to select the most adequate to their preferences.

Khalili-Damghani's work [13] shows how flexible frameworks developed to solve PPS problems can be. In this case, an evolutionary algorithm (EA) is combined with a data envelope analysis model (DEA) to create the structure of a fuzzy rule-based (FRB) system that measures the suitability of all available candidate project portfolios.

Mira et al. [14] evaluates the performance of a DSS framework by solving a real-life simulation of a PPS problem and comparing the cumulative controlled risk value obtained by the DSS with respect to the controlled risk value obtained by a manual-based portfolio selection method. The results show a 10% improvement of the DSS framework over manual-based selection.

Mohammed [15] proposes the use of various strategies to find appropriate solutions to PPS problems within fuzzy environments. For this purpose, his work relies on the Analytic Hierarchy Process (AHP) [16] and TOPSIS [17] adapted to work using fuzzy strategies, which use a set of vectors of relative criteria weights. In this case, the used strategies incorporate preferences in the decision-making process before the system proceeds to generate a recommendation.

Recently, DSS frameworks have proven to be an adequate alternative to solve PPS problems focused on sustainability. Dobrovolskiene and Tamosiuniene [18] propose integrating the analysis of a sustainability index of each project within a Markowitz risk-return scheme [19]. This incorporation aims to find better portfolios based on a risk-return assessment that at the same time considers the DM's responsibility towards the surrounding environment with a long-term focus on the well-being of society.

Debnath et al. [20] propose a DSS framework supported by a hybrid multi-criteria decision support method. This hybrid system combines strategies such as sensitivity analysis with grey-based Decision-Making Trial and Evaluation Laboratory [21] and Multi-Attributive Border Approximation area Comparison [22] to solve PPS problems focused on the development, quality, and distribution of genetically modified agricultural products considering sustainability under social, beneficial, and differential criteria.

Verdecho et al. [23] present another proposal focused on sustainability. In this case, an AHP is used to solve a PPS problem related to supply chains, whose objectives are focused on financial, environmental, and social sustainability. Their framework also seeks to optimize supply-related processes and customer satisfaction.

2.2. Interactive Systems for Optimization Problems

A common scenario when using DSS frameworks to solve any optimization problem, such as the PPS problem, is that while they present a solution based on preferences defined by a DM, the next step in the decision-making process is not considered. This step consists on determining the acceptance (or rejection) of the recommended solution by the DM, as well as updating the DM's preferences. The preferences of the DM may change during solution process. It must be considered that the DM may be a single agent or group whose preferences are susceptible to social, political or economical-related changes surrounding the problem to be solved. Therefore, it is desirable that the framework is able to adapt to a new preferences.

For this situation, it is advisable to use an interactive process between the DM and the recommender system for decision-making support. The proposal of this work focuses on a bidirectional interaction. This feature is not present in the PPS problem nor in the optimization problems mentioned in this subsection.

Miettinen et al. [24] present a study that focuses on solving multi-objective optimization problems (MOPs) while using an interactive system that allows the system and the user to exchange information. Their study mentions that there are three main stopping criteria for these systems: the DM accepts the solution, the DM stops the process manually or an algorithmic stopping criterion is reached. Also, according to their work and [25] the interactive process can be divided in two phases. First, a learning phase where the DM obtains knowledge regarding the problem. The second phase is the decision phase, where the system identifies the most suitable solution according to the current information and DM must accept or reject it.

The *Flexible and Interactive Tradeoff* method [26] is a proposed approach implemented into a DSS to solve MOPs. This proposal considers that it is easier for the DM to compare results from multiple alternatives based on a definition of strict preferences between criteria rather than on indifference. This approach considers that the DM needs to establish a preferential lexicographic order for all criteria.

The InDM2 algorithm [27] is a recent work that allows interaction between the DM and the recommender system. The DM initially establishes a reference point which reflects to reflect his/her preferences. During the solution process, InDM2 shows the user the best candidate solutions it has found that match the current preferences. The DM is able to accept the solutions obtained, wait for the system to provide new solutions or stop the process at any time. InDM2 also allows the DM to update the current reference point or propose a new one, allowing the system to obtain new information based on the DM's new preferences.

Azabi et al. [28] propose an interactive optimization framework supported by a low fidelity flow resolver and an interactive Multi-objective Particle Swarm Optimization (MOPSO) for the optimization of the aerodynamic shape design of aerial vehicle platforms. As InDM2, the DM can incorporate preferences before and during the solution process. This interaction allows their framework to define and update a region of interest, accelerating the process. The results of their experiments show that the interactive MOPSO outperforms a non-interactive MOPSO, proving that a constant user-system interaction can provide better results.

There are also interactive framework proposals focused on solving PPS problems. Strummer et al. [29] proposes an interactive framework using a strategy based on identifying Pareto optimal solutions to determine a set of optimal portfolios to present to the DM. The system first solves a PPS problem and presents the best solutions under the current preferences and constraints defined by to the DM. The DM can then interact with the system to determine its preference for a particular criterion or set new constraints.

A study in Nowak et al. [30] states that several frameworks presented in the literature assume that the DM has a high-level knowledge of the problem, the methods used to solve, and has a well-defined set of preferences. These assumptions are obviously not always true. Therefore, the recommender system has to be as user-friendly as possible and understand that the user might have little knowledge of the problem prior using the system. Their study also notes a lack of consideration of dynamic elements, such as changes in preferences or the problem environment. The authors propose a general structure for the development of frameworks to solve PPS problems. This structure considers the criteria to be evaluated, the data needed to evaluate the projects, an analysis and evaluation of the projects, as well as the construction of project portfolios. The proposed structure also is capable of obtaining new information from the DM during each iteration, allowing the DSS to adapt to every change and focusing their search towards the new preferences.

Interactive systems can use graphical visualization to support the decision-making process of optimization problems. The work of Haara et al. [31] performs a study to evaluate several interactive data visualization techniques to support the solution of multi-objective forest planning problems. The DM uses visual elements to ease the process of correctly identifying and defining his/her preferences. The authors mention that these interactive systems can be used for proposal management problems. PPS problems are management problems. Therefore, it possible to think that these proposals could work successfully for PPS problems.

The *Your Own Decision Aid* (YODA) framework is an interactive recommender system proposed in Kurttila et al. [32] to solve PPS problems. YODA focuses on working with DMs composed of multiple people, where each user defines his/her preferences and acceptable candidate projects. All available projects are separated in subsets based on the level of group acceptance that each project has. The projects with the highest level of acceptance will have priority when the system defines a candidate project portfolio. Each user is able to update their preferences or define a project acceptance threshold to allow rejected projects that are close enough to their acceptance standards to be considered to be acceptable alternatives.

2.3. Characterization of Cognitive Tools to Improve User-System Interaction

The interactive process between the user and the DSS should not be limited to a series of commands simulating a master-slave structure. The interaction requires the characterization of cognitive tasks to become an entity that not only receives information but also provides new knowledge of the problem to the user.

The problem of characterizing cognitive tasks in the decision support process has been addressed using different approaches. Some representative works on providing explanations to accompany a recommended solution in each interaction are shown in this section. Some works, described below, are based on argumentation to model human argumentation and dialogue processes. Their description includes limitations.

The proposals using artificial intelligence provide a recommendation by learning the user preferences for particular products. They do not seek to recommend a solution for an optimization problem. Instead, they use the AI methods to optimize the recommender systems (e.g., identify similar users). Other related works use queries to obtain information from the DM related to the currently presented solution, but they do not explain the result [33].

Labreuche [34] describes how to use the argumentation theory to perform a pairwise comparison between alternatives, using an MCDA method based on a weight vector. It establishes four different situations, which involve two candidate solutions x and y and a weight vector w for six different criteria representing the DM preferences. These situations use pairwise evaluations based on criteria weights to present an argument in favor or against the statement "x is preferred over y".

Ouerdane [35] extends Labreuche's work, presenting an approach to provide underlying reasons for supporting an alternative selected by a recommendation system. For the process of justification, the argumentation theory and decision support are combined with an established language to enable communication. It also proposed a hierarchical structure of argument schemes to decompose the decision process into steps whose underlying premises are made explicit, allowing identifying when the dialogue should incorporate the information into the dialogue with the DM. This structure has only been tested for a low-dimensional choice problem (CHP), where the decision options are known from the beginning. Said proposal analyzed the required elements to perform a dialogue game with the user, not only to defend an established recommendation by the system but also to obtain new preferences and statements provided by the DM. The new information obtained could change dialogue-related elements, leading the system to provide a new recommendation if necessary.

The work presented in Cruz-Reyes et al. [36] proposes a framework design for generating DSSs focused on a PPS problem by characterizing arguments and dialogue using argumentation theory and rough sets theory. The framework has a justification module for the recommended solution shown to the user; the justification is supported by argument schemes and decision rules generated with rough sets. The process starts by obtaining available preferential information provided by the DM and selecting the appropriate multi-criteria method to evaluate the available portfolios. After, it generates a recommendation and its justification interactively. This work focuses more on the decision rules generated through rough sets. The argumentation theory is a complementing element of the architectural design, and it is presented only in a conceptual form.

Sassoon et al. [37] use argumentation theory by using argumentation schemes within a chatbot. Their chatbot establishes a dialogue between the user and a DSS focused on medical consultation. It considers the user's symptoms, medical history, and a list of available treatments to recommend the most appropriate medication. This DSS can attempt to justify its response through arguments. This system only considers current feasible information and does not use the user's preferences.

The studies conducted by Morveli-Espinoza et al. [38–40] focus on the solution of goal selection problems. Their research uses artificial intelligence and argumentation semantics to select goals that are not in conflict and produce the best results considering a set of premises added before carrying out the solution process. The interface developed in their study is able to answer the "Why?" and "Why not?" questions for each goal, generating arguments based on the semantics used.

Recommender systems for e-commerce often rely on artificial intelligence (AI). The use of advanced AI related methods allow the system to ease the user-system interaction and recommend higher quality e-services and online products more closely related to the DM's preferences [33].

Recently, interactive DSS frameworks that accept arguments from the DM have been proposed to solve PPS problems . Vayanos et al. [41] presents a framework that focuses on obtaining the preferences of the DM before and during the solution of the problem. The system generates a set of preferences based on a moderate number of queries presented to the DM. Each query provides a pairwise comparisons between two solutions. The system is based on weak-preference concept. This means that even if the user shows a preference on a certain criterion, this is not considered to be an absolute factor to determine dominance between solutions and is instead taken as a support by the framework when performing a portfolio project selection.

The previous query-based interactive system research extends in another study [42]. This investigation considers two and multi-stage robust optimization problems, including R&D PPS problems. The DSS framework interacts with the user before making a recommendation by performing a series of queries where the DM must define a value that reflects the level of attractiveness towards a particular item. The system uses these values to elicit preferences considering one of two possible models: maximize worst-case utility or minimize the worst-case regret of the item recommended.

Another DSS interactive framework has been recently proposed in Nowak & Trzaskalik [43]. Their work presents a MCDSS which interacts with the DM during each interaction and allows the user to redefine his/her preferences and constraints to solve dynamic PPS problems. Their DSS considers two possible sources that lead to a change in the problem environment: a time-dependent variable and a change in the DM's preferences and constraints.

These last three proposals allow the user to provide new preferences by using queries. The proposal presented in this paper aims to provide an interactive recommender system that receives new preferential information from the DM and adapts its recommendation. The proposed system is also able to provide the DM with new information based on the knowledge obtained. In addition, the system can argue and defend its proposed project portfolio selection through arguments, with the objective that the DM understands, through dialogue, that the recommendation presented by the system is the most appropriate based on the current information.

The intention of allowing the proposed DSS framework to defend its recommendation through arguments is to allow the user to learn in detail the characteristics and properties of the problem to solve. This also allows the DM to see thoroughly the reasons for the portfolio selection made by the DSS. This paper focuses on the use of argumentation theory to not only support the solution process for a PPS problem, but also to allow the system to defend the recommended solution. Additionally, this paper incorporates two newly proposed STDs, argumentation schemes and a proof standard (TOPSIS [17]) different from those proposed by Ouerdane [35].

3. Background

This section reviews the most relevant concepts related to the proposed work, necessary to understand the said proposal and how it operates. For this, the revised concepts focus on the decision-making problem, several of the most relevant approaches, and recommendation systems, and the argumentation theory.

3.1. Multi-Objective Optimization Problem

As mentioned in Section 1, many cases in which decision problems arise involve multiple objectives to be satisfied and usually in conflict with each other. Equation (1) presents the definition of a multi-objective optimization problem (MOP). This particular example presents a maximization MOP, looking to obtain the variable decision vector $\vec{x}$ that obtains the highest possible value for the M objectives within the function set F. However, it is also necessary to mention that it is possible to define minimization MOPs or combine both maximization and minimization for a subset of objectives.

$$\max F(\vec{x}) = f_1(\vec{x}), f_2(\vec{x}), ..., f_M(\vec{x}) \quad \text{s.t. } g(\vec{x}) > 0, h(\vec{x}) = 0 \ . \tag{1}$$

Each MOP has a set of inequality (g) and equality (h) constraints that define the solutions' feasibility. Based on the above scenario, it is understandable to believe that there are cases in which defining a single solution as optimal over all the other candidates is impossible. At this point, it falls to the decision-maker to carry out the selection of the most appropriate solution (or set of solutions) based on his preferences.

3.1.1. The Decision Making Problem

In real-life situations, the DM may be represented by a person or group which seeks to improve their profits. However, the DM might not have enough resources to support all available alternatives simultaneously. This leads to what can be defined as a decision-making problem. It is necessary to search for actions that meet the current goals in the best way possible, using the available resources and maximizing profit.

Decision-making problems present four basic elements [44]: A set of one or several objectives to solve; a set of candidate solutions to achieve all objectives within the set; a set of factors that define the environment that surrounds the problem; and a set of utility values associated with each solution when they interact with the current environment.

In these cases, DMs might use multi-criteria decision support systems (MCDSS) to support their decisions. MCDSS uses computational techniques used to analyze highly complex decision problems in a reasonable computational time [45]. The multi-criteria decision analysis (MCDA) is a collection of concepts, methods, and techniques that seek to help individuals or groups make decisions involving conflicting points of view and multiple stakeholders [46]. MCDA methods are relevant components of MCDSS. Five elements are involved in these methods: Goal, decision-maker, alternatives or actions, preferences, and a solution set based on preferences.

3.1.2. Project Portfolio Selection Problem

An example of a decision-making problem can be seen in the project portfolio selection (PPS) problem. A project is defined as a temporary, unique, and unrepeatable process that pursues a specific set of objectives [47]. A project portfolio is a set of projects selected for future implementation.

In this case, a person or organization has a set of projects to carry out. These projects share the resources currently available, and there is the possibility that several of those projects complement each other, as they are effective in the same area. Therefore, it is necessary to know which project portfolio meets an organization's demands, maximizing its profit.

Equations (2)–(4) present a formal definition of the PPS problem. Let N be the number of available projects. A project portfolio $\vec{x}$ is an N sized binary vector. The projects that have been selected are given a value of 1, while the non-selected projects are given a value of 0. The value of a project portfolio for an objective i is defined by the sum of each selected portfolio's profit towards the said objective. The profit matrix p contains the respective profit obtained by the jth project for the ith objective.

Two main constraints restrict the PPS problem. First, the budget threshold, which is presented in Equation (3). The cost vector c defines how much each project costs, while B defines the maximum available current budget. The sum of all the selected projects' costs must be equal to or lower than B.

The second constraint refers to all the areas involved in the problem. Thus, it is necessary to consider several A areas and a binary project-area matrix a, which defines which projects are assigned on each area. Each area has lower and upper investment thresholds L_k and U_k, respectively. The sum of all selected projects' costs involved in each area must be between those two thresholds to be considered a feasible portfolio.

$$\max f_i(\vec{x}) = \sum_{j=1}^{N} x_j p_{i,j}. \tag{2}$$

Such as

$$\sum_{i=1}^{N} x_i c_i \leq B, \tag{3}$$

$$L_k \leq \sum_{i=1}^{N} x_i c_i a_{k,i} \leq U_k \quad k = 1, 2, ..., A. \tag{4}$$

3.2. Recommender System

By solving a PPS problem using a method such as genetic or exact algorithms, it is possible to generate a set of good quality candidate solutions. However, a prevalent issue at this step lies in presenting the DM too many potential solutions, which may be too many to carry out an analysis using only the human capability. It is also necessary to consider that the DM's preferences might have changed during the problem's solution, making the decision-making process even more difficult.

A recommender system is a potential alternative for this situation. This system relies on the DM's preferences and a set of various heuristics to direct its search and define which solutions from the set may be more attractive to the DM [48]. Specifically, in the PPS problem, a set of solutions, global and area budget constraints, and DM preferences can be used to determine the most appropriate project portfolios.

However, there is a possibility that DM is not entirely convinced and needs to know the reasons behind the decision made by the recommender system. Other possible situations that the system might face when presenting a solution to the DM are related to the human factor. For example, the DM may not know how to express his preferences correctly, may not fully know the details of the problem, and may even directly reject the system's recommendation without waiting for a justification. For these reasons, it is desirable to establish a quick relationship with the DM. The theory of argumentation offers an alternative to carry out this relationship.

3.3. Argumentation Theory in Decision Making

The argumentation theory is within the field of artificial intelligence. It can be defined as the process of constructing and evaluating arguments to justify conclusions. This allows decision-making to be carried out in a justified manner. This theory is based on non-monotonic reasoning. This means that the conclusions obtained may be modified and even rejected when new information is presented [35].

The most relevant elements to consider within the argumentation theory are cognitive artifacts, proof standards, and argumentation schemes.

3.3.1. Cognitive Artifact

Cognitive artifacts human-made objects that seek to help or enhance cognition. Its use is not only focused on supporting memory but also to set reasoning towards classifications and comparisons among several alternatives [49]. The support to the decision-making process presented by the argumentation theory can be seen as a set of cognitive artifacts used sequentially. This sequence occurs through an interaction between an expert and a client. According to [50], this process uses four cognitive artifacts: a representation of the problem, a formulation of the problem, a model of evaluation, and a final recommendation. This work addresses these last two artifacts.

3.3.2. Proof Standard

In argumentation theory, all statements must be analyzed to determine their truthfulness and their effect on a possible conclusion the DM desires to reach [35]. Proof standards are methods and techniques that allow the unification of a set of arguments for and against a certain conclusion. These proof standards analyze and determine each argument's strength and value to solve the conflict between them by accepting or rejecting the established conclusion.

A basic example of a proof standard is the simple majority. This standard takes a statement such as "project x is better than project y". For this case, the M objectives are considered, and the values obtained by each one for both projects are analyzed. If x has more objectives with better value than y, then the conclusion is true. This expression can be formally defined as presented in the following Equation (5), where S_i represent the dominance factor for objective i

$$x \succeq y \leftrightarrow |\{i \in M : xS_iy\}| \geq |\{i \in M : yS_ix\}|. \tag{5}$$

3.3.3. Argumentation Scheme

Argumentation schemes can be defined as argumentative structures capable of detecting common and stereotypical patterns of human reasoning [51]. They are based on a set of inference rules in which the existence of certain premises can lead to a conclusion. The structure of the schemes is based on non-monotonic reasoning, allowing the entry of new information, altering the state of the conclusion.

An argumentation scheme is composed of three main elements:

1. Premises: arguments for or against the conclusion. The status of each premise can be considered to be true or false until proven otherwise or to require further evidence for consideration;
2. Conclusion: statement to be confirmed or rejected based on the premises and a proof standard;
3. Critical questions: questions related to the structure of the argumentation scheme that, if not answered adequately, can falsify the veracity of an argument within it.

Argumentation schemes are not necessarily complex. For example, the cause to effect scheme [52] is based on two premises: If event A occurs, event B occurs as a consequence, and A has occurred. Therefore, the conclusion defines that B will occur. Critical questions focus on the strength of the relationship between A and B, whether if it is strong enough evidence to warrant this event, and if there exist other relevant factors that also provoke B to occur.

3.4. Dialogue Game

One possible form to represent argumentation theory within decision-making problems is through the use of dialogue games. These games model verbally or in-writing the interaction between two or more individuals, called players. The dialogue game intends to exchange arguments both for and against a statement between the players to reach a satisfactory conclusion [53].

Multiple elements must be considered for the dialogue game, such as the players and their respective roles, objectives, limitations, etc. Like any game, a set of rules must be established that defines which actions are acceptable or not during the dialogue. Also, it is necessary to define a system to determine the movements that each participant is allowed to perform at the different stages of the dialogue game.

3.4.1. Dialogue Game Rules

The dialogue game rules establish how the game is performed, defining criteria such as the starting and ending points of the game, the movements allowed for each player. These rules also define the criteria necessary to allow a coherent dialogue between the players. Each one can provide statements, arguments, and premises considered acceptable by the other participants, avoiding fallacies and dialogue loops that would stall the dialogue at a certain point [53].

There are four different types of dialogue game rules.

1. Locution rules: define the set of movements allowed for the entire dialogue game;
2. Compromise rules: define the set of statements and arguments each player is compromised to defend until proven right or wrong;
3. Dialogue rules: define the set of available movements a player has during the current state of the dialogue;
4. Termination rules: define the scenario or state that needs to be reached for the dialogue game to end.

3.4.2. State Transition Diagram

Based on the defined dialogue game rules, it is possible to identify which movements are allowed for each player and when he/she can use them. A state transition diagram (STD) can represent the evolution of the dialogue game graphically. An STD allows the players to visualize each of the different states where the dialogue can be located and the player currently in turn and what their available movements are. Similarly, an STD represents the starting and ending points of the game. With this, the four different types of rules of the dialogue game are effectively represented.

4. Proposed Work

This section describes the methodology and the different cognitive components defined for DAIRS. Afterward, a prototype proposed in this paper implements this methodology, which allows a user-system interaction through a dialogue game. This work focuses on two cognitive tasks: the evaluation model of the alternatives based on proof standards and the construction of arguments for the proposed recommender system's recommendation using argumentation schemes and a dialogue game.

4.1. Dairs Methodology

The evaluation of alternatives is the process of evaluating a set of alternatives based on their attributes, indicators, or dimensions of those alternatives [50]. In this case, the alternatives are the feasible project portfolios for the PPS problem. Each portfolio is evaluated considering its performance on each objective and set of constraints. A criteria weight vector or a criteria hierarchy order is commonly used to solve evaluate alternatives. Therefore, DAIRS also considers these two elements when evaluating portfolios to create a recommendation

Using the previous information regarding the properties of the problem provided by the DM, a proof standard is selected considering said properties and used to evaluate all the feasible portfolios. Then, the recommender system defines an initial recommendation supported by the information provided by the DM and an abductive inference argumentation scheme based on the information obtained by the proof standard used. Therefore, before the dialogue game has begun, the system already has an initial portfolio recommendation to present to the user according to his/her preferences and arguments to defend said recommendation.

The recommendation system presented in this work requires defining a set of crucial elements for its operation: A set of proof standards, argumentation schemes, and a dialogue structure that defines how both user and system will perform a bidirectional interaction using a dialogue game.

4.1.1. Proof Standards

To carry out a proper dialogue game between the user and the system, it is necessary to define methods that allow correctly collecting and analyzing the arguments for and against the current statement to reach a reasonable conclusion. Proof standards allow performing such collection and analysis.

The recommender system is capable of using a large number of proof standards. For this work, the orientation of the set of proof standards selected aims towards defining a solution for the PPS problem and is based on Ouerdane's work [35].

DAIRS considers proof standards that use a criteria preference hierarchy. These standards allow the user to define strict preferences between objectives. The recommender system focuses its search on the criteria defined as most relevant by the DM.

Simple majority: As explained in Section 3.3.2 and presented in Equation (5), this standard evaluates the truth of the statement "x is better than y" based on the number of objectives this statement holds.

Lexicographic order: This proof standard uses a hierarchical order established in the criteria. A project x is better than a project y if, and only if, x has a better value on a criterion of higher priority than y. The criteria hierarchy establishes that a higher-order criterion is infinitely more important than those in a lower position. Therefore, this method disregards the value of any other criterion of lower priority.

There are cases where even when the DM has a higher preference over specific criteria, this preference might not be strict. Instead, there is a certain threshold of acceptance for criteria with lower priority if their improvement is significant in these cases. Therefore, DAIRS considers proof standards that analyze each project portfolio supported by a criteria weight vector, determining each objective's relevance. These standards allow the system to identify possible significant improvements in criteria with different levels of importance for the DM.

Weighted majority: This method follows a similar strategy than simple majority. However, it relies on the weights of each criterion to evaluate. In this case, a criteria weight vector w assigns a weight to each criterion i (w_i). Portfolio x has a preference over portfolio y if the sum of the weights of the criteria where x is better than y is greater than the sum of the weights of the criteria where y is better than x.

$$x \succeq y \leftrightarrow W_{xy} = \sum_{xS_iy} w_i \geq W_{yx} = \sum_{yS_ix} w_i. \tag{6}$$

Weighted sum: This method defines a single fitness value S_x for each portfolio based on w and the fitness value f obtained on each objective i ($f_i(x)$). Let N be the number of criteria for the current problem. Equation (7) presents a formal definition of the previous statement. Portfolio x is preferred over portfolio y if, and only if, the sum of x is greater than the sum of y ($S_x > S_y$).

$$S_x = \sum_{i=1}^{N} w_i f_i(x). \tag{7}$$

TOPSIS: This proof standard is based on a method proposed in [17], which considers both the distance to the ideal solution, also known as utopia point, and the distance towards the negative ideal solution or nadir point. The solution that is closer to the former and furthest from the latter is the one that takes precedence.

The selection of the proper proof standard is essential to obtain a successful recommendation that follows both the DM's preferences and the quality of the solution itself. This process has a very relevant impact on the dialogue game. Each proof standard can have a set of properties defined, making them unique compared to the other set standards. During the dialogue game, both the user and system can define which properties are suitable to be considered or not in the discussion to obtain better recommendations or enhance the dialogue game's quality, based on the information provided by both players. The properties considered for the proof standard selection are:

1. Ordinality: Only the ordinal information about the performance is relevant.
2. Anonymity: There is no specific preference order for the criteria.
3. Additivity with respect to coalitions: It is possible to formulate additive values regarding the importance of a criteria subset.
4. Additivity with respect to values: The value of a solution is obtained by the sum of each criterion's values.
5. Veto: A solution must improve another over a certain veto threshold to be accepted.
6. Distance to the worst solution. The best solution is determined not only by its closeness towards the best possible solution but also by how far it is from the worst possible solution

This set of properties is based on the recommendations provided multiple works in the literature [35,52]. Table 1 shows the properties belonging to each proof standard. It should be noted that both simple majority and weighted majority methods can be used with or without a veto threshold.

Table 1. Proof Standards used for DAIRS and their properties.

Proof Standard	Ordinality	Anonymity	Add. w.r.t Coalitions	Add. w.r.t Values	Veto	Distance
Simple majority	✓	✓	✓		✓*	
Lexicographic	✓		✓			
Weighted majority	✓		✓		✓*	
Weighted sum				✓		
TOPSIS				✓		✓

* These proof standards can be used with or without using a veto threshold.

4.1.2. Argumentation Schemes

In addition to determining the proof standards to be used, it is necessary to define which human behavior patterns to consider for a dialogue within the system. The intention of defining the patterns to be identified is to regulate the system responses based on these patterns and establish boundaries in the dialogue to avoid situations such as infinite dialogue loops or loss of focus. For this reason, it is necessary to establish a set of argumentation schemes, which allow the process of identifying behavioral patterns to be carried out.

This work seeks to incorporate the proof standards selected in the previous subsection to strengthen and facilitate premise analysis and to define a conclusion for the current statement in the dialogue through argumentation schemes. These schemes are chosen considering proposals provided in previous related works [35,52]:

Abductive reasoning argument: This argumentation scheme allows the system to select the most suitable proof standard according to the current properties identified based on the information provided by user and the system.

Argument from position to know: The system performs an initial recommendation using this argumentation scheme after the system chooses a proof standard. This scheme also provides recommendations for the dialogue game's first cycles. With this, the system does not consider itself an expert yet as it has only obtained the initial information given by the list of available projects, DM's preferences, and budget threshold.

Argument from an expert opinion: After several cycles have passed in the dialogue game, surpassing a certain number of cycles, defined as *cycle threshold*, the system considers that it has obtained enough information from the user to position itself as an expert for the problem analyzed. Under this scheme, the system is more assertive in its arguments, as it has more information to defend them instead of just expecting to obtain new data from the user.

Multi-criteria pairwise comparison: The system compares the current recommendation against other alternatives, as well as solutions picked by the user that might attract his/her interest. The proof standard currently being used supports this scheme to form arguments to either defend the recommendation or select the user-picked solution if the new information provided proves that the DM's selection outperforms the system's recommendation under the current proof standard.

Practical argument from analogy: Sometimes, two solutions might be similar to a high degree. Therefore, it is necessary to consider if the previous solutions considered by either the system or the user can be considered recommendations for the dialogue game's current stage.

Ad ignorantiam: The current state of the system is unable to make inferences. All the information known by the system is considered valid by it. Meanwhile, all unknown information is considered false. The user can provide the system with new information regarding the problem in discussion at any point during the dialogue game.

Cause to effect: A change in the state of a proof standard property or the value of a criterion affects the current state of the system's recommendation. Whenever a change is detected, the system performs a reevaluation of the current solutions based on the new information. Then, it provides the user with a new recommendation, and the dialogue game continues.

From bias: The system considers this fallacy as the user might be biased towards a particular solution. While one of the recommender system's objectives is to provide the most suitable solution, user satisfaction is also a very relevant factor that a system must consider. Therefore, the system allows the user to set the recommended solution as the alternative the user picks. However, the system constantly reminds the DM that his/her choice might be biased and not the best available.

During the dialogue game, the system uses an argumentation scheme selected depending on the activities carried out in its current state by either the system or the user. Therefore, it is necessary to properly establish the dialogue game structure to use the correct argumentation scheme to characterize the arguments and premises used in the dialogue's current state.

The system relies on argumentation schemes to accept or reject a statement and obtain information, leading to changes in the problem's criteria values or the state of the proof standard properties. As previously mentioned, argumentation schemes can define the most suitable proof standard according to the current information provided.

4.1.3. Dialogue Game Rules

DAIRS aims to use a dialogue game to establish a two-dimensional interaction between the user and the system. This interaction allows both participants to provide statements to strengthen the information to ease the decision-making process.

Before carrying out a dialogue game between the user and the system, it is necessary to define the set of rules that the players will follow in the game. As previously mentioned in Section 3.4, there are four types of dialogue game rules: locution, compromise, dialogue, and termination.

The compromise, dialogue, and termination rules followed in this work are established in [35]. However, the locution rules provide two main additions. First, the system can reject an argument presented by the user if it does not satisfy the current evaluation criteria. Second, the user is allowed to reject the system's recommendation at multiple points during the dialogue. These additions focus on the system's capability to defend its recommendation and user's satisfaction. Table 2 presents the locution rules used in this work. Let ϕ be the current statement, C a critical question, and "type" refers to C being an assumption or exception.

Table 2. Locution rules for the dialogue game used in this work.

Movement	Replies	Surrenders	Status

assert(ϕ)	challenge(ϕ)
	which proof standard?
	pose(C, type, ϕ, scheme)
		accept(ϕ)
		reject(ϕ)
			stated

challenge(ϕ)	argue(ϕ, premises, scheme)	retract(ϕ)	questioned

argue(ϕ, premises, scheme)	challenge(premise)
	pose(C, type, ϕ, scheme)
	assert(ϕ)
	which proof standard?
		accept(ϕ)
		accept(premise)
		reject(ϕ)
			accepted
			rejected

pose(C, type, ϕ, scheme)	challenge(not C)
	assert(C)
	assert(not C)
		retract(ϕ)	rejected

accept(ϕ)			accepted
retract(ϕ)			rejected
which proof standard?	current proof standard		

4.1.4. State Transition Diagrams

Once the dialogue game rules are defined, it is possible to design state transition diagrams (STDs). An STD can graphically represent how the dialogue flow will carry out. A noticeable advantage in using STDs is that they offer an easy method to identify and regulate how the dialogue transpires. Also, STDs show the movements available to both players at each stage of the interaction.

The recommender system proposed uses two STDs. Before a dialogue game begins, the system will select one of these diagrams to establish a user-system interaction for the current instance. The factor considered to define which STD to use is whether the DM establishes criteria preference hierarchy before the dialogue game begins. Depending on which scenario occurs, the system will use a particular STD and a different proof standard according to which properties are considered active.

The reasoning for using different STDs based on the DM's preferences is to take advantage of the amount of information and knowledge the user has regarding the problem. DAIRS provides a learning-focused dialogue if the DM has little knowledge of the problem. Meanwhile, the system provides a more assertive and portfolio selection-driven dialogue if the DM has an acceptable level of knowledge and it is possible to skip or shorten the learning phase.

State Transition Diagram 1 (STD1): This diagram is chosen whenever the initial information available about the problem does not provide an explicit preference hierarchy regarding the problem criteria. This STD follows the structure defined in [35] while adding the additional locution rules mentioned previously. In particular, it adds a move that allows the system to reject the user's suggestion if there are no additional reasons for supporting his or her statement after a certain number of dialogue cycles have passed. A *dialogue cycle* can be defined as the point in the dialogue game when it reaches the initial state (1) once again. This system explains to the user that the reason for this rejection is to avoid a dialogue loop and continue the recommendation process. Figure 1 shows the structure of STD1.

State Transition Diagram 2 (STD2): The second STD is used when there is an explicit user-defined preference hierarchy for the criteria before the dialogue game begins. The system seeks to exploit this situation to use and obtain as much information as possible from the early state of the dialogue game. Also, it allows the user to present critical questions from the beginning, which is not allowed when using STD1. STD2 provides more flexibility for the user by allowing him/her to reject the recommendation since the initial states of the dialogue. Figure 2 shows the structure of STD2.

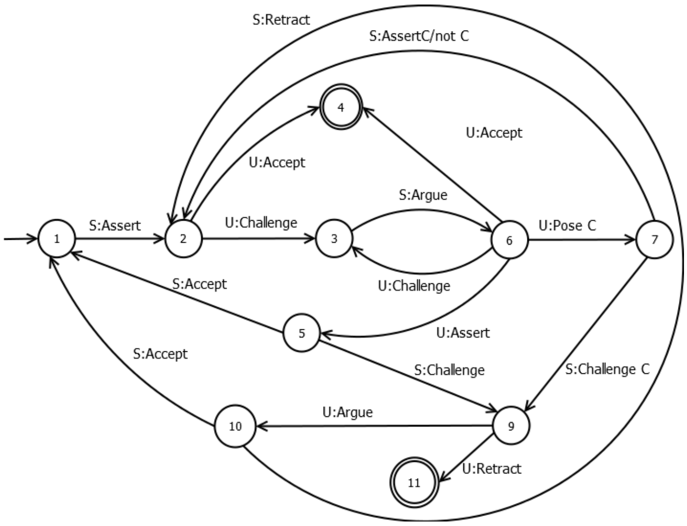


Figure 1. State Transition Diagram 1, used when there is not an explicit criteria preference hierarchy defined on the initial information.

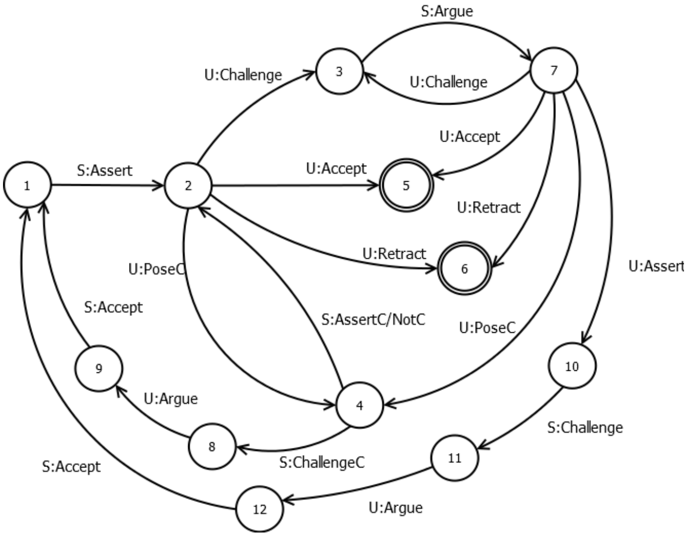


Figure 2. State Transition Diagram 2, used when there is an explicit criteria preference hierarchy defined on the initial information.

4.1.5. System Modules

The next step is to incorporate the dialogue game and all its necessary procedures to be carried out properly within DAIRS. Previously, four main processes were identified as necessary to be implemented in the system to execute a recommendation process properly [36]. Figure 3 presents the structure of these models.

Load instance module: Reads the information concerning an instance to be solved by the recommender system. The DM uploads a file containing initial instance data to the system. This file contains information such as the number of candidate solutions and criteria, a criteria weight vector, a solution/objective matrix, budget threshold, veto threshold (if required), and criteria hierarchy. For the PPS problem, it is also necessary to insert additional data, such as the project portfolio matrix, representing the projects selected by each portfolio.

Configuration module: The system analyzes the information from the instance obtained in the load instance module to determine the initial configuration of all the elements required to start a dialogue game, such as the dialogue game rules, the state transition diagram, and the initial proof standard. This setup will allow the system to provide an initial recommendation to start the dialogue with the user.

Dialogue module: The user and the system start the dialogue game. The system's main objective is to convince the user to accept the recommendation provided by it. However, the user can reject the current recommendation or add new information and modify the initial configuration. This process will provide new information to the system, which the system will use to generate a new recommendation.

Recommendation acceptance/rejection module: The user can accept or reject the system's recommendation. This module determines a final step in the dialogue. The proposed recommender system attempts to consider the human factor by allowing the user to reject the solution at several stages of the dialogue, even if the solution recommended is the most suitable according to the current information provided by both the instance and the user. This option aims towards the user's satisfaction. As previously mentioned, while the recommender system's objective is to provide a high-quality recommendation, it is also desirable that the user feels satisfied with his/her final decision. User satisfaction is also an objective that any recommender system must pursue.

These modules adequately represent a recommender system's structure supported by concepts related to argumentation theory, such as argumentation schemes. For this reason, the development of the recommender system presented in this paper uses the previously mentioned structure.

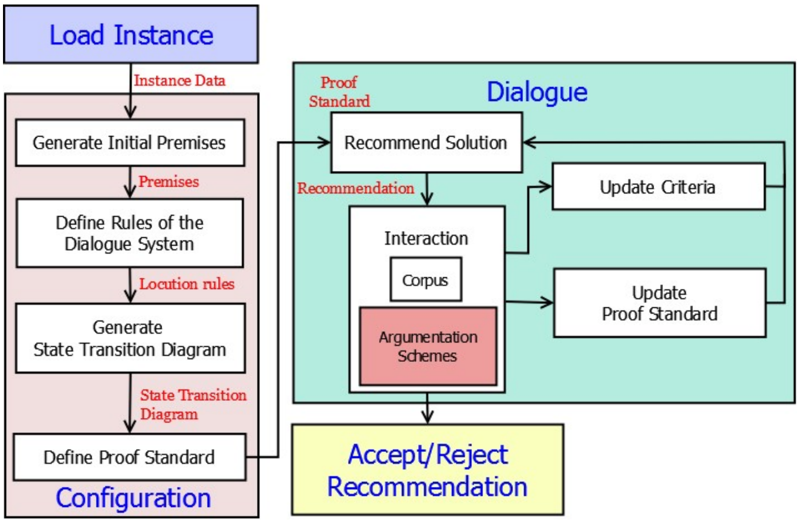
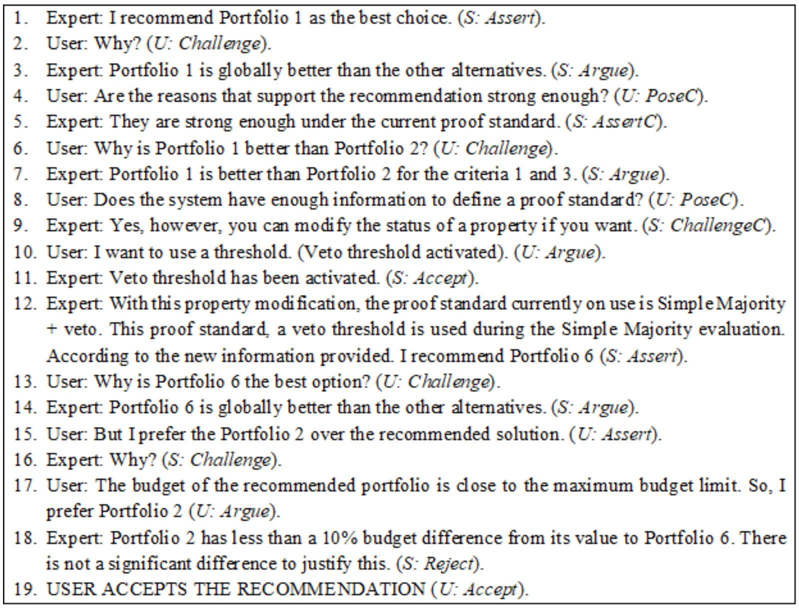


Figure 3. Diagram module of the proposed recommender system, argumentation scheme are used within the dialogue module.

4.2. Interactive Prototype

The next step in developing the proposed recommender system is the implementation of a prototype, which incorporates all the previously mentioned elements (argumentation schemes, proof standards, dialogue games, and STDs). The proposed methodology intends to properly carry out a dialogue game, following the dialogue structures defined and represented in the STDs. The development of this prototype allows a user to directly contact the recommender system and to evaluate the usability of the framework designed in the previous subsection.

Figure 4 shows a dialogue game carried out between two users following the proposed structure. This dialogue shows an interaction between a recommender system (which plays the role of an expert) and the DM (who plays the role of the user). While the system presents recommendations, the user can question them, challenge them, or argue. The end of the dialogue relies on the user's final decision to accept or reject the system's recommendations. Note that the system can evolve its recommendation into a new one when the information provided presents valid arguments to justify the change.



1. Expert: I recommend Portfolio 1 as the best choice. (*S: Assert*).
2. User: Why? (*U: Challenge*).
3. Expert: Portfolio 1 is globally better than the other alternatives. (*S: Argue*).
4. User: Are the reasons that support the recommendation strong enough? (*U: PoseC*).
5. Expert: They are strong enough under the current proof standard. (*S: AssertC*).
6. User: Why is Portfolio 1 better than Portfolio 2? (*U: Challenge*).
7. Expert: Portfolio 1 is better than Portfolio 2 for the criteria 1 and 3. (*S: Argue*).
8. User: Does the system have enough information to define a proof standard? (*U: PoseC*).
9. Expert: Yes, however, you can modify the status of a property if you want. (*S: ChallengeC*).
10. User: I want to use a threshold. (Veto threshold activated). (*U: Argue*).
11. Expert: Veto threshold has been activated. (*S: Accept*).
12. Expert: With this property modification, the proof standard currently on use is Simple Majority + veto. This proof standard, a veto threshold is used during the Simple Majority evaluation. According to the new information provided. I recommend Portfolio 6 (*S: Assert*).
13. User: Why is Portfolio 6 the best option? (*U: Challenge*).
14. Expert: Portfolio 6 is globally better than the other alternatives. (*S: Argue*).
15. User: But I prefer the Portfolio 2 over the recommended solution. (*U: Assert*).
16. Expert: Why? (*S: Challenge*).
17. User: The budget of the recommended portfolio is close to the maximum budget limit. So, I prefer Portfolio 2 (*U: Argue*).
18. Expert: Portfolio 2 has less than a 10% budget difference from its value to Portfolio 6. There is not a significant difference to justify this. (*S: Reject*).
19. USER ACCEPTS THE RECOMMENDATION (*U: Accept*).

Figure 4. Example of a dialogue game between user and system following the defined STDs.

4.2.1. Bidirectional Interaction Algorithm

Algorithm 1 corresponds to the proposed method for bidirectional interaction between the user and DAIRS. The objective is to present the user with recommended solutions and an explanation of the recommendation while receiving the DM's preferences. The system must define several argumentation elements before the user-system interaction within the prototype may begin: A set of proof standards PS and its properties $PSProperties$, the initial set of premises $Premises$, the argumentation scheme set used for the dialogue game $Schemes$, the dialogue game rules D, and the set of available state transition diagrams STD. The output of this algorithm is a portfolio recommendation r_p

This algorithm also requires a file of the instance *(file)*. This file must contain a set of elements as part of the initial input: An alternative/criteria value matrix (C), a criteria preference hierarchy ($PrefC$), a criteria weight vector (W), a veto threshold vector for all criteria (V), a set of available project portfolios (P), its respective cost ($Pcost$), and the maximum allowed budget (B). Appendix A shows in more detail the information that this file should contain.

The algorithm begins using the *Load instance* module to load an instance in step 1, obtaining all the data necessary to proceed to the *Configuration* module. From steps 2 to 7, this module defines each element's values for the cognitive decision tasks and dialogue game, according to the information provided by the instance.

Then, the *Dialogue* module is used from step 8 to step 18, establishing an interaction with the user in step 9, which could result in a possible modification of the values of the alternative/criterion value matrix, the active set of the proof standard properties, or the selected proof standard, as well as an update on the set of premises according to with the new information given by the user during that step.

Algorithm 1 Bidirectional interaction of DAIRS

```
1: ${C, PrefC, W, V, P, Pcost, B} \leftarrow$ load_instance($file$)
2: $Schemes' \leftarrow$ select_schemes($PrefC, Premises, Schemes$)
3: $D' \leftarrow$ select_locution_rule_subset($D, PrefC$)
4: $std \leftarrow$ select_std($STD, D'$)
5: $PSProperties' \leftarrow$ set_properties($PSProperties, PrefC, W, V, Premises$)
6: $ps \leftarrow$ proof_standard_selection($PSProperties', PS$)
7: $r_p \leftarrow$ recommend_portfolio($ps, P, C, PrefC, W, V$)
8: do
9: ${Premises, C, PSProperties', ps} \leftarrow$ interaction($std, D, Schemes', Premises$)
10: if modified($C$) then
11:    update_criteria($C$)
12:    $r_p \leftarrow$ recommend_portfolio($ps, P, C, PrefC, W, V$)
13: else if modified($PSProperties'$) then
14:    $PSProperties' \leftarrow$ set_properties($PSProperties', PrefC, W, V, Premises$)
15:    $ps \leftarrow$ proof_standard_selection($PSProperties', PS$)
16:    $r_p \leftarrow$ recommend_portfolio($ps, P, C, PrefC, W, V$)
17: else if modified($ps$) then
18:    $r_p \leftarrow$ recommend_portfolio($ps, P, C, PrefC, W, V$)
19: end if
20: while !accept_reject($r_p$)
```

Then, the system checks whether there was a change that could affect the current recommendation. Steps 11 and 12 are executed if there is a change in the alternative/criterion matrix values. These steps update the matrix and use the current proof standard to evaluate all the available portfolios again. Steps 14 to 16 are performed if either the user or the system has modified the proof standard's properties. These steps update the set of active proof standard properties, select the most appropriate proof standard and reevaluate the set of portfolios. The system can directly change the proof standard to offer the user a more flexible system if the user desires. If so, then step 18 is executed, using the chosen proof standard to generate a new recommendation.

The algorithm repeats this process until the user reaches a final state of acceptance or rejection of the system's recommendation. When that happens, DAIRS reaches the *Recommendation acceptance/rejection* module, considering the dialogue game finished and ending the interaction.

4.2.2. Graphical User Interface

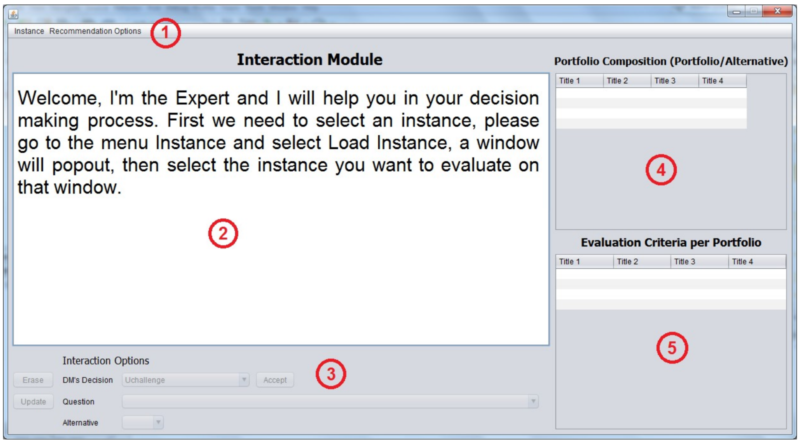
The graphical user interface of the proposed prototype seeks to allow the user to interact with the system in multiple ways. From the definition of the instance to work with, establish a dialogue with the system, edit values of the profit obtained by each available project, and manipulate the status of the proof standard properties considered by the system to match the user's preferences better. This interface is composed of a set of windows that allow the user to perform the activities previously mentioned.

Figure 5 presents the graphical user interface (GUI) of DAIRS; the primary areas in this interface are:

1. The menu bar. A set of menus that allow the user to perform actions related to the instance and its properties. It contains two sub-menus. The first sub-menu,

named *Instance*, allows the user to read, start and restart instances. The second submenu, named *Recommendation Options*, lets the user update criteria values, visualize information regarding all available portfolios, the current state of the dialogue game, and even provides the user a Help window with any necessary additional information regarding the GUI.

2. The dialogue area. Displays the recommendations and arguments presented by DAIRS, questions posed, or changes in the user's information. This section of the windows presents the arguments provided by both players during the current and previous steps in the dialogue game.
3. The interaction area. This area allows the user to perform a dialogue with the system. The user can determine his next move within the dialogue game, the statement or question that follows that movement, and, if necessary, the chance to select an alternative portfolio that accompanies the presented statement.
4. The portfolio composition area. Shows information about the portfolios and their selected projects by presenting a portfolio/alternative binary matrix.
5. The evaluation criteria area. Shows the information regarding each criterion's values for every portfolio by presenting a portfolio/criteria matrix.


Instance Recommendation Options
1
Interaction Module
2
Portfolio Composition (Portfolio/Alternative)
4
Evaluation Criteria per Portfolio
5
Interaction Options
3
Erase
DM's Decision
Uchallenge
Accept
Update
Question
Alternative

Welcome, I'm the Expert and I will help you in your decision making process. First we need to select an instance, please go to the menu Instance and select Load Instance, a window will popout, then select the instance you want to evaluate on that window.

Title 1	Title 2	Title 3	Title 4

Title 1	Title 2	Title 3	Title 4

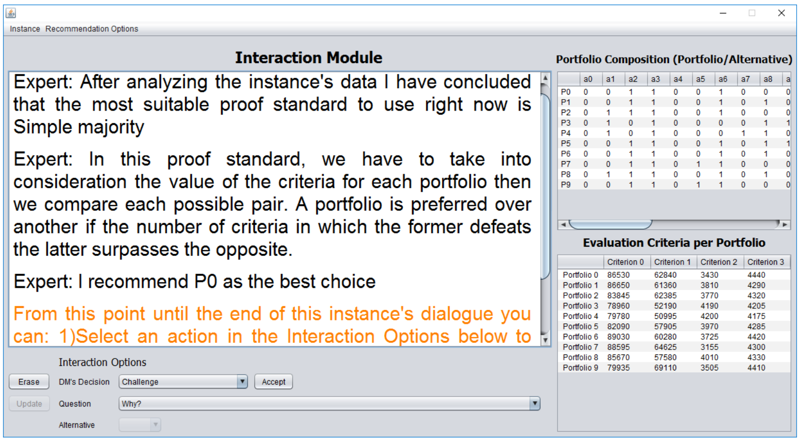
Figure 5. Main window of the DAIRS GUI. The GUI allows the user to read the dialogue, perform actions and see the available portfolios.

As previously mentioned, the recommender system prototype proposed in this work focuses on the PPS problem. As previously mentioned, the GUI intends to allow the user to interact with a recommender system using the proposed structure. The *Load instance* module reads the information about a PPS problem instance from a file. This file includes all the required information necessary to initiate a dialogue game between the user and system in DAIRS, as explained in Algorithm 1.

The *Configuration* module allows the user to select proper parameters to start the dialogue game seeking to aid the DM in his/her decision for the uploaded PPS problem. The initial premises and arguments that both the user and system are available to select from are determined based on the information. The dialogue game rules are defined. Then, DAIRS generates an STD following the structure of said rules. In this case, if there is not a criteria hierarchy defined on the instance file, then STD1 is used. However, if there is a preference order defined in the file, then STD2 is used. Finally, a proof standard is selected based on the information provided by said instance.

After this, the *Dialogue* module is reached. In the first step, the system provides an initial recommendation to the user based on the selected proof standard and all available information regarding the candidate portfolios. Figure 6 shows this process. From this point, the dialogue game begins, the user can accept or reject the said recommendation, provide his arguments to counter the system's proposal, or even introduce additional information which affects weights or veto thresholds of the criteria, the impact of an alternative in a criterion, or the proof standard. Figure 7 presents a screenshot reflecting these movements.

During the dialogue game, it is expected that the prototype's interface allows changes within the instance. The user can change the value of any criterion for each available project. The user can do so until the process reaches the *Recommendation acceptance/rejection* module when the DM accepts or rejects the current recommendation and considers it a final decision. By doing so, the dialogue game reaches its end.



Interaction Module

Expert: After analyzing the instance's data I have concluded that the most suitable proof standard to use right now is Simple majority

Expert: In this proof standard, we have to take into consideration the value of the criteria for each portfolio then we compare each possible pair. A portfolio is preferred over another if the number of criteria in which the former defeats the latter surpasses the opposite.

Expert: I recommend P0 as the best choice

From this point until the end of this instance's dialogue you can: 1)Select an action in the Interaction Options below to

Interaction Options

DM's Decision: Challenge

Question: Why?

Alternative

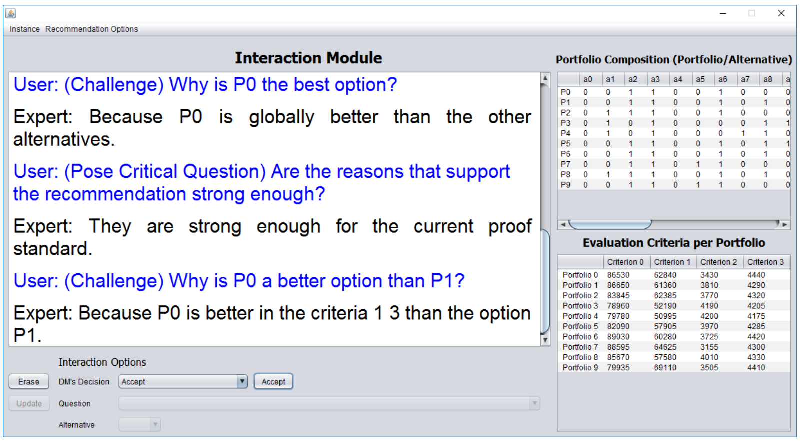
Portfolio Composition (Portfolio/Alternative)

	a0	a1	a2	a3	a4	a5	a6	a7	a8	a
P0	0	0	1	1	0	0	1	0	0	0
P1	0	0	1	1	0	0	1	0	1	0
P2	0	1	1	1	0	0	1	0	0	0
P3	0	1	0	1	0	0	0	0	1	1
P4	0	1	0	1	0	0	0	1	1	0
P5	0	0	1	1	0	0	1	0	1	1
P6	0	0	1	1	0	0	1	0	1	0
P7	0	0	1	1	0	1	1	0	0	0
P8	0	1	1	1	0	0	1	0	1	0
P9	0	0	1	1	0	1	1	0	0	0

Evaluation Criteria per Portfolio

	Criterion 0	Criterion 1	Criterion 2	Criterion 3
Portfolio 0	86530	62840	3430	4440
Portfolio 1	86650	61360	3810	4290
Portfolio 2	83845	62385	3770	4320
Portfolio 3	78960	52190	4190	4205
Portfolio 4	79780	50995	4200	4175
Portfolio 5	82090	57905	3970	4285
Portfolio 6	89030	60280	3725	4420
Portfolio 7	88595	64625	3155	4300
Portfolio 8	85670	57580	4010	4330
Portfolio 9	79935	69110	3505	4410

Figure 6. Initial recommendation from the system. DAIRS analyzes all the information provided by the instance file and provides a recommendation.



Interaction Module

User: (Challenge) Why is P0 the best option?

Expert: Because P0 is globally better than the other alternatives.

User: (Pose Critical Question) Are the reasons that support the recommendation strong enough?

Expert: They are strong enough for the current proof standard.

User: (Challenge) Why is P0 a better option than P1?

Expert: Because P0 is better in the criteria 1 3 than the option P1.

Interaction Options

DM's Decision: Accept

Question

Alternative

Portfolio Composition (Portfolio/Alternative)

	a0	a1	a2	a3	a4	a5	a6	a7	a8	a
P0	0	0	1	1	0	0	1	0	0	0
P1	0	0	1	1	0	0	1	0	1	0
P2	0	1	1	1	0	0	1	0	0	0
P3	0	1	0	1	0	0	0	0	1	1
P4	0	1	0	1	0	0	0	1	1	0
P5	0	0	1	1	0	0	1	0	1	1
P6	0	0	1	1	0	0	1	0	1	0
P7	0	0	1	1	0	1	1	0	0	0
P8	0	1	1	1	0	0	1	0	1	0
P9	0	0	1	1	0	1	1	0	0	0

Evaluation Criteria per Portfolio

	Criterion 0	Criterion 1	Criterion 2	Criterion 3
Portfolio 0	86530	62840	3430	4440
Portfolio 1	86650	61360	3810	4290
Portfolio 2	83845	62385	3770	4320
Portfolio 3	78960	52190	4190	4205
Portfolio 4	79780	50995	4200	4175
Portfolio 5	82090	57905	3970	4285
Portfolio 6	89030	60280	3725	4420
Portfolio 7	88595	64625	3155	4300
Portfolio 8	85670	57580	4010	4330
Portfolio 9	79935	69110	3505	4410

Figure 7. Advanced stage of the dialogue game. The user is questioning reasoning behind the system's recommendation and the system is able to respond.

4.2.3. Definition of the Dialogue Game Rules

Within the prototype, once the instance to read containing the PPS problem's information has been defined, it is necessary to determine how DAIRS will carry out the dialogue game between the user and the system. For this purpose, the system must define the dialogue game rules.

For this prototype, the compromise, dialogue, and termination rules are identical in all possible scenarios where an interaction between the players occurs to aid the decision-making process of a PPS problem, as shown in Section 4.1.3. However, it is necessary to define the locution rules that each dialogue game will use, based on whether or not there is a hierarchy order established for the criteria.

As mentioned in Section 4.1.4, DAIRS uses two STDs. The first one, STD1, does not require an initial preference hierarchy and focuses on obtaining information regarding the PPS problem on the dialogue game's initial cycles. The second diagram, STD2, is used when the DM defines preferences before the dialogue game begins. In this case, the system is more flexible to the user since DAIRS considers that both players have a better understanding of the problem as there is enough information to determine a hierarchy.

Once the dialogue game rules and STD are defined, the user can communicate with the system through the main window's interaction area after the system has presented an initial recommendation. Considering the structure presented by Figure 5, the user has at his disposal a set of available actions that allow him to interact with the system before and during the dialogue game. These actions are the ones that allow the creation of bidirectional interactions between the user and the system:

a. DM's Decision. Dialogue move performed by the DM.
b. Question. Statement related to the decision taken by the DM.
c. Alternative. This option will active to allow the user to select a portfolio from the available candidates if the statement selected requires selecting an alternative (for example, comparing the recommendation and another portfolio).
d. Accept button. Executes the DM's decision.
e. Erase button. Deletes the text from the dialogue text box.
f. Update button. Allows the user to update the values of the project/criteria matrix.

The options available for *DM's decision* depend on the current state of the dialogue within the STD and the locution rules defined. The user can accept the current recommendation (Accept) or reject it (Retract), ending the dialogue game. He/She can also present an argument to challenge the system's actions (Challenge), create an argument for or against the recommendation (Argue), suggest a recommendation for the system to analyze (Assert), or present a critical question that can modify the current proof standard used and its properties (Pose Critical Question).

4.2.4. Use of Argumentation Schemes

DAIRS uses an argumentation scheme based on the current state of the STD and the DM's action in the interaction area. The argumentation schemes used in this prototype are those reviewed in Section 4.1.2. This subsection briefly explains the conditions and events that trigger the use of each scheme.

The *abductive reasoning argument* scheme is used in the prototype when the system reads an instance before generating an initial recommendation, the user uses the GUI to start a dialogue game, after posing a critical question, and when the user argues to have a preference towards a particular criterion.

DAIRS uses the *argument from position to know* scheme after setting the initial proof standard, when the dialogue game starts using the GUI, when the system provides a new recommendation and the dialogues cycles have not surpassed the *cycle threshold*. The system also uses this scheme if the user challenges a system's argument and when the user poses a critical question.

Meanwhile, the recommender system uses the *argument from an expert opinion* scheme under the same scenarios as *argument from position to know*. However, it is only used when the number of dialogue cycles has surpassed the *cycle threshold*, which implies that the system has a more profound knowledge about the instance.

Whenever the user wishes to compare the profit or budget of two portfolios, DAIRS uses the *multi-criteria pairwise comparison* scheme. When the difference between two compared portfolios has no significant difference, DAIRS uses the *practical argument from analogy* argumentation scheme to support its decision.

For the fallacy-based argumentation schemes, the system uses the *ad ignorantiam* scheme at all times, as all information not introduced into the system is considered false by it. Meanwhile, DAIRS uses the *from bias* scheme when there is a criteria hierarchy defined or if the user decides to define a hierarchy.

Lastly, the prototype uses the *cause to effect* argumentation scheme when the user poses a critical question, asserts a preference towards a specific portfolio, or defines a preference towards a particular criterion as an argument to justify his/her preference for a specific portfolio.

4.2.5. Proof Standard Selection

The last step in the *configuration* module before presenting an initial recommendation is to select the initial proof standard. To do the system performs a two-stage method. The first stage corresponds to the definition of the proof standard properties before starting the dialogue game.

In DAIRS, different considerations determine if a property is set as active or inactive. Ordinality is always active unless the value of one of the weight vector values is equal to or exceeds 0.6 under a normalized value. Anonymity is active when there is not an explicit criteria preference hierarchy order defined. Additivity with respect to coalitions and with respect to values are only active if ordinality is active as well. Veto and distance to the worst solution are inactive by default.

The veto property is defined as inactive by default for the definition of the initial proof standard since the simple majority and weighted majority proof standards can have both veto and non-veto versions. Therefore, the user has the choice to activate this property during the dialogue game. Distance to the worst solution is set as inactive as the system seeks to use basic comparisons between all portfolios during the initial recommendation. The user is allowed to activate this property and access more complex proof standards during the dialogue.

After the proof standard properties setup, the system selects the most suitable standard based on the active properties. The DAIRS prototype analyses each proof standard, choosing the one with the most significant number of related properties active at that time in the system.

Following the proof standard selection, the process moves towards the *dialogue* module and performs an interaction between the user and the system using the DAIRS prototype. In this module, the user can modify the current state of all properties during the argument exchange between him/her and the system. Providing new information can also cause said properties to become active or inactive. There are three conditions in which the system can modify the status of each proof standard property:

1. If the user explicitly indicates he/she wishes to modify the state of a property (see steps 9 to 11 in Figure 4).
2. If the user indicates that he/she has a preference for a particular criterion.
3. If the user directly selects the proof standard by posing a critical question asking if the current proof standard is the best available option, in which the system responds by allowing the user to edit the status of a property or choose a new standard directly.

If any previous scenarios occur, the recommender system selects the new proof standard to use by considering the active properties or using the standard that the user directly chose. After doing so, the system analyses the available portfolios under the selected proof standard and presents a new recommendation. This process is what the system considers a *dialogue loop*.

5. Experimentation and Analysis

This section defines an experimental design to evaluate the effect of the developed prototype on various users. Generally, the measurement of recommender systems comes through quantitative measures. However, human factors affect the acceptance of the recommendation that must be evaluated in interactive systems [54], such as user satisfaction and confidence in the results.

This experiment seeks to analyze the usability of the recommender system under a real-life simulation of a PPS problem in a controlled environment, where users interact with the system. Under these considerations, this work performs a usability test to evaluate the proposed prototype.

The analysis presented in this section will allow a study of the effects on user overall satisfaction by using argumentation theory concepts, such as argumentation schemes, proof standards, and dialogue games on an MCDSS. This study will also compare the effects on user satisfaction under the two STDs presented in this proposal.

5.1. Experimental Design

A study is conducted on two groups of seven individuals to evaluate the performance of the DAIRS prototype built; each group includes people with different degrees of computational and mathematical knowledge, from people that have a basic level of computer knowledge to master degree and Ph.D. students.

Each member plays the role of a user and interacts with the system. in a dialogue game. The interaction period given to the users to work with the prototype has a maximum limit set to 50 min. The experimentation process performed by each group consists of the following steps:

(1) Introduction to the system: Users are shown the recommender system prototype and explained how it works. The users receive a detailed explanation about the different components DAIRS has, the possible actions they can perform on the system at any given time, and how the system reacts to each of the user's movements (maximum time length: 5 min).

(2) Initial use of the system to solve a sample PPS problem: In this step, the user directly interacts with the DAIRS prototype for the first time. Users face a real-life simulation of a small-sized PPS problem in terms of the number of available projects. Then, the evaluator asks each user to carry out a set of steps: Create a project/profit matrix of the PPS problem presented, analyze a set of previously made project portfolios to solve this problem, manually select the portfolio he/she believes is the best choice. After that, the users create a file of this PPS problem using the structure mentioned in Section 4.2 and upload it to the recommender system in its GUI. Finally, each user analyzes and compares their decision against the system's recommendation and engages in a brief dialogue game (maximum time length: 10 min).

The introductory PPS problem for this step is a simple example that presents the following scenario: *"You have got $20,000 in savings, and there are some necessities of life and work that you want to cover which have the following costs:"*

- Laptop—$9000.
- Desktop computer—$7000.
- Air conditioner for your room—$3000.
- Car repairs—$10,000.
- New smartphone—$2000.

"However, your savings do not allow you to buy everything, so you must select a subset. You must select which of them to choose taking into account four equally important criteria:"

- Study support
- Personal satisfaction
- Recreation outside of study
- Comfort

(3) Simulation of a complex real-life PPS problem: To fully evaluate the prototype's capabilities, both groups perform a simulation of a real-life PPS under a different environment. The first group works with an instance without a criteria preference hierarchy order defined. Meanwhile, the second group uses an instance with a criteria hierarchy. The PPS problem to solve presents the following scenario:

Four neighboring cities are planning to apply 25 social projects to improve the citizens' quality of life. However, before these projects were budgeted, a natural disaster severely depleted these towns' funds. Because of this, the cities can implement only a subset of the projects. Each city provided a list defining a level of satisfaction provided to the city by each project. Meanwhile, an analyst was hired, who generated a set of possible combinations of the projects that the cities could execute.

The first group users, which manage an instance without a criteria hierarchy, are given the objective: *Determine which project portfolio is the most adequate to best satisfy the four cities.*

The second group users, which manages an instance *with* criteria preference hierarchy, are given the objective: *Determine which project portfolio is the most adequate. The user must* *consider that a council composed of members from all four cities has decided to satisfy mainly one of the four cities as it is the one that generates the most income for all of them.*

At the beginning of this problem's study, each user expresses preferences through a criteria weight vector. The group supervisor makes a consensus using Borda counting [55,56], obtaining a weight vector that characterizes the group. Appendix A presents the information of the PPS problem file.

Each user receives a file containing the initial information of the PPS problem. The file follows their respective group's structure. The user must upload the file into the recommender system's GUI and engage in a dialogue game to obtain the most suitable solution to satisfy their respective objectives. Since the first group's problem setup does not include a criteria hierarchy, their dialogue game will follow the structure defined in STD1. Meanwhile, the second group will follow the structure of STD2 as there is a predefined criteria ranking (maximum time length: 30 min).

(4) Application of an evaluation to measure the usability of the prototype and to obtain user feedback for potential future work (maximum time length: 5 min).

5.2. Usability Evaluation

This work uses a usability test to evaluate the performance of DAIRS based on the user's opinion and satisfaction. The usability test analyses six critical elements related to a recommender system's quality: Design, functionality, ease of use, learning capacity, user satisfaction, and result and potential future use. The designed usability evaluation questions and structure are based on the models presented by Lewis and Zins et al.'s works [57,58]. This test uses a score between 0 and 10 as a measure, where 0 means complete disagreement and 10 means complete agreement. Each of the analyzed elements features a subset of questions to evaluate it:

Design:

1. I am pleased with the system's GUI.
2. The organization of the information provided by the system was clear.
3. The interface was simple to use.

Functionality:

4. The system has all the functions and capabilities I expect.
5. The information collected by the system helped me complete my activities.
6. The projects recommended by the system are suitable for my investment.
7. Being able to select my solution, disregarding the recommendation presented by the system, was helpful.

Ease of use:

8. The system was simple to use.
9. It was easy to find the information I needed.
10. The Help window provided clear information.
11. Overall, the system is easy to use.

Learning:

12. It was easy learning to use this system.
13. The information provided by the system was easy to understand.
14. The reasoning provided by the system in the dialogue eases my decision-making.
15. I consider that previous system information is required to use it.

Satisfaction:

16. I felt comfortable using this system.
17. I enjoy building my investment plan using this system.
18. Overall, I am satisfied with this system.

Result and future use:

19. I was able to complete the tasks using this system.
20. I was able to complete the tasks quickly using this system.

21. I was able to complete the tasks efficiently using this system.
22. I think that I could become more productive quickly using this system.
23. The system was able to convince me that the recommendations had value.
24. With my experience using the system, I think I would use it regularly.

5.3. Results and Analysis

The results obtained in the usability evaluation for each user were added to a total value per group. Then, the average values were obtained per question and for each of the question subsets representing the elements considered relevant for a recommender system as mentioned in the previous subsection.

Table 3 presents the results obtained on average for each element of the recommendation system considered. A Wilcoxon statistical test [59] was performed to determine whether there is a significant difference between the values obtained by the groups. The greatest difference is found within the satisfaction criterion, while the learning section shows the smallest difference between the two groups.

Figure 8 shows graphically the average values obtained in each section by each of the groups. As mentioned above, the satisfaction section shows the most remarkable difference between the two groups, while the least remarkable difference is in the learning section. Another observation that this figure presents is that the average value for all sections is higher than 8.

Table 4 presents the average value obtained by question for each group and the difference between them. The results presented in this table allow a specific visualization of the main strengths of each STD based on the users' evaluation. Figures 9 and 10 graphically show the results for each question in STD1 and STD2, respectively. These resources allow seeing which elements had a more relevant impact on user satisfaction in each of the analyzed sections of the prototype for each STD and compare them.

Table 3. Difference between the average value obtained per section for both groups. A statistical test is used to find if there is a significant difference between both groups.

Section	Group 1	Group 2	Difference
Design	9.09524	8.80952	0.28571
Functionality	9.28571	8.96429	0.32143
Ease of use	8.46429	8.89286	0.42857
Learning	8.92857	8.85714	0.07143
Satisfaction	9.52381	8.71429	0.80952 *
Result and future use	9.52381	8.73810	0.78571 *

* indicates there is a significant difference between values using Wilcoxon Test.

Based on this, it is possible to assume that STD1, which is selected if the instance does not have a defined criteria hierarchy before establishing a user-system interaction, used in a dialogue game provides better functionality and satisfaction for the users. Also, according to the values obtained for the questions related to the results and future use, it could be implied that test users would prefer to conduct a dialogue game using the structure in STD1 over the structure in STD2.

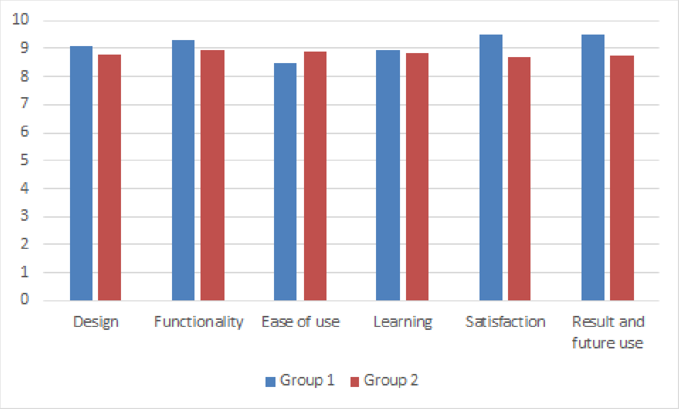


Figure 8. Bar graph comparison of the average value obtained per section by both groups.

Table 4. Difference between the average per question for both groups. This comparison allows analyzing and understanding the specific elements in which users were most satisfied.

Question	Group 1	Group 2	Difference
1	9.00000	8.42857	0.57143
2	9.28571	9.00000	0.28571
3	9.00000	9.00000	0.00000
4	9.42857	8.85714	0.57143
5	9.14286	9.00000	0.14286
6	9.14286	9.28571	0.14286
7	9.42857	8.71429	0.71429
8	7.85714	9.00000	1.14286
9	8.42857	8.71429	0.28571
10	9.00000	8.71429	0.28571
11	8.57143	9.14286	0.57143
12	8.57143	9.42857	0.85714
13	9.28571	9.00000	0.28571
14	9.42857	9.00000	0.42857
15	8.42857	8.00000	0.42857
16	9.28571	8.71429	0.57143
17	9.57143	8.57143	1.00000
18	9.71429	8.85714	0.85714
19	9.85714	8.71429	1.14286
20	9.57143	9.14286	0.42857
21	9.42857	8.85714	0.57143
22	9.71429	8.42857	1.28571
23	9.28571	9.00000	0.28571
24	9.28571	8.28571	1.00000

However, the overall user satisfaction while using STD2, which is used if there is a defined criterion preference hierarchy defined from the initial step of the dialogue game, is also acceptable. STD2 presents advantages over STD1 over certain aspects based on the results obtained, specifically regarding the ease of use, which is one of the main objectives of the STD2 design. Therefore, neither can be discarded as both have potential utility within the prototype and can provide new relevant information to the user during the dialogue.

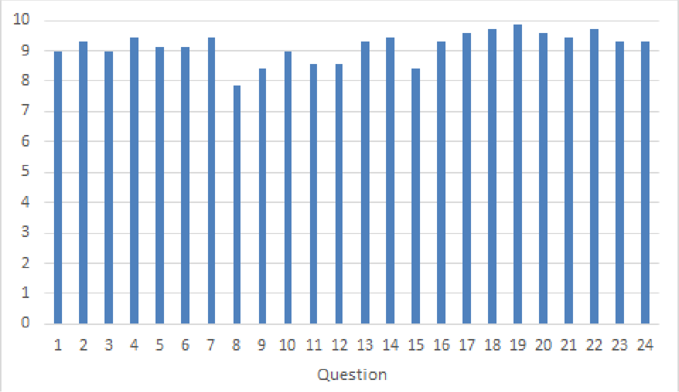


Figure 9. Bar graph showing the average obtained per question by the first group (STD1).

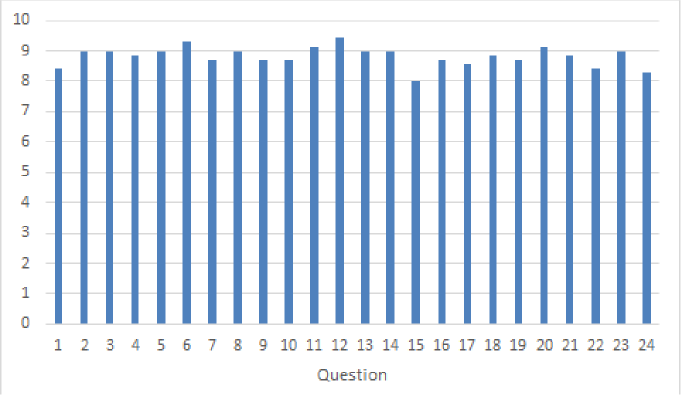


Figure 10. Bar graph showing the average obtained per question by the second group (STD2).

Analyzing the values obtained concerning design, users generally felt comfortable using the GUI presented. According to the answers obtained by the questions made regarding this section, this comfort is because each of the main window parts and the available options and windows given by the menus provide the necessary content without saturating the user with too much unnecessary information. The most common observation regarding the prototype's design is the need to add more colored details to make the relevant elements of the dialogue more noticeable. With this, it is possible to consider that the DAIRS GUI design is easy and straightforward to use. These graphical advantages are intended to favor the flow of the dialogue and thus obtain a better recommendation.

Both groups were also satisfied with the prototype's functionality, as shown from the values in Table 3. Most users think that the system can adequately support the decision-making process for PPS problems and that it has the necessary tools to execute this task. For this analyzed section, the most significant difference between the groups focuses on the users' ability to select their solution. This result suggests that users prefer an assertive but learning-focused interface. Another observation from the users is again focused on graphical aspects since both groups proposed the use of graphs to represent the criteria profit and budgets.

The results regarding ease of use imply that some users in the first group had difficulty adapting to the prototype's operation at the beginning of the test. In fact, concerning the questions related to this section, the most notable difference is presented for the users' opinion regarding the system's simplicity. Although the interface is considered to be simple and accessible for most users, some users from the first group believe that starting a dialogue game can be complicated. The obtained results can conclude that the prototype, though easy to use, requires a certain degree of prior knowledge for proper usage. Users with more previous knowledge of the problem (STD2) quickly adapted to the use of DAIRS. However, the overall results are satisfactory for both groups. As a relevant observation, the users would desire to have access to a user manual that explains each of the prototype components' operations in detail.

An interesting fact that should be mentioned is that, although some users reported issues adapting at the beginning of the test, the results concerning learning, in general, show that the conveniences and information offered by the system allow them to learn how to use it properly quickly.

Most users in both groups conclude that the prototype offers easy and simple ways to learn how to use the system, although the initial impact can present a steep learning curve at the beginning. Similarly, users generally feel that DAIRS adequately supports them in obtaining information and learning quickly and effectively about the PPS problem. Based on the learning section results, it is possible to believe that DAIRS offers the user an advantage to solve a problem, as it offers an effective problem-learning methodology supported by argumentation theory.

The two sections of questions in which there was a statistically significant difference between the two groups were "satisfaction" and "results and future use". Although mentioning the difficulty of adapting at the beginning of using the prototype in general, the first group felt largely satisfied and comfortable using the system by the end of the test. On the other hand, although the users from the second group of users were satisfied with the prototype, they consider it advisable to reduce the system's dialogue game duration. Although users are satisfied with the system's functionality and interface, they feel that the time needed for the user and the system to reach an agreement on a portfolio recommendation could be improved.

Based on the information previously presented, it is possible to say that the definition of bidirectional interaction between the user and DAIRS is effective since users feel generally satisfied with the recommendation obtained to solve a problem by using an interactive recommender system supported by MCDA methods and argumentation theory, using argumentation schemes, proof standards, and dialogue games.

Also, the results obtained in the "satisfaction" and "results and future use" sections can assume that the learning-oriented approach, given by STD1, offers higher user satisfaction in the results obtained compared to a recommendation-oriented approach, as presented by STD2. This assumption agrees with the conclusions obtained by the results in the other analyzed sections, where users using STD2 feel more comfortable using DAIRS when carrying out a dialogue following this diagram but preferred that the system would focus more on continuing the learning process of the problem.

In general, there is a better overall evaluation by the first group, being only the ease of use question subset the only exception. However, there is only a significant difference in the satisfaction and results and future use sections.

All analyzed sections obtained an average value higher than 8, and, except for the analysis for the ease of use on the first group, these values were never lower than 8.5. Based on this, it is possible to consider that the prototype had a satisfactory degree of acceptance by both groups and that the future implementation of all the presented observations could further improve its quality.

The system received an average score of 89.91%. Therefore, it is possible to conclude that this evaluation is satisfactory enough to consider DAIRS as a promising alternative. However, the results and observations from the users evidence the necessity to introduce visual resources; although the plain text could be enough for some people, others prefer a representation using images and graphs.

Most state-of-the-art works presented in Section 2 propose MCDSS frameworks to solve optimization problems and establish an interaction between the user and the system. However, this interaction only allows users to incorporate new information, and the system does not establish a deep interaction with the DM that goes beyond receiving such information and generating new recommendations. Experimentation with DAIRS shows that it is possible to generate an MCDSS to solve PPS problems capable of establishing a bidirectional interaction. In this interaction, both participants generate and obtain new information. The system's defense of the recommendation and the user's statements use argumentation theory in a dialogue game supported by argumentation schemes and proof standards.

6. Conclusions and Future Work

This work studied the characterization of cognitive tasks involved in the decision-aiding process. The cognitive tasks involved in the process were defined, identifying those that could generate an interaction between the user and the system. This paper addressed two cognitive tasks to create a final recommendation: the evaluation model for the alternatives and the argument construction. For the first cognitive task, the proposed recommender system used proof standards to define a method to evaluate and select the best fitting alternative. For the second cognitive task, the system used argumentation schemes and a dialogue game to support the preferred alternative and establish a possible user-system interaction.

One of this work's main contributions is the development of the Decision Aid Interactive Recommender System (DAIRS), an MCDSS framework focused on solving PPS multi-objective problems. The framework is based on the characterization of cognitive tasks through argumentation schemes, dialogue game rules, state transition diagrams, and proof standards. These elements are incorporated into DAIRS to allow the recommender system to perform a bidirectional interaction between it and a user. This work proposed and developed a DAIRS experimental prototype that provides an environment to aid the decision-making to validate the proposed system.

Another contribution is the proposal and design of two state transition diagrams (STDs) to determine the flow of a dialogue game between the DM and DAIRS. These STDs allow two-way interaction between both participants, meaning that both can obtain and provide information. Also, the proposed STDs have two relevant components. First, the user can reject the proposal; this defines a new dialogue stopping criterion aiming towards the user's satisfaction. Second, the system is able to defend its arguments and reject the user's statements if there is not enough information to support them. The first STD, STD1, focuses on a more learning-oriented dialogue. Meanwhile, the second STD, named STD2, assumes that the user has an acceptable degree of knowledge about the problem to solve and focuses on providing recommendations to the user and engaging in a dialogue game focused on said recommendations.

Some of the most relevant features in the DAIRS prototype are designing and implementing several concepts related to argumentation theory within an MCDSS. The first of these concepts is a set of proof standards based on several known MCDA methods. Also, DAIRS incorporates multiple argumentation schemes from the literature on its process, supported by proof standards. Another relevant feature in DAIRS is the use of these elements by employing a dialogue game that uses one of the STDs proposed in this paper to direct the flow of the user-system interaction. DAIRS consider the three standard stopping criteria for a DSS interaction: user acceptance, manual stop, and algorithmic stop. The user can accept or reject the final decision; this considers user acceptance and manual stop. For the last stopping criterion, DAIRS implements a method to avoid loops in a dialogue game using multiple argumentation schemes.

Considering the strategies used by several state-of-the-art works, DAIRS uses proof standards based on a criteria hierarchy and a criteria weight vector. These considerations positively impacted users' overall satisfaction when using the proposed prototype since it considers their preferences using methods focused on qualitative (hierarchy) and quantitative (vector of weights) strategies, which allowed for a more flexible dialogue.

A usability evaluation analyzed the proposed system in this work to measure the quality of the developed DAIRS based on the experience of multiple test users after using it to solve a PPS problem that simulated a real-life situation. This evaluation studied the user experience regarding DAIRS by considering human factors that affect the acceptance or rejection of a recommendation. The results obtained were satisfactory enough as the system received an average approval of 89.91% and an overall acceptance in several critical elements such as design, functionality, ease of use, learning capability, satisfaction, and future use. Users were satisfied using the proposed GUI due to its simple design, ease to learn, use, interaction, and capability to obtain problem information.

On the other hand, the results for users using STD1 were often better than STD2. However, in both cases, the conclusions were primarily positive. These observations allow understanding that users are looking for an interactive system that assertively establishes recommendations, but with a focus directed towards learning about the problem with the objective that both the user and the system gain new knowledge to find a better solution.

The results show that the design of a bidirectional interactive recommender system allows users to successfully and effectively select a suitable recommendation for PPS problems. DAIRS presents a novel approach to the generation of recommendations for this type of problem not previously explored in the literature, to the authors' knowledge.

Considering the research area related to this work and all the observations and comments provided by the test users, multiple areas offer potential future work. First, the use of the proposed system on real-life problems different than the PPS problem. Second, adding new elements that make the recommender system capable of receiving new user-made portfolios during the dialogue game. Currently, the system uses only one STD per dialogue. Therefore, future work could focus on using more than one STD per dialogue game, looking to improve the dialogue game's quality. Also, there exists a wide variety of MCDA methods in the state-of-the-art, opening the possibility of using different methods as proof standards. Finally, the following versions of the prototype could provide the user with a more friendly looking GUI, featuring graphs and a more colorful environment.

Author Contributions: Conceptualization: T.M.-E., L.C.-R. and C.M.-T.; methodology: T.M.-E., C.M.-T. and N.R.-V.; software: T.M.-E. and N.R.-V.; validation: T.M.-E., L.C.-R. and C.G.-S.; formal analysis: T.M.-E., C.M.-T. and H.F.-H.; investigation: T.M.-E., L.C.-R., C.M.-T. and C.G.-S.; resources: T.M.-E., L.C.-R. and H.F.-H.; data curation: T.M.-E.; writing—original draft preparation: T.M.-E. and C.M.-T.; writing—review and editing: T.M.-E., L.C.-R., C.M.-T. and C.G.-S.; visualization: L.C.-R.; supervision: L.C.-R.; project administration: L.C.-R.; funding acquisition: T.M.-E., L.C.-R., C.M.-T. and H.F.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to a completely voluntary and conscientious acceptance by each of the users to participate in the tests carried out to perform the usability evaluation. All users only interacted with the system during the test periods established in advance notice.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: A DAIRS prototype is available at: https://www.dropbox.com/sh/j1yfblg011a7m0w/AABYrgbKBDWEFBg_vRcGE6dja?dl=0 (accessed on 20 April 2021).

Acknowledgments: Authors thank to CONACYT for supporting the projects from (a) Cátedras CONACYT Program with Number 3058. (b) Project CONACyT A1-S-11012 from Convocatoria de Investigación Científica Básica 2017–2018 and CONACYT Project with Number 312397 from Programa de Apoyo para Actividades Científicas, Tecnológicas y de Innovación (PAACTI), a efecto de participar en la Convocatoria 2020-1 Apoyo para Proyectos de Investigación Científica, Desarrollo Tecnológico e Innovación en Salud ante la Contingencia por COVID-19. (c) T. Macias-Escobar would like to acknowledge CONACYT, Mexico National Grant System, Grant 465554.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MCDA	Multi-criteria Decision Analysis
DAIRS	Decision Aid Interactive Recommender System
DM	Decision Maker
PPS	Project Portfolio Selection
DSS	Decision Support Systems
CHP	Choice Problem
MOP	Multi-objective Optimization Problem
MCDSS	Multi-Criteria Decision Support Systems
STD	State Transition Diagram
GUI	Graphical User Interface

Appendix A. PPS Problem Test Case

This appendix presents the information in the PPS problem file required for the *load instance module*. This information is also used to perform step 3 of the experimental design.

- Problem type: Maximization
- Number of available projects: 25
- Number of available project portfolios: 10
- Criteria size: 4
- Criteria hierarchy (for the second group): {2,3,1,4}
- Criteria weight vector (for the second group): {0.20 0.39 0.36 0.05}
- Budget threshold: 80,000
- Veto threshold per criterion {1500,1200,75,75}

Table A1. Criteria-profit matrix.

Project	Criterion 1	Criterion 2	Criterion 3	Criterion 4
Project 1	3200	2000	165	300
Project 2	6255	1640	385	390
Project 3	5680	6940	270	445
Project 4	8965	4195	355	415
Project 5	6550	6560	315	440
Project 6	6740	6290	150	350
Project 7	9055	7165	375	485
Project 8	4170	3015	410	285
Project 9	9735	2860	480	330
Project 10	3350	4210	400	315
Project 11	8595	7270	150	265
Project 12	9070	2430	455	360
Project 13	9930	5825	420	385
Project 14	4675	4505	425	490
Project 15	8065	4030	165	425
Project 16	7910	7665	240	320
Project 17	9860	6265	415	350
Project 18	3175	6240	225	445
Project 19	9660	655	320	475
Project 20	1150	4500	415	400
Project 21	3245	5950	105	275
Project 22	5350	6750	480	460
Project 23	6050	2505	285	305
Project 24	9190	2395	160	290
Project 25	9615	4340	100	480

List of available project portfolios and required budget (0 indicates that a project has not been included by the portfolio, 1 indicates that a project has been included):

- Portfolio 1 {0,0,1,1,0,0,1,0,0,0,1,1,1,1,0,1,1,1,0,0,0,0,0,0,1} Total cost: 79,290
- Portfolio 2 {0,0,1,1,0,0,1,0,1,0,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0} Total cost: 79,575
- Portfolio 3 {0,1,1,1,0,0,1,0,0,0,1,1,1,0,0,1,1,1,0,0,0,1,0,0,0} Total cost: 79,875
- Portfolio 4 {0,1,0,1,0,0,0,0,1,1,1,1,1,1,0,0,1,1,0,0,0,1,0,0,0} Total cost: 79,525
- Portfolio 5 {0,1,0,1,0,0,0,1,1,0,1,1,1,1,0,0,1,1,0,0,0,1,0,0,0} Total cost: 79,905
- Portfolio 6 {0,0,1,1,0,0,1,0,1,1,1,1,1,1,0,0,1,1,0,0,0,0,0,0,0} Total cost: 79,885
- Portfolio 7 {0,0,1,1,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,1,0,0,1} Total cost: 79,410
- Portfolio 8 {0,0,1,1,0,1,1,0,0,0,1,1,1,0,0,1,1,1,0,0,0,0,0,0,1} Total cost: 79,585
- Portfolio 9 {0,1,1,1,0,0,1,0,1,0,1,1,1,0,0,0,1,1,0,0,0,1,0,0,0} Total cost: 79,230
- Portfolio 10 {0,0,1,1,0,1,1,0,0,0,1,0,1,1,0,1,1,1,0,0,0,1,0,0,0} Total cost: 79,825

References

1. Keeney, R.L.; Raiffa, H.; Meyer, R.F. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*; Cambridge University Press: Cambridge, UK, 1993.
2. Fernández González, E.; López Cervantes, E.; Navarro Castillo, J.; Vega López, I. Aplicación de metaheurísticas multiobjetivo a la solución de problemas de cartera de proyectos públicos con una valoración multidimensional de su impacto. *Gestión Política Pública* **2011**, *20*, 381–432.
3. Bechikh, S.; Kessentini, M.; Said, L.B.; Ghédira, K. Preference incorporation in evolutionary multiobjective optimization: A survey of the state-of-the-art. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 2015; Volume 98, pp. 141–207.
4. Fernandez, E.; Lopez, E.; Lopez, F.; Coello, C.A.C. Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method. *Inf. Sci.* **2011**, *181*, 44–56. [CrossRef]
5. Ishizaka, A.; Nemery, P. *Multi-Criteria Decision Analysis: Methods and Software*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
6. Chu, P.Y.V.; Hsu, Y.L.; Fehling, M. A decision support system for project portfolio selection. *Comput. Ind.* **1996**, *32*, 141–149. [CrossRef]
7. Bana e Costa, C.A.; De Corte, J.M.; Vansnick, J.C. On the mathematical foundations of MACBETH. In *Multiple Criteria Decision Analysis*; Springer: New York, NY, USA, 2016; pp. 421–463.
8. Hummel, J.; Oliveira, M.D.; Bana e Costa, C.A.; IJzerman, M.J. Supporting the project portfolio selection decision of research and development investments by means of multi-criteria resource allocation modelling. In *Multi-Criteria Decision Analysis to Support Healthcare Decisions*; Springer: Cham, Switzerland, 2017; pp. 89–103.
9. Archer, N.P.; Ghasemzadeh, F. An integrated framework for project portfolio selection. *Int. J. Proj. Manag.* **1999**, *17*, 207–216. [CrossRef]
10. Ghasemzadeh, F.; Archer, N.P. Project portfolio selection through decision support. *Decis. Support Syst.* **2000**, *29*, 73–88. [CrossRef]
11. Hu, G.; Wang, L.; Fetch, S.; Bidanda, B. A multi-objective model for project portfolio selection to implement lean and Six Sigma concepts. *Int. J. Prod. Res.* **2008**, *46*, 6611–6625. [CrossRef]
12. Smith, B. Lean and Six Sigma–a one-two punch. *Qual. Prog.* **2003**, *36*, 37.
13. Khalili-Damghani, K.; Sadi-Nezhad, S.; Lotfi, F.H.; Tavana, M. A hybrid fuzzy rule-based multi-criteria framework for sustainable project portfolio selection. *Inf. Sci.* **2013**, *220*, 442–462. [CrossRef]
14. Mira, C.; Feijão, P.; Souza, M.A.; Moura, A.; Meidanis, J.; Lima, G.; Bossolan, R.P.; Freitas, Ì.T. A project portfolio selection decision support system. In Proceedings of the 2013 10th International Conference on Service Systems and Service Management, Hong Kong, China, 17–19 July 2013; pp. 725–730.
15. Mohammed, H.J. The optimal project selection in portfolio management using fuzzy multi-criteria decision-making methodology. *J. Sustain. Financ. Investig.* **2021**, 1–17. [CrossRef]
16. Saaty, R.W. The analytic hierarchy process—What it is and how it is used. *Math. Model.* **1987**, *9*, 161–176. [CrossRef]
17. Hwang, C.L.; Yoon, K. Methods for multiple attribute decision making. In *Multiple Attribute Decision Making*; Springer: Berlin/Heidelberg, Germany, 1981; pp. 58–191.
18. Dobrovolskienė, N.; Tamošiūnienė, R. Sustainability-oriented financial resource allocation in a project portfolio through multi-criteria decision-making. *Sustainability* **2016**, *8*, 485. [CrossRef]
19. Markowitz, H. *Portfolio Selection: Efficient Diversification of Investments*; Yale University Press: New Haven, CT, USA, 1959.
20. Debnath, A.; Roy, J.; Kar, S.; Zavadskas, E.K.; Antucheviciene, J. A hybrid MCDM approach for strategic project portfolio selection of agro by-products. *Sustainability* **2017**, *9*, 1302. [CrossRef]
21. Bai, C.; Sarkis, J. A grey-based DEMATEL model for evaluating business process management critical success factors. *Int. J. Prod. Econ.* **2013**, *146*, 281–292. [CrossRef]
22. Pamučar, D.; Ćirović, G. The selection of transport and handling resources in logistics centers using Multi-Attributive Border Approximation area Comparison (MABAC). *Expert Syst. Appl.* **2015**, *42*, 3016–3028. [CrossRef]

23. Verdecho, M.J.; Pérez-Perales, D.; Alarcón-Valero, F. Project portfolio selection for increasing sustainability in supply chains. *Econ. Bus. Lett.* **2020**, *9*, 317–325. [CrossRef]
24. Miettinen, K.; Hakanen, J.; Podkopaev, D. Interactive nonlinear multiobjective optimization methods. In *Multiple Criteria Decision Analysis*; Springer: New York, NY, USA, 2016; pp. 927–976.
25. Miettinen, K.; Ruiz, F.; Wierzbicki, A.P. Introduction to multiobjective optimization: Interactive approaches. In *Multiobjective Optimization*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 27–57.
26. De Almeida, A.T.; de Almeida, J.A.; Costa, A.P.C.S.; de Almeida-Filho, A.T. A new method for elicitation of criteria weights in additive models: Flexible and interactive tradeoff. *Eur. J. Oper. Res.* **2016**, *250*, 179–191. [CrossRef]
27. Nebro, A.J.; Ruiz, A.B.; Barba-González, C.; García-Nieto, J.; Luque, M.; Aldana-Montes, J.F. InDM2: Interactive dynamic multi-objective decision making using evolutionary algorithms. *Swarm Evol. Comput.* **2018**, *40*, 184–195. [CrossRef]
28. Azabi, Y.; Savvaris, A.; Kipouros, T. The interactive design approach for aerodynamic shape design optimisation of the aegis UAV. *Aerospace* **2019**, *6*, 42. [CrossRef]
29. Stummer, C.; Kiesling, E.; Gutjahr, W.J. A multicriteria decision support system for competence-driven project portfolio selection. *Int. J. Inf. Technol. Decis. Mak.* **2009**, *8*, 379–401. [CrossRef]
30. Nowak, M. Project portfolio selection using interactive approach. *Procedia Eng.* **2013**, *57*, 814–822. [CrossRef]
31. Haara, A.; Pykäläinen, J.; Tolvanen, A.; Kurttila, M. Use of interactive data visualization in multi-objective forest planning. *J. Environ. Manag.* **2018**, *210*, 71–86. [CrossRef]
32. Kurttila, M.; Haara, A.; Juutinen, A.; Karhu, J.; Ojanen, P.; Pykäläinen, J.; Saarimaa, M.; Tarvainen, O.; Sarkkola, S.; Tolvanen, A. Applying a multi-criteria project portfolio tool in selecting energy peat production areas. *Sustainability* **2020**, *12*, 1705. [CrossRef]
33. Zhang, Q.; Lu, J.; Jin, Y. Artificial intelligence in recommender systems. *Complex Intell. Syst.* **2021**, *7*, 439–457. [CrossRef]
34. Labreuche, C. Argumentation of the decision made by several aggregation operators based on weights. In Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'06), Paris, France, 2–7 July 2006; pp. 683–690.
35. Ouerdane, W. Multiple Criteria Decision Aiding: A Dialectical Perspective. Ph.D. Thesis, University of Paris-Dauphine, Paris, France, 2011.
36. Cruz-Reyes, L.; Medina-Trejo, C.; Morales-Rodríguez, M.L.; Gómez-Santillan, C.G.; Macias-Escobar, T.E.; Guerrero-Nava, C.A.; Pérez-Villafuerte, M. A Dialogue Interaction Module for a Decision Support System Based on Argumentation Schemes to Public Project Portfolio. In *Nature-Inspired Design of Hybrid Intelligent Systems*; Springer: Cham, Switzerland, 2017; pp. 741–756.
37. Sassoon, I.; Kökciyan, N.; Sklar, E.; Parsons, S. Explainable argumentation for wellness consultation. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*; Springer: Cham, Switzerland, 2019; pp. 186–202.
38. Morveli-Espinoza, M.; Nieves, J.C.; Possebom, A.; Puyol-Gruart, J.; Tacla, C.A. An argumentation-based approach for identifying and dealing with incompatibilities among procedural goals. *Int. J. Approx. Reason.* **2019**, *105*, 1–26. [CrossRef]
39. Espinoza, M.M.; Possebom, A.T.; Tacla, C.A. Argumentation-based agents that explain their decisions. In Proceedings of the 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), Salvador, Brazil, 15–18 October 2019; pp. 467–472.
40. Morveli-Espinoza, M.; Tacla, C.A.; Jasinski, H.M. An Argumentation-Based Approach for Explaining Goals Selection in Intelligent Agents. In *Brazilian Conference on Intelligent Systems*; Springer: Cham, Switzerland, 2020; pp. 47–62.
41. Vayanos, P.; McElfresh, D.; Ye, Y.; Dickerson, J.; Rice, E. Active preference elicitation via adjustable robust optimization. *arXiv* **2020**, arXiv:2003.01899 .
42. Vayanos, P.; Georghiou, A.; Yu, H. Robust optimization with decision-dependent information discovery. *arXiv* **2020**, arXiv:2004.08490.
43. Nowak, M.; Trzaskalik, T. A trade-off multiobjective dynamic programming procedure and its application to project portfolio selection. *Ann. Oper. Res.* **2021**, 1–27. [CrossRef]
44. Chernoff, H.; Moses, L.E. *Elementary Decision Theory*; Courier Corporation: Chelmsford, MA, USA, 2012.
45. López, J.C.L.; González, E.F.; Alvarado, M.T. Special Issue on Multicriteria Decision Support Systems. *Computación y Sistemas* **2008**, *12*. Available online: http://www.scielo.org.mx/pdf/cys/v12n2/v12n2a1.pdf (accessed on 20 April 2021).
46. Belton, V.; Stewart, T. *Multiple Criteria Decision Analysis: An Integrated Approach*; Springer Science & Business Media: Berlin, Germany, 2002.
47. Carazo, A.F.; Gómez, T.; Molina, J.; Hernández-Díaz, A.G.; Guerrero, F.M.; Caballero, R. Solving a comprehensive model for multiobjective project portfolio selection. *Comput. Oper. Res.* **2010**, *37*, 630–639. [CrossRef]
48. Resnick, P.; Varian, H.R. Recommender systems. *Commun. ACM* **1997**, *40*, 56–58. [CrossRef]
49. Wilson, R.A.; Keil, F.C. *The MIT Encyclopedia of the Cognitive Sciences*; MIT Press: Cambridge, MA, USA, 2001.
50. Tsoukiàs, A. On the concept of decision aiding process: An operational perspective. *Ann. Oper. Res.* **2007**, *154*, 3–27. [CrossRef]
51. Walton, D.N. *Argumentation Schemes for Presumptive Reasoning*; Psychology Press: New York, NY, USA, 1996. .
52. Walton, D.; Reed, C.; Macagno, F. *Argumentation Schemes*; Cambridge University Press: Cambridge, UK, 2008.
53. Walton, D.N. *Logical Dialogue—Games and Fallacies*; University Press of America: Lanham, MD, USA, 1984
54. He, C.; Parra, D.; Verbert, K. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Syst. Appl.* **2016**, *56*, 9–27. [CrossRef]
55. Van Newenhizen, J. The Borda method is most likely to respect the Condorcet principle. *Econ. Theory* **1992**, *2*, 69–83. [CrossRef]
56. Orouskhani, M.; Shi, D.; Cheng, X. A Fuzzy Adaptive Dynamic NSGA-II With Fuzzy-Based Borda Ranking Method and Its Application to Multimedia Data Analysis. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 118–128. [CrossRef]

57. Lewis, J.R. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *Int. J. Hum. Comput. Interact.* **1995**, *7*, 57–78. [CrossRef]
58. Zins, A.H.; Bauernfeind, U.; Del Missier, F.; Venturini, A.; Rumetshofer, H.; Frew, A. *An Experimental Usability Test for Different Destination Recommender Systems*; Springer-Verlag New York Inc.: New York, NY, USA, 2004; pp. 228–238.
59. Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in Statistics*; Springer: New York, NY, USA, 1992; pp. 196–202.