Using Choquet Integral as Preference Model in Interactive Evolutionary Multiobjective Optimization

Juergen Branke[a], Salvatore Corrente[b], Salvatore Greco[b,e], Roman Słowiński[c,d], Piotr Zielniewicz[c]

[a] *Warwick Business School, The University of Warwick, Coventry, CV4 7AL, United Kingdom*
[b] *Dept. of Economics and Business, University of Catania, Corso Italia, 55, 95129, Catania, Italy*
[c] *Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland*
[d] *Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland*
[e] *University of Portsmouth, Portsmouth Business School, Centre of Operations Research and Logistics (CORL), Portsmouth PO1 3DE, United Kingdom*

Abstract

We propose an interactive multiobjective evolutionary algorithm that attempts to discover the most preferred part of the Pareto-optimal set. Preference information is elicited by asking the user to compare some solutions pairwise. This information is then used to curb the set of compatible user's value functions, and the multiobjetive evolutionary algorithm is run to simultaneously search for all solutions that could potentially be the most preferred. Compared to previous similar approaches, we implement a much more efficient way of determining potentially preferred solutions, that is, solutions that are best for at least one value function compatible with the preference information provided by the decision maker. For the first time in the context of evolutionary computation, we apply the Choquet integral as a user's preference model, allowing us to capture interactions between objectives. As there is a trade-off between the flexibility of the value function model and the complexity of learning, a faithful model of user's preferences, we propose to start the interactive process with a simple linear model but then to switch to the Choquet integral as soon as the preference information can no longer be represented using the linear model. An experimental analysis demonstrates the effectiveness of the approach.

Keywords: Multiobjective Optimization, Evolutionary Algorithms, Interaction, Choquet Integral.

---

1. Introduction

Multiobjective optimization involves several conflicting objectives that compete for the best solution in a constrained multidimensional space of decision variables. In general, there is no single optimal solution (as in single-objective optimization), but a set of alternatives for which it is not possible to improve one objective without deteriorating another one, called Pareto-optimal solutions. Despite the existence of multiple Pareto-optimal solutions, in practice, usually only one of these solutions is to be chosen. Thus,

---

Email addresses: `Juergen.Branke@wbs.ac.uk` (Juergen Branke), `salvatore.corrente@unict.it` (Salvatore Corrente), `salgreco@unict.it` (Salvatore Greco), `roman.slowinski@cs.put.poznan.pl` (Roman Słowiński), `piotr.zielniewicz@cs.put.poznan.pl` (Piotr Zielniewicz)

in multiobjective optimization, there are two equally important tasks: an optimization task for finding Pareto-optimal solutions by a search procedure, and a decision aiding task for recommending a single most preferred solution. The "most preferred" refers to the value system of a particular user, also called decision maker (DM). Thus, decision aiding necessitates some preference elicitation from the user.

As to procedures searching for Pareto-optimal solutions, in the last two decades we have been able to observe a growing popularity of algorithms adopting the principles of natural evolution. A distinguishing feature of these evolutionary algorithms is that they work with a population of solutions. This is of particular advantage in the case of multiobjective optimization, as they can search for several Pareto-optimal solutions simultaneously in one run, providing the user with a set of feasible solutions to choose from. In the early stage of development of evolutionary algorithms for multiobjective optimization, the efforts were focused on efficient generation of the whole set of Pareto-optimal solutions (or of a good approximation thereof), leaving the decision aiding task to the post-optimization stage. Later, researchers started interlacing the optimization and preference handling in interactive procedures, which allows to converge more quickly to the most preferred region of the Pareto-optimal front [8].

Most interactive procedures for multiobjective evolutionary optimization assume a particular mathematical model of user's preferences. This model drives the search procedure towards the most preferred Pareto-optimal solutions. The model building involves preference information supplied by the user. In case of simple preference models, one may expect that the user can provide directly the values of model parameters. However, simple models, like the weighted sum of objectives, fail to represent more subtle user's preferences (see Section 1.3). For this reason, there is a tendency to use more complex preference models built from indirect preference information which is much easier to elicit by the user than the direct preference information. In many previous studies on interactive multiobjective optimization, the indirect preference information had the form of pairwise comparisons of some solutions from a current population [10, 11, 28, 37]. In particular, the NEMO-I[1] method [10] is defining a set of value functions compatible with the preferences elicited from the DM, expressed in terms of pairwise comparisons of solutions. These compatible value functions are used in Robust Ordinal Regression (ROR) [17, 27] to build a *necessary preference relation* ($\succsim^N$) on the current population of solutions. More precisely, $a \succsim^N b$ if a is at least as good as b for all compatible value functions. NEMO-I is adopting the scheme of NSGA-II [19], however substituting the dominance relation by the necessary preference relation in the ranking. While NEMO-I has shown a satisfactory convergence to the best compromise solution, the calculation of the necessary preference relation requires a considerable computational effort. Therefore, in this paper, we are using an alternative method called NEMO-II, which overcomes the problem of prohibitive computational effort. NEMO-II accepts any type of value function. We are considering four types of value functions within NEMO-II: linear, additive piecewise-linear, general additive, and (for the first time in combination with evolutionary multiobjective algorithms) the Choquet integral [14, 23].

---

[1] NEMO: Necessary preference enhanced Evolutionary Multiobjective Optimizer.

The paper is organized as follows. In Section 2, we review interactive evolutionary multiobjective algorithms (MOEAs). Then, in Section 3, we present the general scheme of NEMO-II. Further, in Section 4, we describe the four above mentioned value functions. The new procedure, called NEMO-II-Ch, is introduced in Section 5. A computational experiment with the proposed procedure and its main competitors on a set of benchmark problems is presented in Section 6. Section 7 summarizes our conclusions and suggests avenues for future research.

1.1. Interactive Evolutionary Multiobjective Optimization

Evolutionary multiobjective optimization (EMO) has become very popular because of its ability to generate a set of non-dominated solutions in one run, from which the DM can choose a favorite solution without eliciting any preference information a priori. Nonetheless, in recent years, there has been a growing interest in EMO algorithms that are able to take into account user's preference information in the search process. This is motivated by the following expected advantages.

1. Instead of a diverse set of solutions (many of them clearly irrelevant to the DM) a search based on the DM's partial preferences will provide a more suitable sample of all Pareto-optimal solutions. It could either be a smaller set of only the most relevant solutions, or a more fine-grained resolution of the relevant parts of the Pareto frontier.
2. By focusing the search onto the relevant part of the search space, one may expect the optimization algorithm to find these solutions more quickly.
3. As the number of objectives increases, it becomes more and more difficult to identify and represent the complete Pareto-optimal frontier. This is partly because of the increasing number of Pareto-optimal solutions, but also because with an increasing number of objectives almost all solutions in the population become non-dominated, rendering dominance as selection criterion useless. User's preference information allows re-introducing the necessary selection pressure.

The literature contains today quite a few techniques that allow the incorporation of full or partial preference information into MOEAs, and previous surveys on this topic include [5, 6, 15, 16, 39].

Many of the techniques integrate partial user's preferences a priori, e.g., by allowing the DM to specify a reference point [21, 29, 41], maximal and minimal trade-offs [7], or desirability functions [45]. In the following, we focus on the literature that is most related to our paper, namely *interactive* approaches that learn user's preferences over the course of the optimization based on *the DM's (partial) ranking of small sets of solutions*. They allow to accumulate preference information and thus refine the internal preference model over time, and because they engage the DM in the optimization process, they initiate a learning process on the DM's side as well. These approaches are surveyed in the following, divided into methods that attempt to learn a representative user's value function, and those that focus on the set of value functions compatible with the elicited preference information. Some interactive approaches based on other paradigms include [18, 31, 41, 43].

We will use the following notation: A is the set of solutions in a considered population; $A^R \subseteq A$ is the set of reference solutions in population A; $f_1, \ldots, f_n$ are n objective functions such that each solution $a \in A$ is associated with the vector of evaluations $(f_1(a), \ldots, f_n(a))$. For simplicity, we sometimes only write a_j instead of $f_j(a)$. Unless specified otherwise, we suppose, without loss of generality, that the objective functions have to be maximized.

1.1.1. Approaches to learn a value function representing user's preferences

The approaches in this subsection use the elicited preference information to derive a single value function to approximate user's preferences. Value functions can have different complexity, ranging from simple linear functions to the highly non-linear functions considered in the non-parametric approaches, such as artificial neural networks or support vector machines. Most approaches simply use the derived value function for ranking individuals, sometimes as secondary criterion after non-dominance, but other uses can also be found.

Phelps and Köksalan [37] proposed an interactive evolutionary algorithm that periodically asks the DM to compare pairs of solutions. Assuming **linear** value functions (actually, the objectives are modified before the optimization to the squared distance from a reference value, which effectively results in ellipsoidal iso-utility curves), the method determines the most discriminant weight vector compatible with the preference information. Most discriminant here means the weight vector that maximizes the minimum value difference over all pairs of solutions compared by the DM.

Denote by $\succ_p$ the binary relation on the set A^R, representing the preference information provided by the user in terms of pairwise comparisons. Then, the following linear program (LP) identifies the most discriminant value function:

$$\left.\begin{array}{l} \max \varepsilon, \text{ subject to} \\ \displaystyle\sum_{j=1}^{n} w_j f_j(a) - \sum_{j=1}^{n} w_j f_j(b) \geq \varepsilon, \quad \text{for all } a \succ_p b \\ \displaystyle\sum_{j=1}^{n} w_j = 1, \quad w_j \geq 0. \end{array}\right\} \tag{1}$$

The resulting weight vector is then used for ranking individuals in the evolutionary algorithm that works as a single objective evolutionary algorithm between user interactions. If the LP is overconstrained and no feasible solution is found, the oldest preference information is discarded until feasibility is restored.

Deb et al. [20] derive a **polynomial** value function model. The user is shown a set of (five in the paper) solutions and asked to (at least partially) rank them. Then, similar to the approach by Phelps and Köksalan [37], the most discriminant value function is determined. However, due to the polynomial value function model, fitting the model to the specified preferences is a non-linear optimization problem, and the authors propose to use sequential quadratic programming to solve it. The most discriminant value function is used in the MOEA's ranking of individuals. Basically, the objective space is separated into two areas: all individuals with an estimated value (according to the approximated value function) better than the solution ranked *second* by the DM are assumed to dominate all the solutions with an estimated value worse than the solution ranked second. The authors additionally use the approximated value function to perform a local single-objective optimization starting with the solution ranked best by the DM.

Todd and Sen [44] use **artificial neural networks** to represent the DM's value function. Periodically, they present the DM with a set of solutions and ask for a score. The set of solutions is chosen such that they represent a broad variety regarding the approximated value function, and the estimated best and worst individual of the population are always included. Information from several interactions is accumulated after normalizing preference scores.

Another model that allows representation of complex value functions are **support vector machines** (SVM). Battiti and Passerini [4] use SVMs in the setting of an interactive MOEA. Periodically, the DM is presented with a set of solutions and asked to (at least partially) rank them. This information is then used to train the SVM, with cross-validation employed to select an appropriate kernel. The derived approximate value function is then used to sort individuals in the same non-dominance rank based on their value according to the learned value function. The paper examines the influence of the number of solutions shown to the DM (assuming full ranking) and the number of interactions with the DM. The results suggest that a relatively large number of solutions need to be ranked for the SVM to learn a useful value function (around 10-20), but only two interactions with the DM seem sufficient to come very close to results that would have been obtained had the DM's "*true value function*" been known from the beginning. The authors recommend to not start interaction until the MOEA has found a reasonable coverage of the entire Pareto frontier, which somewhat defeats the purpose of narrowing down the search early on. In [12], the approach's robustness to incorrect (noisy) DM preferences is examined and it is shown that the algorithm can cope well with noise.

Branke et al. [11] have recently compared various ways to define a representative value function and found that the function that maximizes the sum of values of individuals in the population actually performed slightly better than the most discriminant value function, and much better than a value function that minimizes slope changes. They showed that their approach, called NEMO-0, which is able to learn arbitrary monotonic additive value functions, can perform well in cases where a linear value function model is not sufficient to represent the user's preferences.

1.1.2. Approaches to learn a set of value functions representing user's preferences

Rather than deriving a single value function, Jaszkiewicz [30] notes that there may be several value functions compatible with the specified user's preferences and samples the preference function used in each generation from the set of preference functions (in this case **linear** weightings are assumed). The proposed approach uses the value function also for local search. In the interactive version, preference information from pairwise comparisons of solutions is used to reduce the set of possible weight vectors.

Greenwood et al. [28] suggested the imprecise value function approach which considers all compatible **linear** value functions *simultaneously.* The procedure asks the user to rank a few solutions, and from this derives constraints for the weightings of the objectives consistent with the given ordering. Then, these are used to check whether there is a feasible linear weighting such that solution a would be preferred to solution b.

Then, to compare any two solutions a and b from set A, one has to consider all value functions compatible with the user's preferences: a is considered at least as good as b if for all compatible value functions a gets a value not smaller than b. To make this conclusion, the following linear program (LP) has to be solved:

$$\begin{array}{l} \varepsilon^* = \max\ \varepsilon, \ \text{ subject to} \\ \left.\begin{array}{l} \displaystyle\sum_{j=1}^{n} w_j f_j(b) - \sum_{j=1}^{n} w_j f_j(a) \geq \varepsilon \\ \displaystyle\sum_{j=1}^{n} w_j f_j(c) - \sum_{j=1}^{n} w_j f_j(d) \geq \varepsilon, \ \text{ for all } \ c \succ_p d \\ \displaystyle\sum_{j=1}^{n} w_j = 1, \quad w_j \geq 0. \end{array}\right\} E^N(a,b) \end{array} \tag{2}$$

If the set of constraints $E^N(a,b)$ is infeasible or $\varepsilon^* \leq 0$, it can be concluded that there is no compatible value function such that b would be strictly preferred to a, and therefore a is at least as good as b for all compatible value functions. If $\varepsilon^* > 0$, then we know that b is possibly preferred over a, and we would proceed by checking whether b is always at least as good as a by solving $E^N(b,a)$. Overall, the method requires to solve one or two LPs for each pair of solutions in the population.

In [34], the value function model is only implicit. Under the assumption of **quasi-concave** value functions, specified preferences between solutions can be generalized to preference cones. This idea is used by Fowler et al. [22] to partially rank the non-dominated solutions in an MOEA. The DM is asked to consider a set of six solutions and specify the best and worst. From this information, six preference cones are derived (five 2-point cones involving the best and any of the other solutions, and one 6-point preference cone specifying that five solutions are better than the worst). All generated cones are kept throughout the optimization run, even if the solutions defining the cone are deleted from the population. The solutions shown to the DM are selected from the set of non-dominated solutions that cannot already be ranked with the existing cones.

Branke et al. [9, 10] proposed a method called NEMO-I. It is similar to the imprecise value function approach by Greenwood et al. [28], but rather than being restricted to linear value functions, it allows for **piecewise-linear** [10] or **general monotonic additive** [9, 10] value functions. NEMO-I replaces the use of the dominance relation in the non-dominance sorting step of NSGA-II by the necessary preference relation. Additionally, it computes a representative value function used for scaling in the crowding distance calculation.

The procedure used in NEMO-I is computationally very expensive because it requires solving at least one LP for each pair of solutions $(a,b) \in A \times A$. This means that, for a population composed of s solutions, one has to solve up to $s(s-1)$ LP problems in each iteration where we get new preference information. For this reason, in this paper, we are using a new variant, called NEMO-II, which requires significantly less computational effort. It has first been proposed in [11] as part of a general framework but it is implemented here for the first time.

1.2. General scheme of NEMO-II

In this section, we shall introduce the NEMO-II method presented as Algorithm 1. The procedure starts, as a classical MOEA, by randomly generating a population of solutions. Then, after ordering the population into fronts by using the dominance relation, two solutions in the first non-dominated front are chosen randomly to be compared by the user. The preference of the user on this pair of solutions is converted into a linear inequality involving the unknown value function. After introducing these constraints to the set of constraints defining the set of value functions compatible with the user's preferences, one has to check if the augmented set of constraints is feasible, which means that there exists at least one value function compatible with the preferences of the user. If this is not the case, a sufficient number of constraints being the cause of the inconsistency should be removed. For this reason, the procedure starts with removing the constraints representing the oldest pairwise comparisons. After the set of constraints becomes feasible, the method tries to reintroduce the removed constraints that were not the cause of the infeasibility in a reverse order (therefore starting from the newest pairwise comparison) as long as the feasibility is maintained.

The ordering of the population is done as in NSGA-II, putting the solutions into different fronts but, differently from NSGA-II, NEMO-II does not use the dominance relation. For each solution a in the current population A, NEMO-II checks whether there exists at least one compatible value function for which a is the most preferred solution in the current population, by solving the following LP problem

$$\begin{array}{l}\varepsilon_a = \max \varepsilon, \ \text{ subject to} \\ \left.\begin{array}{ll} U(a) - U(b) \geq \varepsilon \ \text{ for all } b \in A \setminus \{a\}, & [C1] \\ U(c) - U(d) \geq \varepsilon \ \text{ for all } \ c \succ_p d, & [C2] \\ \text{monotonicity and normalization constraints,} & [C3] \end{array}\right\} E_a\end{array}$$

where constraints [C1] are used to impose that a is the most preferred among the considered solutions; constraints [C2] translate the preferences of the user, while constraints [C3] are monotonicity and normalization constraints depending on the type of the adopted value function that shall be described in the next section.

If E_a is feasible and $\varepsilon_a > 0$, then there exists at least one value function for which the solution a is the most preferred solution and, therefore, it is included in the first front. After removing all solutions going in the first front, the same procedure is applied to build the other fronts until each solution is assigned to a front. Within the same front, the solutions are ordered by using the crowding distance on the objective space[2].

So, although NEMO-I and NEMO-II both take into account all value functions compatible with the user's

---

[2] The crowding distance of a solution a is the sum of distances between $a's$ left and right neighbor in each dimension, and infinity if a is an extreme solution [19].

preference information, NEMO-I makes pairwise comparisons between solutions, while NEMO-II compares each solution to all other solutions in the current population. The NEMO-II method has two advantages. First it substantially reduces the computational effort required. While in NEMO-I the construction of a front in a population of s individuals requires the solution of up to $s(s-1)$ LPs, in NEMO-II, the same can be done by solving only s LP problems, one for each solution in the population. Second, NEMO-II is slightly more precise in the sense that it only puts solutions in the best rank for which there exists a value function that makes them most preferred compared to all other solutions in the population, whereas NEMO-I may include some solutions for which no compatible value function exists that would prefer them over all other solutions, as long as no other solution is necessarily preferred in pairwise comparisons. Each front in NEMO-II is a subset of the front in NEMO-I that, in turn, is a subset of the non-dominated front.

A small example may illustrate the difference. Consider the case of three solutions a, b, c, evaluated with respect to the value functions U^1 and U^2 as follows: $U^1(a) = 1$, $U^1(b) = 0.5$ and $U^1(c) = 0$; $U^2(a) = 0$, $U^2(b) = 0.5$ and $U^2(c) = 1$. If U_1 and U_2 were the only value functions compatible with the user's preferences, NEMO-II would not put solution b into the first rank, as it would not be the most preferred under either value function. According to NEMO-I however, neither a nor c are necessarily preferred over b (because $U^2(b) > U^2(a)$ and $U^1(b) > U^1(c)$) and thus NEMO-I would put solution b into the first rank.

For these reasons, we base our new algorithm on the NEMO-II paradigm. The procedure is repeated until the assumed number of iterations has been reached.

Algorithm 1 Basic NEMO-II method

```
Generate initial population of solutions.
Elicit user's preferences by asking DM to compare two randomly selected non-dominated solutions.
Rank individuals into fronts by iteratively identifying all solutions that are most preferred for at least one
compatible value function. Rank within each front using crowding distance.
repeat
  Select individuals for mating.
  Generate offspring using crossover and mutation and add them to the population.
  if Time to ask the DM then
    Elicit user's preferences by asking DM to compare two randomly selected non-dominated solutions.
    if There is no value function remaining compatible with the user's preferences then
      Remove information on pairwise comparisons, starting from the oldest one, until feasibility is
      restored and reintroduce them in the reverse order as long as feasibility is maintained.
    end if
  end if
  Rank individuals into fronts by iteratively identifying all solutions that are most preferred for at least
  one compatible value function. Rank within each front using crowding distance.
  Reduce population size back to initial size by removing worst individuals.
until Stopping criterion met.
```

1.3. Preference Modeling Using Value Functions

1.3.1. Additive Preference Models

Among many preference models considered in the literature, the most popular is an *additive value function* defined on A, such that

$$U(a) = \sum_{j=1}^{n} u_j(f_j(a)) = \sum_{j=1}^{n} u_j(a_j), \tag{3}$$

where u_j are non-decreasing marginal value functions, $u_j : G_j \rightarrow \mathbb{R}$, $j \in G$, G_j is a value set of objective $f_j, j = 1, \ldots, n$, and G is the set of all indices of the objectives.
This model assumes one of the two forms of the marginal value functions $u_j(a_j)$:

(i) piecewise-linear,

(ii) general, non-decreasing.

In case (i), the ranges $[\alpha_j, \beta_j]$ are divided into $\gamma_j \geq 1$ sub-intervals $[a_j^0, a_j^1]$, $[a_j^1, a_j^2]$, ..., $[a_j^{\gamma_j - 1}, a_j^{\gamma_j}]$, where $a_j^k = \alpha_j + \frac{k}{\gamma_j}(\beta_j - \alpha_j)$, $k = 0, \ldots, \gamma_j$, and $j \in G$, while α_j and β_j are the worst and the best performances on objective f_j, respectively. The marginal value of solution $a \in A$ with respect to objective f_j is obtained by linear interpolation,

$$u_j(a_j) = u_j(a_j^k) + \frac{a_j - a_j^k}{a_j^{k+1} - a_j^k}(u_j(a_j^{k+1}) - u_j(a_j^k)), \tag{4}$$

if $a_j \in [a_j^k, a_j^{k+1}]$, where $k \in \{0, ..., \gamma_j - 1\}$.

The piecewise-linear additive model is completely defined by the marginal values at the breakpoints, i.e., $u_j(a_j^0) = u_j(\alpha_j)$, $u_j(a_j^1)$, $u_j(a_j^2)$, $\cdots$, $u_j(a_j^{\gamma_j}) = u_j(\beta_j)$. Considering this type of preference function, one has to consider the following monotonicity and normalization constraints:

- for all $j \in G$ and for all $k = 0, \ldots, \gamma_j - 1$, $u_j(a_j^k) \leq u_j(a_j^{k+1})$;
- for all $j \in G$, $u_j(\alpha_j) = 0$, and $\sum_{j \in G} u_j(\beta_j) = 1$.

In case (ii), the characteristic points of marginal value functions $u_j, j \in G$, are fixed in evaluation points of considered solutions. Let τ_j be the permutation on the set of indices of solutions from A that reorders them according to non-decreasing evaluation on objective j, i.e.,

$$a_{\tau_j(1)} \leq a_{\tau_j(2)} \leq \ldots \leq a_{\tau_j(m-1)} \leq a_{\tau_j(m)}, \; j \in G, m = |A|.$$

The general non-decreasing additive model is completely defined by the marginal values at the characteristic points, i.e., $u_j(\alpha_j) = u_j(a_{\tau_j(1)})$, $u_j(a_{\tau_j(2)})$, $\ldots$, $u_j(a_{\tau_j(m)}) = u_j(\beta_j)$. Note that in this case, no linear interpolation is required to express the marginal value of any reference solution.

Considering this type of value function, the monotonicity constraints have the form $u_j(a_{\tau_j(k)}) \leq u_j(a_{\tau_j(k+1)})$, for all $k = 1, \ldots, m-1$, while normalization constraints are the same as in case (i).

1.3.2. The Choquet Integral Preference Model

The simplest additive value function model is the weighted sum, obtained by assigning a non-negative weight w_j to each objective $f_j, j \in G$, and giving to each $a \in A$ the value

$$U(a) = \sum_{j=1}^{n} w_j f_j(a) = w_1 f_1(a) + \ldots + w_n f_n(a). \tag{5}$$

The weighted sum has some limitations in representing user's preferences, which are shown in the following example. Let us underline that in this example we shall speak of evaluation criteria, being in Multiple Criteria Decision Aiding (MCDA) the equivalent of the objective functions in Multiobjective Optimization.

Example. The manager of an international company wants to rank three candidates (*Smith*, *Johnson* and *Brown*), taking into account their performances on criteria experience (Ex) and age (Ag), given on a $[0, 10]$ scale (see Table 1).

Table 1: Experience (Ex) and age (Ag) of three candidates

	Ex	Ag
Smith(S)	6	10
Johnson(J)	8	8
Brown(B)	10	6

Since candidates having good experience are not necessarily young, and vice versa, if there is a good performance on one of the two criteria, one does not expect a good performance also on the other criterion. Consequently, a candidate being good both on experience and age is well appreciated. Therefore, in the manager's mind there is a positive interaction (synergy) between the performance on experience and the performance on age. In other words, the two criteria are not preferentially independent [32]. For this reason, the manager prefers *Johnson* to *Smith* and *Brown*.

If one would like to represent the preferences expressed by the manager using the weighted sum model, the following inequalities should be satisfied:

$$\begin{aligned} w_{Ex} \cdot 6 + w_{Ag} \cdot 10 &< w_{Ex} \cdot 8 + w_{Ag} \cdot 8, \\ w_{Ex} \cdot 10 + w_{Ag} \cdot 6 &< w_{Ex} \cdot 8 + w_{Ag} \cdot 8, \end{aligned}$$

where w_{Ex} and w_{Ag} are the weights of experience and age, respectively. It can be easily verified that the above inequalities are contradictory since:

$$w_{Ex} \cdot (-2) + w_{Ag} \cdot 2 < 0 < w_{Ex} \cdot (-2) + w_{Ag} \cdot 2.$$

Thus we have to conclude that, due to the positive interaction between the performances on experience and age, the weighted sum is not able to represent the manager's preferences. $\square$

In order to represent preferences in case of interaction between criteria, non-additive integrals are often used [25]. The best-known non-additive integral in the literature is the Choquet integral [14]. The Choquet integral is based on the concept of capacity (fuzzy measure) that assigns a weight to each subset of criteria rather than to each single criterion. More precisely, denoting by 2^G the power set of G (i.e., the set of all subsets of G), the function $\mu : 2^G \rightarrow [0,1]$ is called capacity on 2^G if the following properties are satisfied:

1a) $\mu(\emptyset) = 0$ and $\mu(G) = 1$ (boundary conditions),

2a) $\forall\, T \subseteq S \subseteq G,\ \ \mu(T) \leq \mu(S)$ (monotonicity condition).

Intuitively, for all $T \subseteq G$, $\mu(T)$ can be interpreted as a comprehensive importance of the criteria from T considered as a whole.

Example (continuation). To represent the importance and the interaction of the performances on experience and age, one can set $\mu(\{Ex\}) = 0.4$, $\mu(\{Ag\}) = 0.3$ and $\mu(\{Ex, Ag\}) = 1$. The difference $\mu(\{Ex, Ag\}) - \mu(\{Ex\}) - \mu(\{Ag\}) = 0.3$ represents the positive interaction between experience and age because it measures how much greater is the importance of experience and age considered as a whole ($\mu(\{Ex, Ag\})$) comparing to the sum of their importances when they are considered separately ($\mu(\{Ex\}){+}\mu(\{Ag\})$). $\square$

The Choquet integral involving capacity μ assigns to each alternative $a \in A$ the following value:

$$C_\mu(a) = \sum_{j=1}^{n} \left[f_{(j)}(a) - f_{(j-1)}\left(a\right) \right] \mu\left(N_j\right), \tag{6}$$

where $_{(\cdot)}$ stands for a permutation of the indices of criteria, such that

$$f_{(0)}(a) \leq f_{(1)}\left(a\right) \leq f_{(2)}\left(a\right) \leq \ldots \leq f_{(n)}\left(a\right), \tag{7}$$

$N_j = \{(j), \ldots, (n)\}$ and $f_{(0)}(a) = 0$.

Observe that (7) requires that the values taken by objective functions $f_j, j = 1, \ldots, n$, have to be non negative. If this is not the case, one can recode the values $f_j(a), a \in A$, with a translation $f_j^*(a) = f_j(a) + c$ with $c \geq -\min\limits_{j \in G,\, b \in A} f_j(b)$, so that we get $f_j^*(a) \geq 0$ for all $j = 1, \ldots, n$ and all $a \in A$.

A meaningful reformulation of the capacity μ and of the Choquet integral can be obtained by means of the Möbius representation of the capacity μ (see [40]) which is a function $m : 2^G \rightarrow \mathbb{R}$ [42] defined as follows:

$$\mu(S) = \sum_{T \subseteq S} m(T).$$

Note that if S is a singleton, i.e., $S = \{j\}$ with $j = 1, \ldots, n$, then $\mu(\{j\}) = m(\{j\})$. Moreover, if S is a pair of criteria, i.e., $S = \{i, j\}$, then $\mu(\{i, j\}) = m(\{i\}) + m(\{j\}) + m(\{i, j\})$.
The Möbius representation $m(S)$ of capacity $\mu(S)$ can be obtained as follows:

$$m(S) = \sum_{T \subseteq S} (-1)^{|S-T|} \mu(T).$$

In terms of the Möbius representation, properties **1a)** and **2a)** are, respectively, restated as (see [13]):

1b) $m(\emptyset) = 0, \ \sum_{T \subseteq G} m(T) = 1,$

2b) $\forall \, j \in G$ and $\forall S \subseteq G \setminus \{j\}, \ \sum_{T \subseteq S} m(T \cup \{j\}) \geq 0,$

while the Choquet integral may be reformulated as follows:

$$C_\mu(a) = \sum_{T \subseteq G} m(T) \min_{j \in T} f_j(a). \tag{8}$$

Geometrically, in the case of two objectives ($G = \{f_1, f_2\}$), the iso-value curve of the Choquet integral can be decomposed into two linear functions, one above the line $f_1 = f_2$ and one below the line $f_1 = f_2$. Using (8) we can write

$$C_\mu(a) = \begin{cases} (m(\{1\}) + m(\{1,2\}))f_1(a) + m(\{2\})f_2(a), & \text{if } f_1(a) \leq f_2(a) \\ m(\{1\})f_1(a) + (m(\{2\}) + m(\{1,2\}))f_2(a), & \text{if } f_1(a) \geq f_2(a) \end{cases}$$

Example (continuation). The value assigned to *Smith* (S) by the Choquet integral with capacity μ is the following:

$$C_\mu(S) = f_{Ex}(S) \cdot \mu(\{Ex, Ag\}) + (f_{Ag}(S) - f_{Ex}(S)) \cdot \mu(\{Ag\}) = 7.2. \tag{9}$$

This value can be explained as follows. The performance $f_{Ex}(S) = 6$ is attained by the two criteria and thus it is multiplied by $\mu(\{Ex, Ag\})$ which is the weight assigned to experience and age considered as a whole. The performance $f_{Ag}(S) = 10$ is attained only on criterion age and therefore the difference $f_{Ag}(S) - f_{Ex}(S)$ is multiplied by $\mu(\{Ag\})$ which is the weight assigned to age considered alone. Analogously, we get $C_\mu(Johnson) = 8$ and $C_\mu(Brown) = 7.6$, so that we have $C_\mu(Johnson) > C_\mu(Smith)$ and $C_\mu(Johnson) > C_\mu(Brown)$, and thus we can conclude that the Choquet integral is able to represent the manager's preferences.

Observe, moreover, that the Möbius representation m of the capacity μ gives $m(\{Ex\}) = 0.4$, $m(\{Ag\}) = 0.3$ and $m(\{Ex, Ag\}) = 0.3$. Therefore, the Choquet integral referring to *Smith* can be reformulated as follows in terms of the Möbius representation m:

$$C_\mu(S) = f_{Ex}(S) \cdot m(\{Ex\}) + f_{Ag}(S) \cdot m(\{Ag\}) + \min(f_{Ex}(S), f_{Ag}(S)) \cdot m(\{Ex, Ag\}) = 7.2.$$

This value can be explained as follows. The performances on experience and on age are multiplied by $m(\{Ex\})$ and $m(\{Ag\})$, respectively, representing the relative weights of the two criteria. However, the value obtained by summation of the two weighted components has to be corrected by adding $\min(f_{Ex}(S), f_{Ag}(S)) \cdot m(\{Ex, Ag\})$ representing the positive interaction between the performance on experience and the performance on age. The Choquet integral of *Johnson* and *Brown* can be analogously reformulated in terms of the Möbius representation. $\square$

In order to reduce the number of parameters to be elicited and to avoid an overprecise description of the interactions among criteria, Grabisch [24] introduced the concept of fuzzy k-additive capacity. A capacity is called k-additive if $m(T) = 0$ for $T \subseteq G$, such that $|T| > k$. In particular, in case of a 1-additive capacity, the Choquet integral is the standard weighted sum model.

In MCDA, it is easier and more straightforward to consider 2-additive capacities, since then the users have to express preference information on positive and negative interactions between two criteria only, neglecting possible interactions among three, four and, generally, r criteria, $r = 2, \ldots, n$. Moreover, by considering 2-additive capacities, the computational effort needed to determine the parameters is reduced since only $n + \binom{n}{2}$ coefficients have to be assessed; specifically, in terms of the Möbius representation, a value $m(\{i\})$ for every criterion i, and a value $m(\{i, j\})$ for every pair of criteria $\{i, j\}$. The value that a 2-additive capacity μ assigns to a set $S \subseteq G$ can be expressed in terms of the Möbius representation as follows:

$$\mu(S) = \sum_{i \in S} m\left(\{i\}\right) + \sum_{\{i,j\} \subseteq S} m\left(\{i, j\}\right), \ \forall S \subseteq G. \tag{10}$$

With regard to 2-additive capacities, properties **1b)** and **2b)** have, respectively, the following forms:

1c) $m\left(\emptyset\right) = 0, \ \sum_{i \in G} m\left(\{i\}\right) + \sum_{\{i,j\} \subseteq G} m\left(\{i, j\}\right) = 1,$

2c) $\begin{cases} m\left(\{i\}\right) \geq 0, \ \forall i \in G, \\ m\left(\{i\}\right) + \sum_{j \in T} m\left(\{i, j\}\right) \geq 0, \text{ for all } i \in G, \text{ and for all } T \subseteq G \setminus \{i\}, \ T \neq \emptyset. \end{cases}$

In this case, the Choquet integral of $a \in A$ is calculated as:

$$C_\mu(a) = \sum_{i \in G} m\left(\{i\}\right) f_i\left(a\right) + \sum_{\{i,j\} \subseteq G} m\left(\{i, j\}\right) \min(f_i\left(a\right), f_j\left(a\right)). \tag{11}$$

As one can observe, the use of the Choquet integral is based on several parameters (capacity $\mu(T)$ for each subset $T \subseteq G$ or a value $m(T)$ for each subset $T \subseteq G$ in case of the Möbius representation of capacity μ). To determine these parameters, a direct and an indirect technique known from the literature can be applied. In the direct technique, the user has to provide the parameters directly, while in the indirect technique the user has to provide some preference information from which parameters compatible with this information are retrieved by ordinal regression. The latter technique is much more realistic than the former because it requires less cognitive effort from the user. The indirect technique for the Choquet integral has been firstly proposed in [35]. When using the indirect technique, it is possible that more than one set of parameters is compatible with the preference information given by the user. For this reason, selection of only one of these compatible sets of parameters is somewhat arbitrary. To take into account all sets of parameters compatible with the user's preferences, Robust Ordinal Regression (ROR) [27] has been recently proposed. Taking into account all the sets of parameters compatible with the preferences of the user, ROR presents a recommendation in terms of *necessary* or *possible* preference relations which, for a pair of alternatives a and b, hold if a is at least as good as b for all or for at least one set of compatible parameters, respectively. ROR has been applied to the Choquet integral in [2] under the name of Non Additive Robust Ordinal Regression (NAROR).

Observe also that, besides determination of the capacity, the use of the Choquet integral involves another specific problem that is the construction of a common scale for the considered criteria permitting to compare the performances on different criteria and to compute in a meaningful way their difference. Indeed, looking at the definition of the Choquet integral in (6), we can observe that on one hand, permutation $_{(\cdot)}$ of the considered performances on different criteria is required, while on the other hand, the computation of the Choquet integral requires also that the differences between the performances on criteria $g_{(i)}$ and $g_{(i-1)}$, $i = 1, \ldots, n$, are meaningful. In the provided example this is obvious because, for instance, considering candidate *Smith*, the performace of 10 on age is clearly more valuable than the performance of 6 on experience, and their difference is 4. But, considering the case of a decision about cars, which is more valuable between a maximum speed of 200 km/h and a price of 35,000 euros? This means that in case of criteria with heterogeneous scales, the performances on all criteria have to be mapped to a common scale which permits to compare them and also to compute their difference. Very often a normalization of performances on each criterion is done considering an "unacceptable" and an "optimal value" for each criterion and considering a linear interpolation between these two extremes (see, e.g., [26]). A more sophisticated methodology permitting to construct a common scale and a capacity for the Choquet integral on the basis of preference information supplied by the DM has been proposed in [1] and further developed in [3]. In this paper, as we shall explain in detail in Section 1.4, we consider an intermediate approach consisting in first normalizing the performances on each criterion, and then rescaling them through multiplication by a set of weights that ensure comparability between performances on different criteria. We explain this procedure in the continuation of the previous example, in which the performances on considered criteria are already normalized but they need to be rescaled so that the preferences of the DM can be represented by the Choquet integral.

Example (continuation). Suppose that two new candidates have to be added to the three previously considered. Their performances with respect to experience and age are presented in table 2.

Table 2: Two new candidates evaluated on experience and age

	Ex	Ag
Baker	7	9
Miller	9	7

After reflecting a little, the manager arrived at the conclusion that he has the following preferences with respect to the five candidates:

$$Baker \succ Johnson \succ Miller \succ Brown \succ Smith.$$

When trying to apply the Choquet integral to represent the current manager's preferences, we realize that it is not possible. Indeed, by computing the Choquet integral of the performances of the candidates *Brown* and *Smith* we get

$$6 \cdot \mu(\{Ex, Ag\}) + (10 - 6) \cdot \mu(\{Ex\}) > 6 \cdot \mu(\{Ex, Ag\}) + (10 - 6) \cdot \mu(\{Ag\}) \tag{12}$$

while comparing *Baker* and *Miller* we get

$$7 \cdot \mu(\{Ex, Ag\}) + (9 - 7) \cdot \mu(\{Ag\}) > 7 \cdot \mu(\{Ex, Ag\}) + (9 - 7) \cdot \mu(\{Ex\}). \tag{13}$$

From Eq. (12) we get $\mu(\{Ex\}) > \mu(\{Ag\})$ while from Eq. (13) we get $\mu(\{Ag\}) > \mu(\{Ex\})$, which are of course incompatible.

Observe, however, that if you rescale the criteria experience and age multiplying the relative performances of the candidates by 0.56 and 0.44, respectively, we get the performances shown in Table 3. Computing the Choquet integral of the five candidates considering the capacity previously defined ($\mu(\{Ex\}) = 0.4$, $\mu(\{Ag\}) = 0.3$ and $\mu(\{Ex, Ag\}) = 1$), we get the values in the last column of Table 3 which represent the preferences of the manager. □

Table 3: The five candidates evaluated on experience and age after rescaling

	Ex	Ag	CI
Smith	3.36	4.4	3.67
Johnson	4.48	3.52	3.90
Brown	5.6	2.64	3.82
Baker	3.92	3.96	3.93
Miller	5.04	3.08	3.86

Let us conclude this section discussing the use of Choquet integral in case the DM prefers smaller objectives function values, i.e. for all $a, b \in A$, if $f_j(a) \leq f_j(b)$, then a is at least as good as b with respect

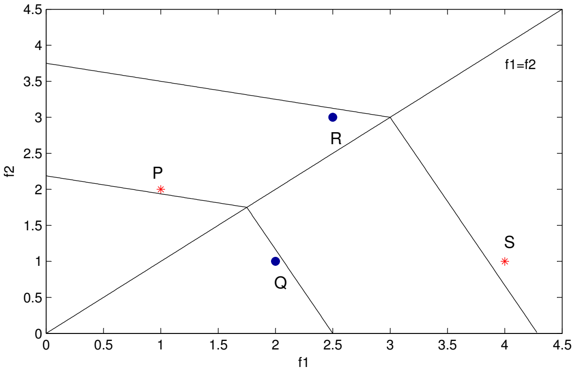

Figure 1: In this case, $Q \equiv (2,1)$ is preferred to $P \equiv (1,2)$ and $R \equiv (2.5,3)$ is preferred to $S \equiv (4,1)$. Trying to translate these preferences by using a linear model (5), we get the contradictory inequalities $w_1 < w_2$ and $w_1 > \frac{4}{3}w_2$. By using the Choquet integral preference model (8) to translate these preferences, we get the inequalities $m(\{1\}) < m(\{2\})$ and $m(\{1\}) > \frac{4}{3}m(\{2\}) + m(\{1,2\})$ being not in contradiction. For example, by considering $m(\{1\}) = 0.7$, $m(\{2\}) = 0.8$ and $m(\{1,2\}) = -0.5$ these inequalities are satisfied.

to objective f_j. In this case, the objective functions f_j can still be aggregated using the Choquet integral, but then solutions with a smaller Choquet integral would be preferred, i.e., for all $a, b \in A$ if $C_\mu(a) \leq C_\mu(b)$, then a is comprehensively at least as good as b. In this case, the aim is to minimize rather than to maximize the value of the Choquet integral.

1.4. The NEMO-II-Ch method

Because most benchmark problems in evolutionary multiobjective optimization are *minimization* problems, in the following description of NEMO-II and of the empirical analysis, we assume that objective functions as well as a supposed users' utility are to be minimized. Compared to the discussion above, the only difference is that while with maximization, the preference of an alternative a over an alternative b was translated into the constraint $U(a) \geq U(b) + \varepsilon$, now the preference is translated into the constraint $U(a) + \varepsilon \leq U(b)$. In case of indifference between a and b, the corresponding constraint is $U(a) = U(b)$.

Let us first consider a problem with two objectives f_1 and f_2 to be minimized. As observed in Section 1.3.2, due to some interactions between the considered criteria it could happen (as in the case shown in Figure 1), that the linear model is not able to represent the preferences of the DM. For this reason, we suggest using the Choquet integral preference model that is able to take into account interactions between criteria. In [38], it has been shown experimentally that the Choquet integral has a greater capacity of representing the preferences of a DM than the weighted sum model.

While a greater flexibility of the preference model allows to capture more complicated user preference information and is thus desirable, usually, it also has more parameters, and more preference information is required before the set of compatible value functions is curbed sufficiently to be useful in narrowing down the search. For this reason, we propose to keep the complexity of the preference model low as long as it is

Algorithm 2 Basic NEMO-II-Ch method

```
Current preference model = LINEAR.
Generate initial population of solutions.
Elicit user's preferences by asking DM to compare two randomly selected non-dominated solutions.
Rank individuals into fronts by iteratively identifying all solutions that are most preferred for at least one compatible value function. Rank within each front using crowding distance.
repeat
  Select individuals for mating.
  Generate offspring using crossover and mutation and add them to the population.
  if Time to ask the DM then
    Elicit user's preferences by asking DM to compare two solutions that are the most preferred for at least one compatible value function.
    if There is no value function remaining compatible with the user's preferences then
      if Current preference model = LINEAR then
        Preference model = CHOQUET.
      else
        Remove information on pairwise comparisons, starting from the oldest one, until feasibility is restored and reintroduce them in the reverse order as long as feasibility is maintained.
      end if
    end if
  end if
  Rank individuals into fronts by iteratively identifying all solutions that are most preferred for at least one compatible value function. Rank within each front using crowding distance.
  Reduce population size back to initial size by removing worst individuals.
until Stopping criterion met.
```

sufficient to capture all preference information, but switch to a more complex preference model when this is no longer the case, following the procedure described in Algorithm 2. In particular, we start with assuming a linear preference model. Once we can no longer find a linear value function compatible with all elicited preference relations, we switch to a 2-additive Choquet integral.

To check whether there exists a set of weights $\mathbf{w} = (w_1, \ldots, w_n)$ such that the linear model is able to restore the preferences of the DM, one has to solve the following LP,

$$
\begin{array}{l}
\max \varepsilon, \ \text{ subject to} \\
\left.\begin{array}{l}
U(a) - U(b) + \varepsilon \leq 0 \ \text{ for all } \ a \succ_p b, \\
\displaystyle\sum_{j=1}^{n} w_j = 1 \\
w_j \geq 0, \ \text{for all } j = 1, \ldots, n
\end{array}\right\} E_{DM}^{linear}
\end{array}
$$

where $U(a) - U(b) \leq \varepsilon$ are the constraints translating the preferences of the DM while the other two constraints are used to ensure that weights are non-negative and normalized.

There is a $\mathbf{w}$ compatible with the preferences of the DM if and only if E_{DM}^{linear} is feasible and $\varepsilon^{linear} > 0$ where $\varepsilon^{linear} = \max \varepsilon$ subject to E_{DM}^{linear}. In this case, one can proceed to order the population by using the same procedure described in the previous section, checking, for each solution x, whether there exists a set of weights $\mathbf{w} = (w_1, \ldots, w_n)$ such that x is the best among the considered solutions. To this end, one has to solve the following LP.

$$
\begin{array}{l}
\max \varepsilon, \ \text{ subject to} \\
\left.\begin{array}{l}
U(a) - U(b) + \varepsilon \leq 0 \ \text{ for all } \ a \succ_p b, \\
U(x) - U(y) + \varepsilon \leq 0 \ \text{ for all } y \in A \setminus \{x\}, \\
\displaystyle\sum_{j=1}^{n} w_j = 1 \\
w_j \geq 0, \ \text{for all } j = 1, \ldots, n
\end{array}\right\} E_{x}^{linear}
\end{array}
$$

where constraints $U(x) - U(y) \leq \varepsilon$ ensure that x is preferred to all other solutions in A.

If E_x^{linear} is feasible and $\varepsilon_x^{linear} > 0$ where $\varepsilon_x^{linear} = \max \varepsilon$ subject to E_x^{linear}, then there exists a set of weights $\mathbf{w}$ such that x is the preferred solution and therefore it is included in the first front. After ordering all the solutions into different fronts, the solutions in the same front are ordered by computing the classical crowding distance of NSGA-II.

If there exists no vector $\mathbf{w}$ such that the linear model would be able to restore the preference information provided by the DM, we need to use a more complex model such as the Choquet integral in order to represent the preferences expressed by the DM. As we shall justify later, we use the 2-additive Choquet integral here that has a parameter $m(\{j\})$ for each objective j and a parameter $m(\{i, j\})$ for each pair of objectives $\{i, j\}$.

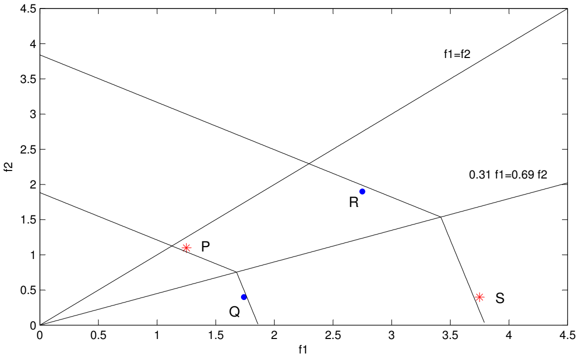

Figure 2: Example for a case where for capturing a user's preferences with the Choquet integral it is necessary to move the line $f_2 = f_1$ by multiplying both objectives by weights w_1 and w_2 such that $w_1 + w_2 = 1$.

As already discussed in Section 1.3.2, the use of the Choquet integral assumes that all objectives are expressed on the same scale. Indeed, when using the original formulation of the Choquet integral involving the capacity μ (see Eq. (6)), for each solution x we need to order all values of each individual's objective from the worst to the best. Analogously, using the Choquet integral expressed by means of the Möbius decomposition m (see Eq. (8)), for each solution x and for each subset of objective functions, we need to know the minimum value. Since we cannot assume that the scales of the different objectives are comparable, the scaling becomes part of the model. In other words, in addition to the usual parameters of the Choquet integral, we also need to consider a scaling weight for each objective.

Let us consider the example shown in Figure 2. If the user states that $Q \equiv (1.75, 0.4)$ is preferred to $P \equiv (1.25, 1.05)$ and $R \equiv (2.75, 1.9)$ is preferred to $S \equiv (3.75, 0.4)$, there is no Choquet integral compatible with these preferences. The preferences correspond to the inequalities $0.5m(\{1\}) - 0.65m(\{2\}) - 0.65m(\{1,2\}) < 0$ and $m(\{1\}) - 1.5m(\{2\}) - 1.5m(\{1,2\}) > 0$ which contradict the monotonicity constraints 2c). This is also apparent from the fact that all four alternatives are located under the line $f_1 = f_2$, where the iso-utility function of the Choquet integral is linear, and no linear model is able to reflect the preferences.

However, if we scale the objectives appropriately (which "moves" the $f_1 = f_2$ line), then it is possible to represent the preference information using a Choquet integral. For example, let us multiply the two objectives by the weights $w_1 = 0.31$ and $w_2 = 0.69$. The four points become $Q \equiv (0.5425, 0.276)$, $P \equiv (0.3875, 0.7245)$, $R \equiv (0.8525, 1.311)$ and $S \equiv (1.1625, 0.276)$ while the constraints translating the preferences of the user become $0.155m(\{1\}) - 0.4485m(\{2\}) - 0.1115m(\{1,2\}) < 0$ and $0.31m(\{1\}) - 1.035m(\{2\}) - 0.5765m(\{1,2\}) > 0$, being compatible with the Möbius decomposition $m(\{1\}) = 0.9$, $m(\{2\}) = 0.4$ and $m(\{1,2\}) = -0.3$.

Mathematically, determining whether the preferences can be represented by an appropriate scaling and Choquet integral translates into the following non-linear program:

$$\max \varepsilon, \text{ subject to}$$

$$\left.\begin{array}{l} C_\mu(w_1 f_1(a),\ldots,w_n f_n(a)) \\ \quad -C_\mu(w_1 f_1(b),\ldots,w_n f_n(b)) \leq \varepsilon \text{ for all } a \succ_p b, \\ \displaystyle\sum_{j=1}^{n} w_j = 1 \\ m(\emptyset) = 0, \\ \sum_{j=1}^{n} m(\{j\}) + \sum_{\{i,j\}\subseteq\{1,\ldots,n\}} m(\{i,j\}) = 1, \\ m(\{j\}) \geq 0, \text{ for all } j = 1,\ldots,n, \\ m(\{j\}) + \displaystyle\sum_{i \in T} m(\{i,j\}) \geq 0, \text{ for all } j = 1,\ldots,n, \text{ and} \\ \quad \text{for all } T \subseteq \{1,\ldots,n\} \setminus \{j\}, T \neq \emptyset. \end{array}\right\} E_{DM}^{Ch} \tag{14}$$

The user's preferences can be represented if an only if the solution to this optimization problem results in $\varepsilon_{DM}^{Ch} > 0$, where $\varepsilon_{DM}^{Ch} = \max \varepsilon$ subject to E_{DM}^{Ch}.

Since the optimization problem is non-linear, to solve it, we use the Nelder-Mead algorithm [36] to search the space of weights while maximizing ε, with an LP being solved in every iteration to determine the best Möbius parameters for the weights in the current iteration. The algorithm is aborted as soon as an $\varepsilon > 0$ has been found. If after a few iterations (we chose 40 in the experiments below), no such weight/Choquet coefficient combination has been found, we stop the search and remove some of the DM's preference information, starting from the oldest, until the feasibility is regained.

Once it has been found that the preference information provided by the DM can be represented by using the Choquet integral preference model (possibly after removing some pieces of information), the solutions are put into different fronts by using the same procedure as described previously. For each solution $x \in A$, one has to check whether this solution might be the most preferred one by solving the following optimization problem.

$$\max \varepsilon, \text{ subject to}$$

$$\left.\begin{array}{l} C_\mu(w_1 f_1(x),\ldots,w_n f_n(x)) \\ -C_\mu(w_1 f_1(y),\ldots,w_n f_n(y)) \leq \varepsilon, \text{ for all } y \in A \setminus \{x\}, \\ E_{DM}^{Ch} \end{array}\right\} E_x^{Ch} \tag{15}$$

Because, again, the optimization problem (15) is non-linear, in a first step we try to simplify the solution by fixing the vector of weights $\mathbf{w}' = (w_1',\ldots,w_n')$ such that $w_1' f_1(x) = \ldots = w_n' f_n(x)$ and then checking, by using linear programming optimization, if there exists a set of Choquet coefficients such that $\varepsilon_x^{Ch} > 0$, where $\varepsilon_x^{Ch} = \max \varepsilon$ s.t. E_x^{Ch} (indeed, if we fix the weights $(w_1,\ldots,w_n)$, then the optimization problem (15) becomes an LP problem). We found that in many cases, this will find a feasible solution if there exists one. If not, then we use the Nelder-Mead method explained above to check whether the optimization problem allows for $\varepsilon_x^{Ch} > 0$.

Note that for more than 2 dimensions, we still restrict our model to 2-additive Choquet.

1.5. Experimental Results

In this section, the algorithms introduced before are compared empirically. We start with a comparison of NEMO-I and NEMO-II to justify our use of NEMO-II for the remainder of the paper. Next, we look at the effect of model complexity, and demonstrate the benefit of starting with a simple model but then switching to a more complex model if it is necessary to represent the user's preferences. Finally, we compare a number of algorithms on various 2 to 5-dimensional benchmark functions.

For the algorithms tested, we use the following notation:

- **NEMO-I-L:** The NEMO-I algorithm with a *linear* additive preference model.
- **NEMO-II-L:** The NEMO-II algorithm with a *linear* additive preference model.
- **NEMO-II-PL2:** The NEMO-II algorithm with a piecewise linear additive preference model, consisting of two linear pieces. The breakpoint was chosen to be the median of the values in the population for each objective.
- **NEMO-II-G:** The NEMO-II algorithm with a general monotonic additive preference model as often used in ROR [27].
- **NEMO-II-Ch:** The proposed NEMO-II algorithm that starts with a simple linear preference model and switches to a 2-additive Choquet preference model when the linear model is no longer able to account for the user's preference information.

Furthermore, the following are provided as benchmarks:

- **NSGA-II:** The standard NSGA-II algorithm not using preference information.
- **EA-UVF:** A single-objective EA that uses the true user's value function for ranking individuals (information that is not available to the other algorithms). EA-UVF shows the performance that could be expected if the user's value function was fully known to the evolutionary algorithm from the beginning. Clearly, this is an idealized setting and only serves as a reference.
- **Optimum:** This is the best value of feasible solutions according to the true user's value function.

Let us point out that comparison between different interactive methods is difficult since they use different preference information. Consequently, we decided to compare our method with NSGA-II only because NEMO-II-Ch is based on NSGA-II.

All algorithms use a real valued representation, generate offspring by simulated binary crossover with crossover probability of 0.9 and $\eta_c = 15$, and Gaussian mutation with mutation probability $\frac{1}{v}$ (where v is the number of variables and depends on the considered problem) and step size $\sigma = 0.1$. Mating selection is done by tournament selection. We run the algorithm for a pre-specified number of 400 (in case of 2 objectives) or 600 (in case of 3 or 5 objectives) generations. The population size has been set to 30, a value smaller than usual in MOEAs, but we do not aim to find the whole Pareto frontier but only the most preferred solutions. The DM is asked to provide some preference information about one pair of randomly picked non-dominated solutions every 10 generations.

The "true" user's value function assumed in this study is the Chebyshev function, i.e., the user's goal is to maximize $U_{DM}(a) = -\max\{w_1 f_1(a), \ldots, w_n f_n(a)\}$, which is equivalent to minimizing $U^-_{DM}(a) = \max\{w_1 f_1(a), \ldots, w_n f_n(a)\}$. The parameters $w_1, \ldots, w_n$ depend on the problem and are defined below. For the sake of simplicity, with a slight abuse of the terminology, when we speak of the user's value functions we refer to their opposite forms, and thus we aim at minimizing the Chebyshev-like value function $U^-_{DM}(a) = \max\{w_1 f_1(a), \ldots, w_n f_n(a)\}$. When showing the plots of convergence and tables, the terms "Convergence indicator" and "Convergence curve" mean the value of $U^-_{DM}(\cdot)$.

We measure performance based on the user's true utility of the best individual in the population, i.e., the minimum of $U(x)$ over all individuals in the population. This assumes that the final population is returned to the DM, and the DM is able and willing to spend the effort to identify the most preferred solution among the set. All results shown in all tables and figures have been averaged over 50 independent runs.

1.5.1. Comparison of NEMO-I and NEMO-II

Since we implement here for the first time an idea that has been proposed in [11] as NEMO-II, we would like to assert that it is not only much more efficient than NEMO-I, but also competitive in terms of solution quality. To do this, we have compared NEMO-I and NEMO-II with a simple linear preference model on the simple convex 2-dimensional test problem ZDT1. The user's true value function was set to $U(x) = f_1(x) + f_2(x)$, i.e., the linear model is sufficient to represent the user's preferences. Knowing the user's true value function, one can calculate the solution with the best true user value for a given problem. This best value will be called "optimum". Figure 3 shows the convergence curve corresponding to this experiment. One can observe that both methods converge to the optimum ($f_1(x) = \frac{1}{4}$ and $f_2(x) = \frac{1}{2}$, with $U(x) = \frac{3}{4}$), and that the convergence of NEMO-II-L is slightly quicker than that of NEMO-I-L. We thus conclude that NEMO-II is not only computationally much more efficient than NEMO-I, but at least as good, and we will focus on NEMO-II in the remainder of this paper.

1.5.2. Model complexity switching

Determining an appropriate model complexity is a challenging issue. Obviously, a higher complexity of the algorithm's preference model allows the algorithm to model more complex user's preferences. On the other hand, higher model complexity usually means more parameters to be set, and thus more preference information is required for the model to become sufficiently restricted to be useful in guiding the evolutionary process. Besides, more parameters also means a higher computational cost for solving each LP in NEMO-II. For these reasons, we have proposed in Section 1.4 NEMO-II-Ch that starts with a simple linear preference

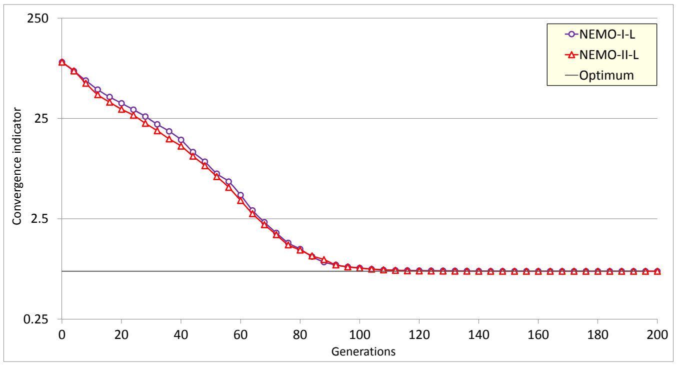

Figure 3: ZDT1-2D Linear $(f_1(x) + f_2(x))$

model, and switches to a 2-additive Choquet integral when the simple linear model is no longer able to account for the user's preference information.

Figure 4 compares the proposed NEMO-II-Ch with NEMO-II-L, with NEMO-II that uses the 2-additive Choquet integral from the beginning, and with NEMO-II that starts using the linear model, but then switches to the full Choquet model rather than the 2-additive one. The test problem is the 5-dimensional DTLZ1, and the user preference function to be minimized is the Chebyshev function:

$$U(x) = \max\left\{0.1f_1(x), 0.15f_2(x), 0.2f_3(x), 0.25f_4(x), 0.3f_5(x)\right\}. \tag{16}$$

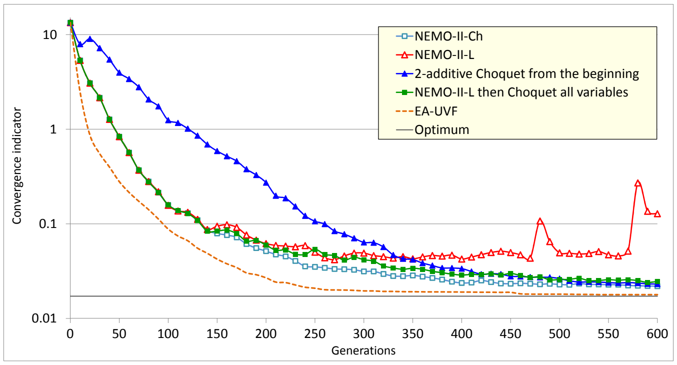

Figure 4: DTLZ1-5D Chebyshev with different types of Choquet integral as preference model.

As can be observed, the algorithm variants starting with a linear preference model converge much more quickly. This makes intuitive sense, as they have fewer parameters to estimate and thus can narrow down the search more quickly. Around iteration 70, the convergence curves for NEMO-II-L, NEMO-II-Ch and "NEMO-II-L then Choquet all variables" diverge, i.e., the linear model is sometimes no longer sufficient to capture the user's preference information, and switching to the Choquet model is required (note that not all runs switch at the same time). The difference between switching to the full Choquet integral or 2-additive Choquet integral is relatively small, but the full Choquet integral is somewhat slower, probably because it increases the number of parameters which slows down convergence. There doesn't seem to be a difference in final solution quality. Sticking to the linear model eventually results in poor and unstable behavior, as the model can no longer capture the user's preferences and old preference information has to be discarded.

In the following, we will thus stick to NEMO-II-Ch as our most promising procedure. To demonstrate the advantage of learning user's preferences, we compare it on various benchmark problems with NSGA-II. To further demonstrate the particular advantages of the Choquet integral over alternative value function models, we also report on results obtained with NEMO-II-L, NEMO-II-PL2, and NEMO-II-G.

Results in 2D

As 2D test problems, we use ZDT1 (with a convex Pareto front) and ZDT2 (with concave Pareto front) and the parameters of the Chebyshev function given in Table 4.

Table 4: Parameters of the user's value function in 2D

	w_1	w_2
ZDT1-middle	0.6	0.4
ZDT1-extreme	0.15	0.85
ZDT2-middle	0.6	0.4
ZDT2-extreme	0.15	0.85

As one can observe in Figures 5 and 6, the differences of final solution quality of the various approaches are very small, and all get very close to the optimal solution. This is not very surprising, since although the linear model can not represent the Chebyshev user preference directly, given that ZDT1 has a convex Pareto front, even the linear model is able to converge to any solution on the frontier. The fact that NEMO-II-Ch and NEMO-II-L lines overlap almost completely indicates that a linear model was almost always able to respect the user's preferences. Where the user prefers an extreme solution, the convergence of NSGA-II and NEMO-II-G is somewhat slower.

Moving to ZDT2, the situation is slightly different. NEMO-II-L is no longer able to converge to the correct point because of the concavity of the Pareto front of the test function as can be noticed from the erratic behavior of the corresponding curve in Figure 7 and especially in Figure 8. NSGA-II and NEMO-II-G again converge more slowly in the case of the user preferring an extreme solution.

For both two-dimensional test problems and both user's preference functions, the convergence curves for NSGA-II and NEMO-II-G are very similar, indicating that NEMO-II-G was not able to use the provided preference information to substantially enrich the standard non-dominated sorting procedure.

Rather than comparing the value obtained after a specific number of generations, Table 5 shows the

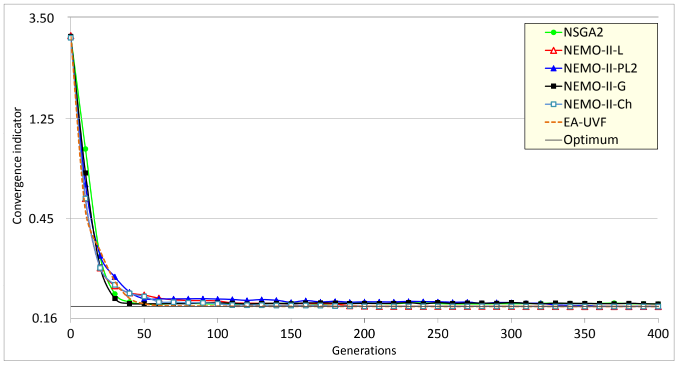

Figure 5: ZDT1-middle

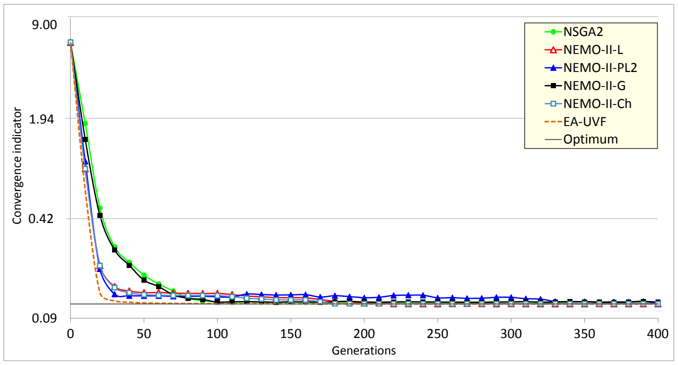

Figure 6: ZDT1-extreme

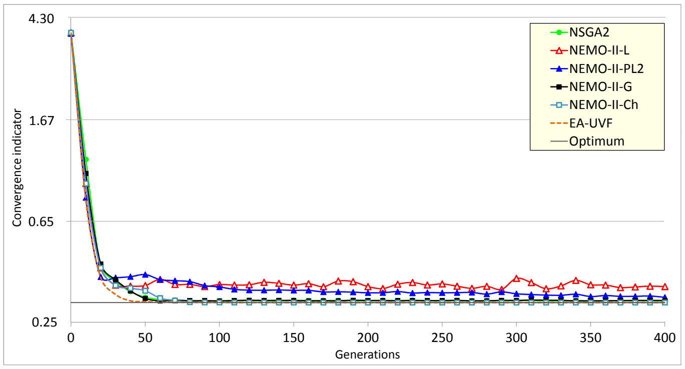

Figure 7: ZDT2-middle

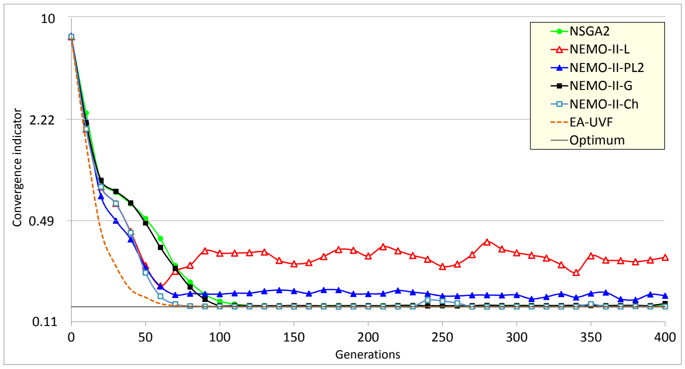

Figure 8: ZDT2-extreme

Table 5: Area under the curve for 2-dimensional problems, mean ± std. err. The difference of all results to the results of NEMO-II-Ch is significant except for NEMO-II-L in ZDT1.

	ZDT1	
	middle	extreme
NSGA-II	141.38 ± 1.88	200.03 ± 4.12
NEMO-II-L	127.32 ± 1.76	164.48 ± 4.09
NEMO-II-PL2	138.27 ± 1.51	180.14 ± 3.04
NEMO-II-G	139.38 ± 0.94	196.22 ± 2.57
NEMO-II-Ch	**126.73** ± 1.62	**163.73** ± 4.11
	ZDT2	
	middle	extreme
NSGA-II	199.96 ± 2.02	255.08 ± 4.67
NEMO-II-L	207.80 ± 1.95	275.34 ± 5.24
NEMO-II-PL2	205.17 ± 1.40	254.36 ± 3.79
NEMO-II-G	199.99 ± 1.08	252.38 ± 2.48
NEMO-II-Ch	**189.55** ± 1.61	**225.27** ± 4.56

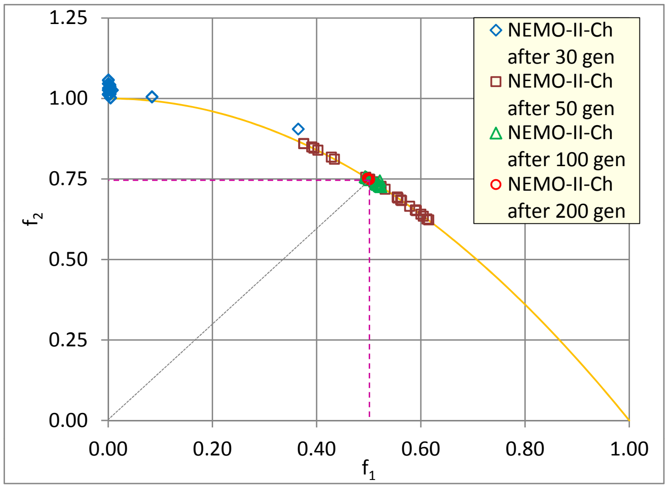

Figure 9: Convergence to the best preferred solution by NEMO-II-Ch for ZDT2-middle

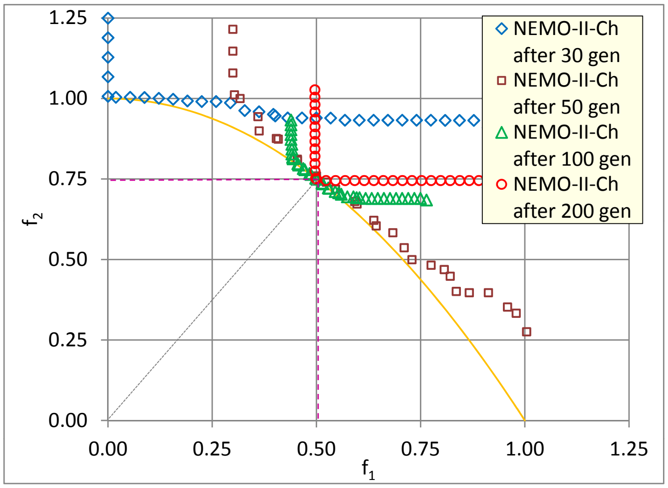

Figure 10: Attainment surfaces by NEMO-II-Ch for ZDT2-middle

area under the convergence curve over all 400 generations as a measure of the overall performance of the algorithm. Based on a Mann-Whitney-U test with 5% significance level, the differences between NEMO-II-Ch and each of the other tested algorithms are significant, except for ZDT1 and the difference to NEMO-II-L. The latter is not surprising, given that for such a convex problem a linear model is able to find any solution on the frontier.

To illustrate the convergence of NEMO-II-Ch in the objective space, we show in Figure 9 the population of solutions of ZDT2-middle after 30, 50, 100 and 200 generations. One can observe that after 30 generations the population of solutions is located in the upper left corner of the Pareto front which corresponds to one of the two minima obtained by a linear value function model. Then, in subsequent generations the method discovers that the linear model is not able to represent the growing set of pairwise comparisons and switches to the Choquet integral model. From generation 50 on, the method generates solutions grouped around the point most preferred by the artificial user. In generations 100 and then 200, the population is almost exactly focused on this point.

It is also interesting to observe the convergence process using the attainment surface plots [33] for the same problem. They are shown in Figure 10. The median attainment surface shows the area that is dominated by solutions from the population of a given generation in 50% of the runs. In Figure 10 the dominated area is located on the upper right side of the curves. The obtained surface after 30 generations has almost conic shape because the population of solutions is concentrated in the left upper corner of the Pareto front. After switching from the linear to the Choquet integral model the attainment surfaces become more and more conic and, finally, in generation 200 it is perfectly orthogonal. This means that the whole population is focused on one, most preferred point.

Results in 3D

In three dimensions we have compared the five methods on the benchmark problems DTLZ1 and DTLZ2 considering the user's Chebyshev value function with the parameters given in Table 6.

Table 6: Parameters of the user's value function in 3D

	w_1	w_2	w_3
DTLZ1-3D-middle	0.3	0.4	0.3
DTLZ1-3D-extreme	0.2	0.3	0.5
DTLZ2-3D-middle	0.3	0.4	0.3
DTLZ2-3D-extreme	0.2	0.3	0.5

What we can observe for DTLZ1 in Figures 11 and 12 is that only NEMO-II-Ch is able to get very close to the optimal solution. Indeed, after generation 200 the curve of NEMO-II-Ch merges with that of EA-UVF, which means the NEMO-II-Ch behaves like a single-objective EA using the true user's value function. NSGA-II is the second closest, but converges significantly slower. NEMO-II-L and NEMO-II-PL2 show erratic behavior, and NEMO-II-G converges very slowly, apparently unable to make use of the provided preference information. Note that the peaks in the plots of NEMO-II-L and NEMO-II-PL2 appear when in order to get a compatible model some of the oldest preference information has to be removed. This removal deteriorates temporarily the value of the best solution in the current population.

From Figures 13 and 14, one can notice that on DTLZ2, NEMO-II-Ch is slower than NSGA-II or even NEMO-II-G in the beginning, although it takes over eventually and finds better solutions. We are not sure what characteristics of DTLZ2 cause this behavior. Something seems to mislead the linear model in the wrong direction, which is doing really poorly overall. However, NEMO-II-Ch is able to recover from a bad start when switching to the Choquet integral, and eventually yields the best results.

As the numerical results in Table 7 confirm, overall, NEMO-II-Ch is still best in three out of the four scenarios.

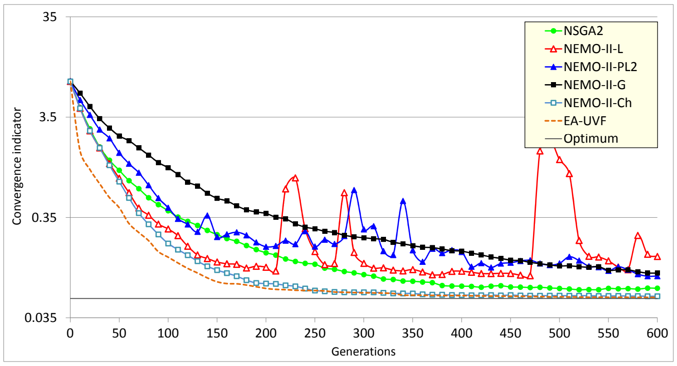

Figure 11: DTLZ1-3D-middle

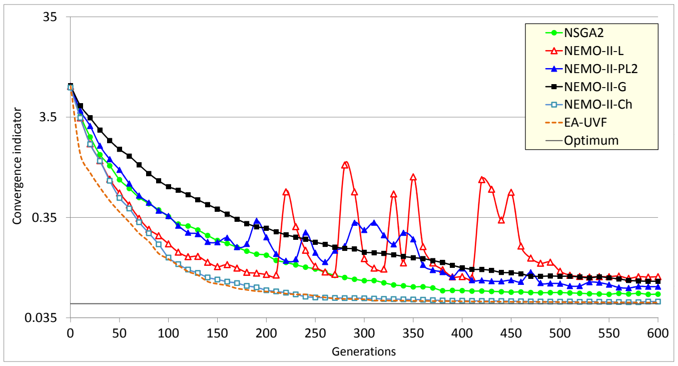

Figure 12: DTLZ1-3D-extreme

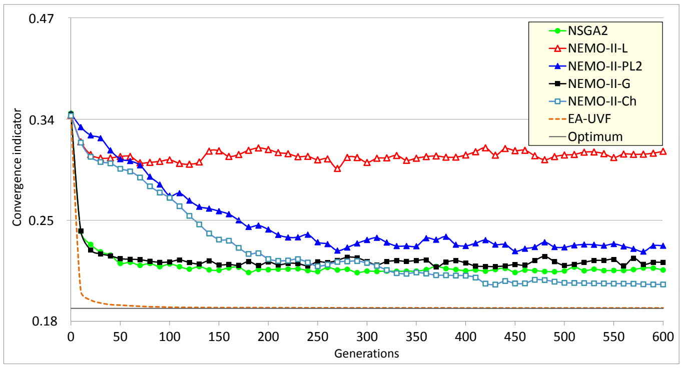

Figure 13: DTLZ2-3D-middle

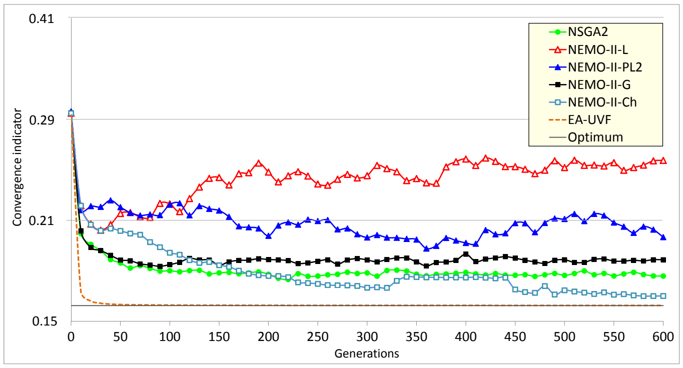

Figure 14: DTLZ2-3D-extreme

Table 7: Area under the curve for 3-dimensional problems, mean ± std. err. The following differences to NEMO-II-CH are not significant based on a Mann-Whitney-U test with 5% significance level: DTLZ2 extreme and NSGA-II, DTLZ2 middle and NEMO-II-G.

	DTLZ1	
	middle	extreme
NSGA-II	334.78± 9.64	291.60 ± 7.93
NEMO-II-L	391.65± 41.33	326.68 ± 27.10
NEMO-II-PL2	446.30±12.74	361.94 ± 9.14
NEMO-II-G	562.98 ± 13.34	439.64 ± 10.40
NEMO-II-Ch	**288.17** ± 11.89	**244.99** ± 8.99
	DTLZ2	
	middle	extreme
NSGA-II	**128.98** ± 0.09	107.01 ± 0.10
NEMO-II-L	182.87 ± 2.01	144.03 ± 6.04
NEMO-II-PL2	148.71 ± 0.64	124.69 ± 0.78
NEMO-II-G	131.58 ± 0.07	111.24 ± 0.08
NEMO-II-Ch	137.96 ± 2.31	**106.90** ± 1.33

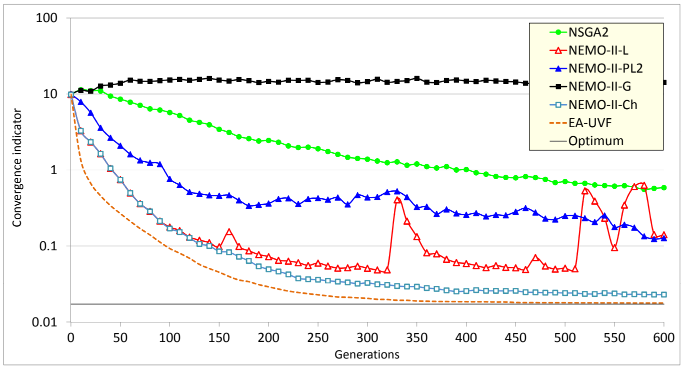

Figure 15: DTLZ1-5D-extreme$_1$

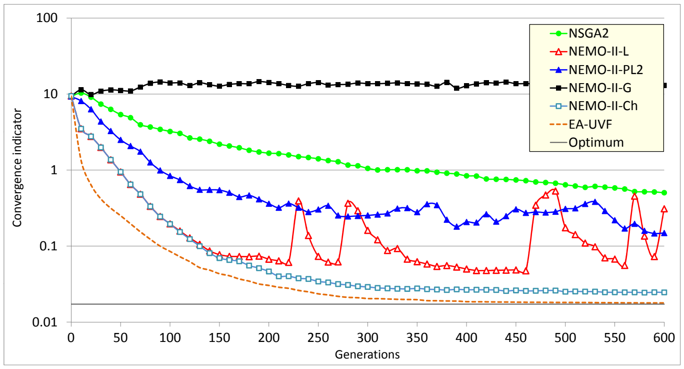

Figure 16: DTLZ1-5D-extreme$_2$

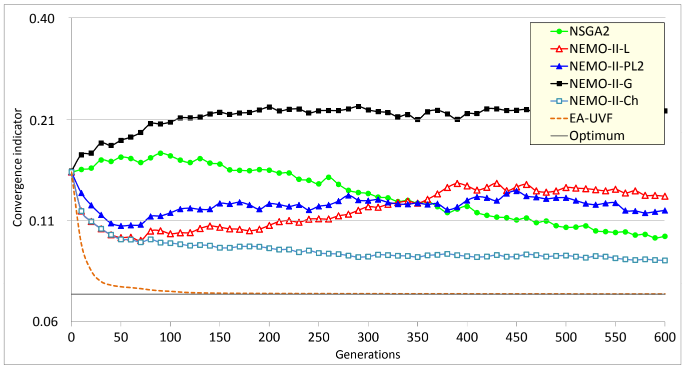

Figure 17: DTLZ2-5D-extreme$_1$

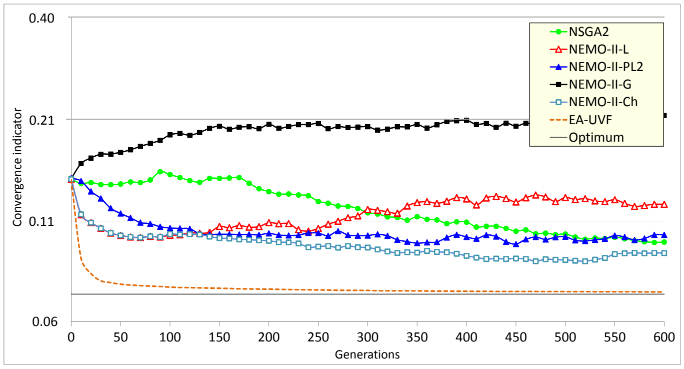

Figure 18: DTLZ2-5D-extreme$_2$

Results in 5D

In 5D we have considered the DTLZ1 and DTLZ2 benchmark functions for two user's Chebyshev value functions with parameters given in Table 8.

Table 8: Parameters of the user's value function in 5D

	w_1	w_2	w_3	w_4	w_5
DTLZ1-5D-extreme$_1$	0.1	0.15	0.2	0.25	0.3
DTLZ1-5D-extreme$_2$	0.3	0.25	0.2	0.15	0.1
DTLZ2-5D-extreme$_1$	0.1	0.15	0.2	0.25	0.3
DTLZ2-5D-extreme$_2$	0.3	0.25	0.2	0.15	0.1

Different from the 2D and 3D cases, in which the performance of NSGA-II was good, in the 5D case, NSGA-II performs quite poorly. This is not very surprising, as it is known that the non-dominance ranking does not work effectively in more than 3 dimensions. In such cases, preference information is hugely beneficial as it allows to substantially enrich the non-dominated ranking. Figures 15-18 thus show that NEMO-II-Ch

Table 9: Area under the curve for 5-dimensional problems, mean ± std. err. NEMO-II-Ch is significantly better than each other algorithm except for the comparison with NEMO-II-L on DTLZ1.

	DTLZ1	
	extreme$_1$	extreme$_2$
NSGA-II	1747.43 ± 70.29	1271.34 ± 50.33
NEMO-II-L	251.67 ± 19.80	250.81 ± 18.19
NEMO-II-PL2	586.27 ± 22.49	299.10 ± 21.89
NEMO-II-G	8569.95 ± 40.16	7531.91 ± 38.14
NEMO-II-Ch	**207.84** ± 9.75	**208.96** ± 9.25
	DTLZ2	
	extreme$_1$	extreme$_2$
NSGA-II	81.86 ± 1.34	74.20 ± 1.03
NEMO-II-L	73.47 ± 2.50	70.45 ± 1.95
NEMO-II-PL2	74.97 ± 0.51	63.59 ± 0.54
NEMO-II-G	129.65 ± 0.67	119.30 ± 0.71
NEMO-II-Ch	**57.02** ± 0.65	**58.06** ± 0.74

obtains much better results than any of the other algorithms on both benchmark problems and for both user preference functions. The two linear models NEMO-II-L and NEMO-II-PL2 again show an erratic behavior, presumably because they are not sufficiently complex to reflect the user's preference information. A bit surprising is the very poor behavior of NEMO-II-G. Apparently, in 5D, the general monotonic additive preference model is not helpful. Again, when looking at the numerical results in Table 9, we see that NEMO-II-Ch yields better results than all other algorithms in all cases.

1.5.3. Average utility

Finally, Table 10 looks at the *average* utility of all the individuals in the final population. This measure provides some information on how well the algorithm was able to focus the search onto the most preferred region of the search space, as keeping some individuals with poor utility in the population would hurt this performance measure.

Again, NEMO-II-Ch performs best for most scenarios. It is marginally worse than NEMO-II-L and NEMO-II-PL2 for ZDT1-middle where a linear value function model is sufficient to capture the user's preferences. And it is slightly worse than NEMO-II-PL2 on the 5-dimensional DTLZ2-extreme$_2$ problem, despite the fact that according to Figure 18, the best solution in the population of NEMO-II-PL2 is clearly worse than the best solution in the population of NEMO-II-Ch. The likely explanation is that although NEMO-II-PL2 converged to an inferior solution, it converged fully, whereas NEMO-II-Ch still had some diversity in the population that degraded the average utility.

1.6. Conclusions

In this paper, we presented the NEMO-II-Ch method. NEMO-II-Ch is an interactive evolutionary multiobjective procedure guided by user's preferences towards the most preferred part of the Pareto-optimal set. The novelties brought by the method consist in the following features:

- It is the first implementation and empirical evaluation of the NEMO-II idea.

Table 10: Average and stdv of the utilities for the individuals in the last population for all considered methods

	ZDT1-middle	ZDT1-extreme	ZDT2-middle	ZDT2-extreme
NSGA-II	0.3415 ± 0.117	0.3756 ± 0.2233	0.4247 ± 0.0868	0.5079 ± 0.2487
NEMO-II-L	$\mathbf{0.1825 \pm 0.0082}$	0.1182 ± 0.0643	0.4077 ± 0.1033	0.4304 ± 0.3214
NEMO-II-PL2	0.1853 ± 0.0155	0.1192 ± 0.0138	0.3364 ± 0.0374	0.3581 ± 0.1952
NEMO-II-G	0.3458 ± 0.0136	0.3740 ± 0.0299	0.4301 ± 0.0278	0.5298 ± 0.0519
NEMO-II-Ch	0.1856 ± 0.0262	$\mathbf{0.1169 \pm 0.0071}$	$\mathbf{0.3010 \pm 0.0106}$	$\mathbf{0.1418 \pm 0.0406}$

	DTLZ1-3D-middle	DTLZ1-3D-extreme	DTLZ2-3D-middle	DTLZ2-3D-extreme
NSGA-II	0.1252 ± 0.0373	0.129 ± 0.0545	0.2886 ± 0.049	0.3131 ± 0.1041
NEMO-II-L	3.5691 ± 16.8329	2.7865 ± 12.9957	0.3162 ± 0.0411	0.2812 ± 0.1038
NEMO-II-PL2	0.1237 ± 0.0521	0.1211 ± 0.0313	0.2454 ± 0.0404	0.2260 ± 0.0450
NEMO-II-G	0.1411 ± 0.0284	0.1384 ± 0.0098	0.3054 ± 0.0085	0.3324 ± 0.0113
NEMO-II-Ch	$\mathbf{0.0635 \pm 0.0197}$	$\mathbf{0.0541 \pm 0.0102}$	$\mathbf{0.2078 \pm 0.0395}$	$\mathbf{0.1705 \pm 0.0269}$

	DTLZ1-5D-extreme$_1$	DTLZ1-5D-extreme$_2$	DTLZ2-5D-extreme$_1$	DTLZ2-5D-extreme$_2$
NSGA-II	1.4225 ± 1.0363	1.9122 ± 1.7195	0.2168 ± 0.0864	0.1977 ± 0.0812
NEMO-II-L	3.0879 ± 13.0158	2.2674 ± 10.7805	0.1622 ± 0.0657	0.1381 ± 0.0465
NEMO-II-PL2	0.1922 ± 0.1720	0.1623 ± 0.3116	0.1491 ± 0.0468	$\mathbf{0.1214} \pm 0.0374$
NEMO-II-G	16.754 ± 4.772	17.494 ± 4.815	0.3806 ± 0.0620	0.3517 ± 0.0509
NEMO-II-Ch	$\mathbf{0.0368 \pm 0.0209}$	$\mathbf{0.0417 \pm 0.0205}$	$\mathbf{0.1449 \pm 0.0666}$	0.1353 ± 0.0687

- It does not work by considering only one model to translate the preferences of the DM but it starts from the simplest one (the linear model) and passes to a more complex one (the 2-additive Choquet integral model) when it is not possible to represent the DM's preferences using the linear model.

- The use of the Choquet integral preference model has never been considered in the evolutionary multiobjective optimization field for its relative complexity, however, we have demonstrated that it is able to deal efficiently with problems where preferences involve interactions among criteria, which additive preference models are unable to represent.

In order to demonstrate the effectiveness of the presented method, as well as the quality of its solutions, we have compared NEMO-II-Ch with NSGA-II and a variant of Greenwood's method (NEMO-II-L) on a variety of benchmark problems in 2D, 3D and 5D.

In almost all performed simulations, NEMO-II-Ch clearly obtained better results than the other tested methods.

Further developments will include to study how the increase of the number of interacting objectives in the k-additive Choquet integral for $k > 2$ influences the performance of the interactive procedure based on this preference model.

Acknowledgment: The second and the third authors wish to acknowledge funding by the "Programma Operativo Nazionale", Ricerca & Competitività "2007-2013" within the project PON04a2 E SINERGREEN-RES-NOVAE.
The last two authors wish to acknowledge financial support of the Polish National Science Center, grant no. DEC-2015/18/A/ST6/00171.

References

[1] S. Angilella, S. Greco, F. Lamantia, and B. Matarazzo. Assessing non-additive utility for multicriteria decision aid. *European Journal of Operational Research*, 158(3):734–744, 2004.

[2] S. Angilella, S. Greco, and B. Matarazzo. Non-additive robust ordinal regression: A multiple criteria decision model based on the Choquet integral. *European Journal of Operational Research*, 201(1):277–288, 2010.

[3] S. Angilella, S. Corrente, and S.. Greco. Stochastic multiobjective acceptability analysis for the Choquet integral preference model and the scale construction problem. *European Journal of Operational Research*, 240(1):172–182, 2015.

[4] R. Battiti and A. Passerini. Brain-computer evolutionary multiobjective optimization: A genetic algorithm adapting to the decision maker. *IEEE Transactions on Evolutionary Computation*, 14(5):671–687, 2010.

[5] J. Branke. Consideration of user preferences in evolutionary multi-objective optimization. In J. Branke, K. Deb, K. Miettinen, and R.Słowiński, editors, *Multiobjective Optimization - Interactive and Evolutionary Approaches*, volume 5252 of *LNCS*, pages 157–178. Springer, 2008.

[6] J. Branke. MCDA and Multiobjective Evolutionary Algorithms. In S. Greco, M. Ehrgott, and J. Figueira, editors, *Multiple criteria decision analysis: state of the art surveys*, chapter 22. Springer, 2nd edition, To appear.

[7] J. Branke, T. Kaußler, and H. Schmeck. Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32:499–507, 2001.

[8] J. Branke, K. Deb, K. Miettinen, and R. Słowiński. *Multiobjective Optimization: Interactive and Evolutionary Approaches*, volume 5252. Springer, 2008.

[9] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz. Interactive evolutionary multiobjective optimization using robust ordinal regression. In M. Ehrgott et al., editors, *International Conference on Evolutionary Multi-Criterion Optimization*, volume 5467 of *LNCS*, pages 554–568. Springer, 2009.

[10] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz. Interactive evolutionary multiobjective optimization driven by robust ordinal regression. *Bulletin of the Polish Academy of Sciences - Technical Sciences*, 58(3):347–358, 2010.

[11] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz. Learning Value Functions in Interactive Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 19(1):88–102, 2015.

[12] P. Campigotto and A. Passerini. Adapting to a realistic decision maker: experiments towards a reactive multi-objective optimizer. In *International Conference on Learning and Intelligent Optimization*, volume 6073 of *LNCS*, pages 338–341. Springer, 2010.

[13] A. Chateauneuf and J.Y. Jaffray. Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion. *Mathematical Social Sciences*, 17:263–283, 1989.

[14] G. Choquet. Theory of capacities. *Annales de l'institut Fourier*, 5:131–295, 1954.

[15] C. A. Coello Coello. Handling preferences in evolutionary multiobjective optimization: A survey. In *Congress on Evolutionary Computation*, volume 1, pages 30–37. IEEE, 2000.

[16] C.A. Coello Coello, D.A. Van Veldhuizen, and G.B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer, 2002.

[17] S. Corrente, S. Greco, M. Kadziński, and R. Słowiński. Robust Ordinal Regression in preference learning and ranking. *Machine Learning*, 2(93):381–422, 2013.

[18] K. Deb and S. Chaudhuri. I-MODE: An interactive multi-objective optimization and decision-making using evolutionary methods. Technical Report KanGAL Report No. 2007003, Indian Institute of Technology Kanpur, 2007.

[19] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI*, pages 849–858. Springer, 2000.

[20] K. Deb, A. Sinha, P.J. Korhonen, and J. Wallenius. An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE Transactions on Evolutionary Computation*, 14(5):723–739, 2010.

[21] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, 1993.

[22] J. W. Fowler, E. S. Gel, M. M. Köksalan, P. Korhonen, J. L. Marquis, and J. Wallenius. Interactive evolutionary multi-objective optimization for quasi-concave preference functions. *European Journal of Operational Research*, 206:417–425, 2010.

[23] M. Grabisch. The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research*, 89:445–456, 1996.

[24] M. Grabisch. *k*-order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, 92: 167–189, 1997.

[25] M. Grabisch and C. Labreuche. Fuzzy measures and integrals in MCDA. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 563–604. Springer, Berlin, 2005.

[26] M. Grabisch, C. Labreuche, and J.-C. Vansnick. On the extension of pseudo-boolean functions for the aggregation of interacting criteria. *European Journal of Operational Research*, 148(1):28–47, 2003.

[27] S. Greco, V. Mousseau, and R. Słowiński. Ordinal regression revisited: multiple criteria ranking using a set of additive value functions. *European Journal of Operational Research*, 191(2):416–436, 2008.

[28] G. W. Greenwood, X. S. Hu, and J. G. D'Ambrosio. Fitness functions for multiple objective optimization problems: combining preferences with Pareto rankings. In *Foundations of Genetic Algorithms*, pages 437–455. Morgan Kaufmann, 1997.

[29] A. Lopez Jaimes, A. Arias Montano, and C. A. Coello Coello. Preference incorporation to solve many-ojective airfoil design problems. In *Congress on Evolutionary Computation*, pages 1605–1612. IEEE, 2011.

[30] A. Jaszkiewicz. Interactive multiobjective optimization with the Pareto memetic algorithm. *Foundations of Computing and Decision Sciences*, 32(1):15–32, 2007.

[31] I. Karahan and M. Köksalan. A territory defining multiobjective evolutionary algorithms and preference incorporation. *IEEE Transactions on Evolutionary Computation*, 14(4):636–664, 2010.

[32] R.L. Keeney and H. Raiffa. *Decisions with multiple objectives: Preferences and value tradeoffs.* J. Wiley, New York, 1993.

[33] J. Knowles. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on*, pages 552–557. IEEE, 2005.

[34] P. Korhonen, J. Wallenius, and S. Zionts. Solving the discrete multiple criteria problem using convex cones. *Management Science*, 30:1336–1345, 1984.

[35] J.L. Marichal and M. Roubens. Determination of weights of interacting criteria from a reference set. *European Journal of Operational Research*, 124(3):641–650, 2000.

[36] J.A Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[37] S. Phelps and M. Köksalan. An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science*, 49(12):1726–1738, 2003.

[38] M. Pirlot, H. Schmitz, and P. Meyer. An empirical comparison of the expressiveness of the additive value function and the Choquet integral models for representing rankings. In *Proceedings of the 25th Mini-EURO conference on uncertainty and robustness in planning and decision making, URPDM, Coimbra, Portugal.* 2010.

[39] L. Rachmawati and D. Srinivasan. Preference incorporation in multi-objective evolutionary algorithms: A survey. In *Congress on Evolutionary Computation*, pages 3385–3391. IEEE, 2006.

[40] G.C. Rota. On the foundations of combinatorial theory I. Theory of Möbius functions. *Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 2:340–368, 1964.

[41] L. B. Said, S. Bechikh, and K. Ghedira. The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making. *IEEE Transactions on Evolutionary Computation*, 14(5):801–818, 2010.

[42] G. Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, 1976.

[43] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina. A preference-based interactive evolutionary algorithm for multiobjective optimization. *Evolutionary Computation Journal*, 17(3):411–436, 2009.

[44] D. S. Todd and P. Sen. Directed multiple objective search of design spaces using genetic algorithms and neural networks. In W. Banzhaf et al., editor, *Genetic and Evolutionary Computation Conference*, pages 1738–1743. Morgan Kaufmann, San Francisco, California, 1999.

[45] T. Wagner and H. Trautmann. Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *IEEE Transactions on Evolutionary Computation*, 14(5):688–701, 2010.