Pareto Landscapes Analyses via Graph-Based Modeling for Interactive Decision-Making

Ofer M. Shir[1,2⋆], Shahar Chen[1,3], David Amid[1], Oded Margalit[1], Michael Masin[1], Ateret Anaby-Tavor[1], and David Boaz[1]

[1] IBM Research, Haifa, Israel
[2] Computer Science Department, Tel-Hai College, Upper Galilee, Israel
[3] Computer Science Department, Technion - Israel Institute of Technology

Abstract. We consider two complementary tasks for consuming optimization results of a given multiobjective problem by decision-makers. The underpinning in both exploratory tasks is analyzing Pareto landscapes, and we propose in both cases discrete graph-based reductions. Firstly, we introduce interactive navigation from a given suboptimal reference solution to Pareto efficient solution-points. The proposed traversal mechanism is based upon landscape improvement-transitions from the reference towards Pareto-dominating solutions in a *baby-steps fashion* – accepting relatively small variations in the design-space. The Efficient Frontier and the archive of Pareto suboptimal points are to be obtained by population-based multiobjective solvers, such as Evolutionary Multiobjective Algorithms. Secondly, we propose a framework for automatically recommending a preferable subset of points belonging to the Frontier that accounts for the decision-maker's tendencies. We devise a line of action that activates one of two approaches: either recommending the top *offensive team* – the gain-prone subset of points, or the top *defensive team* – the loss-averse subset of points. We describe the entire recommendation process and formulate mixed-integer linear programs for solving its combinatorial graph-based problems.

Keywords: Pareto landscapes, multi-criterion decision-making, design-space, interactive recommender systems, prospect theory, graph traversals, vertex covering, submodular functions.

1 Introduction

The goal of multiobjective optimization is to output the Efficient Frontier and the Pareto Optimal Set, whose points are mathematically indifferent with respect to each other. Attaining the Efficient Frontier of a multiobjective optimization problem can be either treated by means of algorithms utilizing mathematical programming solvers, e.g., the so-called Diversity Maximization Approach [1] employing specific solvers, or approximated by population-based heuristics, such as Evolutionary Multiobjective Optimization Algorithms [2]. The selection

⋆ The first two authors contributed equally to this work.

Nomenclature

Term	Description	Notation
model	multiobjective optimization model	$\mathcal{M}$
Efficient Frontier	non-dominated points in the m-dimensional objective space	$\mathcal{F}$
Pareto Optimal Set	pre-images of the Frontier in the search space	$\mathcal{X}$
archive	history of solution-points obtained during optimization	$\mathcal{A}$
solution	a feasible solution to the given multiobjective problem	ψ
reference	a solution provided by the user to become an initial point	ψ_0
transition	an ordered pair of solution-points a and b	$(a \rightsquigarrow b)$
path	a sequence of transitions from the reference to the Frontier	$p,\ \mathcal{P}[:,:]$
metric	symmetric distance metric amongst solution-points	Δ
hop limit	maximally acceptable distance for transitions	$\delta_{\max}$
preference	a preference relation between two solution-points	$\preceq_{\text{pref}}$
graph	a graph with vertices V and directed edges E	$\mathcal{G} = \{V, E\}$
edge	a directed edge between node u to node v with weight ω	$(u \rightsquigarrow v,\ \omega)$

phase, entitled Multi-Criterion Decision Making (MCDM) [3], is to be subjectively driven by the human decision-maker (DM) based upon their preferences. This phase may involve exploration of the Frontier, and eventually, the challenge in selecting a solution is to account for gains and losses while adhering to the personal preferences. The current work is concerned with devising automated exploration recipes in two directions: (i) utilizing a *traversal mechanism* to discover (possibly suboptimal) points that may be of interest because of their design-space specification, and (ii) identifying recommended subsets of solution-points that meet the DM's tendencies. We propose here a common ground to treat both exploration tasks, that is, by reducing the challenge of landscape analyses to discrete graph-based formulations. While the traversal challenge is reduced to a *shortest path* problem on the archive of suboptimal solutions, the recommendation challenge is reduced to a *vertex covering problem* of an outranking graph. In the latter we show how to transform the analysis problem into mixed-integer linear programs as well as a formulation of maximizing a *submodular function* over the graph vertices. The Nomenclature for this paper is outlined in the enclosed table.

Related Work In essence, work related to the considered tasks lies in the areas of MCDM and interactive recommender systems. Preference elicitation and derivation of expected utility functions is rooted in the Multi-Attribute Utility Theory [4]. Analytical Hierarchy Process [5] introduced a questioning protocol for manual decision-making, based upon pairwise comparisons, and derived a calculation procedure featuring matrix algebra. The ELECTRE family of methods [6], on the other hand, defined a preference relation amongst solution-points, and proposed a procedure to construct a preference graph – these methods eventually recommend to the DM the solution-points that lie in this graph's *kernel*. Various approaches for reducing the cardinality of the Efficient Frontier and computing a subset of the "most interesting" solution-points have been proposed from different angles, e.g., in *preference ordering* [7] or in *smart filtering* [8]. Interactive recommender systems that suggest candidate solution-points to the user while accounting for multiple attributes (objectives) have long been proposed, e.g., in the realm of e-Commerce [9]. These methods typically infer the user's preferences that are communicated via so-called *critiques* in order to ad-

just the next suggested solution. Another approach, that is somewhat related to our proposed traversal, is the so-called Pareto Navigator [10], which interactively incorporates user preferences to the course of optimization. Finally, employing a graph-based approach to navigate amongst suggested solution-points has been proposed by Hadzic and O'Sullivan [11]. Their study aimed to select a product from a catalogue (e-Commerce) without considering explicit Pareto relations, and suggesting candidate solution-points by analyzing the graph.

2 Traversing over Pareto Landscapes

In order to promote acceptance of machine-driven optimization results by DMs and gain their confidence in such candidate solutions, we present here a novel exploration framework for the multiobjective domain, which constitutes a traversal over Pareto landscapes. The primary idea behind the Pareto traversal is to offer a hopping mechanism that begins at the DM's suboptimal reference point and terminates at the Efficient Frontier or prior to that, at any point satisfactory to the DM. In short, each hop is directed towards Pareto-improvements, and at the same time is meant to constitute a *baby-step* in the **design (decision) space**, i.e., transitions are improvements in the prescribed objectives but are only relatively small variations with respect to the current candidate design. The rationale behind this constraint is the fact that human perception tends to adjust better to small changes rather than to large. At the same time, a typical limitation on human *mental resource* upon reaching a decision [12] poses a tradeoff between the need for baby-step adjustments to the available mental resource (time and energy to decide).

The strict Pareto domination relation can be relaxed by means of a generalized preference relation between two solution-points. While we keep our formal definition of the Efficient Frontier as the output of the optimization process, we would like to consider a relaxed preference relationship amongst solution-points in the archive. Examples of such relations are described in [13], entitled therein (ε, λ)-dominance or the preference-elicited $\preceq_{\Theta}$ relation. In what follows, we shall denote a preference relation in this framework as $\preceq_{\text{pref}}$, and it is assumed to be prescribed by the user. The careful reader would note that this relation is not necessarily a partial order, and a preference graph that is based upon $\preceq_{\text{pref}}$-relations may possess cycles.

2.1 Proposed Method I: Batch Navigation

A successful Pareto optimization process yields the Efficient Frontier as well its Pareto Optimal Set, but may support in parallel the **archiving** of all its intermediate feasible solutions along this process. The obtained archive, whose non-dominated set is the Efficient Frontier, is likely to contain a vast majority of dominated solutions with diverse ranks. **This archive is of particular interest to the current study, since it encompasses the potential to bridge between a reference solution to Pareto optimal solutions by means**

`shortestPathToFrontier` (reference ψ_0, Frontier $\mathcal{F}$, Archive $\mathcal{A}$, metric Δ, preference $\preceq_{\text{pref}}$, lim $\delta_{\max}$)

$\mathcal{G} \longleftarrow$ `constructTraversalGraph`$(\mathcal{A}, \Delta, \preceq_{\text{pref}}, \delta_{\max})$
$\{\mathcal{D},\ \mathcal{P}\} \longleftarrow \mathcal{G}$.`Dijkstra`$(\psi_0, \mathcal{F})$
idx $\longleftarrow \arg\min(\mathcal{D})$ /* minimal accumulated distance to the Frontier */
return $\{\mathcal{P}[\text{idx}], \text{idx}\}$

of a guided hopping mechanism. Our proposed Pareto navigation system offers 3 optional tracks of guidance to the DM. In all cases, a symmetric distance metric is provided as input:

1. **Closest** : given a reference solution and an archive, return a Pareto efficient solution that is closest to the reference point in the scope of the design space.
2. **Shortest** : given a reference solution, an archive, a preference relation $\preceq_{\text{pref}}$, and the hop limit $\delta_{\max}$, construct the directed graph of all possible pathways, and return the shortest path from the reference to the Frontier. The shortest path can be either defined as the shortest sequence of hops to the Frontier (in terms of number of moves), or as the smallest accumulated distance to the Frontier. The design distance between each pair of consecutive solutions is bounded by $\delta_{\max}$. See `shortestPathToFrontier` for the pseudo-code.
3. **Classic** : given a reference solution, an archive, a preference relation, a maximal distance, and maximal number of suggested solutions, iteratively return a set of dominating solutions, close enough to the reference (bounded by $\delta_{\max}$ in the design space), lying on paths to the Frontier and of which the DM is requested to select. See `traverseToFrontier` for the pseudo-code.

We discuss in what follows the various components of the **Classic** traversal.

Shortest Path Calculations We view the archive of the optimization process $\mathcal{A}$ as a *directed graph*: each solution-point constitutes a vertex, and there is an edge from solution-point a to solution-point b if and only if $b \preceq_{\text{pref}} a$ holds and the design distance between a and b does not exceed $\delta_{\max}$. For instance, if $\preceq_{\text{pref}}$ is taken to be the standard Pareto domination – and if $\Delta(a,b) \leq \delta_{\max}$ and b dominates a both hold – then the graph will possess an edge $(a \rightsquigarrow b,\ \Delta(a,b))$. See `constructTraversalGraph` for the pseudo-code. By the current construction, a valid sequence of solutions to the Frontier is a valid path from the source vertex (ψ_0, the reference solution provided by the user) to one of the Frontier vertices, i.e., a solution-point in $\mathcal{F}$. In practice, we consider Dijkstra's shortest path algorithm [14], utilized on a directed graph $\mathcal{G}$, which receives a source ψ_0 and the Efficient Frontier vertices $\mathcal{F}$. Its implementation returns the shortest paths from the source to each of the vertices in $\mathcal{F}$, with both time and space complexity of $\mathcal{O}\left(|\mathcal{A}|^2\right)$. Its output, denoted in our notation as $\mathcal{G}$.`Dijkstra`, comprises the calculated distances $\mathcal{D}$ and the description of the paths $\mathcal{P}$. As mentioned earlier, $\mathcal{G}$ may possess cycles depending upon the definition of $\preceq_{\text{pref}}$, e.g., a cycle of

`constructTraversalGraph` (Archive $\mathcal{A}$, metric Δ, preference $\preceq_{\text{pref}}$, lim $\delta_{\max}$)

1: $V \longleftarrow \mathcal{A},\ E \longleftarrow \emptyset$
2: **for** $i = 1 \ldots |\mathcal{A}|$ **do**
3: **for** $j = 1 \ldots |\mathcal{A}|$ **do**
4: **if** $\mathcal{A}[i] \preceq_{\text{pref}} \mathcal{A}[j] \ \wedge\ \Delta(i,j) \leq \delta_{\max}$ **then**
5: $E \longleftarrow E \cup (j \rightsquigarrow i,\ \omega = \Delta(i,j))$
6: **return** $\mathcal{G} = \{V, E\}$

indifferent solution-points when $\preceq_{\text{pref}}$ is a weak Pareto-dominance relation. In such cases, Dijkstra's algorithm is still guaranteed to calculate the shortest paths to $\mathcal{F}$, but a feature to eliminate revisiting nodes iteration-wise is then needed (since Dijkstra is executed independently in each iteration), e.g., by means of Tabu-list elimination. A discussion on paths calculation efficiency in large-scale archives will follow.

The Interactive Hopping Mechanism Each iteration is meant to hold a specific solution, ψ_t, starting with the original reference ψ_0 at the beginning. At iteration t, the proposed method presents several candidate hops, and more specifically, it presents a limited number of solutions ($N_{\max}$ at most) that meet all the following **3 requirements**:

1. Satisfying the preference relation $\preceq_{\text{pref}}$ with regard to ψ_t (objective-wise)
2. Being within $\delta_{\max}$ from ψ_t (design-wise with respect to Δ)
3. Frontier-reachable: on a pathway of valid hops (i.e., via nodes that satisfy (1)+(2)) to the Efficient Frontier

The peak of each iteration is reached when the DM is asked by the machine to select the next hop within the proposed set, which we denote as $\mathcal{H}_{\mathcal{G}}$; this step is entitled `getDMSelection`: $\psi_{t+1} \longleftarrow$ `getDMSelection` $(\mathcal{H}_{\mathcal{G}}(\psi_t))$. Note that ψ_t may alternatively be defined as a set, rather than an individual solution, and the procedure can be adapted accordingly. A set perspective may suit certain DMs that are capable of simultaneously consuming multiple pathways.

Selection Criteria When calculating the shortest paths from the current solution-point ψ_t to the Frontier $\mathcal{F}$, as in `traverseToFrontier`, attention must be paid to the selection criterion amongst the paths. This is of particular relevance in scenarios where a large number of pathways exist, and the interaction with the DM should comprise a relatively small number of representative candidates. The default selection criterion (line 6 of `traverseToFrontier`) considers the minimization over the accumulated design space distance between ψ_t to $\mathcal{F}$ and the selection of the top $N_{\max}$ minimizers. While this criterion constitutes the primary motivation behind this newly proposed technique, there could be alternative criteria, or secondary criteria that would follow it. Possible ideas may comprise minimizing the total number of hops to the Frontier (i.e., considering

`traverseToFrontier` (reference ψ_0, Frontier $\mathcal{F}$, Archive $\mathcal{A}$, metric Δ, preference $\preceq_{\text{pref}}$, lim $\delta_{\max}$, suggestions $N_{\max}$)

$\mathcal{G} \longleftarrow$ `constructTraversalGraph`$(\mathcal{A}, \Delta, \preceq_{\text{pref}}, \delta_{\max})$
$t \longleftarrow 0,\ p \longleftarrow \emptyset$
repeat
  $\{\mathcal{D},\ \mathcal{P}\} \longleftarrow \mathcal{G}$.`Dijkstra`$(\psi_t, \mathcal{F})$ /* obtain distances and paths to $\mathcal{F}$ */
  $\mathcal{I} \longleftarrow$ `sort`$(\mathcal{D}$, 'ascend') /* hold the post-sorting permutation indices */
  $\mathcal{H}_{\mathcal{G}}(\psi_t) \longleftarrow \{v \mid v \in \mathcal{P}[1, \mathcal{I}(1 : N_{\max})]\}$
  $\boxed{\psi_{t+1} \longleftarrow \texttt{getDMSelection}(\mathcal{H}_{\mathcal{G}}(\psi_t))}$ /* get input from the user */
  $p[t] \longleftarrow (\psi_t \rightsquigarrow \psi_{t+1})$
  $t \longleftarrow t + 1$
until `terminatedByDM`()
`print`(p)
return ψ_t

the shortest path subject to graph's edge lengths uniformly set to 1), maximizing the overall improvements in all objectives (e.g., considering a global utility of all objective functions and accounting for its largest ascend), maximizing the *design diversity* in the target solution-points on the Frontier, or maximizing the objective diversity amongst the $N_{\max}$ selected points. The careful reader may foresee scenarios in which no paths exist per certain $\delta_{\max}$ values. Alternatively, the possibility of paths that comprise too many hops also exists. In both cases, we propose an interactive approach that may either iteratively adapt $\delta_{\max}$ based on feedback from the user, self-adapt it according to a heuristic, or hybridize the two approaches. At the same time, scenarios in which the local neighborhood of the current solution-point possesses an excessive number of candidates may arise. In these scenarios, we propose to apply secondary selection criteria in order to trim its size.

Space-Complexity in Large-Scale Archives A practical problem arises when the graph cardinality becomes excessively large (e.g., order of 20000 nodes). In order to address this issue, we aimed to exploit the tradeoff between memory and run-time and implemented a lighter version of Dijkstra that does not hold the entire graph in memory but rather calculates edge lengths *on-the-fly*. Consequently, the **space complexity** is reduced to $\mathcal{O}(|\mathcal{A}|)$ in comparison to $\mathcal{O}\left(|\mathcal{A}|^2\right)$ in the original algorithm. On the other hand, the graph edges are generated during the execution of the function, and therefore the graph construction is also performed more than once. Assuming the graph construction runs in t_1 seconds, and Dijkstra runs in t_2 seconds (both are $\mathcal{O}\left(|\mathcal{A}|^2\right)$), then k executions of Dijkstra-Light will take $k \cdot (t_1 + t_2)$, instead of $t_1 + k \cdot t_2$. Let $c = \frac{t_2}{t_1}$ denote the run-time ratio between graph construction and Dijkstra execution. Thus we obtain:

$$\frac{k \cdot (t_1 + t_2)}{t_1 + k \cdot t_2} = \frac{k \cdot (c+1)}{1 + k \cdot c} < \frac{k \cdot (c+1)}{k \cdot c} = 1 + \frac{1}{c}. \tag{1}$$

`onlineTraversal` (model $\mathcal{M}$, reference ψ_0, metric Δ, lim $\delta_{\max}$)

1: $t \longleftarrow 0$
2: $p \longleftarrow \emptyset$
3: **repeat**
4: $\quad \mathcal{R} \longleftarrow$ `formMO`$(\mathcal{M}, \psi_t, \Delta, \delta_{\max})$ /* formulate a local multiobjective problem */
5: $\quad \mathcal{N}_{\mathcal{G}}(\psi_t) \longleftarrow$ `solveLocal`$(\mathcal{R})$
6: $\quad \boxed{\psi_{t+1} \longleftarrow \texttt{getDMSelection}(\mathcal{N}_{\mathcal{G}}(\psi_t))}$ /* get input from the user */
7: $\quad p[t] \longleftarrow (\psi_t \rightsquigarrow \psi_{t+1})$
8: $\quad t \longleftarrow t+1$
9: **until** `terminatedByDM`()
10: `print`(p)
11: **return** ψ_t

In practice, graph construction tends to be significantly faster than Dijkstra execution, implying a loss of at most factor-2 in run-time with our so-called Dijkstra Light. Other heuristic techniques to treat large-scale archives may comprise filtering-out the following subsets: (a) non-dominating archive solutions (solutions that do not Pareto dominate the reference solution are not reachable and thus should be excluded from the graph), (b) unreachable vertices (i.e., solutions that are not reachable within $\delta_{\max}$ steps from ψ_0), and (c) vertices that do not belong to paths ending on the Frontier $\mathcal{F}$.

An illustration of the batch navigation process is provided in Fig. 1 on a 5-objective problem by means of Parallel-Coordinates visualization.

2.2 Proposed Method II: Online Navigation

This approach does not perform any calculations in advance, but rather solves the optimization model *on-the-fly* with interactive guidance by the DM. In what follows, reference is made to the pseudo-code entitled `onlineTraversal`. Here, a multiobjective optimization model $\mathcal{M}$ is provided as input to the algorithm, which then dynamically formulates a variant model $\mathcal{R}$ (line 4) in each iteration and solves it (line 5) on-the-fly to obtain the local Efficient Frontier with regard to the current solution-point ψ_t. The locality of the attained solutions in the design space has to satisfy the constraints prescribed by $\mathcal{R}$, e.g., $\Delta(\psi_t, \boldsymbol{x}) \leq \delta_{\max}$. An example for a multiobjective optimization approach that accounts for decision space constraints was reported in [15]. The selection of the decision maker is recorded (line 6) and becomes the solution-point of the consecutive iteration.

3 Automated Recommendation

In this section we are interested in formulating an automated methodology for recommending specific solution-points within the Efficient Frontier with minimal

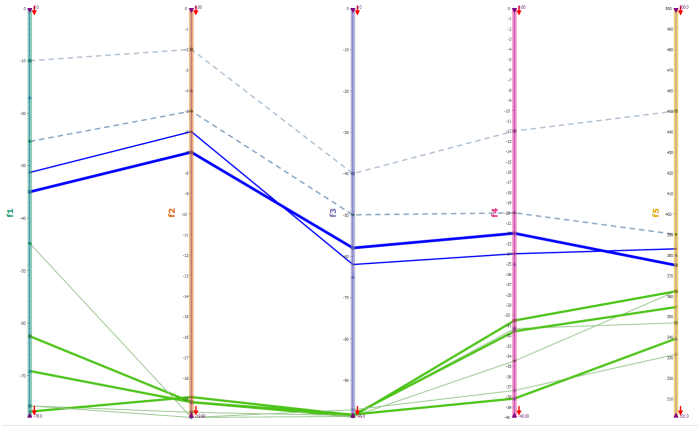

Fig. 1: A realization of the batch navigation process for a 5-objective minimization problem, depicted by means of Parallel Coordinates, with arbitrary units. The dotted lines represent the already traveresed path, whereas the solid blue lines represent two candidate solution-points being proposed to the DM at the current iteration. Furthermore, the blue solid **thick** line represents the currently examined solution-point, and the green solid **thick** lines are solution-points that may be reached from the examined option via calculated feasible pathways.

a priori information elicited from the DM. The Efficient Frontier is assumed to be computed in a satisfactory manner in advance and to be provided as input to the proposed process (Generate-First-Choose-Later fashion). One of the principal directions of this study is the consideration of Prospect Theory [16], whose core involves two perspectives, namely gain-prone versus loss-aversive, in human decision-making. While being gain-prone and striving to maximize profit is the economically rational perspective, it is argued that being loss-aversive and avoiding temporary losses/risk at any cost often *takes-over* human decision-making. Here, the general idea is to adopt this consideration and form two tracks of recommendation – *loss-averse* (LA) and *gain-prone* (GP) perspectives – and introduce them to recommender systems in MCDM. We coin terms, which are borrowed from American Football, and consider respectively two *teams*: the LA team, formed by the solution-points excelling in *defense*, alongside the GP team, formed by the solution-points excelling in *offense*. This partition to tracks is not revealed to the DMs, and the recommender system is to select the appropriate track according to their elicited tendencies. Given an Efficient Frontier of size N, we propose to either consider it as a whole or partition it to clusters, and treat the tasks of either *global* or *local* recommendations, respectively. The partitioning of the Efficient Frontier, by means of unsupervised clustering, is to be followed by the identification of so-called winners per each partition, yielding overall n recommended solutions. The framework is summarized by the following steps:

1. Partitioning the Efficient Frontier, $\mathcal{F} = \left\{\boldsymbol{f}^{(i)}\right\}_{i=1}^{N}$, into κ clusters of sizes $N_1, N_2, \ldots, N_\kappa$.
2. For each cluster i, constructing a **complete directed graph** with N_i vertices, each representing a solution-point. Pairwise outranking calculations obtain the degree to which a solution is preferred over another (if at all), and consequently, a weighted edge is constructed between them directed toward the outranked solution. The weight, $w_e \in [0, 1]$, quantifies the degree of preference of one solution over the other, by construction. The calculation of this degree is inspired by *Fuzzy Logic K-Optimality* [17], and certain variants of the ELECTRE family of methods [6].
3. Selecting a "good subset" of vertices for each cluster, referred to as the "winners" or the "top team".

Fig. 2 summarizes the entire recommendation process that we envision. Also, see `recommend` for the pseudo-code of the proposed recommendation recipe.

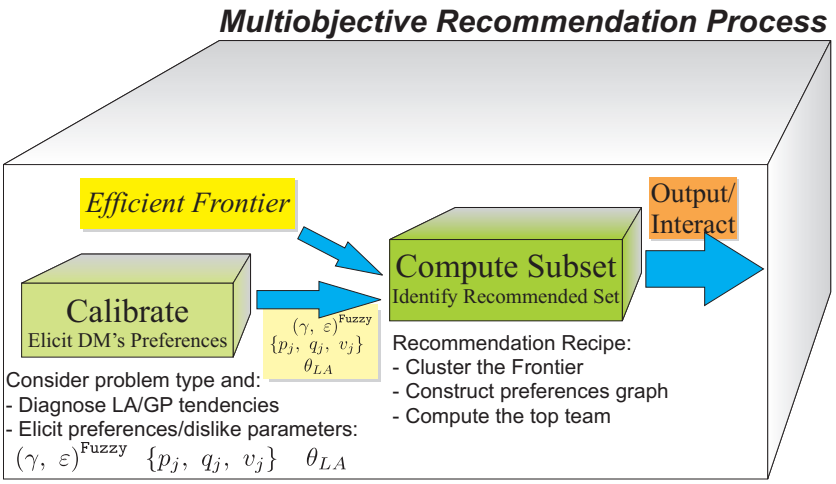

Fig. 2: Summary of the envisioned recommendation process. The current study focuses on the subset attainment, whereas the calibration is not addressed here.

3.1 Clustering

Clustering is meant to reduce the complexity of dealing with a potentially large Efficient Frontier into treatment of its semantically-derived subsets. We propose to divide the Frontier into smaller regions and let the DM focus only on areas that they find particularly interesting. We consider each cluster as an independent set of solution-points, and recommend n_i out of N_i solutions that reside in the i^{th} cluster ($\frac{n_i}{N_i} \approx \frac{n}{N}$). A possible realization is Lloyd's k-means clustering [18].

`recommend`(Efficient Frontier $\mathcal{F}$, numClusters κ, int `mode`)

$\Gamma \longleftarrow$ `cluster` $(\mathcal{F},\ \kappa)$
for $i = 1 \ldots \kappa$ **do**
  $\mathcal{G}_i \longleftarrow$ `calcOutrankingGraph` $(\mathcal{F}(\Gamma(i,:)))$
  if `mode`==**GP then**
    $\mathcal{W}_i \longleftarrow$ `selectOffensiveTeam` $(\mathcal{G}_i)$
  else
    $\mathcal{W}_i \longleftarrow$ `selectDefensiveTeam` $(\mathcal{G}_i)$
return $\{\mathcal{W}_i\}_{i=1}^{\kappa}$ /* top teams per cluster */

`calcOutrankingGraph`(solutions $\mathcal{F}$)

`initialize pairwise preference matrix` $\Omega = (\omega_{i,j}) \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|},\ \omega_{i,i} = 0$
$V \longleftarrow \mathcal{F}$
$E \longleftarrow \emptyset$
for $i = 1 \ldots |\mathcal{F}|$ **do**
  for $j = 1 \ldots |\mathcal{F}|$ **do**
    switch (`mode`)
    case `K-OPT`**:**
      $\omega_{i,j} \longleftarrow 1/\min_k \left(\boldsymbol{f}^{(i)} \prec_k \boldsymbol{f}^{(j)}\right)$ /* see Eq. 3 */
    case `ELECT-III`**:**
      $\omega_{i,j} \longleftarrow \sigma\left(\boldsymbol{f}^{(i)},\ \boldsymbol{f}^{(j)}\right)$ /* see Eq. 4 */
    case `ELECT-IS`**:**
      $\omega_{i,j} \longleftarrow \max_s \left(\boldsymbol{f}^{(i)} \preceq_s \boldsymbol{f}^{(j)}\right)$ /* see Eq. 5 */
    end switch
    $E \longleftarrow E \cup (i \rightsquigarrow j,\ w = \omega_{i,j})$
return $\mathcal{G} = \{V, E\}$ /* complete directed graph */

3.2 Outranking Relations: Pairwise Comparisons

As a second step, the machine performs a series of pairwise comparisons amongst the solution-points in each cluster. Comparing two solutions a and b and determining what is the confidence level that a is preferable over b is somewhat a simpler task than conducting global prioritization over a set of solutions. Thus, we perform $\sum_{1\leq i\leq\kappa} 2\binom{N_i}{2}$ pairwise comparisons (rather than $2\binom{N}{2}$ in the global perspective, which is $\sim \kappa$ faster), and a directed weighted graph encompassing this entire information is then constructed per each cluster. In essence, an edge in this graph, directed from vertex a toward vertex b, represents the confidence level regarding the assertion "solution-point a is favorable over solution-point b". We consider three estimation techniques for quantifying such confidence levels: one is based upon K-Optimality and *Fuzzy Logic* [17], while the other two stem from the so-called *ELECTRE-III* and *ELECTRE-IS* techniques [6]. See `calcOutrankingGraph` for the pseudo-code of the graph construction.

Preference Estimation with K-Optimality and Fuzzy Logic Following [17], we adopt the k-dominance relation, which we shall define in what follows concerning solution-points $\boldsymbol{f}^{(1)}$ and $\boldsymbol{f}^{(2)}$ subject to vector minimization. First, consider the following functions,

$$
\begin{aligned}
n_b\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right) &\equiv \left|\left\{i \in [0 \dots m] \,\middle|\, f_i^{(1)} < f_i^{(2)}\right\}\right| \\
n_e\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right) &\equiv \left|\left\{i \in [0 \dots m] \,\middle|\, f_i^{(1)} = f_i^{(2)}\right\}\right| , \\
n_w\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right) &\equiv \left|\left\{i \in [0 \dots m] \,\middle|\, f_i^{(1)} > f_i^{(2)}\right\}\right|
\end{aligned}
$$

counting better/equal/worse coordinates in a pairwise vector comparison. Let m denote the objective-space dimensionality, then the following straightforward equality and inequality hold on any **two different solution-points residing on the Efficient Frontier**: $n_b + n_e + n_w = m, \quad 0 \leq n_b, n_e, n_w < m$. We employ *Fuzzy Logic* to relax this standard definition. When conducting pairwise comparisons, accounting for the *level of improvement* between two solutions may become an important factor. In order to encompass this aspect, the important concept of fuzziness is discussed here in the form of fuzzy membership functions. The definition of n_b, n_e and n_w can be revisited upon consideration of the fuzzy membership functions μ_b, μ_e and μ_w (note the aggregated subscripts):

$$
n_{\{b,e,w\}}\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right) \equiv \sum_{i=1}^{m} \mu_{\{b,e,w\}}^{(i)}\left(f_i^{(1)} - f_i^{(2)}\right)
$$

The specific shape of membership is to be decided upon, e.g., by setting it to linear fuzzy membership functions. Given these $\{n_b, n_e, n_w\}$ measures, evaluated by means of the direct or fuzzy notion, we would like to form hierarchy of subclasses of solutions within the Efficient Frontier. We state that solution-point $\boldsymbol{f}^{(1)}$ k-dominates solution-point $\boldsymbol{f}^{(2)}$, denoted as $\boldsymbol{f}^{(1)} \prec_k \boldsymbol{f}^{(2)}$, if and only if

$$
\boldsymbol{f}^{(1)} \prec_k \boldsymbol{f}^{(2)} \Longleftrightarrow n_e\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right) < m \ \wedge \ \frac{n_w\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right)}{n_b\left(\boldsymbol{f}^{(1)}, \boldsymbol{f}^{(2)}\right)} \leq k \tag{2}
$$

with $0 \leq k \leq 1$. Accordingly, k-optimality can be then defined: given $0 \leq k \leq 1$, $\boldsymbol{v}^*$ is the k-optimum if and only if there is no other solution $\boldsymbol{v}$, such that $\boldsymbol{v}$ k-dominates $\boldsymbol{v}^*$. Eq. 2 with $k = 0$ is essentially reduced to the formal Pareto domination relation, i.e., all the solution-points on the Efficient Frontier are 0-optimal in the strict (non-fuzzy) notion. We introduce the *degree of k-optimality* of a given solution-point $\boldsymbol{v}$ as the maximal k for which $\boldsymbol{v}$ is k-optimal. Hence, the "resistance" of a given solution increases as its degree of optimality increases, and therefore, a solution with a larger degree is more likely to be preferred over others. In order to compare amongst solution-points, we **define** the following pairwise preference function regarding solution-points a and b lying on an Efficient Frontier ($k > 0$):

$$
F_{\texttt{K-OPT}}(a, b) = \begin{cases} 1/\min_k (a \prec_k b) & \text{if } \texttt{min non-empty} \\ 0 & \text{otherwise} \end{cases} . \tag{3}
$$

Preference Estimation with ELECTRE The *ELECTRE* family of techniques [6] introduces a different approach for conducting automated pairwise comparisons. Generally speaking, ELECTRE focuses on quantifying the degree of loss/inferiority when comparing between solution-points a and b. Accordingly, as the level of loss increases, the assertion that a is preferred over b weakens. Here, the first value to be computed is the *concordance*, denoted as $c_{(q)}(a,b)$, which is defined as the fraction of objectives in which the values in solution-point a outrank the values in solution-point b with a tolerated error of q. This parameter, as others, may be utilized either with or without a coordinate subscript, indicating that it is either per-coordinate or global, respectively. At the same time, a *discordance* index $d^{(j)}_{(p,v)}$ is defined as the complementary measure to the concordance index; it is formulated for the j^{th} coordinate, and it accounts for losses between p_j to v_j. Losses beyond the latter parameter, which is entitled the *veto threshold*, would eliminate the outranking assertion. In other words, $d^{(j)}_{(p,v)} \in [0,1]$ is proportional to the loss in $[-p_j, -v_j]$ – if the loss exceeds $-v_j$ then $d_j = 1$ and veto is applied; if the loss does not exceed $-p_j$ then $d_j = 0$. We are interested in two particular ELECTRE variants, as specified in what follows.

ELECTRE-III Here, the score for the assertion that a outranks b is defined using a *credibility index* $\sigma(a,b)$, which quantifies the certainty of that assertion:

$$\sigma(a,b) = c(a,b) \cdot \prod_{j:d^{(j)}_{(p,v)}(a,b) > c_{(q)}(a,b)} \frac{1 - d^{(j)}_{(p,v)}(a,b)}{1 - c_{(q)}(a,b)}$$

We propose this index as a **definition** for a pairwise preference function:

$$F_{\texttt{EL-III}}(a,b) = \sigma(a,b) \tag{4}$$

ELECTRE-IS In this variant, on the other hand, a threshold s is given, and the assertion that a outranks b is quantified by means of a boolean index, denoted as $a \preceq_s b$, and holds if and only if

$$\begin{cases} c_{(q)}(a,b) \geq s \\ \forall j \quad d^{(j)}_{(p,v)} \geq -v + (v-p) \cdot w\left(s, c_{(q)}(a,b)\right) \end{cases},$$

where $w\left(s, c_{(q)}(a,b)\right) = \frac{1 - c_{(q)}(a,b)}{1-s}$. Equivalently to the k-optimality in Eq. 3, we **define** a pairwise preference function:

$$F_{\texttt{EL-IS}}(a,b) = \max_s \left(a \preceq_s b\right). \tag{5}$$

Outranking Aftermath A directed graph with edge weights possessing minimal values of 0 (indicating that a is certainly not better than b) and otherwise larger positive values (indicating the degree to which a is preferred over b) is then constructed based upon one of the three approaches (Eqs. 3, 4 or 5). Note that

the outcomes of the three proposed pairwise preference functions are dependent upon their parameter settings: calibrating the *fuzzy membership functions*, or setting the parameters $\{p_j, q_j, v_j\}$ for ELECTRE is critical and is likely to be problem-dependent. While the *fuzzy scoring* concept constitutes a strong tool, the *k-optimality* approach as a whole seems to lack the consideration of the LA perspective, but rather to strongly reflect the GP perspective. Also, the measures $\{n_b, n_e, n_w\}$ do not entirely estimate the DM's utility function because summation over losses versus gains does not capture well one's preferences. On the other hand, the ELECTRE techniques do capture the LA perspective and possess the potential to identify incredibility. At the same time, when a solution-point a is potentially preferred over b, the credibility index becomes inaccurate – especially in cases of a few objectives (up to 4 or 5): minor *loss* variations may change the concordance value (affecting s or σ, respectively).

A Novel Hybrid Perspective In order to address those effects and yet to benefit from both methods, we propose to hybridize ELECTRE with *fuzzy k-optimality* as follows: utilize ELECTRE to measure incredibility (accounting for LA), and employ *fuzzy* membership functions to evaluate preference (accounting for GP). Here is the proposed pairwise preference function:

$$F_{\texttt{HYB}}(a,b) = \begin{cases} 0 & \text{if } \sigma(a,b) < \theta_{LA} \\ F_{\texttt{GP}}(a,b) & \text{otherwise} \end{cases} . \tag{6}$$

We are now left with setting the LA threshold, θ_{LA}, and devise a fuzzy function that estimates the GP preference. To this end, we propose a fuzzy linear function that computes the average gain as an estimator for the DM's utility function: $F_{\texttt{GP}}(a,b) = \min\left\{0, \frac{n_b - n_w}{m}\right\}$.

3.3 Selection

In the final step of our proposed framework we are to obtain the subset of recommended solutions given the preferences graph. As a reference, in some ELECTRE variants the recommended subset is chosen to be the *graph kernel* $\mathcal{K}$[4], yet following a different graph construction. In our reckoning, kernels do not directly fit the current framework for several reasons. Firstly, they are defined for *unweighted graphs* while our preference graph is *weighted*; secondly, their size is fixed and the proposed recipe wishes to freely set the size of the recommended subset; and finally, not every graph has a kernel (for instance, a directed cycle with $2n-1$ vertices has an empty kernel). We consider two selection tracks:

1. GP track: solutions that "win the most" form the *top offensive team*
2. LA track: solutions that "lose the least" form the *top defensive team*

[4] The kernel $\mathcal{K}$ of a graph is a subset of vertices that is both independent (that is, if $u, v \in \mathcal{K}$, then $(u,v), (v,u) \notin E$), and dominating (that is, for any $v \notin \mathcal{K}$, there exists $u \in \mathcal{K}$ such that $(u,v) \in E$).

A naïve line of action would examine the edges around each vertex, and calculate its *degree of optimality*. In the GP track, it refers to the maximal gain over other solution-points, i.e., $deg(v) = \max_{e \in \delta_{out}(v)} w_e$. In the LA track, it refers to minimizing the maximal preference of another solution-point over it, $deg(v) = \max_{e \in \delta_{in}(v)} w_e$. Eventually, solution-points with the highest degree are recommended. The problem with this naïve line of action is that each solution is independently scored, and the obtained subset is not likely to be diverse. We would like to aim for a subset of recommended solutions being good representatives of the Frontier, and we do not wish to solely rely on clustering in achieving diversity. Next, we propose selection lines of action per each track.

GP Track: The Top Offensive Team Motivated by the graph kernel idea, we attempt at finding a suitable definition to our recommendation case. Intuitively, the definition of recommendation seems to be about selecting a solution subset of a given size n, which optimally "covers" all solution-points. Thus, we try to relax the notion of a *Dominating Set* for weighted graphs as follows. For each set $\mathcal{D}$, we define the *covering degree* of each vertex as,

$$cvr\left(\mathcal{D}, v\right) = \begin{cases} 1 & \text{if } v \in \mathcal{D} \\ \max\limits_{u \in \mathcal{D}, (u,v) \in E} w(u,v) & \text{otherwise} \end{cases}. \tag{7}$$

We define the *covering degree* of each set as the total degree of all vertices,

$$cvr\left(\mathcal{D}\right) = \sum_{v \in V} cvr\left(\mathcal{D}, v\right). \tag{8}$$

Although the independence property of the kernel is not explicitly expressed here, the choice of maximization, rather than summation, in the definition of $cvr\left(\mathcal{D}, v\right)$ prevents us from choosing two similar solutions (the marginal gain of the second solution would be relatively small). An alternative approach that we do not pursue here is to promote diversity by selecting solution-points with "victories" in all coordinates (e.g., by satisfying the $\bigvee_{i=1}^{m}$ operator). We associate the recommended subset with the solution of the following optimization problem:

$$\max_{|\mathcal{D}| \leq n} cvr\left(\mathcal{D}\right) \tag{9}$$

We formalize this problem by means of an integer program. Given the desired number of recommended solutions N_{rec}, let $w(u,v)$ be the (preference) weighted edge between solution-points u and v. We consider binary decision variables $\texttt{sol}[1..|V|]$ to represent the selection of a solution-point (vertex), and binary decision variables $\texttt{cover}[1..|V|][1..|V|]$ where $\texttt{cover}[v][u] = 1$ translates to selecting vertex v to cover vertex u. **P1** in Fig. 4, which is a mixed-integer linear program (MILP), realizes Eq. 9. Even though global domination over the entire graph is guaranteed, there is a chance that the "most dominant" solutions will not be selected upon solving **P1**, as illustrated in Fig. 3: Evidently, the best single vertex to be picked therein is b since it possesses a covering degree of 3. At the same time, when targeting a subset of size 2, any solution that contains b has a degree of at most 4.6, while the degree of $\{a, c\}$ is 4.7.

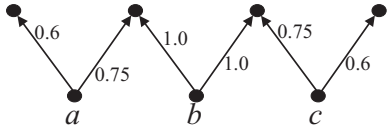

Fig. 3: Illustrating a scenario in the SubDominatingSet selection mode (Eq. 9) where the "most dominating" solution-point is not necessarily picked.

Maximizing the Submodular Coverage Function in a Greedy Approach An alternative approach would be to generate $\mathcal{D}$ greedily, i.e., to select in each iteration a vertex that maximizes the covering degree considering the solutions that were already chosen. This may degrade the total quality of the subset, but would not miss out the most dominant solutions. At the practical level, the *greedy* approach possesses a straightforward implementation, featuring the following step for each iteration $k = 0 \dots N_{\text{rec}} - 1$ (setting $\mathcal{D}^0 \leftarrow \emptyset$):

$$\mathcal{D}^{k+1} \leftarrow \mathcal{D}^k \cup \left\{ \arg\max_{v \in V} \left\{ cvr\left(\mathcal{D}^k \cup \{v\}\right) - cvr\left(\mathcal{D}^k\right) \right\} \right\} \tag{10}$$

Since solving Eq. 9 constitutes maximization of a *submodular monotone function*, **utilizing Eq. 10 guarantees a** $(1 - 1/e)$**-approximation to it [19]**.

LA Track: The Top Defensive Team An alternative selection of a recommended subset of solutions would be the defensive angle. Here, we aim at identifying a subset of solutions, ideally diverse, which are not outranked altogether by a single solution. For each *resisting set* $\mathcal{R}$, we define the degree of each vertex as

$$deg\left(\mathcal{R}, u\right) = \sum_{v \in \mathcal{R} \setminus u} w\left(u, v\right). \tag{11}$$

The *resistance degree* of each set is then defined as the maximal degree of all vertices (i.e., the *strongest offense* on $\mathcal{R}$):

$$res\left(\mathcal{R}\right) = \max_{u \in V} deg\left(\mathcal{R}, u\right). \tag{12}$$

[P1] maximize $\sum_{v \in V} \sum_{u \in V} w(u,v) \cdot \texttt{cover}[v][u]$
subject to:
$\sum_{v \in V} \texttt{sol}[v] \leq N_{\text{rec}}$
$\sum_{v \in V} \texttt{cover}[v][u] \leq 1 \ \forall u \in V$
$\texttt{cover}[v][u] \leq \texttt{sol}[v] \ \forall u \in V \ \forall v \in V$

[P2] minimize $\texttt{t}$
subject to:
$\sum_{v \in V} \texttt{sol}[v] \geq N_{\text{rec}}$
$\sum_{v \in V} w(u,v) \cdot \texttt{sol}[v] \leq \texttt{t} \ \forall u \in V$

Fig. 4: Mixed-integer linear programs realizing Eqs. 9 and 13.

In a min-max fashion, we describe the recommended resisting subset as the solution to the following optimization problem:

$$\min_{|\mathcal{R}|\geq n} res\left(\mathcal{R}\right). \tag{13}$$

Given the desired number of solutions N_{rec}, let $w(u,v)$ be the weighted edge between solution-points u and v. We consider binary decision variables $\texttt{sol}[1..|V|]$ to represent a selection of a solution-point, and a dummy continuous decision variable $\texttt{t}$. **P2** in Fig. 4 constitutes a realization to Eq. 13. A *greedy* approach can also be employed in the defensive context. In an analogous way, it is easy to compute the subset of solution-points having the strongest resistance degrees with the following step for each iteration $k = 0 \ldots N_{\text{rec}} - 1$ (setting $\mathcal{R}^0 \leftarrow \emptyset$):

$$\mathcal{R}^{k+1} \longleftarrow \mathcal{R}^k \cup \left\{ \arg\min_{v} \sum_{\mathcal{R}^k \cup \{v\}} w\left(u,v\right) \right\} \tag{14}$$

A Note on Duality One may think that LA is a dual of GP in a sense that any method to solve one problem can easily be transformed to the other (e.g., one can switch the direction of the edges and get a solution to a GP problem with an LA-solver). This intuition fails as the following example shows – assume a star sub-graph (i.e., there exists a vertex with edges directed at all other vertices), then a GP with $N_{\text{rec}} = 1$ would select it as a singleton and guarantee an optimal solution regardless of the edges between all the other vertices. Upon reversing the edges' direction and solving for LA we would get some subset of the other vertices depending upon the edges amongst them.

4 Summary

We introduced a novel interactive approach for traversing over multiobjective landscapes post-optimization. The proposed traversal is meant to facilitate guided exploration from a reference suboptimal solution to Pareto dominant solutions, which constitute improvements in the objective-space and yet are within baby-steps in the design-space. This proposed navigation, which was derived in both batch and online versions, is targeted at gaining the DM's confidence in the machine-driven Pareto-optimal solutions. At the same time, the inherent tradeoff between the need for adjusting to new conceptual designs to the limited mental resource that characterize DMs was raised. We then proposed a novel MCDM automated recommendation process that has the capacity to facilitate both LA and GP states-of-mind. We derived multiple candidate formulations for describing the outranking relations amongst solution-points on the Efficient Frontier, and devised a recipe for identifying the top subsets by means of combinatorial optimization or greedy approximations. It should be stressed that determining the DM tendency to be GP or LA is beyond the scope of the current work, and is left for future work on the cognitive aspects of such recommender systems.

References

1. Masin, M., Bukchin, Y.: Diversity maximization approach for multiobjective optimization. Operations Research **56**(2) (2008) 411–424
2. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation **7**(2) (2003) 117–132
3. Köksalan, M., Wallenius, J., Zionts, S.: Multiple Criteria Decision Making: From Early History to the 21st Century. World Scientific (2011)
4. Keeney, R., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs. John Wiley and Sons, New York (1976)
5. Saaty, T.L.: The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. McGraw-Hill (1980)
6. Roy, B.: The Outranking Approach and the Foundations of ELECTRE Methods. Theory and Decision **31** (1991) 49–73
7. Das, I.: A Preference Ordering Among Various Pareto Optimal Alternatives. Structural Optimization **18** (1999) 30–35
8. Mattson, C., Mullur, A., Messac, A.: Smart Pareto Filter: Obtaining a Minimal Representation of Multiobjective Design Space. Engineering Optimization **36**(4) (2004) 721–740
9. Stolze, M., Ströbel, M.: Dealing with Learning in eCommerce Product Navigation and Decision Support: the Teaching Salesman Problem. In: Proc. of the 2nd Interdiscip. World Congress on Mass Customization and Personalization. (2003)
10. Eskelinen, P., Miettinen, K., Klamroth, K., Hakanen, J.: Pareto Navigator for Interactive Nonlinear Multiobjective Optimization. OR Spectrum **32**(1) (2010) 211–227
11. Hadzic, T., O'Sullivan, B.: Critique Graphs for Catalogue Navigation. In: Proceedings of the 2008 ACM conference on Recommender systems. RecSys '08, New York, NY, USA, ACM (2008) 115–122
12. Levav, J., Heitmann, M., Herrmann, A., Iyengar, S.S.: Order in Product Customization Decisions: Evidence from Field Experiments. Journal of Political Economy **118**(2) (2010) 274–299
13. Marinescu, R., Razak, A., Wilson, N.: Multi-Objective Influence Diagrams. In: Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI). (2012) 574–583
14. Dijkstra, E.: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik **1**(1) (1959) 269–271
15. Zadorojniy, A., Masin, M., Greenberg, L., Shir, O.M., Zeidner, L.: Algorithms for Finding Maximum Diversity of Design Variables in Multi-Objective Optimization. Procedia Computer Science **8** (2012) 171–176 Conf. on Sys. Eng. Research.
16. Kahneman, D., Tversky, A.: Prospect Theory: An Analysis of Decision Under Risk. Econometrica **47**(2) (March 1979) 263–291
17. Farina, M., Amato, P.: Fuzzy Optimality and Evolutionary Multiobjective Optimization. In: Evolutionary Multi-Criterion Optimization. Volume 2632 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2003) 58–72
18. Lloyd, S.: Least Squares Quantization in PCM. Information Theory, IEEE Transactions on **28**(2) (mar 1982) 129 – 137
19. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An Analysis of Approximations for Maximizing Submodular Set Functions - I. Mathematical Programming **14**(1) (1978) 265–294