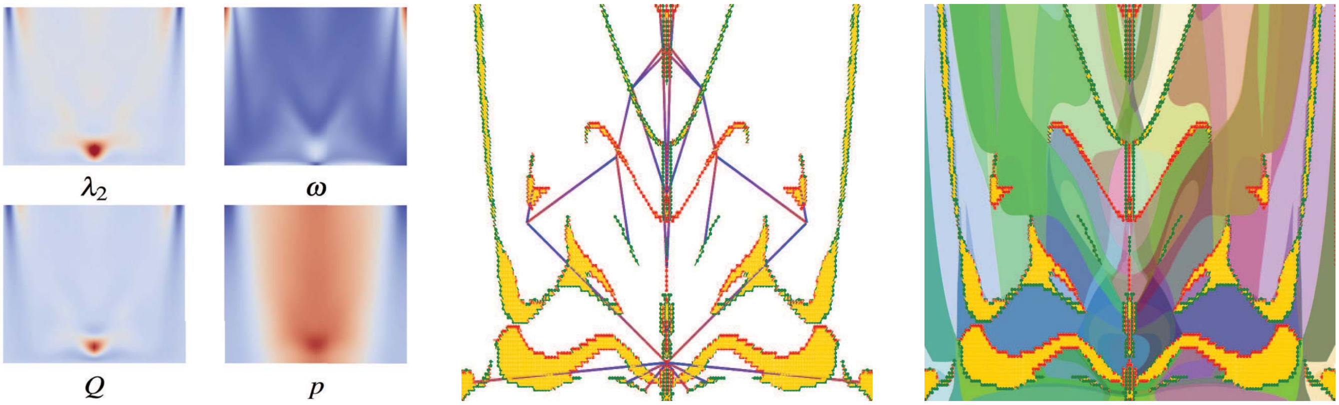
Decomposition and Simplification of Multivariate Data using Pareto Sets

Lars Huettenberger, Student Member, IEEE, Christian Heine, Member, IEEE Computer Society, and Christoph Garth, Member, IEEE



(a) scalar fields (b) reachability graph (c) domain decomposition

Fig. 1. An illustration of the reachability graph over a real-world data set with four scalar fields. (a) depicts color maps of the scalar fields. The reachability graph shown in (b) provides the decomposition of the domain indicated by color in (c).

Abstract—Topological and structural analysis of multivariate data is aimed at improving the understanding and usage of such data through identification of intrinsic features and structural relationships among multiple variables. We present two novel methods for simplifying so-called Pareto sets that describe such structural relationships. Such simplification is a precondition for meaningful visualization of structurally rich or noisy data. As a framework for simplification operations, we introduce a decomposition of the data domain into regions of equivalent structural behavior and the reachability graph that describes global connectivity of Pareto extrema. Simplification is then performed as a sequence of edge collapses in this graph; to determine a suitable sequence of such operations, we describe and utilize a comparison measure that reflects the changes to the data that each operation represents. We demonstrate and evaluate our methods on synthetic and real-world examples.

Index Terms—Multivariate Topology, Pareto Set, Simplification, Decomposition

1 Introduction

Topological visualization techniques capture extremal structures in data and identify intrinsic features in terms of these structures and associated points, to provide an abstract understanding of data sets.

Especially in the context of increasingly complex and large data sets that result from modern scientific computing methods, an abstract depiction that reduces the data to important features are of increasing importance. This abstraction however must include all relevant information while concealing the unimportant aspects of the data such that intuitive understanding and insight are still possible.

Established methods for topological analysis and visualization, such as *contour trees*, identify critical points and regions as well as their connections. Areas of similar behavior are determined to derive a segmentation of the data domain. For univariate data like scalar or vector fields, i.e. where for each position only one scalar value or vector is given, such methods are already introduced and well studied even in high-dimensional domains, cf. e.g. [2] for the scalar case.

A few techniques also exist for the case of multivariate data, where for each position multiple scalar variables are provided. These include *Jacobi sets* [5], *joint contour nets* [1], and *Pareto sets* [10]. Some of these techniques are summarized in Section 2 to provide a context to the work of this paper. The application of these techniques is however, as is typical for topological approaches, difficult when the data to be analyzed is noisy or structurally rich; the resulting topological structures can be extremely complex, and adequate insight cannot be drawn from corresponding visualizations.

A straightforward idea in this setting would be to apply common topological simplification techniques to the individual scalar fields of a multivariate data set before application of multivariate topological analysis. However, such an approach would neglect the functional relationship that might exist between the separate functions, as well as the information given through any multivariate topology. Hence, it appears sensible that structural simplification should be performed on the multivariate structure.

In this context, we propose an approach to achieve structural simplification of Pareto sets over multivariate data. To this purpose, after recapitulating the basic concepts underlying the Pareto set approach in Section 3, this paper provides the following contributions:

- We present a scheme, inspired by previous work [10], to compute a *reachability graph* between Pareto sets of a multivariate function (Section 4).

- *Lars Huettenberger is with TU Kaiserslautern. E-mail: l_huette@cs.uni-kl.de.*
- *Christian Heine is with ETH Zurich. E-mail: cheine@inf.ethz.ch.*
- *Christoph Garth is with TU Kaiserslautern. E-mail: garth@cs.uni-kl.de.*

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014. Date of publication 11 Aug. 2014; date of current version 9 Nov. 2014.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.
Digital Object Identifier 10.1109/TVCG.2014.2346447

- We describe operations on the reachability graph that simplify its overall structure, discuss the implementation of these changes in terms of the underlying function, and propose a measure that determines an order of simplification steps (Section 4).

- To obtain a basis for comparative evaluation, we extend an approach on the simplification of Jacobi sets [17] to multivariate data with an arbitrary number of variables (Section 5).

We illustrate and compare both simplifications on analytic examples and a typical application from the domain of flow visualization in Section 6, aimed at understanding the differences between vortices as described by multiple scalar indicators. The paper concludes with a discussion of the relative merits of each scheme, as well as an outlook towards promising future work.

2 Related Work

There is a very large body of work on topological visualization techniques. We therefore focus specifically on surveying topological approaches to univariate and multivariate data that are related to the approach presented in this paper.

For scalar fields, a common topological representation is the so-called *Reeb graph*, or, in the case of simply connected domains, the *contour tree* [4, 13, 12]. Both track the evolution of connected components of the isosurfaces or level sets, i.e. changes to the preimages of scalar values in the data range. Individual components give rise to edges in the graph, and birth, death, merging, and splitting of components is represented by critical points, i.e. the graph's nodes. Another approach by Chiang et al. [3] constructs contour trees through traversal along monotone ascending and descending paths.

A second framework for the topological analysis of scalar data are Morse-Smale complexes [8, 15]. Here, loosely speaking, the data domain is decomposed based on common behavior with respect to gradient ascent or descent. In this context, common behavior means that two points reach the same maximum and minimum if ascending, respectively descending a path parallel to the scalar gradient field.

For both approaches, simplification techniques have been proposed and studied, with the aim of allowing topological visualization of structurally rich data. In many applications, structural richness may result from noise or sampling inaccuracies, and robust topological visualization should ideally be impervious to such phenomena.

For example, the concept of *persistence* in general captures the existence interval of topological features as scalar values change. For two adjacent nodes in a Reeb graph, this corresponds to scalar value difference between the corresponding critical points [7]. Under the assumption that unimportant features or those resulting from noise have low persistence, i.e. tend to disappear under slight changes to the data, they can then be safely omitted from visualization. Such simplification must be consistent in the sense that it captures the topological structure of hypothetically modified data that is close to the original. Recent work introduced a flooding-based method that "fills" a feature up without introducing new ones [19]. For further literature towards univariate simplification, start for example with Gyulassy et al. [9].

For multivariate (scalar) data, topological visualization is still in its infancy. A general approach taken is to extend the theoretical framework from the univariate case to multiple scalar fields. Edelsbrunner et al. extended Reeb graphs into *Reeb spaces* [6] where the now high-dimensional range space is traversed. Another approach by Edelsbrunner and Harer is *Jacobi sets* [5] which work with the intersection and restriction of level sets and are based on the folding of smooth mappings [21]. A general disadvantage of Jacobi sets is that their applicability is limited to the case of $k \leq d$ functions over a d-dimensional domain. Inspired by mathematical work on optimization problems of Stadler and Flamm [16], Pareto sets as a tool for the structural visualization of multivariate data were introduced by Huettenberger et al. [10]. They are defined for an arbitrary number of functions.

Schneider et al. proposed a comparison through normalized spatial overlap of the *largest contours*, sets of maximal contours containing only one critical point each, from the separate functions [14]. Schneider et al. receive a weighted graph to which they apply graph clustering. This however does not typically result in data simplification or decomposition of the domain.

Joint contour nets [1] make use of contour lines to discretize the domain into slabs, which are regions with constant scalar values. Slab adjacency is captured in a graph which provides a useful visualization. As outlined by Carr and Duke in their paper, joint contour nets can implicitly be simplified through the coarseness of the quantized data on which is used to construct the net. So far, this is the only known stategy for simplifying joint contour nets. However, this approach is global and does not admit the simplification of local structures.

For multiple vector fields, Morse decomposition is proposed by Szymczack [18] in connection with supertransition graphs. Here, a given grid is refined into smaller subgrids, each represented by a node. Any vector inducing movement from one subgrid to another is recorded as an edge in the graph. Again, the simplicity of the graph can be controlled through the refinement of the subgrids. While this can be applied to almost any problem, a coarser grid might not represent important topological information. A simplification through removal or merge of unimportant information would therefore be desirable.

An approach by Tricoche et al. [20] for single vector fields that also works for multivariate data is to identify critical points in each function and then cluster them in the common domain to reduce the structural complexity. The clustering is nonetheless based on geometric distances between the points and not on any topological structure.

However, so far only limited results have been discussed toward the simplification of multivariate structural visualization. A promising method is the simplification of Jacobi sets through a local difference measure, described by Suthambhara and Natarajan [17] for two functions f and g on a 2-manifold embedded in $\mathbb{R}^3$. Their measure is based on $\nabla f(x) \times \nabla g(x)$ to describe the similarity between the two functions. Though, it might be possible to expand this measure to multiple functions, the limitation for Jacobi sets still exists, namely $k \leq d$.

3 Pareto Sets

In the following, we recapitulate briefly the concepts underlying the Pareto set visualization approach. We largely follow Huettenberger et al. [10], who extended the ideas from the work of Stadler and Flamm [16] to the piecewise linear, simplicial setting.

Let $\mathbb{S}$ be a d-dimensional simplical complex $\mathbb{S} \subset \mathbb{R}^d$, $d > 0$, and $f : \mathbb{S} \mapsto \mathbb{R}^k$, $k > 0$, a multivariate function such that $f = (f_1, \ldots, f_k)$ is given at each vertex and extended to $\mathbb{S}$ by barycentric interpolation.

To compare two points x and y in $\mathbb{S}$, a partial ordering is defined in such a way that *x (weakly) dominates y* (denoted $x \succeq y$) if $f_i(x) \geq f_i(y)$ for all $i = 1, \ldots, k$. Furthermore, *x strictly dominates y* (written $x \succ y$) if $x \succeq y$ and there is at least one index i such that $f_i(x) > f_i(y)$. If either $x \succeq y$ or $y \succeq x$ holds, the points x and y are called *comparable*, and *incomparable* otherwise. Intuitively, comparability states that, when changing from x to y, all functions f_i behave similar – they all either increase or all decrease.

This gives rise to a definition of *extremal* points of f using this dominance relation: a point x is called a *Pareto optimum* if there exists an open neighborhood $U(x)$ with $x \in U(x)$ such that all $y \in U(x) \setminus \{x\}$ are incomparable. x is a *Pareto minimum* if all points $y \in U(x) \setminus \{x\}$ are

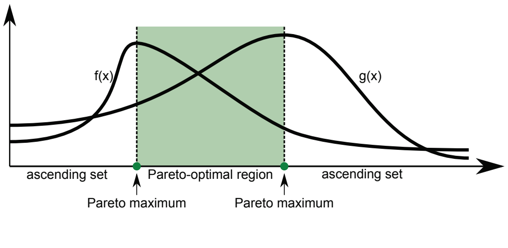


Fig. 2. Pareto set for two 1D functions.

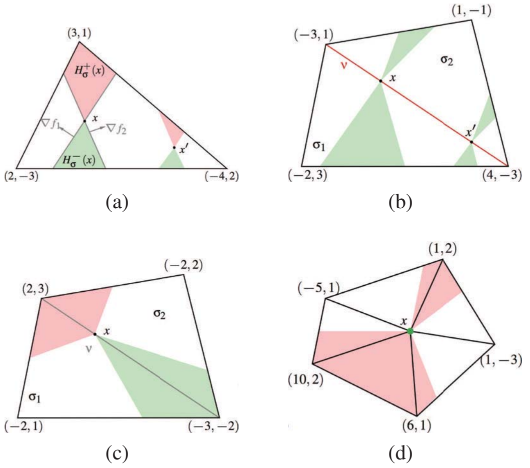


Fig. 3. Illustration of Pareto extrema and regular points in linearly interpolated simplices. (a) Dominated and dominating points w.r.t. x in σ are colored green and red, respectively. (b) For x on an edge, both adjacent simplicies are to be considered. Here, x is Pareto maximal; the same holds for every point x' on the edge v. (c) x has both dominating and dominated points and is thus regular; the same is true for every $x' \in v$. (d) A Pareto-minimal vertex.

either incomparable or weakly dominating and x is a *Pareto maximum* if those comparable points are all weakly dominated by x.

Pareto optima, Pareto minima, and Pareto maxima are collectively called *Pareto extrema* while all other points are considered *regular*. The definitions are illustrated for $k = 2$ functions in Fig. 2. The set of all Pareto extrema is denoted as the *Pareto set* $\mathbb{P}(f)$ of f.

Note that in a flat region, an open neighborhood U for which $f(x) = f(y)$ for all $x, y \in U$ holds, every point is Pareto maximal and also Pareto minimal. This does not provide a conflict with our definitions or the results in the rest of the paper. We, nevertheless, assume in the following all functions to be free of flat regions.

Additionally, Huettenberger et al. defined the *local ascending sets* for $x \in \operatorname{int}\sigma$ as

$$H^+_\sigma(x) \;:=\; \bigcap_{i=1}^{k} \{\, y \in \sigma \;\mid\; f_i(x) \le f_i(y) \,\}.$$

And analogously, the *local descending sets* $H^-_\sigma(x)$.

For the special case that $x \in \sigma$ lies on a face of σ we refer to the illustrations in Fig. 3 and to the previous work [10] for further details. In this case σ needs to be extended to a set of simplices. However, for the sake of simplicity, we refrain from additional notation.

For the remainder of this work, we denote the set of connected components over $\mathbb{P}(f)$ with $\mathbb{E}(f)$.

4 Reachability Graph and Simplification (RGS)

To simplify multivariate data, we apply an idea similar, but not equivalent, to contour trees as constructed by Chiang et al. [3] for univariate scenarios. We extend the ascending and descending sets from simplices to domain-wide connections between Pareto extrema. This results in a global structure, visualized through a graph showing the relations between the elements of $\mathbb{E}(f)$. We discuss the simplification operations for this graph and how those operations are translated to actual modifications to the data. To conclude this simplification idea we define a suitable criterion to find a good sequence of graph operations.

Ascending and Descending Sets

As a reminder, for a (univariate) scalar field $s : \mathbb{S} \mapsto \mathbb{R}$ over the simplicial complex $\mathbb{S}$, prior work usually aimed for a decomposition of the data into regions of common behavior. Each regular point $x \in \mathbb{S}$ can be associated with a set of critical points, local minima, maxima, and saddles of s. The graph-based visualization of the decomposition provides an abstract view on the data and supports its simplification.

We combine this decomposition idea with the *Barrier tree* method introduced by Stadler and Flamm [16] to obtain a global structure of a multivariate field and use it as base for the simplification. We therefore extend Stadler and Flamm's definition, given for graphs, to piecewise-linear data to define *reachability* between Pareto extrema.

Let now $f : \mathbb{S} \mapsto \mathbb{R}^d$ and $x, y \in \mathbb{S}$ be two points with $x \succ y$. We say that x *reaches* y, denoted with $x \rightsquigarrow y$, if there is a continuous path $p : [0,1] \mapsto \mathbb{S}$ such that $p(0) = x$, $p(1) = y$, and $p(i) \succ p(j)$ for $0 \le i < j \le 1$. Note that this definition excludes $x \rightsquigarrow x$. We further define a *global ascending set* for x by

$$H^+(x) \;:=\; \{\, y \in \mathbb{S} \;\mid\; x \rightsquigarrow y \,\}$$

and analogously the *global descending set* $H^-(x)$.

Note that, if x is regular, $H^+(x)$ contains at least one Pareto maximal point, however in general it contains subsets of the Pareto extrema in $\mathbb{E}(f)$. Hence, we use the terminology that x reaches or is reached by a component $Y \in \mathbb{E}(f)$ if and only if $x \rightsquigarrow y$ or $y \rightsquigarrow x$ for some $y \in Y$, respectively. We usually have a proper subset relation since a regular point cannot reach or be reached by a Pareto optimum. To see this, assume for a contradiction that x reaches a Pareto optimum y and, without loss of generality, let $x \succ y$, then a path p exists such that $x = p(0) \succ p(1) = y$ but also $p(1-\varepsilon) \succ p(1)$ for every $1 > \varepsilon > 0$. Hence for every sufficiently small neighborhood around y we find a $z = p(1-\varepsilon)$ such that z is inside the neighborhood and comparable with y. This is a contradiction to y is Pareto optimal.

We extend this nomenclature from points to connected components $S \in \mathbb{E}(f)$ of the Pareto set and define $H^+(S)$ as the union of all $H^+(x)$ for all $x \in S$. If furthermore for two components $S, T \in \mathbb{E}(f)$, $H^+(S) \cap T \neq \emptyset$ is true, we extend the above terminology and write $S \rightsquigarrow T$.

Each point x which is not a Pareto optimum is associated with at least one element of $\mathbb{E}(f)$ which reaches or is reached by x. Pareto optima on the other hand do not have comparable points in their vicinity, therefore no continuous path can start from or end at them. Let

$$\begin{aligned} A(x) \;&:=\; \{\, S \in \mathbb{E}(f) \;\mid\; x \rightsquigarrow S \,\} \qquad \text{and} \\ D(x) \;&:=\; \{\, S \in \mathbb{E}(f) \;\mid\; S \rightsquigarrow x \,\}, \end{aligned}$$

then we define the equivalence relation

$$x \sim y \quad\Longleftrightarrow\quad A(x) = A(y) \;\text{ and }\; D(x) = D(y). \tag{1}$$

Note that all Pareto optima are equivalent, since $A(x) = D(x) = \emptyset$; for Pareto maxima $A(x) = \emptyset$ holds, while for Pareto minima $D(x) = \emptyset$ holds. Of further interest are points for which $A(x) \cap D(x) \neq \emptyset$, i.e. points that indicate paths from some elements in $\mathbb{E}(f)$ to itself, a loop. Specifically, when $A(x) = D(x)$ only contains a single element of $\mathbb{E}(f)$,

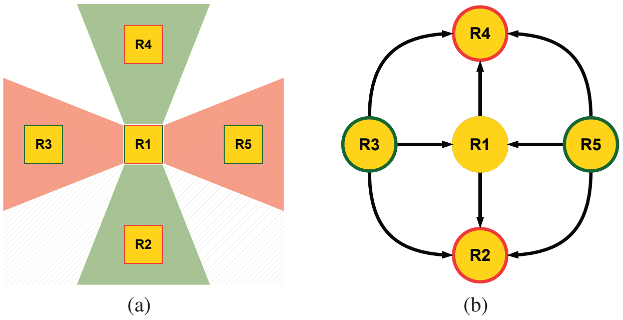


Fig. 4. An example configuration of Pareto extrema illustrating the reachability graph, the ascending and descending sets of the connected component $R1$ (shaded red and green), as well as the descending sets of $R2$ (hatched).

this indicates an isolated region inside the data which is only reached by this element, for example, when a region of regular points is completely enclosed by Pareto extrema.

The connected components of the equivalence classes based on those sets provide a decomposition of the domain $\mathbb{S}$. As illustration, Fig. 8(c) shows a data set with four functions over a 2-dimensional domain with the Pareto optima, maxima and minima, colored in yellow, green, and red, respectively. For each connected component, ascending and descending sets are uniquely color coded and transparently overlaid, illustrating the separate equivalence classes.

As you can see in Fig. 8(c) and 1(c) the decomposition is very confusing. Hence, analogously to contour trees [3], we aim to create an abstract view on the decomposition through a graph-based approach.

Reachability Graph

Given the global ascending and descending sets for each connected component in $\mathbb{P}(f)$ we construct a *reachability graph*. The graph contains a directed edge from $R1$ to $R2$, if and only if $R1$ reaches $R2$. Hence, the edge represents the set of all points $x \in \mathbb{S}$ with $R1 \in D(x)$ and $R2 \in A(x)$.

Fig. 4 sketches a scenario with four functions, each with a univariate maximum in $R2, R4$, a minimum in $R3, R4$, and a saddle in $R1$. The ascending and descending sets of $R1$ are colored in red and green, the descending set for $R2$ is hatched to indicate reachability. The second image shows the reachability graph for this scenario. In later sections, we visually hide edges that are already given transitively, like for example the edge $(R5, R2)$, to clean up the diagram and provide more comprehensible illustrations. Note however that reachability among the connected components is not transitive and we applied this approach in later diagrams only for visual purpose. Our simplification methods consider all edges including those hidden in the visualization.

So far we are able to classify points as Pareto maximal, Pareto minimal, and Pareto optimal while the elements in $\mathbb{E}(f)$ can only be denoted as Pareto extrema since they typically consist of a mix of these.

Also note that loops in the graph are possible. Some are based on numerical errors due to the abstracted view of simplical complexes but not all. For a simple example consider the input data as presented in Fig. 5, a ring of four vertices $\{v_1, v_2, v_3, v_4\}$ connected by edges $(v_1, v_2), (v_2, v_3), (v_3, v_4)$, and (v_4, v_1). The function values of two functions are printed directly on the vertices while all other points along the edges are linearly interpolated. $f(v_1) = (1,1)$, $f(v_2) = (2,2)$, $f(v_3) = (3,3)$, and $f(v_4) = (0,4)$. Note that v_1 and v_4, and also v_2 and v_4 are incomparable such that these vertices and all points included in the two edges between them are Pareto extremal and form a connected component in $\mathbb{P}(f)$. However, v_1 reaches v_3 via a path over v_2 such that the resulting reachability graph contains only one node and a loop to itself.

As an aside, note that an edge (g,h) in the reachability graph implies that the connected component G reaches H, i.e. that $C_{G,H} = \{x \in \mathbb{S} \mid G \in D(x)$ and $H \in A(x)\} \neq \emptyset$ but it is possible that this set consists of multiple connected components in $C_{G,H}$. Hence, while not in the focus of this paper, a multigraph with multiple edges between the nodes might be a more suited visualization. This and other aspects, like for example a good placement of the edges according to $C_{G,H}$ is a possible topic of future work while for the simplification approach in this paper the graph as presented above is sufficient.

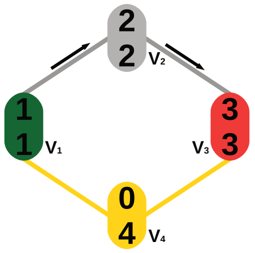


Fig. 5. A simple example bivariate data set with four vertices and four edges. Vertex v_1 is a Pareto minimum, colored green, v_3 is Pareto maximal, colored red, and yellow colored vertex v_4 and adjacent edges are Pareto optimal. v_1, v_3, and v_4 and all points along the mentioned edges build on connected component S, i.e. one element in $\mathbb{E}(f)$. Note that this component reaches itself reaches itself via the ascending path along the edges (v1,v2) and (v2,v3) which results in a loop.

Graph Simplification

Given the reachability graph, there are two types of possible operations that yield simplification of the graph and are therefore suitable to simplify the structure of $\mathbb{E}(f)$. All other operations to a graph structure can be done through a sequence of these two or their reverse actions.

Merge two adjacent nodes into a new one, ”which inherits the old nodes' edges/neighbors.

Purge a node and remove it and every adjacent edge from the graph.

In detail, let $G = (V,E)$ be a directed graph with nodes V and edges $E \subseteq V \times V$. To merge $a, b \in V$ into some $c \notin V$, we create the new graph $G' = (V', E')$ with $V' = \{c\} \cap V \setminus \{a,b\}$ and $E' = E|_{a\leftarrow c, b\leftarrow c} \setminus \{(a,b),(b,a)\}$. The formula $E|_{a\leftarrow c, b\leftarrow c} \setminus \{(a,b),(b,a)\}$ means we replace every appearance of a or b in the set of edges E with c and remove every loop we might create through this renaming.

To purge $a \in V$, we would create the new graph $G = (V', E')$ with $V' = V \setminus \{a\}$ and $E' = \{(x,y) \in E \mid x, y \neq a\}$. However, the removal of any node with more than one adjacent edge can result in a disconnected graph, which implies disconnected data. This is something we have to avoid in our simplification approaches. The only allowed purge is the removal of a leaf which is topologically equivalent to a merge operation of the leaf and its adjacent node. Hence, for our graph simplification approach, we only need to discuss the merge of nodes.

Interpretation

A **merge** operation in the reachability graph can have different interpretations in terms of how the function f is changed such that the new reachability graph is isomorphic to the previous one after the merge operation. For the contour tree in the univariate case there are several ways as, for example, presented by Tierny and Pascucci [19]. To provide a broad overview, we present three approaches to change f which result in a merge in the reachability graph. Note that more efficient methods are possible but not in the scope of this overview.

Given an edge (R_1, R_2) connecting two connected components of $\mathbb{P}(f)$ $R1$ and $R2$, either $R1$ or $R2$ can be removed, or these components are merged through a connecting region of Pareto extrema. Fig. 6 illustrates the changes to f that are equivalent to a merge in the corresponding reachability graph. Assume that $R1$ and $R2$ are elements in $\mathbb{E}(f)$ with the corresponding nodes R_1 and R_2 and an edge (R_1, R_2). The hatched red and green areas are sections of the ascending and descending set of $R1$, the unhatched are those of $R2$. The blue encircled region describes the equivalence class of regular point $H^+(R1) \cap H^-(R2)$. It contains all ascending paths that start from a point in $R1$ and end somewhere in $R2$. To merge the nodes R_1 and R_2, we either connect $R1$ and $R2$ by Pareto extrema or replace either $R1$ or $R2$ with regular points. In either case, we are not allowed to create any more connected components in $\mathbb{E}(f)$ or loops in the reachability graph. Otherwise, the changes to f do not correspond to the merge operation in the reachability graph.

Fig. 6(b) illustrates the removal of $R2$. It shows the ascending and descending sets of $R1$. Note how all paths from or to $R2$ are now reconnected to $R1$ without creating circular paths or new Pareto extrema.

The functions are modified in such a way that all function values of the points in $R2$ are iteratively moved towards $R1$. Fig. 6(c) illustrates the idea behind this *pushing* analogy on a one-dimensional example with two functions (orange and blue) and two Pareto extrema with the same color scheme as in the other images. The function values in the neighborhood of these moved points are interpolated to create the topological behavior as illustrated in Fig. 6(b).

The second option, removal of $R1$ to merge (R_1, R_2) in the reachability graph, is symmetric to this illustration. However, these two options are not able to handle loops, i.e. when $R1 = R2$. Because the removal results in regular points that do not reach Pareto maxima or Pareto minima. This is a contradiction to our above observation.

Hence, for the third option, i.e. connecting $R1$ and $R2$ through additional Pareto extrema, we modify the functions inside the blue circled area in Fig. 6(a). This area contains all ascending paths from any point of $R1$ to some point of $R2$. While a single line of Pareto extrema from some point in $R1$ to another in $R2$ also connects these components, we are forced to change all points in the encircled regions since we otherwise create loops from this connected component $R1/R2$ to itself. This is however not topologically equivalent to the merge of (R_1, R_2). Hence, all points inside the blue circled area are changed into Pareto extrema through modification of f.

While the complete algorithmic explanation to this approach is outside the scope of this paper, consider the above example with four vertices connected in a circle as a simple and compact illustration. The vertices v_1 reach v_3 via a path through v_2. We choose a point along this path, here v_2, and change at least two of its function values, here f_0 and f_1. The first value is decreased below $\min\{f_0(v_1), f_0(v_3)\}$ and the second value is increase above $\max\{f_1(v_1), f_1(v_3)\}$, e.g. $f'(v_2) = (0,4)$. The function values for v_1 and v_3 remain unchanged while all other points along the path are linearly interpolated. Note that all points along the path become Pareto extremal.

We have in summary two important key stones for the application of this idea to our example in Fig. 6(a). First, given the area that contains all ascending paths from one element of $\mathbb{E}(f)$ to the other – the blue circled area in Fig. 6(a) – all function values at the boundary of that area and outside are not changed. Second, at least two functions are modified in such a way that their gradient vectors are inverted to each other for all points inside that area.

Operation Sequence The last step towards a viable simplification method is to determine a suitable order of merge operations such that simplification addresses small-scale regions (e.g. resulting from noise) with preference.

In the previous paragraph we designed the merge of (a,b) through removal of $R2$, removal of $R1$, or through an additional connection between $R1$ and $R2$. In the first option, we modified only a very small part of $H^+(R1) \cap H^-(R2)$ and all points of $R2$, in the second option only some points in $H^+(R1) \cap H^-(R2)$ and all of $R1$, and for the new connection we required to change all points in $H^+(R1) \cap H^-(R2)$.

To estimate the amount of change to f required to transform a regular point x into a Pareto optimum, we measure the dominated and/or dominating points with respect to x restricted to a small neighborhood of x, i.e. $H^+_\sigma(x) \cup H^-_\sigma(x)$ as defined in Section 3, σ being the simplex containing x. If those points become incomparable to x by small changes to f, x becomes Pareto optimal.

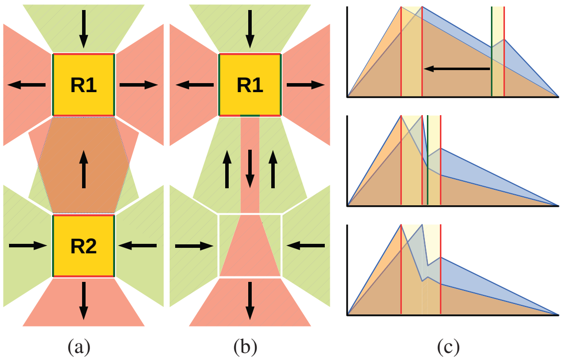


Fig. 6. Illustration of the functional change if an edge corresponding to a connection between $R1$ and $R2$ is merged. (a) and (b) show the result of the changes to the ascending and descending sets of $R1$ while (c) provides implementational ideas through a 1D example.

As a reminder, since f is assumed to be linear in σ, $H^+_\sigma(x)$ is an intersection of k convex half-spaces which are provided through the hyperplanes $\{\ y \in \sigma\ \mid f_i(x) = f_i(y)\ \}$ for $1 \le i \le k$ and σ.

Hence, if x is not Pareto maximal, $H^+_\sigma(x) \setminus \{x\} \neq \emptyset$ is a convex polyhedron and the cone at x has an opening angle which reflects the local quantity of dominated points invariant to the size of σ and the actual scalar values from f. We denote this angle with $\alpha^+_\sigma(x) \in [0,\pi]$. If x lies on a face of some simplex, σ becomes a set of simplices to which x is adjacent and $\alpha^+_\sigma(x)$ is calculated through the sum of opening angle in those simplices. Again, for the sake of simplicity, we refrain from additional notation.

We estimate the amount of alteration to f required to transform all points in $R = H^+(R1) \cap H^-(R2)$ into Pareto extrema through $\int_{y \in R} \alpha^+_{\sigma_y} + \alpha^-_{\sigma_y} dA_y$. However, note that the ascending and descending set for x are point-symmetric at x, therefore $\alpha^+_\sigma(x) = \alpha^-_\sigma(x)$ holds. Hence, $\int_{y \in R} \alpha^+_{\sigma_y} dA_y$ equivalently reflects the alteration to f except for a scalar factor.

To change a Pareto optimum x into a regular point, an optimization problem must be solved for each point, which is only practical for small problem instances. An obvious upper bound with the same approach as α^+_σ is to measure all incomparable points with respect to x restricted to a small neighborhood to x. For Pareto optima, this is obviously the complete neighborhood, therefore an upper bound.

Hence, we augment the reachability graph with the following weights such that we receive a weighted graph $G = (V, E, w)$. For each edge $(S,T) \in E$, corresponding to components $S, T \subseteq \mathbb{E}(f)$, we define $w(s,t) = \int_{y \in R} \alpha^+_{\sigma_y} dA_y + \min\{|S|, |T|\}$ with $R \subseteq H^+(S) \cap H^-(T)$ as described above.

With this definition, $w(R_1, R_2)$ is an estimate of the work needed to remove the component $R2$ in the previous image. Furthermore, as the node R_2 is removed, all incoming and outgoing edges of R_2 become incoming and outgoing edges of R_1. To adjust the weights without a recalculation of the ascending and descending sets, we add $w(R_1, R_2)$ to every relocated edge weight and sum the weights of multiple edges, if those appear. The weights measure how much the functions at a point have to be changed with respect to the halfplanes to turn a regular point into a Pareto extremum. Note that these weights remain conservative estimations since in our piecewise linear setting the change applied to one point also changes all other points of the same simplex.

We apply a greedy algorithm to find a sequence of edge merges until a given threshold is reached. In particular, the algorithm chooses the edge with minimal weight, merges the two adjacent nodes as illustrated before by merging one of the connected components into the other. Then the algorithm recomputes the remaining weights and repeats the procedure until no edge with a weight smaller than the given threshold can be found.

Note that errors in the triangulation are only local, i.e. they usually affect only single simplices and inside those only slightly such that the opening angle $\alpha^+_\sigma(x)$ is rather small. Hence, loops, created through these triangulation errors have a small weight and are chosen first by the greedy algorithm to be removed via a merge of the adjacent node with itself.

Implementation Our current implementation of computing the angle is limited to two dimensional domains, for which we assume a simplicial complex (triangulated grid) with scalar values given at the vertices. We use the same algorithm as [10] to identify the set of Pareto extrema $\mathbb{E}(f)$.

We then compute for each component the ascending and descending set using the following algorithm. We begin at the Pareto minima, respectively Pareto maxima, of the components and propagate a front along the ascending, respectively descending, paths, extending it through individual triangles.

If the front encounters a Pareto extremum and a corresponding edge is not already in the reachability graph, such an edge is added. This is done until every section of the front has stopped at a Pareto extremum. In each triangle, the corresponding section of the global ascending or descending, respectively, set is saved such that in a second step, the weights for each edge can be calculated in each triangle individually

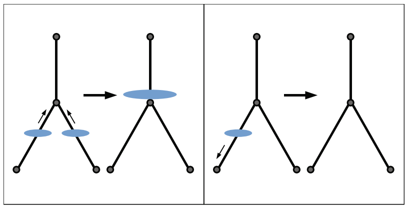


Fig. 7. Illustration of the **Merge** and **Purge** graph operations similar to Suthambhara and Natarajan [17].

and then be accumulated. Note that the sections of the ascending or descending sets, respectively, in the triangles are defined by at most six points, two along each edge of the triangle; in other words, in the worst case, each triangle is visited by the front three times. This makes the calculation of intersections straightforward.

Let $c := |\mathbb{E}(f)|$ be the number of connected components. As the reachability graph G has at most c^2 edges, the calculation of the ascending and descending sets and their intersection can be broken down to m linear problems in each of n triangles such that we obtain a worst-case running time in $O(c^2mn)$. Note that this is a pessimistic estimate, as G has typically fewer than c^2 edges.

5 Simplification Based on the Comparison Measure Contour Tree (CTS)

Simplification based on the reachability graph has a major drawback. If the number of elements $\mathbb{E}(f)$ is relatively large, the simplification algorithm discussed in the previous section is costly, since the ascending and descending sets must be calculated for every component.

The edge weight $w(S,T)$, based on the opening angle $\alpha^+_\sigma(x)$ can however be used to modify a simplification approach presented by Suthambhara and Natarajan [17] for the Jacobi set of two scalar functions such that it works for the Pareto set for arbitrarily many functions.

Suthambhara and Natarajan calculate a local comparison measure that reflects local topological difference between the functions at each point $x \in \mathbb{S}$. This measure yields a scalar field f' from which a contour tree [2] is extracted. Each node $v \in V$ corresponds to connected components in the *level sets*: $\{x \in \mathbb{S} \mid f'(x) = c_v\}$ for some value c_v. Each edge (a,b) corresponds to a connected region in $\{x \in \mathbb{S} \mid c_a \leq f'(x) \leq c_b\}$ such that the level sets corresponding to a and b are subsets. Suthambhara and Natarajan then proved that the hypervolume of this connected region is a conservative estimate of the change in relationship between the two functions if specific graph operations are applied to the contour tree.

We therefore extend the approach of [17] by introducing a similar local comparison measure with the same interpretation as in the work of Suthambhara and Natarajan. Here, we aim to produce a second method for simplification of Pareto sets with reduced computational effort, compare both approaches, and provide a deeper understanding of the connection between Jacobi and Pareto sets.

Comparison Measure

Suthambhara and Natarajan's comparison measure is defined for every point in the domain. It is a scalar invariant value that is also invariant to specific triangulations of the domain, i.e. the size of the simplices. It reflects local topological difference between the underlying functions.

In the context of the Pareto set approach, strong difference between the functions results in incomparable points, reducing the number of comparable ones and produces Pareto optima.

Hence, we suggest to replace the measure from [17] with $2 \cdot (\pi - \alpha^+_\sigma(x))$, the function from the previous section which reflects an upper bound for the difference between the separate functions in f.

As discussed before, α^+_σ is not defined for every point in $\mathbb{S}$ since σ is ambiguous for some x, especially vertices. Following [17], we average α at the vertices based on the values in the adjacent simplices. Through linear interpolation using these average values at the vertices, we receive a continuous, piecewise linear scalar field κ_f for $\mathbb{S}$.

Note that for the Pareto optimal points for which no cone exists, κ_f equals π, which directly implies that Pareto optima are maximal in κ_f. We can easily enforce that every element in $\mathbb{E}(f)$ contains at least one Pareto optimum by adding new functions.

Assume that an element in $\mathbb{E}(f)$ does not contain a Pareto optimum, therefore only contains Pareto maxima and/or Pareto minima. For such a Pareto maximum x with $H^-_\sigma(x) \neq \{x\}$, we add d new functions to f. Every new function is equivalent to f_0 except for a sufficiently small neighborhood $U(x) \cap H^-_\sigma(x)$ around x. Inside that region, the positions with the function values $f_0(x)$ in the new functions are moved by a sufficiently small ε such that these positions form a small d-simplex. The other values for the new functions in $U(x) \cap H^-_\sigma(x)$ are set through linear interpolation. Note that for every point inside that simplex and for every direction from that point we have one of the new function for which the function value increases and another new functions for which it decreases. Hence every point inside that simplex is surrounded by incomparable points and is therefore Pareto extremal.

However, since x is in the connected component and the new Pareto extrema are all adjacent to x, the new functions do not increase $|\mathbb{E}(f)|$ and at most enlarge the component including x. Nor do they change κ_f drastically and only inside the small neighborhood $U(x)$.

Hence, we can assume that every element of $\mathbb{E}(f)$ contains at least one maximum with respect to κ_f. Furthermore, since all Pareto optima have π as value of κ_f, each element in $\mathbb{E}(f)$ can be associated with a unique connected component in the level set $\{x \in \mathbb{S} \mid \kappa_f(x) = \pi\}$.

Note that since the new functions are almost equal to f_0 they can be neglected for the calculation of κ_f except for small neighborhoods around some Pareto optima where those function values actually differ from f_0. Thus almost no extra computational effort is created through those new functions.

Contour Tree

Given κ_f, we follow Suthambhara and Natarajan to construct and simplify a contour tree. For the purpose of illustration, Fig. 7 presents the graph operations provided in [17]. The larger blue nodes are associated with connected components in the Jacobi set. Note the similarity of the Merge and Purge operations to the graph simplification applied in Section 4.

Also note that, under the above assumption, all connected components are associated with leafs in the contour tree such that a merge tree is sufficient for simplification purposes and simpler to compute. Using this insight, as well as the fact that all nodes of interest, i.e. those associated elements of $\mathbb{E}(f)$, are maxima in the merge tree, we employ a simplified version of the greedy algorithm used in [17]. This algorithm sequentially chooses the graph operation with minimal cost, applies this operation to recalculate the weights and repeats this until the minimal cost is above a given threshold or until a number of operations is reached. However, we adopt the basic idea to merge edges with smallest weight first, as is also done in the previous section. Here, for each edge the integral of κ_f over the area swept by the level sets corresponding to an edge in the merge tree are taken as edge weights.

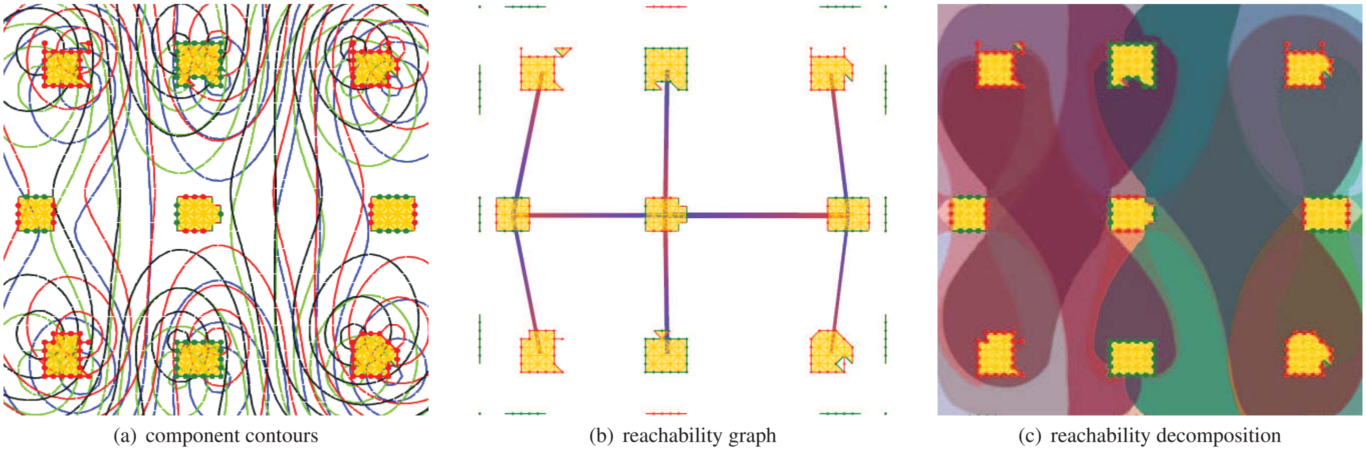
6 Examples

We applied the two strategies, reachability graph simplification (RGS, cf. Section 5) and simplification based on the contour tree of κ_f (CTS, Section 4) to two data sets to provide a proof-of-concept and an initial comparison between these methods. Computation times for the results shown in the following are in the range of seconds to minutes depending on the size of $\mathbb{E}(f)$, as discussed in Section 4.

Synthetic Data

The first data set is a synthetic example. We select a 4×2 set of positions $A_i = \{a_{i,j} \in [0,1]^d | 0 \leq j \leq 4\}$ and $B_i = \{b_{i,j} \in [0,1]^d | 0 \leq j \leq 4\}$ for $1 \leq i \leq 4$ with the restriction that $0 < |a_{s,j} - a_{t,j}| < 0.1$ and $0 < |b_{s,j} - b_{t,j}| < 0.1$ for $1 \leq s,t,j \leq 4$. The functions f_i are defined as

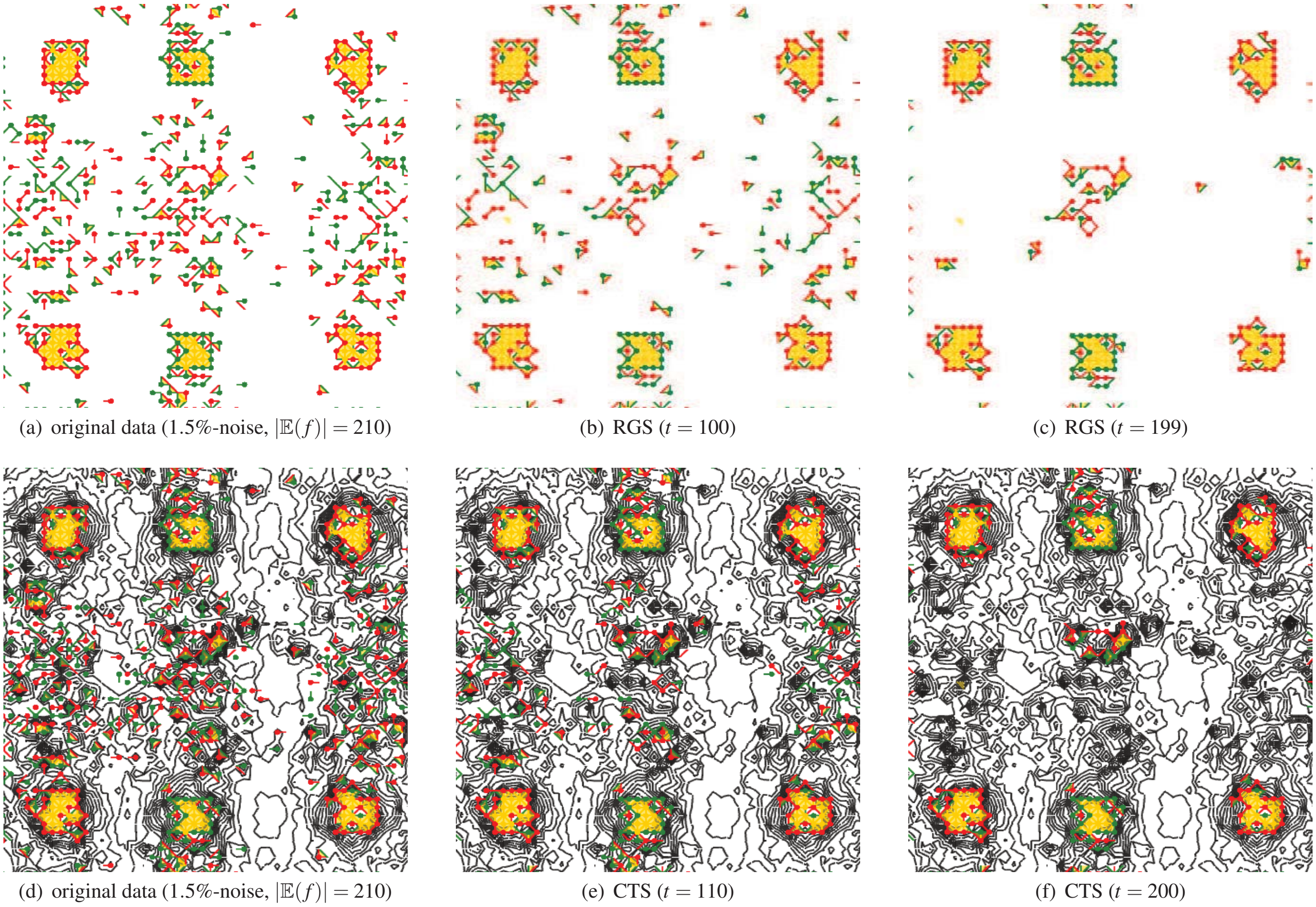
$$f_i(x) := \sum_{j=0}^{4} e^{-|a_{i,j}-x| \cdot 2} - \sum_{j=0}^{4} e^{-|b_{i,j}-x| \cdot 2},$$


(a) component contours

(b) reachability graph

(c) reachability decomposition

Fig. 8. An illustration of the reachability graph over a synthetic data set with four scalar fields. (a) depicts the contours of each of the four scalar component functions. The reachability graph given in (b) provides a decomposition of the domain shown in (c).


(a) original data (1.5%-noise, $|\mathbb{E}(f)| = 210$)

(b) RGS ($t = 100$)

(c) RGS ($t = 199$)

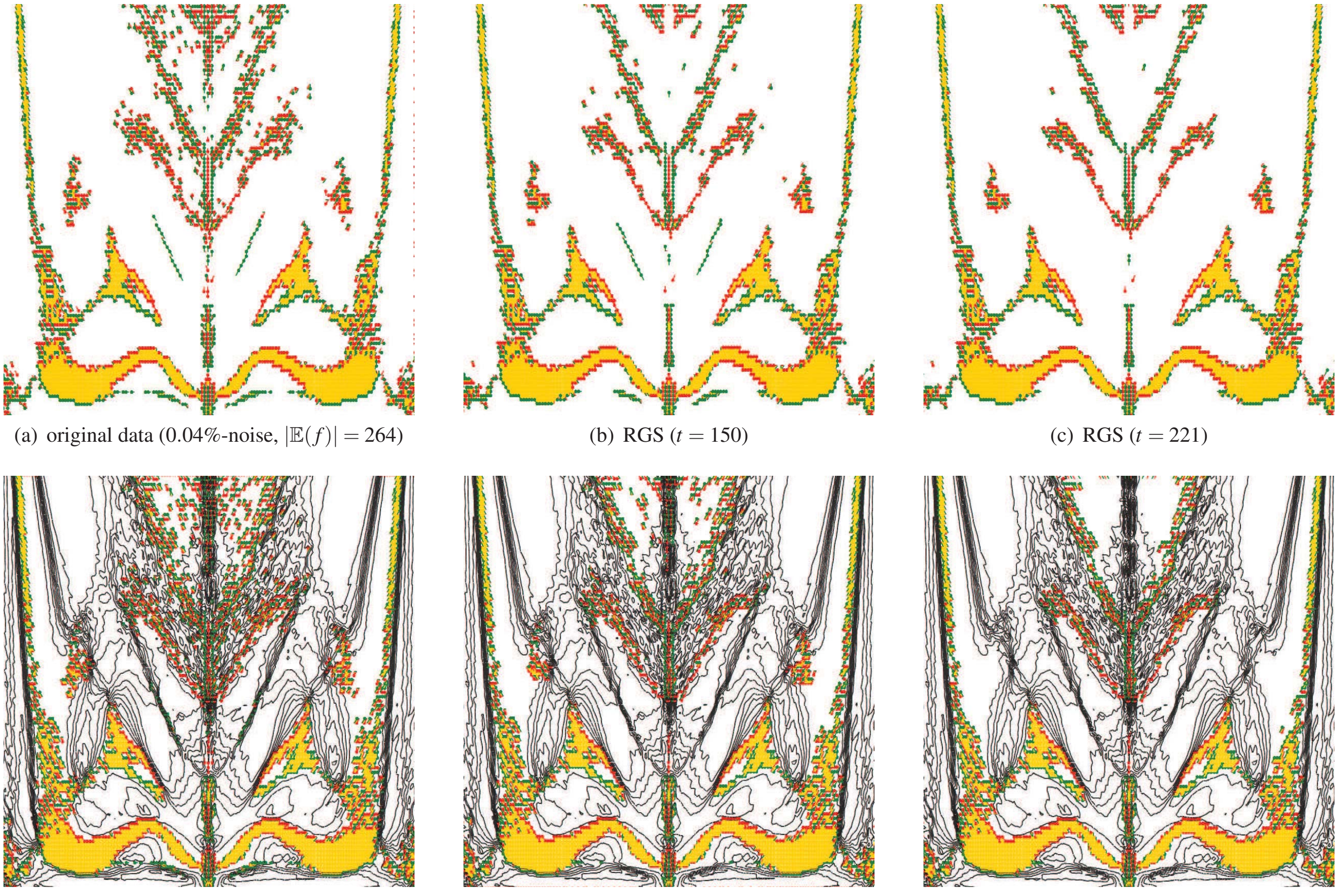
(d) original data (1.5%-noise, $|\mathbb{E}(f)| = 210$)

(e) CTS ($t = 110$)

(f) CTS ($t = 200$)

Fig. 9. A comparison of the two Pareto set simplification techniques over synthetic data with 1.5% random noise. The top images (a) and (c) shows reachability-graph (RGS) simplification for two increasing thresholds. The bottom row illustrates the comparison measure contour tree simplification (CTS) for two increasing thresholds (images (e) and (f)). The first images (a) and (d) present the unsimplified data for the methods.

and combined to yield $f(x) = (f_1(x), \ldots, f_4(x))$. We calculate f at the vertices of a regularly triangulated grid as input to our algorithms. Fig. 8 provides an overview of this example. Each function has a maximum in each of the four connected components of $\mathbb{P}(f)$ in the corners, a minimum in each of the connected components in the top and bottom middle, and a saddle in each of the three connected components in the middle row. Fig. 8(a) shows contour lines for the f_i, color coded in blue, red, green and black, to illustrate their univariate topologies.



(a) original data (0.04%-noise, $|\mathbb{E}(f)| = 264$)

(b) RGS ($t = 150$)

(c) RGS ($t = 221$)

(d) original data (0.04%-noise, $|\mathbb{E}(f)| = 264$)

(e) CTS ($t = 159$)

(f) CTS ($t = 226$)

Fig. 10. A comparison of the two Pareto set simplification techniques over synthetic data with 0.04% random noise. The top images (b) and (c) shows reachability-graph simplification (RGS) for two increasing thresholds. The bottom row illustrates the comparison measure contour tree simplification (CTS) for two increasing thresholds (images (e) and (f)). The unsimplified data is presented in the images (a), respectively (d).

Fig. 8(b) shows the same data and its reachability graph, together with the Pareto sets of f consisting of minima (green), maxima (red), and optima (yellow). Nodes are placed in the center of gravity of the corresponding component. Note that loops are not shown, and edges that are already given transitively are removed to simplify visual understanding of this example. Fig. 8(c) presents the reachability decomposition based on color coding ascending and descending sets. This depiction also contains the reachability graph from the previous image, but is more difficult to comprehend.

This synthetic data has a relatively simple structure. To obtain increased structural richness as a testbed for simplification, we added random noise. At each vertex and for each component, we changed the function values by a uniform random value from the interval $[-p \cdot \delta, p \cdot \delta]$, where δ denotes the range of the corresponding component function and $p << 1$.

Fig. 9(a) and 9(d) show the same synthetic data, both with additional 1.5% noise. Due to the relative flatness of f, even such a low noise level already introduces significant changes to the structure and breaks apart the three connected components in the middle into several smaller components. The six connected components in the top and bottom row remain roughly intact however. Ideally, the simplification procedure would retain these components and remove all other newly introduced components.

Fig. 9(b) and 9(c), and 9(e) and 9(f) show the result of the two approaches after a number of simplification steps. In all four images, the value t indicates the number of removed, respectively merged, elements of $\mathbb{E}(f)$. In both approaches, we remove connected components that correspond to purged or merged nodes in the reachability graph or the contour tree of κ_f, respectively. Hence, neither the form of the components nor the contour lines change among Fig. 9(a) to 9(c) and 9(d) to 9(f), respectively.

Note that both methods are able to remove most of the spurious noise and keep the six connected components in the top and bottom row we identified in the noise-free data in Fi. 8. CTS however removed almost all of the noise-based components too, among them those which we could associate with the three connected compoents in the middle row from Fig. 8.

For the RGS approach, we observe a range of simplification steps for which the desired elements of $\mathbb{E}(f)$ remain. Still, an exact reconstruction of the original synthetic data, i.e. topological equality in the reachability graph, could not be achieved.

An explanation for the results is two-folded. First, the induced noise results only in small, local maxima in κ_f. Hence, the Pareto extrema based on those noise correspond to only low-weighted nodes which are all removed very early in the CTS method. Since we always remove leafs of the contour tree in this method the remaining nodes do not gain weight very fast such that in later simplification steps also Pareto sets which we would consider important are removed. For the RGS approach, on the other hand, the ascending and descending sets of merged Pareto extrema are changed and thereby also all edge weights adjacent to the corresponding nodes in the reachability graph. Hence, in a cluster of noise-based Pareto extrema, like those we see in the middle row of Fig. 8, it is more likely that Pareto extrema remain in the data under RGS than under CTS.

Can Data We have the Can data set as a real-world example which results from a CFD simulation of flow in a fluid-filled cylinder with a rotating lid. Prior work with a similar data set was done by Huettenberger et al. [10]. The data set is a simple example of flow simulation with the aim to identify vortices using several vortex criteria, more specifically, Q-criterion, λ_2, vorticity, and pressure. See, for example, Jeong and Hussain [11] for further details on those criteria. Such simulations are often found in aircraft or car design where the localization of air turbulences and vortices are essential for a fuel efficient design.

All criteria indicate vortices via extremality, however disagree on the exact location of those vortices. Therefore, the Pareto set provides extremal regions, elements of $\mathbb{E}(f)$, on which all criteria agree. The criteria mentioned above are calculated from the simulation data and together form the multivariate scalar function f. The simplicial complex that serves as input to both simplification schemes is generated as a two-dimensional slice parallel to and containing the cylinder's central axis. As the flow is rotationally symmetric, this is sufficient to identify meaningful structures. The images in Fig. 1(a) illustrate the component scalar fields on the slice. The goal in this context is to visualize common topological behavior, agreement between the fields, to identify vortices from multiple criteria simultaneously (cf. [10]).

As for the synthetic data, Fig. 1 illustrates the Pareto sets of f for which Fig. 1(b) and 1(c) depict the reachability graph and decomposition. Note how the number of reachability equivalence classes increases dramatically as the number of extrema regions $|\mathbb{E}(f)|$ increases. Pareto maximal and minimal components, as classified by the reachability graph, are of main interest since they indicate regions in which all considered criteria suggest vortices to be located. All criteria agree that towards those components their value increase, respectively decrease, and thereby agree on extremality of this region.

Edges in the graph furthermore provide a domain expert with information how those regions are connected, even interact with each other. The graph aids the experts to understand global structure and directs them to locations, the Pareto extrema, where a detailed look is necessary. Specifically, an edge represents a set of paths between two connected components on which all criteria agree with respect to the ascending and descending direction, respectively.

It can be observed that this data set is relatively clean and contains little structure. Again, we add noise to gauge the effect of simplification. Fig. 10(a) and 10(d) present the Pareto set for modified data with random 0.04-noise. Note how such small degree of noise already produces a large amount of separate extremal regions.

Fig. 10(b) and 10(c) present the results of RGS for two different numbers of simplification steps, whereas Fig. 10(e) and (f) show the outcome of the CTS method. The value t indicates the number of removed, respectively merge, connected components of $\mathbb{P}(f)$.

As with the previous, synthetic data, we notice that the contour tree approach removes more components than necessary, while in the reachability graph approach noise-based extremal regions remain.

7 Conclusion & Future Work

We presented two techniques for simplifying multivariate scalar data based on structures gained through the calculation of the Pareto set.

The first technique is based on the reachability graph – a weighted graph whose nodes represent connected components of Pareto extrema. Connectivity is determined by the existence of paths along which function values are strictly increasing or decreasing. Edge weights estimate the local stability as measured by the opening angle of the space of incomparable points within a region. For two fields, this measure is a normalized version of Suthambhara and Natarajan [17], originally developed for Jacobi set simplification. However, our measure extends naturally to more than two fields and is furthermore resilient to scaling input field values, e.g. resulting from a change in units. The simplification itself greedily merges or purges node pairs or nodes, respectively, in the order of edge weights until a given threshold is reached. The concept of reachability also lends itself to a decomposition of the domain by defining an equivalence relation among regular points from which the same sets of Pareto extrema can be reached.

The second simplification technique follows more closely along the lines of Suthambhara and Natarajan [17] in that it computes the contour tree of the stability field computed from the original multivariate data and performs topological simplification on that contour tree. Again we replaced the stability measure of Suthambhara and Natarajan by our opening angle measure to support more than two fields.

The results of a case-study-based qualitative comparison of both methods revealed that both techniques managed to remove most of the artificial noise that was introduced during our experiments, but the contour-tree-based method removes more connected components than necessary. The different results may be explained by observing that the CTS essentially turns the multivariate problem into a univariate problem using only local information at each point, whereas the reachability graph considers the functions jointly within regions und thus has more information at its disposal. Compared to the contour tree technique, the reachability graph supports loops and can thereby handle errors based on inconvenient triangulation. However, loops remain difficult to interpret, because they never arise in the univariate setting. In addition, the contour tree technique was slightly faster, but, it does not lend itself to a decomposition of the domain. However since these observation are based on few cases we wish to strengthen our findings by studying further examples in the future.

Both Suthabhara/Natarajan's and our method currently lack precise rules for the change to the input functions such that the simplified combinatorial structure (contour tree and reachability graph, respectively) arises. Suthabhara and Natarajan point out that such a change may be complex and computationally expensive since it must satisfy numerous constraints on the local stability measure. We outlined the feasibility and three models of how to change the functions for our methods, but the study of which model is generally feasible and optimal (e.g. requires minimal changes to the input functions) may depend on the actual application. We thus leave this study for future work.

Although we presented our work only for two-dimensional domains, the computation and simplification of the combinatorial structures extend naturally to higher dimensions. While the combinatorial stucture can always be shown via graph drawing, for more than two dimensions the presentation of the domain decomposition and its relation to the combinatiorial structure become challenging. Future work will therefore investigate display strategies for three-dimensional data sets, as these are of primary interest in scienfic and engineering applications. For example, we might base multivariate transfer function design for volume rendering on our combintorial structure in analogy to Zhou and Takatsuka [22].

While in this paper our aim was to demonstrate the feasibility of Pareto set simplification, a more in-depth evaluation is desirable. Therefore, future work is underway to assess the benefits and drawbacks of simplified Pareto sets over existing multifield visualization methods. Prior work already hinted at the relation between Pareto and Jacobi sets [10], in this paper we found that this similarity extends to topological simplification. We wish to further study those relations; it has the immediate benefit of providing opportunities to transfer research results for both structures and may shed more light on general multifield analysis in the future. Similar benefits may arise through a comparison with Morse-Smale complexes [8] and joint contour nets [1], especially how the quantization of the fields' values interacts with the reachability relation. Because ascending and descending paths are quite unlike steepest ascent and descent the relation to Morse-Smale complexes is not imminent. But Szymczak's work [18] on piecewise constant vector fields bears some algorithmic similarity to the detection of Pareto sets and may provide a link between the two. This similarity may prove fruitful for further reasearch.

Acknowledgments

This work was funded in part by the DFG International Research Training Group 1131 “Visualization of Large and Unstructured Data Sets” and the Marie Curie Actions within the EU FP7 Programme under grant #304099. We are indebted to our colleagues at TU Kaiserslautern for discussion and feedback, and would also like to express our gratitude for constructive comments from reviewers.

REFERENCES

[1] H. Carr and D. Duke. Joint contour nets. *IEEE Transactions on Visualization and Computer Graphics*, 99:1, 2013.

[2] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75 – 94, 2003.

[3] Y.-J. Chiang, T. Lenz, X. Lu, and G. Rote. Simple and optimal output-sensitive construction of contour trees using monotone paths. *Computational Geometry*, 30(2):165–195, 2005.

[4] H. Doraiswamy and V. Natarajan. Computing reeb graphs as a union of contour trees. *IEEE Transactions on Visualization and Computer Graphics*, 19(2):249–262, 2013.

[5] H. Edelsbrunner and J. Harer. Jacobi sets of multiple Morse functions. In *Foundations of Computational Mathematics*, volume 312, pages 37–57. Cambridge University Press, 2004.

[6] H. Edelsbrunner, J. Harer, and A. K. Patel. Reeb spaces of piecewise linear mappings. In *Symposium on Computational Geometry*, pages 242–250, 2008.

[7] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, FOCS '00, pages 454–, Washington, DC, USA, 2000. IEEE Computer Society.

[8] A. Gyulassy, P.-T. Bremer, B. Hamann, and V. Pascucci. A practical approach to morse-smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1619–1626, 2008.

[9] A. Gyulassy, V. Natarajan, V. Pascucci, P. T. Bremer, and B. Hamann. Topology-based simplification for feature extraction from 3d scalar fields. In *Proceedings of IEEE Conference on Visualization*, 2005.

[10] L. Huettenberger, C. Heine, H. Carr, G. Scheuermann, and C. Garth. Towards Multifield Scalar Topology Based on Pareto Optimality. *Computer Graphics Forum*, 32(3):341–350, 2013.

[11] J. Jeong and F. Hussain. On the Identification of a Vortex. *Journal of Fluid Mechanics*, 285:69 – 94, 1995.

[12] V. Pascucci and K. Cole-McLaughlin. Efficient computation of the topology of level sets. In *Visualization, 2002. VIS 2002. IEEE*, pages 187–194, 2002.

[13] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas. Robust on-line computation of reeb graphs: Simplicity and speed. *ACM Trans. Graph.*, 26(3), July 2007.

[14] D. Schneider, C. Heine, H. Carr, and G. Scheuermann. Interactive comparison of multifield scalar data based on largest contours. *Computer-Aided Geometric Design*, (in press), 2012.

[15] N. Shivashankar, M. Senthilnathan, and V. Natarajan. Parallel computation of 2d morse-smale complexes. *IEEE Transactions on Visualization and Computer Graphics*, 18(10):1757–1770, 2012.

[16] P. F. Stadler and C. Flamm. Barrier Trees on Poset-Valued Landscapes. *Genetic Programming and Evolvable Machines*, 4:7–20, 2003.

[17] N. Suthambhara and V. Natarajan. Simplification of jacobi sets. In V. Pascucci, X. Tricoche, H. Hagen, and J. Tierny, editors, *Topological Methods in Data Analysis and Visualization*, Mathematics and Visualization, pages 91–102. Springer Berlin Heidelberg, 2011.

[18] A. Szymczak. Stable morse decompositions for piecewise constant vector fields on surfaces. *Comput. Graph. Forum*, 30(3):851–860, 2011.

[19] J. Tierny and V. Pascucci. Generalized topological simplification of scalar fields on surfaces. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2005–2013, 2012.

[20] X. Tricoche, G. Scheuermann, and H. Hagen. A topology simplification method for 2d vector fields. In *Proceedings of the Conference on Visualization '00*, VIS '00, pages 359–366, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.

[21] H. Whitney. On Singularities of Mappings of Euclidean Spaces. I. Mappings of the Plane into the Plane. *The Annals of Mathematics*, 62(3):374–410, 1955.

[22] J. Zhou and M. Takatsuka. Automatic transfer function generation using contour tree controlled residue flow model and color harmonics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1481–1488, 2009.