

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Misitano, Giovanni

Title: Interactively Learning the Preferences of a Decision Maker in Multi-objective Optimization Utilizing Belief-rules

Year: 2020

Version: Accepted version (Final draft)

Copyright: ©2020 IEEE

Rights: In Copyright

Rights url: <http://rightsstatements.org/page/InC/1.0/?language=en>

Please cite the original version:

Misitano, G. (2020). Interactively Learning the Preferences of a Decision Maker in Multi-objective Optimization Utilizing Belief-rules. In SSCI 2020 : Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (pp. 133-140). IEEE.
<https://doi.org/10.1109/SSCI47803.2020.9308316>

Interactively Learning the Preferences of a Decision Maker in Multi-objective Optimization Utilizing Belief-rules

Giovanni Misitano

*Faculty of Information Technology
University of Jyväskylä*

P.O. Box 35, Agora, 40014, University of Jyväskylä, Finland
giovanni.a.misitano@jyu.fi

Abstract—Many real life problems can be modelled as multi-objective optimization problems. Such problems often consist of multiple conflicting objectives to be optimized simultaneously. Multiple optimal solutions exist to these problems, and a single solution cannot be said to be the best without preferences given by a domain expert. Preferences can be used to find satisfying solutions: optimal solutions, which best match the expert’s preferences. To model the preferences of the expert, and aid him/her in finding satisfying solutions, a novel method is proposed. The method utilizes machine learning combined with belief-rule based systems to adaptively train a belief-rule based system to learn a domain expert’s preferences using preference information gathered during an interactive process. Belief-rule based systems are explainable generalized expert systems, which have not been used before in the manner described in this paper to model preferences of a domain expert for a multi-objective optimization problem. In the case study conducted, the satisfying solutions found using learned preferences are concluded to be compatible with the preferences of the expert, which support the proposed method’s viability as a decision making support tool.

Index Terms—multiple objective optimization, belief-rule based systems, machine learning, Python, preference modelling, decision making

I. INTRODUCTION

Many real-life problems can be modelled as multi-objective optimization problems with multiple conflicting objectives. Such problems can emerge in health care [1] and engineering [2], for example. No single solution exists to these problems and the aid of a domain expert, a decision maker, is involved. The decision maker can give preferences, which can be used to identify satisfying optimal solutions matching the given preferences. To model the preferences of a decision maker, a value function can be used [3]. The value function is not explicitly known and must be inferred from the preferences of the decision maker. If the value function is known, it would allow finding the most satisfying solutions according to the preferences of a decision maker, therefore aiding the decision maker in finding satisfying solutions to a multi-objective optimization problem.

In this paper, a novel method is proposed which utilizes machine learning and belief-rule based systems to learn the value function of a decision maker. Belief-rule based systems are

generalized expert systems able to model non-linear systems. Adaptive training is utilized in the proposed method, where preferences given by a decision maker during an interactive process are utilized to adaptively train a belief-rule based system to model the value function of the decision maker. Belief-rule based systems have been chosen to be utilized in modelling the value function because they can be used to build explainable models [4]. Explainability increases the trust humans have in predictions made by computational models [5]. However, the explainability of the method is not explored in this work, and is left for future research.

Some recent works, where the value function of a decision maker is inferred using machine learning can be found in [6] and [7], for example. For more details on modelling preferences for multi-objective optimization problems, see [8].

This paper is structured in the following manner: in Section II the necessary background is given for understanding how the proposed method works, in Section III the proposed method – the INFRINGER method – is presented, in Section IV the software implementation of the INFRINGER method is briefly discussed, in Section V a case study in forest landscape planning is conducted using the INFRINGER method, and in Section VI the results of the case study are discussed. Finally, in Section VII conclusions are made regarding the feasibility of INFRINGER as a method for solving multi-objective optimization problems, and how successful the method is in learning the preferences of a decision maker. This paper ends with suggestions for future research topics related to the INFRINGER method.

II. BACKGROUND

A. Multi-objective optimization

A multi-objective optimization problem consists of multiple objective functions, which are to be optimized simultaneously under certain constraints. Such a problem can be defined as

$$\max_{\mathbf{x} \in X} \mathbf{f}(\mathbf{x}) = \{f_i(\mathbf{x}) \mid i \in [1, m]\}, \quad (1)$$

where \mathbf{f} is a vector of m objective functions f_i to be maximized. Each f_i expects a decision variable vector \mathbf{x} as its

argument. The decision variable vectors \mathbf{x} with n elements belong to the feasible set $X \subset \mathbb{R}^n$, which is defined by constraints imposed on \mathbf{x} .

The objective functions in (1) are assumed to be conflicting, which means that no solution in X is able to optimize simultaneously all objectives f_i . Instead, multiple optimal solutions $\mathbf{x}^* \in X$ exist, which form a set of optimal objective vectors $\mathbf{f}(\mathbf{x}^*) = \mathbf{z} \in Z$ for all existing optimal solutions \mathbf{x}^* . For two vectors $\mathbf{z}, \tilde{\mathbf{z}} \in Z$, the vector \mathbf{z} dominates $\tilde{\mathbf{z}}$ if, and only if, $z_i \geq \tilde{z}_i$ is true for all $i \in [1, m]$, and $z_i > \tilde{z}_i$ is true for at least one $i \in [1, m]$, where the index i is used to denote the i th element in the vectors. The Pareto optimal set Z^{Pareto} is then defined to be the set containing all the vectors \mathbf{z} , which are not dominated by any other vector in Z . However, in a real world scenario, the whole extent of the Pareto optimal set for a problem is often not known. Therefore, subsets of the *true* Pareto optimal set Z^{Pareto} , known as *representations* of the Pareto optimal set, are used instead. From the definition of the Pareto optimal set, and as long as all relevant objectives are included in the definition of the multi-objective optimization problem, we can assume that the decision maker is only interested in the vectors present in the set Z^{Pareto} .

Additionally, the nadir point \mathbf{z}^{nad} and the ideal point \mathbf{z}^* are defined. The nadir point consists of the worst values of each objective in Z^{Pareto} , and the ideal point consists of the best values respectively. The nadir and ideal points are used to convey information on the attainable objective values of (1). It is worth noting that the ideal point is not feasible, that is, it cannot be computed from a decision vector $\mathbf{x} \in X$. On the other hand, the nadir point might or might not be feasible. In practice, the ideal point is trivial to calculate, but because the nadir point depends on the whole extent of the Pareto optimal set, it is often impossible to calculate the *true* nadir point, which is why estimates of the nadir are often used. In this paper, a payoff table was used in the case study to calculate the nadir point of the forest landscape planning problem. The payoff table alongside an alternative method for approximating the nadir point are discussed in [9].

Methods for solving multi-objective optimization problems can be divided into three categories based on how the methods incorporate a decision maker's preferences. Methods utilizing preferences before solving the multi-objective optimization problem are known as *a priori* methods. Methods that make use of preference information after the optimization process are known as *a posteriori* methods. In this paper, an *interactive* method is developed. In interactive methods, the preference information is used and updated during the optimization process. This requires for the decision maker to actively take part in the optimization process by expressing his/her preferences based for example on candidate solutions, which are computed and presented to him/her during the process. For a more detailed introduction to multi-objective optimization and multi-objective optimization methods, see [3], and for a recent review discussing interactive multi-objective optimization methods, see [10].

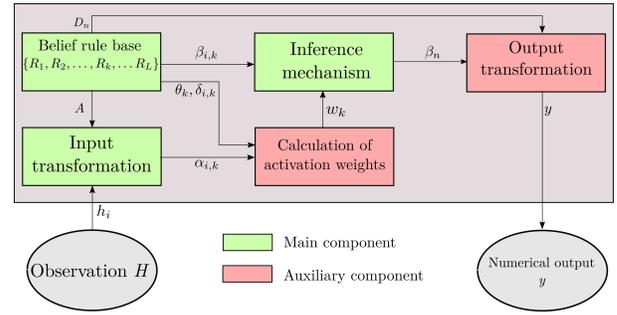


Fig. 1. The general structure of a BRB system and the relation between its components.

B. The value function

To model the preferences of a decision maker, a value function of the form

$$\mathbb{R}^m \rightarrow \mathbb{R} : f_{\text{val}}(\mathbf{z}) \quad (2)$$

is assumed to exist. The value function maps each objective vector \mathbf{z} to a scalar value. The higher the value of (2) is for an objective vector \mathbf{z} , the more preferred \mathbf{z} is according to the preferences of the decision maker. By maximizing (2), the most preferred objective vector(s) according to the decision maker's preferences can be found.

Assuming that a decision maker will always prefer more to less regarding the objectives in a problem (1), the value function (2) can be regarded to be monotonically increasing as a function of the elements z_i . This means that when one of the elements z_i is varied and the others are kept constant, the value of (2) should increase as z_i is increased.

It is unrealistic to assume a decision maker to be able to tell what his/her value function is explicitly. Therefore, the value function is otherwise inferred utilizing preference information given by the decision maker. In this paper, the value function is inferred utilizing a belief-rule based system and pair-wise comparisons of vectors in Z^{Pareto} conducted by a decision maker. Pair-wise comparisons are chosen because they are a relatively simple way for a human decision maker to convey preference information.

C. Belief-rule based systems

Belief-rule based (BRB) systems presented in this subsection are based on the RIMER methodology discussed in [11] and [12]. The BRB systems described in this paper consist of a belief-rule base, an inference mechanism, and a way to transform observed data into inputs used in the rules of the rule base. The BRB system can be trained by formulating a cost function, which is then minimized to find a set of optimal parameters for the BRB system.

The general structure of a rule base is summarized in Figure 1. Concluding this subsection, the cost function and its minimization problem are discussed. The reader is advised that many of the symbols already used and defined in the previous subsections are reused and redefined here. This has been a

$$\beta_n = \frac{\prod_{k=1}^L (w_k \beta_{n,k} + 1 - w_k \sum_{i=1}^N \beta_{i,k}) - \prod_{k=1}^L (1 - w_k \sum_{i=1}^N \beta_{i,k})}{\sum_{j=1}^N \prod_{k=1}^L (w_k \beta_{j,k} + 1 - w_k \sum_{i=1}^N \beta_{i,k}) - (N-1) \prod_{k=1}^L (1 - w_k \sum_{i=1}^N \beta_{i,k}) - \prod_{k=1}^L (1 - w_k)} \quad (5)$$

deliberate choice by the author as not to inflate the number of different symbols used in this paper and to keep the notation similar to the notation found in the source literature.

The rule-base in a BRB system consists of L IF...THEN... rules R_k , where an expression follows the IF, and an action – referred to as a consequent – follows the THEN. An input to a rule is a vector \mathbf{x} of attributes with T elements $x_i, i \in [1, T]$. The attribute values x_i are restricted to the values present in the antecedents of the rule base defined to be a set of T vectors $A_i, i \in [1, T]$ with each vector consisting of referential values. Then, $A_{i,j}$ denotes the j th available referential values for the i th attribute x_i , where the number of these referential values may vary from rule to rule. The vectors of referential values are aggregated in a set of vectors $A = \{A_i \mid i \in [1, T]\}$ called the packet antecedent of the rule base.

Similarly, the output of each rule is restricted to N referential values in a packet consequent $D = \{D_i \mid i \in [1, N]\}$. The element D_i indicates the i th available referential value for the consequents in the rule base.

Each rule R_k is also associated to a vector of belief-degrees $\beta_{i,k}$, where the index k indicates to which rule the degrees are associated to, and the index i indicates the belief-degree associated to the i th referential value $D_i \in D$. The value of the belief-degree indicates the likelihood for a referential value D_i being the consequent of a rule when the expression of the rule is true. Additionally, each rule R_k has an associated rule weight θ_k indicating how important the rule is in the rule base, and an associated set of attribute weights $\delta_{i,k}$, indicating how important the i th attribute is in the expression of the rule.

The ideas presented so far can be summarized in the definition of a single belief-rule with the AND-connective ‘ \wedge ’ aggregating the expressions:

$$\begin{aligned} & \text{IF } x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge \dots \wedge x_{T_k} \text{ is } A_{T_k}^k, \\ R_k : & \text{ THEN } \{(D_i, \beta_{i,k}) \mid i \in [1, N]\}, \\ & \text{with an associated rule weight } \theta_k, \\ & \text{and attribute weights } \{\delta_{i,k} \mid i \in [1, T_k]\}, \end{aligned} \quad (6)$$

where A_i^k is an antecedent referential value vector, such that $A_i^k \in A_i$, and T_k is the number of attributes in the input of rule R_k . The parameters $\beta_{i,k}, \theta_k, \delta_{i,k}$, and A , can be set by an expert or learned.

Observed data H consisting of the elements $h_i, i \in [1, T]$, is transformed to a belief distribution before it can be utilized in a BRB system. Given a packet antecedent A , h_i can be transformed into the belief distribution

$$\alpha_i = \left\{ \min \left(\max \left(\frac{A_{i,j+1} - h_i}{A_{i,j+1} - A_{i,j}}, 0 \right), \max \left(\frac{h_i - A_{i,j-1}}{A_{i,j} - A_{i,j-1}}, 0 \right) \right) \mid j \in [1, |A_i|] \right\}, \quad (7)$$

where $|A_i|$ is the number of elements in A_i , $A_{i,0} = A_{i,T}$, and $A_{i,|A_i|} = A_{i,1}$. That is to say, that the n th element in a belief distribution α_i denoted by $\alpha_{i,n}$ indicates how well an observed attribute h_i matches the referential value $A_{i,n}$ of a rule base.

Once all h_i have been transformed to a belief distribution α_i , an activation weight can be calculated for each rule as

$$w_k = \theta_k \prod_{i=1}^{T_k} (\alpha_{i,j}^k)^{\bar{\delta}_{i,k}} / \sum_{l=1}^L \left[\theta_l \prod_{i=1}^{T_l} (\alpha_{i,j}^l)^{\bar{\delta}_{i,l}} \right], \quad (8)$$

where $\bar{\delta}_{i,k} = (\delta_{i,k}) / (\max_{i \in T_k} (\delta_{i,k}))$. The sub-indices j in the expressions $\alpha_{i,j}^{k/l}$ are determined by each belief-rule R_k so that the values in $\alpha_{i,j}^{k/l}$ convey the information on how the observed input H matches to the rule’s antecedent values A^k . Then, the activation weights indicate how important each rule is given a certain input to the rule base.

Using the computed activation weights, the combined belief degrees β_n can be calculated using (5). A combined belief degree β_n conveys the likelihood for an input \mathbf{x} to be mapped to the consequent referential value $D_n \in D$ in the rule base. Using a function $\mathbb{R} \rightarrow \mathbb{R} : u(D_i)$, the combined belief degrees can be transformed into a numerical output as

$$y = \sum_{i=1}^N u(D_i) \beta_i. \quad (9)$$

The belief-degrees $\beta_{i,k}$, the rule weights θ_k , the attribute weights $\delta_{i,k}$, and the antecedent referential values A_i , can be denoted by $P(\beta_{i,k}, \theta_k, \delta_{i,k}, A_i)$. A minimization problem is then defined

$$\min_{P \in \mathbf{P}} \xi(P), \quad (10)$$

where ξ is a cost function used to train the BRB system, and \mathbf{P} denotes the set of feasible parameters P , which is defined by the following constraints:

$$\sum_{i=1}^L \theta_i = 1, \quad (11)$$

$$\sum_{i=1}^L \beta_{i,k} = 1, k \in [1, L], \quad (12)$$

$$0 \geq \delta_{i,k}, i \in [1, T] \text{ and } k \in [1, L], \quad (13)$$

$$A_{i,n} \leq A_{i,n+1}, i \in [1, T] \text{ and } n \in [1, |A_i| - 1], \quad (14)$$

where (11) and (12) are normalizations, (13) limits the attribute weights to positive values to keep the multiplicand in (8) less or equal to one, and (14) is used to keep the antecedent referential values ordered, which is assumed in (7).

By formulating the cost function ξ to reflect a difference between the numerical output (9) of the BRB system and a desired output, an optimization problem defined in (10) can be formed using the formulated cost function, and can then be solved for a set of optimal parameters $\hat{P} \in \mathbf{P}$, which are then used to update the BRB system to better reflect a desired output effectively training the BRB system.

III. METHOD

A. The INFRINGER method

An interactive method for solving multi-objective optimization problems, the INFRINGER¹ method, is proposed. The method models a decision maker's preferences as a value function, which is learned utilizing a BRB system. The BRB system in the method is adaptively trained using a cost function, which is formulated based on preference information gathered during an interactive process where a decision maker is asked to conduct pair-wise comparisons of Pareto optimal objective vectors.

As its input, the INFRINGER method requires: (i) a set of objective vectors representing a Pareto optimal set Z^{Pareto} in a multi-objective optimization problem; (ii) the ideal point \mathbf{z}^* and the nadir point \mathbf{z}^{nad} of the set Z^{Pareto} ; and (iii) a fitness threshold $\gamma_{\text{th}} \in [0, 1]$ used as a termination criterion. The output of the method consists of: (i) a trained BRB system able to model the decision maker's preferences using a learned value function; and (ii) the vector in Z^{Pareto} with the highest value according to the learned value function. It is assumed that the vectors in Z^{Pareto} consists of real valued objectives scaled between $[0, 1]$ using the nadir and ideal points, such that the nadir point's objectives are zero and the ideal point's objectives are one.

In the INFRINGER method, it is assumed that the given set of objective vectors representing the Pareto optimal set is sufficiently diverse and spread-out in such a way that the set can be assumed to represent the whole extent of the Pareto optimal set. This kind of input set is easily computed for a discrete data-based multi-objective optimization problem modelled after a small and limited data set. However, for non-discrete problems or problems modelled using very large amounts of data, it is not always possible to know if a representation of the Pareto optimal set is comprehensive enough to warrant the assumptions made in the INFRINGER method. Therefore, the INFRINGER method may act more as an a posteriori multi-objective optimization method instead of an interactive method, if the given representation of the Pareto optimal set is not comprehensive enough.

B. Initialization

The decision maker is shown the nadir and ideal points and is then asked whether he/she would like to give a reference point $\bar{\mathbf{z}}$, a vector consisting of desired objective values. If the decision maker does not wish to give a reference point, the reference point is set to be in the middle of the nadir and ideal points.

Next, the decision maker is shown the approximation of the Pareto optimal set alongside the reference, nadir, and ideal points. This is done to give the decision maker an idea of the available objective values.

A BRB system is then initialized with a packet antecedent A with m referential value vectors. Each A_i corresponds to the i th objective, and consists of three referential values: $A_i =$

$\{z_i^{\text{nad}}, \bar{z}_i, z_i^*\}$, where the index i refers to the i th component in each of the vectors. The Cartesian product of the referential value vectors in A is then taken resulting in 3^m new vectors labelled as A^k , where $k \in [1, 3^m]$. These vectors are then used to construct $L = 3^m$ belief-rules (6), where the rule R_k has the antecedent referential values A^k .

The referential values for the consequent D in each rule are chosen to be $D = \{0, 0.25, 0.5, 0.75, 1\}$, because it is assumed that the numerical output of the BRB system modelling a value function is limited to the continuous range $[0, 1]$. The choice of five referential values in D has been a result of trial and error – the choice has shown to yield a good balance between computational efficiency and performance of the BRB system.

To define the belief-degrees β_i for each rule R_k , an initial value function (2) must be assumed. The value function is assumed to be

$$f_p(\mathbf{z}) = \sum_{i=1}^m z_i/m, \quad (15)$$

which is chosen, because it is simple and clearly coherent with the assumptions made in Section II-B. The initially assumed value function (15) could also be chosen differently without impairing the described method. Using the principle (15), and an identity function $u(D_i) = D_i$ in (9), a numerical output y^k can be computed for each A^k . This numerical output can then be transformed into the belief degrees of each rule R_k using (7) by substituting α_i with β_i , $A_{i,*}$ with D_* , and h_i with y^k . The rest of the parameters in the initial BRB system are set to be $\theta_k = 1/L$, and $\delta_{i,k} = 1$, for all indices i, k , because no further assumptions are made regarding the modelled value function.

The initial belief-rule base has now been fully defined, and a numerical output can be computed for an objective vector \mathbf{z} utilizing (7), (8), (5), and (9). This output represent a value associated to \mathbf{z} according to the value function modelled by the BRB system. As a short hand notation

$$\mathfrak{B}(\mathbf{z}) = \text{"value for } \mathbf{z}\text{"} \quad (16)$$

is used to indicate the numerical output of the BRB system.

C. Pair-wise comparisons and fitness evaluation

Following the initialization of the BRB system, the decision maker is shown pairs of vectors in Z^{Pareto} . Five pairs are chosen to be shown, because it is a number of pairs low enough to be grasped by the decision maker, and high enough to still offer a decent representation of the available vectors. To choose pairs with enough variety, the standard deviation and mean are calculated from a distribution of values computed using (16) to calculate a value for each vector in Z^{Pareto} . The first three pairs are selected by finding two vectors, both being a multiple of the standard deviation apart from each other, centered on the mean. The multiples of the standard deviation are one, two, and three, for the first three pairs selected, respectively. The fourth pair consists of the vectors with the highest and lowest values in Z^{Pareto} according to (16), and the fifth pair consists of two random vectors in Z^{Pareto} .

The decision maker is then asked for each pair if he/she prefers the first vector, the second, or if the vectors are equally

¹Which stands for *Interactive inference of preferences using belief rules*

preferable. According to his/her answers, it is assumed that the value function of the decision maker results in a higher value for the preferred vector, and an identical value for two equally preferred vectors. This idea is contrasted to the output of the BRB system (16), based on which a fitness γ is calculated. The maximum value of the fitness γ is 1, which indicates that the values of \mathfrak{B} computed for both vectors in all the pairs agree with the preferences of the decision maker. A fitness value of 0 indicates the opposite. For example, in case of five pair-wise comparisons conducted, a fitness value of 0.6 would indicate that the BRB system compares three out of five pairs similarly to how the decision maker compared them. The preference information gathered in the pair-wise comparisons is also used in a cost function described in the next subsection.

D. Adaptive training and visualization

To adaptively train the BRB system to model the decision maker's preferences using a value function model, a cost function is defined and minimized. Three important assumptions can be made regarding the value function being modelled by the BRB system: the value of the nadir point should be worse than any other vector in Z^{Pareto} , the ideal point should correspondingly have the highest possible value, and the value function should be monotonically increasing as a function of the objective values. Additionally, a fourth assumption is also made: the modeled value function should reflect the preferences of the decision maker according to the pair-wise comparisons conducted in the manner described in Section III-C. To reflect the assumptions made, a cost function is defined for each assumption, which has a higher value the greater the disagreement between the modelled value function and the made assumption is. The cost function used in the INFRINGER method is therefore defined as the aggregation of four cost functions as

$$\xi(P \rightarrow \mathfrak{B}) = \xi_{\text{nadir}} + \xi_{\text{ideal}} + \xi_{\text{mono}} + \xi_{\text{pair}}, \quad (17)$$

where the notation $P \rightarrow \mathfrak{B}$ indicates that the underlying BRB system is updated with the parameters P before the cost function is evaluated, ξ_{nadir} has a higher value the greater the absolute difference between $\mathfrak{B}(z^{\text{nad}})$ and 0 is, ξ_{ideal} has a higher value the greater the absolute difference between $\mathfrak{B}(z^{\text{ideal}})$ and 1 is, ξ_{mono} has a greater value the less monotonically increasing \mathfrak{B} is, and ξ_{pair} is the reciprocal of the fitness γ discussed in Section III-C.

After a cost function (17) is defined, a minimization problem (10) is solved to find the optimal parameters \tilde{P} . Using the optimal parameters \tilde{P} in the BRB system will result in a modelled value function \mathfrak{B} congruent with the assumptions made regarding (17) in the previous paragraph. The modelled value function should therefore reflect the preferences of the decision maker based on the pair-wise comparisons conducted, and abide the assumptions made regarding the general nature of a value function discussed in Section II-B.

To further improve the modelled value function in its ability to model the decision maker's preferences, more pair-wise comparisons can be conducted, and the cost function ξ_{pair}

Algorithm 1 The INFRINGER method

INPUT: A representation of the Pareto set Z^{Pareto} , the ideal and nadir points z^{nad}, z^* , and a fitness threshold γ_{th} .

OUTPUT: A BRB system trained to model a decision maker's preferences using a value function, and a desired number of the highest valued vectors in Z^{Pareto} according to the value function modeled by the BRB system.

Step 1: Show the decision maker z^* and z^{nad} , and ask if he/she would like to supply a reference point \bar{z} with objective values bound by z^{nad} and z^* .

Step 2: If \bar{z} was given in Step 1, continue to Step 3. Otherwise define \bar{z} to be an objective vector in the middle of z^{nad} and z^* .

Step 3: Initialize a BRB system as described in Section III-B.

Step 4: Conduct pair-wise comparisons of the selected vectors in Z^{Pareto} as described in Section III-C.

Step 5: Formulate the cost function and minimize it as described in Section III-D. Update the BRB system with the optimal parameters \tilde{P} found.

Step 6: Visualize the ranked Z^{Pareto} to the decision maker as described in Section III-D using the current value function modelled by the BRB system.

Step 7: Conduct a new pair-wise comparisons of vectors in Z^{Pareto} and compute a fitness γ value as described in Section III-C.

Step 8: If the computed $\gamma \geq \gamma_{\text{th}}$ or the decision maker wishes to stop, go to Step 11. Otherwise continue to Step 10.

Step 9: Update the cost function related to the pair-wise comparisons ξ_{pair} in (17) and minimize it as described in Section III-D. Update the BRB system with the optimal parameters \tilde{P} found and go to Step 7.

Step 10: Visualize the highest valued vectors in Z^{Pareto} as described in Section III-E.

in (17) can be updated with the new preference information gathered from the comparisons. Previously conducted pair-wise comparisons are kept in the formulation of ξ_{pair} because it is assumed that the preferences of the decision maker remain constant.

To suggest the decision maker a best solution, and to give him/her an idea of where their region of interest in Z^{Pareto} might be according to their preferences learned by the BRB system, the decision maker is shown a plot depicting a ranking of the vectors in Z^{Pareto} according to the learned value function \mathfrak{B} . The vector with the highest score in Z^{Pareto} is also indicated in the plot. An example of such a plot can be seen in Figure 3, where colors have been used to indicate the value of each vector in a multi-objective optimization problem with three objectives.

E. Termination

The pair-wise comparisons can be conducted again and the BRB system can be adaptively trained multiple times. After

each adaptive training, the fitness γ of the value function being modelled can be re-evaluated. If the calculated fitness is equal or greater compared to the fitness threshold γ_{th} , the INFRINGER method is terminated. Optionally, the method can be terminated if the decision maker decides he/she does not wish to continue – the decision maker gets tired, or is happy with the solution(s) found. The decision maker is then shown a desired number of the top valued vectors found in the representation of Z^{Pareto} according to \mathfrak{B} .

As for the learned value function, the optimal parameter \tilde{P} found before termination can be saved for later use, for example if the modelled value function of the decision maker is needed later. The INFRINGER method is summarized in Algorithm 1.

IV. SOFTWARE

A software framework to build and train BRB systems discussed in Section II-C has been developed in Python [13]. Based on the aforementioned framework, the INFRINGER method described in Section II-C has also been implemented. The developed framework is a first of its kind. The source code for the developed framework and method is available on GitHub. ²

V. CASE STUDY

In the case study described in this section, the developed INFRINGER method, and its software implementation, were used for solving a multi-objective optimization problem in a forest landscape planning problem. A domain expert acted as the decision maker in the case study.

A. The problem

The problem in the case study consisted of choosing different management strategies for different parts of a forest, which are referred to as forest stands. A strategy could have consisted of chopping down all the trees in the stand and sell the timber, for example. Three indicators were chosen to represent the consequences for the whole forest resulting from choosing different strategies for each forest stand. These indicators were: total average income, average stored carbon-dioxide, and average combined habitat suitability index. The listed indicators were chosen to be the three objectives of a problem being solved and are to be maximized.

Simulated data were available in three CSV³-files, with values for the three objectives for each forest stand. There were 59 available managements strategies, with data for 1475 forest stands whose development over a one hundred year time horizon was simulated for all the available strategies using SIMO [14]. In the case study, only 20 out of the 59 available strategies were considered to lower the computational time for calculating a Pareto optimal objective vector set from the data.

A comprehensive representation of the Pareto optimal set was calculated based on the available data. The nadir and ideal points of the Pareto optimal set were also computed.

The original data used and the source code to compute the Pareto optimal set is available on GitHub⁴.

B. Solving the problem using INFRINGER

In what follows, each time a reference is made regarding a Step followed by a number, it refers to a Step present in Algorithm 1. Analyst refers to the author of this paper who was in charge of operating the INFRINGER method since the case study was conducted over a video call.

The fitness threshold γ_{th} required as one of the inputs to the INFRINGER method was set to be 1 by the analyst. This choice was made in an attempt to avoid premature termination of the method and guarantee the BRB systems to be trained at least a couple of times. This was done to improve the accuracy of the modelled value function in modelling the preferences of the decision maker.

According to Step 1, the decision maker was shown the nadir and ideal points, and the Pareto optimal set. The decision maker then wished to give a reference point, which was used for initializing a BRB system according to Steps 2 and 3. Then, pair-wise comparisons were conducted by the decision maker according to Step 4. The interface shown to the decision maker for conducting the pair-wise comparisons can be seen in Figure 2. A spider plot was also available for each pair-wise comparison to aid the decision maker in comparing the two vectors in each pair. An example of a spider plot shown can be seen in Figure 4. Based on the pair-wise comparisons conducted, the cost function was formulated and minimized according to Step 5. The BRB system was updated with the optimal parameters \tilde{P} found.

The decision maker was then shown the ranked Pareto optimal set according to Step 6. A fitness was calculated afterwards and compared to γ_{th} according to Step 8. The fitness was less than the threshold, and the method was continued according to Step 9. An additional five iterations of Steps 6 through 9 were conducted before the decision maker wished to stop the method, and the highest valued vector was shown to the decision maker as seen in Figure 3 according to Step 10.

C. General remarks

During the case study, the decision maker had made a couple of important remarks that can be used to assess the potential of the proposed INFRINGER method as a tool to assist in decision making: (i) the decision maker said to be happy with the highest valued vectors shown in iterations of Step 6, and he said that he felt that the objective values in the vectors shown were closing to what he was aiming at; (ii) the decision maker said that his preferences had changed during the case study; (iii) the decision maker wished for a fourth preference option to be available in the pair-wise comparisons – there should have been an option of *no preference* indicating that no preference can be given based on the two vectors shown in a pair – and he had chosen the option for equal preference to

²<https://github.com/gialmisi/desdeo-brb> (November 26, 2020)

³Comma separated values.

⁴<https://github.com/gialmisi/forest-opt> (November 26, 2020)

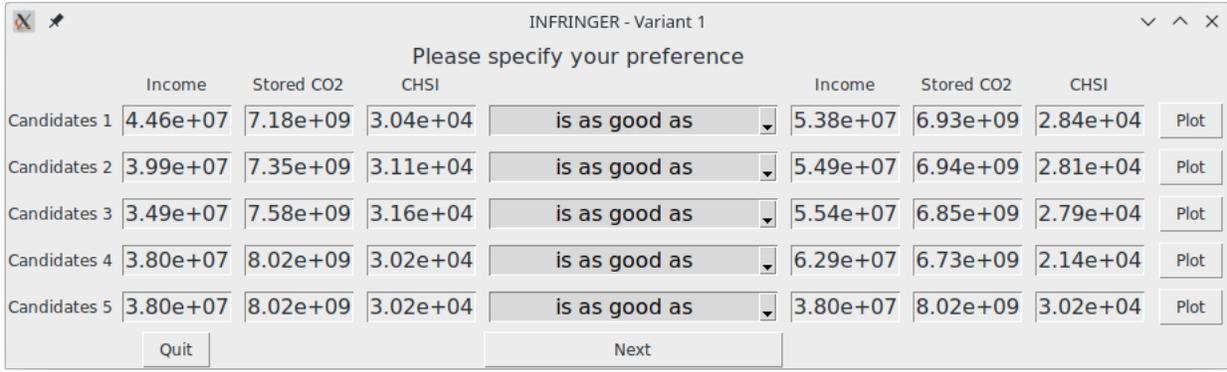


Fig. 2. The interface presented to the decision maker for pair-wise comparison of vectors (candidates) in Z^{Pareto} . For each pair shown, the decision maker may specify a candidate on the left to be either better, worse, or as good as the candidate on the right by choosing an appropriate option from the drop-down menu in the middle. Each pair can be further investigated by clicking the 'Plot' button next to each pair, which will produce a spider plot as shown in Figure 4

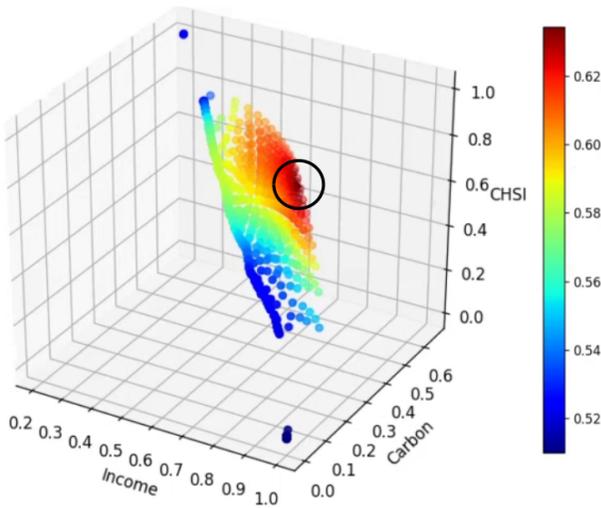


Fig. 3. The ranked Pareto optimal vectors. The highest valued vector, according to the learned preferences, is indicated by the circle. Values are normalized between 0 and 1 according to the limits imposed by the nadir and ideal points. The value of each vector is indicated by the shown color scale. Red tinted vectors have the highest values while blue tinted vectors have the lowest.

indicate absence of preference instead. However, overall the decision maker expressed to be happy in the highest valued objective vectors shown.

Additional graphical and numerical data relevant the case study is available. The material is available by request from the author.

VI. DISCUSSION

The first remark in Section V-C, and the notion of the decision maker being overall happy with the highest valued Pareto optimal objective vectors shown, are clear indicators of the INFRINGER method being able to aid a decision maker to find satisfying solutions. The first remark indicates also that the BRB system was able to learn a value function reflecting the decision maker's preferences in the case study.

However, the decision maker had also clearly stated that his preferences had changed during the case study in the

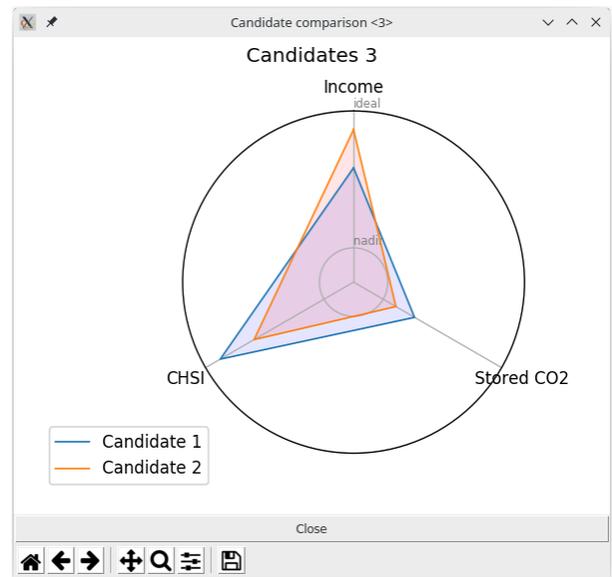


Fig. 4. A spider plot of two vectors in Z^{Pareto} (candidates) shown to the decision maker to aid him/her in comparing the two vectors. In the spider plot shown, the ideal and nadir point's values for each objective are also shown (outer radius and inner radius for the ideal and nadir values, respectively).

second remark – a clear contradiction to the assumption made regarding the preferences of the decision maker made at the end of Section III-D. The change in preferences can be attributed, for example to the decision maker learning about the existing trade-offs between the Pareto optimal objective vectors. This would indicate that incorporating the previously gathered preference information regarding the cost function ξ_{pair} is not correct, and it should be reconsidered whether preference information gathered in previous iterations should be used in training the BRB system in the following iterations as well.

It was also evident from the third remark that an option of no preference should be available in the pair-wise comparisons. During the case study, the option for equal preference had been used, when the decision maker wished to actually give

no preference at all. This has probably lead to some error in the learned value function, and the absence of a *no preference* option should be addressed in future research related to INFRINGER.

Moreover, the comments made in Section III-A suggest that the INFRINGER method can be improved to work as an interactive method in cases where a comprehensive representation of the Pareto optimal set is not available. For example, the INFRINGER method could be enhanced by replacing the input consisting of a pre-computed set of objective vectors representing the Pareto optimal set with an evolutionary search method, which can be used to find a set of non-dominated objective vectors according to the current preference model learned in each iteration of the INFRINGER method.

Lastly, many of the choices regarding the INFINGER method have been a result of trial and error, which has been the method of choice for defining the parameters because of the novel nature of the method – there is no existing literature which would suggest choices for optimal parameters for the method proposed in this paper. Some examples of such choices are the number of vectors shown in the pair-wise comparisons, the number of antecedent and consequent referential values in the BRB system, and the choice of the initially assumed value function (15). Additional research should be conducted to find the optimal choices for these parameters.

VII. CONCLUSION

Based on the results of the case study discussed in Section VI, it is concluded that the proposed novel INFRINGER method is able to aid a decision maker in reaching satisfying solutions in a multi-objective optimization problem, and that the method is able to model the value function of the decision maker. However, it is also evident that the method can be improved further with additional research. Lastly, it must be noted that the BRB system used in the proposed method has the potential to offer explainable results regarding the modelled preferences of the decision maker. The explainability of the learned preferences should therefore also be considered in future works, and may aid in the choosing of the internal parameters of the BRB system.

ACKNOWLEDGMENTS

Thanks for Jian-bo Yang, Ling Xu, and Swati Sachan, for the insightful discussion sessions over Skype.⁵ Thanks for Kaisa Miettinen and Jussi Hakanen for their guidance related to my master's thesis to which this paper is based on, and for their feedback regarding this paper; and for Jose Malmberg whose code, which he developed in his master's thesis has been partly utilized to compute the Pareto set in the case study.⁶ And lastly, thanks for Kyle Eyvindson for acting as the decision maker in the case study.

The research in this paper is related to the thematic research area DEMO: Decision Analytics utilizing Causal Models and

Multi-objective Optimization of the University of Jyväskylä.⁷ This research has been funded by the Academy of Finland: project number 322221.

REFERENCES

- [1] L. G. More, M. A. Brizuela, H. L. Ayala, D. P. Pinto-Roa, and J. L. V. Noguera, "Parameter tuning of CLAHE based on multi-objective optimization to achieve different contrast levels in medical images," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015.
- [2] R. Marler and J. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004.
- [3] K. Miettinen, *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1998.
- [4] S. Sachan, J.-B. Yang, D.-L. Xu, D. Benavides, and Y. Li, "An explainable AI decision-support-system to automate loan underwriting," *Expert Systems with Applications*, 2019.
- [5] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?" in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. ACM Press, 2016.
- [6] R. Battiti and A. Passerini, "Brain-computer evolutionary multiobjective optimization: A genetic algorithm adapting to the decision maker," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 671–687, 2010.
- [7] J. Branke, S. Greco, R. Slowinski, and P. Zielniewicz, "Learning value functions in interactive evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 88–102, 2015.
- [8] R. L. Keeney, H. Raiffa, and D. W. Rajala, "Decisions with multiple objectives: Preferences and value trade-offs," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 7, pp. 403–403, 1979.
- [9] S. Bechikh, L. Ben Said, and K. Ghedira, "Estimating nadir point in multi-objective optimization using mobile reference points," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–9.
- [10] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu, "Interactive multiobjective optimization: A review of the state-of-the-art," *IEEE Access*, vol. 6, pp. 41 256–41 279, 2018.
- [11] J.-B. Yang, J. Liu, J. Wang, H.-S. Sii, and H.-W. Wang, "Belief rule-base inference methodology using the evidential reasoning approach-RIMER," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 36, no. 2, pp. 266–285, 2006.
- [12] Y.-W. Chen, J.-B. Yang, D.-L. Xu, Z.-J. Zhou, and D.-W. Tang, "Inference analysis and adaptive training for belief rule based systems," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12 845–12 860, 2011.
- [13] P. S. Foundation. (2020) Python language reference, version 3.8. [Online]. Available: <https://www.python.org>
- [14] J. Rasinmäki, A. Mäkinen, and J. Kalliovirta, "SIMO: An adaptable simulation framework for multiscale forest resource data," *Computers and Electronics in Agriculture*, vol. 66, no. 1, pp. 76–84, 2009.

⁵From the University of Manchester.

⁶From the University of Jyväskylä

⁷<https://www.jyu.fi/demo> (November 26, 2020)