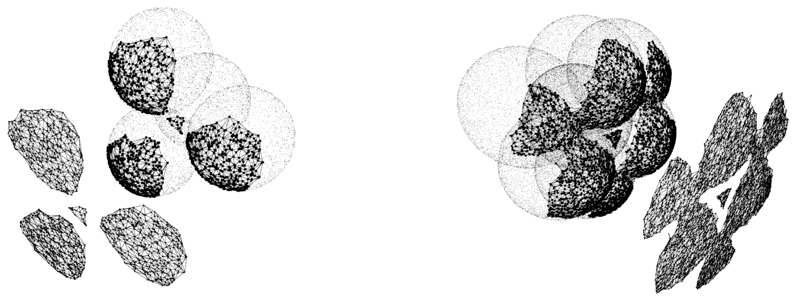
VIPO: Simulation Driven Product and Process Optimization (2021)
https://www.vipo-net.de/

Generating Tessellations of n-dimensional Pareto Frontiers for Visualization and Configuration

Markus Pawellek   Ralf Hohmann   Matthias Schwuchow

ORISA Software GmbH



Abstract

Multiobjective optimization plays a key role in computer-aided design, manufacturing, and engineering. Typical problems need to find a trade-off between many conflicting objectives. Because no single solution exists, human intervention is necessary to provide a preference and choose between different Pareto-optimal solutions. In real-world applications, this possibility should be given by a configuration interface. But in general, solving a multiobjective problem only involves estimating a set of unsorted and nondominated points near the actual Pareto frontier. Hence, a configuration interface for humans, used to choose one of those optimal solutions, is not able to proper visualize solutions or to offer an intuitive user interface. This paper explores the possibility of tessellating n-dimensional Pareto frontiers based on projection, Delaunay triangulation, and a statistical heuristic to generate a hypersurface and make visualization and configuration feasible. The developed algorithm has been tested against standard, constructed, and real test problems.

1 Introduction

In many different areas of science, like engineering, economics, and logistics, multiobjective optimization techniques are used to find trade-offs between multiple conflicting objectives, like maximizing the performance of a computer processor while minimizing its energy consumption. Typically, there is no unique best solution for problems stated in reality.

Multiobjective optimization is able to find multiple so-called nondominated or Pareto-optimal solutions. The set of all possible nondominated solutions is known as the Pareto frontier. There is no mathematical reason to choose only one specific point of such a set. As a consequence, in reality, human intervention is used to subjectively preference one of the found solutions to be able to get parameters which can be used for production.

Selecting Pareto-optimal points for a specific problem is not an easy task and requires further knowledge of the environment, circumstances, and causalities of the problem. Therefore a lot of tools have been developed to support designers and engineers in choosing such a single solution. But in general, multiobjective optimization methods only return an unsorted set of points near the Pareto frontier, making a sophisticated visualization and configuration tricky and unnatural.

Our idea involves the actual construction of the surface of the Pareto frontier. Using an appropriate curved surface allows high-quality rendering techniques, such as rasterization, for visualization to be used. Furthermore, approximations between similar solutions are suddenly allowed and easy to carry out. Configuration would become simpler.

The main problem is that typical Pareto frontiers provide strong discontinuities and are not convex. Thus, the generation of Pareto surfaces is not trivial. This is especially true for more than two objective values. Whereas the Pareto frontier for two objectives in general breaks down into a discrete set of one-dimensional curves in the plane where every point already has unique neighbors and the problem reduces to decide which of those are actually connected, three objective values may already result in a complicated set of one- and two-dimensional curved manifolds embedded in three-dimensional space.

For $n \in \mathbb{N}$ with $n > 2$ objective values, our method uses the $n-1$-dimensional Delaunay tessellation by projecting all points onto a hyperplane to provide a triangulation which allows every point in the set to have two or more neighbors. However, the Delaunay triangulation always constructs the convex hull of all given points and will nearly every time result in too many connections.

At this point, every number of objectives can be handled the same way by using a statistical heuristic and the assumption that the generated solutions are uniformly distributed on the Pareto surface. By estimating the mean and standard deviation of the distribution of distances between two neighboring points, we are able to use a hypothesis test to remove connections when their distance is too large and may exhibit discontinuities.

Applying this algorithm to the standard test problems of multiobjective optimization, as seen in figure 1, given in the literature and further constructed test problems, we will empirically confirm its correctness and robustness.

2 Background

2.1 Multiobjective Optimization

The general goal of mathematical optimization is to quantify an optimal set of input parameters, which we call a configuration, that minimizes or maximizes a given cost function. For single-objective cost functions whose range is the space of real numbers, the natural total ordering of the real axis is intuitively used to decide which configuration leads to better results by comparing their outcome of the cost function.

In multi-objective optimization the range of a cost function is multi-dimensional. Because there is no natural ordering of multi-dimensional real vector spaces, we have to clearly define what it means for a configuration to be more optimal than another one.

> **DEFINITION:** Domination
>
> *For $n \in \mathbb{N}$, let $x, y \in \mathbb{R}^n$. Then we say x dominates y or $x \prec y$, if the following is true.*
>
> *(1)* $\forall k \in \mathbb{N}, k \leq m: \quad x_k \leq y_k$
>
> *(2)* $\exists k \in \mathbb{N}, k \leq m: \quad x_k < y_k$

This induces only a partial ordering for multi-dimensional spaces but seems to suffice for the typical optimization problems. Based on this definition, we can go even further and are able to define optimal values for multi-dimensional subsets.

> **DEFINITION:** Pareto Optimality
>
> *Let $n \in \mathbb{N}$ and $X \subset \mathbb{R}^n$. A point $x \in X$ is said to be non-dominated or (Pareto) optimal, if there is no other point $y \in X$ which is dominating it. We denote $\mathrm{Pareto}(X)$ as the set of all (Pareto) optimal solutions in X.*

With the concept of Pareto optimality, multi-objective optimization problems can be defined in the typical sense of single-objective optimization.

> **DEFINITION:** Multi-Objective Optimization Problem
>
> *Let $n \in \mathbb{N}$ be the number of configuration variables, $m \in \mathbb{N}$ with $m \geq 2$ be the number of objective values, and $X \subset \mathbb{R}^n$ the configuration space or the set of feasible solutions. Furthermore, let $f \colon X \to \mathbb{R}^m$ be the cost function. Then we call the following the*

multi-objective optimization problem $\mathcal{P}$.

$$\mathcal{P} := \underset{x \in X}{\text{minimize}}\, f(x)$$

The solution $\mathcal{S}(\mathcal{P})$ to this problem is then given by the following subset of the graph of f based on the configurations which map to the Pareto frontier.

$$\mathcal{S}(\mathcal{P}) := \left\{ (x, f(x)) \;\middle|\; \begin{array}{c} x \in X, \\ f(x) \in \operatorname{Pareto}(\operatorname{im} f) \end{array} \right\}$$

Typical multi-objective optimization problems which are also used as benchmarks for numerical solvers are given in table 1 and a visualization of their respective Pareto frontiers is given in figure 1.

2.2 Delaunay Triangulation

3 Problem

3.1 Input

A discrete set of points $\mathcal{S}$ in $\mathbb{R}^n$ with $n \in \mathbb{N}$ is given. For our purposes, $\mathcal{S}$ will be the result of a multi-objective optimization in n dimensions. Hence, most of the given points will form the Pareto frontier.

3.2 Output

Based on the given point set $\mathcal{S}$, a triangulation or tessellation shall be computed which enables the user to securely interpolate between Pareto points, to visualize the resulting surfaces, and to localize new points fast for configuration. The typical problem that has to be taken care of is that Pareto frontiers are not continuous in general. Furthermore, multi-objective optimization tends to use many dimensions which are difficult to visualize.

3.3 Solution

A simple solution would be to ignore any kind of surface construction and use the sticky point algorithm. But looking at visualization and configuration this is suboptimal.

1. Estimate a sufficient amount of points near the actual Pareto frontier by using an appropriate multiobjective problem solver, like NSGA-II (Deb et al. 2002), which uses the crowding distance metric or an alternative similar method to provide approximately uniformly distributed points on the Pareto frontier.
2. Project the estimated points from m-dimensional objective vector space to the $m-1$-dimensional hyperplane with respective normal in direction $\sum_{i=1}^{m} e_i$. Because every estimated point is not dominated by other points, the transformation is invertible and makes Delaunay tessellation of the curved Pareto surface possible.
3. Construct the $m-1$-dimensional Delaunay triangulation/tessellation of the projected points by using a state-of-the-art algorithm and apply the same tessellation to the original points in m-dimensional space. Inverting the projection is not needed. It is recommended to use a facet/edge-based data structure, like the quad-edge structure from Guibas and Stolfi (1985) in two dimensions, to store the tessellation scheme. Such a structure has to be general enough to describe arbitrary connections between points even if there is no triangle. Such a structure is easy to store and transmit in binary- and ASCII-based file formats. Following post computation will also become easier and triangles can be constructed by iterating through all connections.
4. Statistically analyze the distances of points to their neighbors and the estimated gradient at their position. This can be done in a global sense for all points or for smaller batches of points.
5. Use a heuristic to combine those two values. Construct an approximating probability distribution for such a heuristic and learn it.
6. Make hypothesis tests based on the learned probability distribution and remove point connections from the tessellation that do not fulfill the test. They could exhibit discontinuities.
7. Visualize the results by projecting the data to two- and three-dimensional space and rendering it.
8. Use the Delaunay tessellation structure for configuration by using fast localization of points (Guibas and Stolfi 1985).

4 Algorithm

4.1 Two Objetives

We have already said that for only two objectives the actual problem of finding the Pareto surface becomes simpler because after sorting the given nondominated points, every point has at least one unique neighbor.

Table 1: This table shows the mathematical formulation of the four standard two-objective test problems given in figure 1.

$$\text{Kursawe}\colon [-5,5]^3 \to \mathbb{R}^2 \qquad \text{Kursawe}(x) \coloneqq \begin{pmatrix} \sum_{i=1}^{2} -10\exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right) \\ \sum_{i=1}^{3} |x_i|^{0.8} + 5\sin x_i^3 \end{pmatrix}$$

$$\text{ZDT3}\colon [0,1]^{30} \to \mathbb{R}^2 \qquad \text{ZDT3}(x) \coloneqq \begin{pmatrix} x_1 \\ g(x)\left(1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)}\sin 10\pi x_1\right) \end{pmatrix}$$

$$g(x) \coloneqq 1 + \frac{9}{29}\sum_{i=2}^{30} x_i$$

$$\text{Poloni}\colon [-\pi,\pi]^2 \to \mathbb{R}^2 \qquad \text{Poloni}(x) \coloneqq \begin{pmatrix} 1 + [A - p(x)]^2 + [B - q(x)]^2 \\ (x_1+3)^2 + (x_2+1)^2 \end{pmatrix}$$

$$A \coloneqq 0.5\sin 1 - 2\cos 1 + \sin 2 - 1.5\cos 2$$

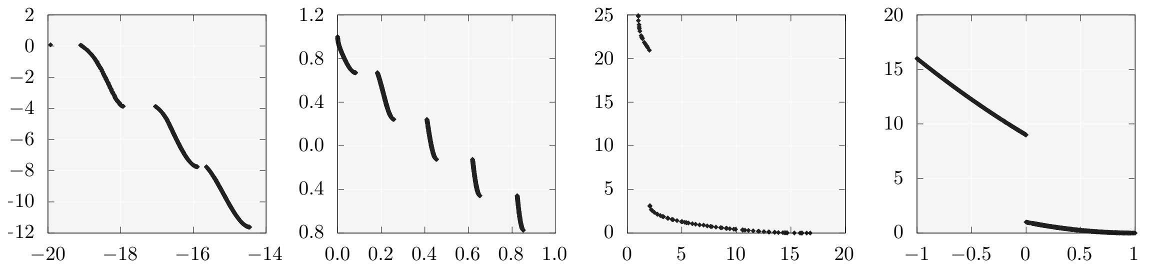
$$B \coloneqq 1.5\sin 1 - \cos 1 + 2\sin 2 - 0.5\cos 2$$

$$p(x) \coloneqq 0.5\sin x_1 - 2\cos x_1 + \sin x_2 - 1.5\cos x_2$$

$$q(x) \coloneqq 1.5\sin x_1 - \cos x_1 + 2\sin x_2 - 0.5\cos x_2$$

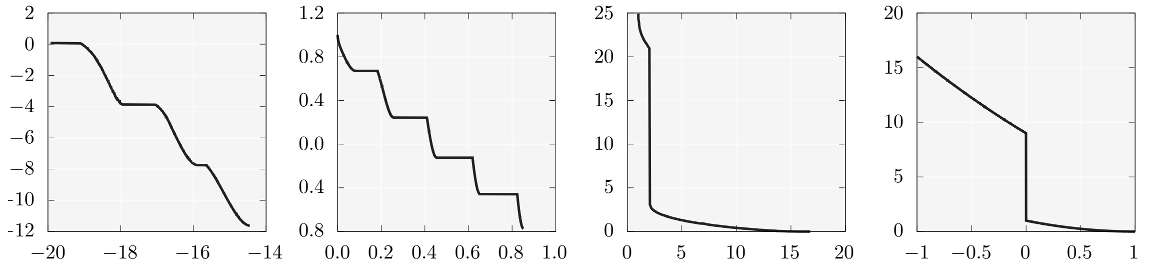
$$\text{Schaffer2}\colon [-5,10] \to \mathbb{R}^2$$

$$\text{Schaffer2}(x) \coloneqq \begin{pmatrix} -x\mathbb{1}_{(-\infty,1]}(x) + (x-2)\mathbb{1}_{(1,3]}(x) + (4-x)\mathbb{1}_{(3,4]}(x) + (x-4)\mathbb{1}_{(4,\infty)}(x) \\ (x-5)^2 \end{pmatrix}$$



(a) Kursawe (b) ZDT3 (c) Poloni (d) Schaffer2

Figure 1: These plots show the discontinuous Pareto frontiers of four standard two-objective test problems. All nondominated solutions were generated by using a custom C++ implementation of the NSGA-II algorithm according to Deb et al. (2002).



(a) Kursawe (b) ZDT3 (c) Poloni (d) Schaffer2

Figure 2: These plots show the naive linear interpolation schemes for the two-objective test problems given by figure 1. Discontinuities of the Pareto frontier are completely ignored. A configurator based solely on linear interpolation would provide impossible solutions.

4.2 Three Objectives

5 Implementation

6 Results

7 Conclusions and Future Work

The assumptions that points are uniformly distributed on the Pareto is a strong one. For many points it suffices, to assume uniformity in a smaller region around every point.

Besides the point distance also the gradient decides on continuity.

References

Bowyer, A. (1981). “Computing Dirichlet Tessellations”. In: *The Computer Journal* 24, pp. 162–166. DOI: 10.1093/comjnl/24.2.162.

Burnikel, Christoph (1998). *Delaunay Graphs by Divide and Conquer*. URL: https://pure.mpg.de/rest/items/item_1819432_4/component/file_2599484/content (visited on 11/07/2020).

Cignoni, P., C. Montani, and R. Scopigno (1998). “DeWall: A Fast Divide-and-Conquer Delaunay Triangulation Algorithm in E^d”. In: *Computer-Aided Design* 30, pp. 333–341. DOI: 10.1016/S0010-4485(97)00082-1.

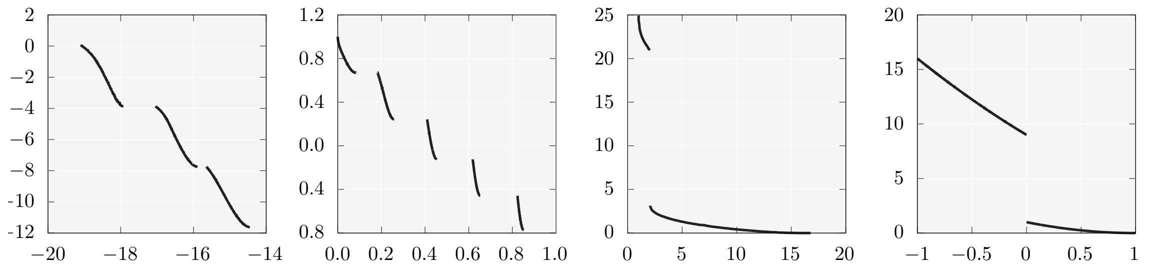
Deb, Kalyanmoy (1998). *Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems*. Tech. rep. CI-49/98. Dortmund, Germany: Department of Computer Science/XI, University of Dortmund. DOI: 10.1162/evco.1999.7.3.205.

Deb, Kalyanmoy and Himanshu Jain (2014). “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Non-Dominated Sorting Approach, Part I: Solving Problems with Box Constraints”. In: *IEEE Transactions on Evolutionary Computation* 18.4, pp. 577–601. DOI: 10.1109/TEVC.2013.2281535.

Deb, Kalyanmoy et al. (2002). “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6, pp. 182–197. DOI: 10.1109/4235.996017.

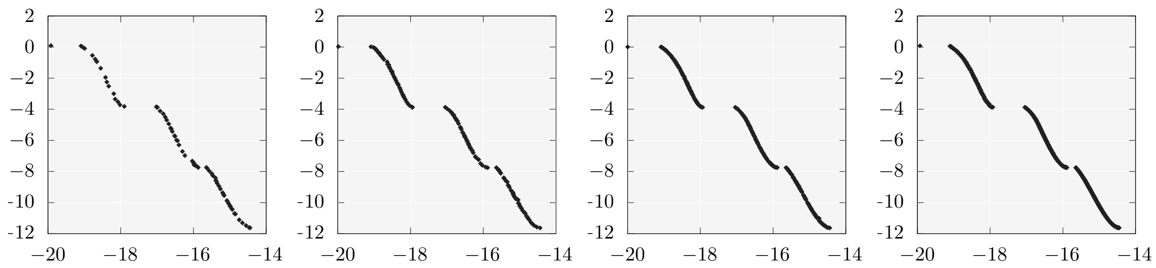
Dwyer, Rex A. (November 1987). “A Faster Divide-and-Conquer Algorithm for Constructing Delaunay Triangulations”. In: *Algorithmica* 2, pp. 137–151. DOI: 10.1007/BF01840356.

Fuetterling, V., C. Lojewski, and F.-J. Pfreundt (2014). “High-Performance Delaunay Triangulation for Many-Core Computers”. In: *High Performance*



(a) Kursawe (b) ZDT3 (c) Poloni (d) Schaffer2

Figure 3: These plots show the linear interpolation schemes for the two-objective test problems given by figure 1 after applying the statistical heuristic to remove connections between neighboring points that may possibly exhibit discontinuities. For all examples, exactly those connections were removed that actually interpolated over a discontinuity.



(a) 100 (b) 200 (c) 400 (d) 1000

Figure 4: These plots show the estimated Pareto frontier of the Kursawe problem given in table 1 and shown in figure 1 for different population sizes used in the underlying algorithm. All nondominated solutions have been computed by using a custom implementation of the NSGA-II algorithm according to Deb et al. (2002).

Graphics 2014, pp. 97–104. DOI: 10.2312/hpg.20141098.

Guibas, Leonidas and Jorge Stolfi (April 1985). “Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams”. In: *ACM Transactions on Graphics* 4, pp. 74–123. DOI: 10.1145/282918.282923. URL: http://sccg.sk/~samuelcik/dgs/quad_edge.pdf (visited on 11/07/2020).

Ibrahim, Amin et al. (2018). “3D-RadVis Antenna: Visualization and Performance Measure for Many-Objective Optimization”. In: *Swarm and Evolutionary Computation* 39, pp. 157–176. DOI: 10.1016/j.swevo.2017.09.011.

Jain, Himanshu and Kalyanmoy Deb (2014). “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Non-Dominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach”. In: *IEEE Transactions on Evolutionary Computation* 18.4, pp. 602–622. DOI: 10.1109/TEVC.2013.2281534.

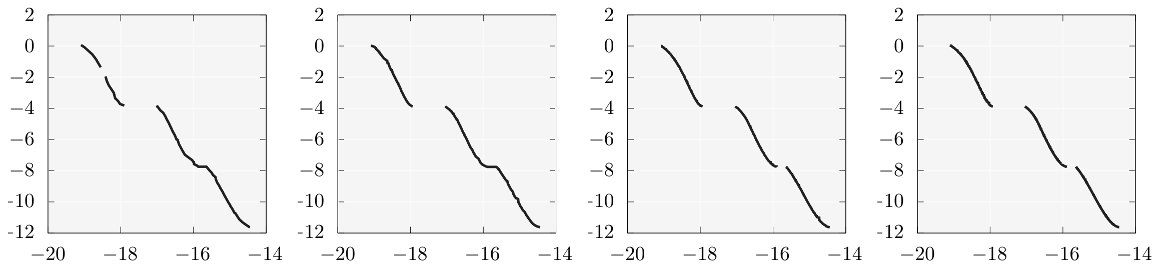
Katajainen, Jyrki and Markku Koppinen (April 1988). “Constructing Delaunay Triangulations by Merging Buckets in Quad-Tree Order”. In: *Fundamenta Informaticae* 11, pp. 275–288.

Konak, Abdullah, David W. Coit, and Alice E. Smith (January 2006). “Multi-objective Optimization Using Genetic Algorithms: A Tutorial”. In: *Reliability Engineering and System Safety* 91, pp. 992–1007. DOI: 10.1016/j.ress.2005.11.018.

Lee, D. T. and B. J. Schachter (1980). “Two Algorithms for Constructing a Delaunay Triangulation”. In: *International Journal of Computer and Information Sciences* 9, pp. 219–242. DOI: 10.1007/BF00977785.

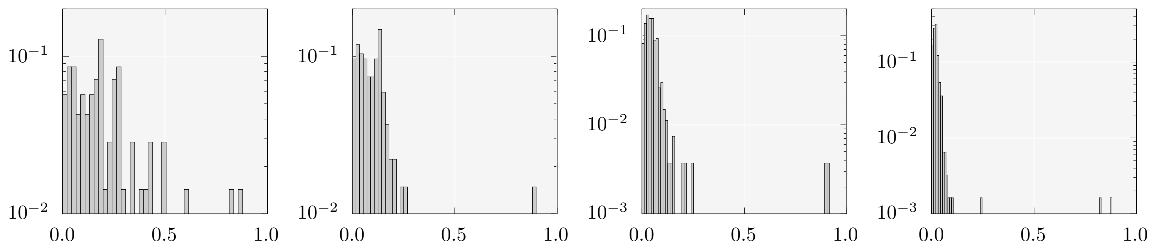
Legriel, Julien et al. (2010). “Approximating the Pareto Front of Multi-Criteria Optimization Problems”. In: *Tools and Algorithms for the Construction and Analysis of Systems*. Springer Berlin Heidelberg, pp. 69–83. ISBN: 978-3-642-12002-2. DOI: 10.1007/978-3-642-12002-2_6.

Lischinski, Dani (1993). *Incremental Delaunay Triangulation*. URL: http://www.karlchenofhell.org/cppswp/lischinski.pdf (visited on 11/07/2020).



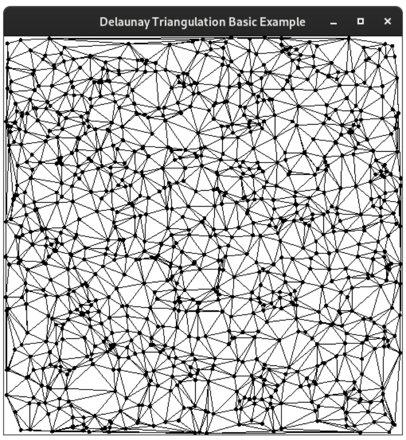
(a) 100 (b) 200 (c) 400 (d) 1000

Figure 5: These plots show the linear interpolation schemes for Pareto frontier of the Kursawe problem shown in figure 4 after applying the statistical heuristic to remove connections between neighboring points that may possibly exhibit discontinuities. Due to the coarse grid of nondominated points for 100 and 200 samples in the population size, the given heuristic was not able to find all discontinuities and even removed a valid connection for 100 samples.



(a) 100 (b) 200 (c) 400 (d) 1000

Figure 6: These plots show the relative frequency histograms of distance distribution of neighboring points in the Pareto frontier of the Kursawe problem for different population sizes according to figure 4.



Shewchuk, Jonathan Richard (n.d.). *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*. URL: https://people.eecs.berkeley.edu/~jrs/papers/triangle.pdf (visited on 11/07/2020).

Srinivas, N. and Kalyanmoy Deb (1994). “Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms”. In: *Evolutionary Computation* 2.3, pp. 221–248. DOI: 10.1162/evco.1994.2.3.221.

Watson, D. F. (1981). “Computing the n-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes”. In: *The Computer Journal* 24, pp. 167–172. DOI: 10.1093/comjnl/24.2.167.

Zitzler, Eckart, Kalyanmoy Deb, and Lothar Thiele (February 2000). “Comparison of Multiobjective Evolutionary Algorithms: Empirical Results”. In: *Evolutionary Computation* 8.2, pp. 173–195. DOI: 10.1162/106365600568202.

Zitzler, Eckart, Marco Laumanns, and Lothar Thiele (July 2001). “SPEA2: Improving the Strength Pareto Evolutionary Algorithm”. In: *TIK-Report* 103. URL: https://people.ee.ethz.ch/~sop/publicationListFiles/zlt2001a.pdf (visited on 01/21/2021).

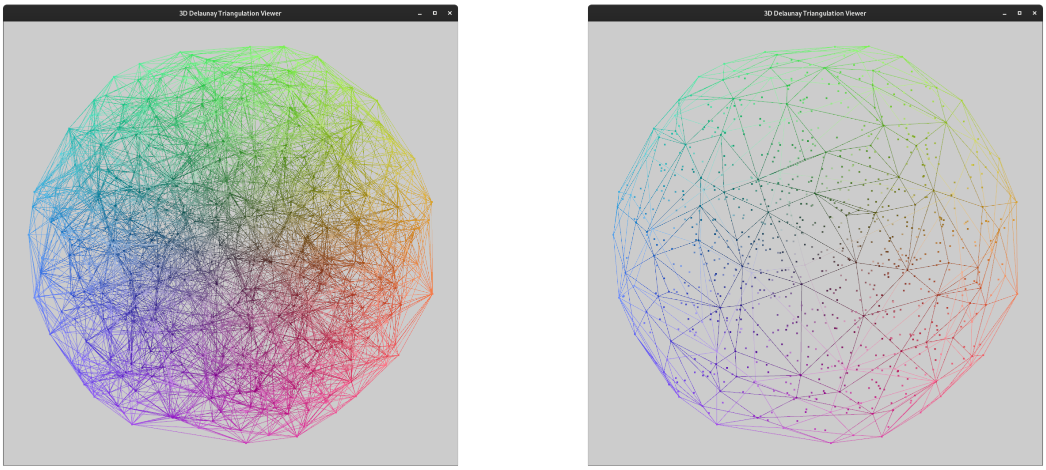


Figure 7: The left image shows a three-dimensional Delaunay tessellation of uniformly distributed points in the unit ball. All points are connected by using tetrahedrons. Using only the surface triangles, one can generate a Delaunay triangulation of the curved surface as seen in the right picture.

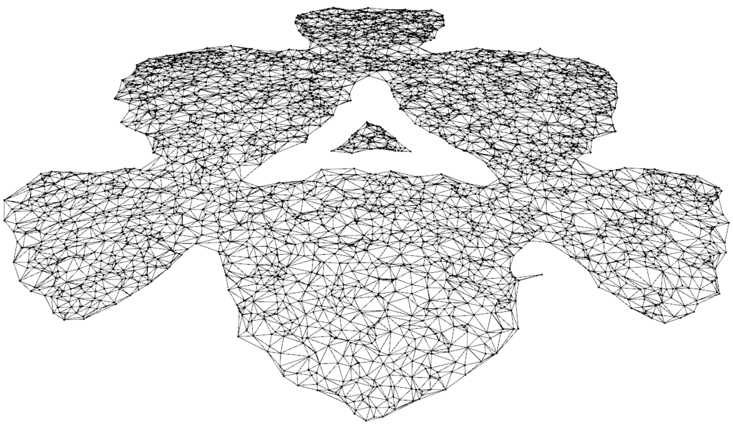


Figure 8: The image shows a two-dimensional Delaunay triangulation of to-a-hyperplane projected curved Pareto surface in three-dimensional space from figure 11. This procedure will be used to construct the surface of general curved Pareto frontiers.

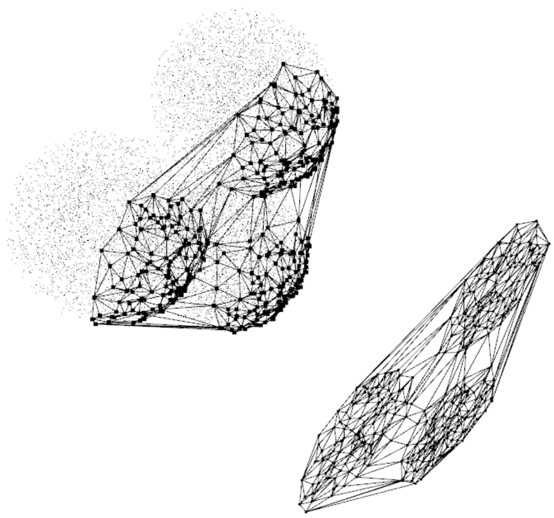


Figure 9: This images shows a three-dimensional domain constructed by three uniform ball distributions and its triangulated Pareto frontier. First, the Pareto frontier has been projected onto the two-dimensional hyperplane with normal $n = \begin{pmatrix}1 & 1 & 1\end{pmatrix}$. Then the Delaunay algorithm was used to generate the triangulation.

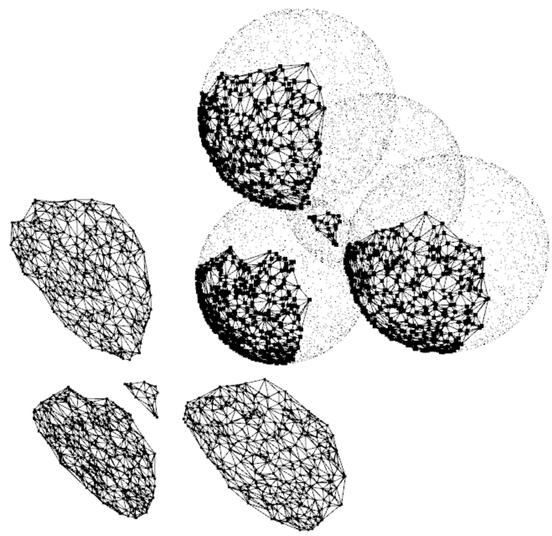


Figure 10: This images shows a three-dimensional domain constructed by four overlapping uniform sphere distributions and its triangulated Pareto frontier. First, the Pareto frontier has been projected onto the two-dimensional hyperplane with normal $n = \begin{pmatrix}1 & 1 & 1\end{pmatrix}$. Then the Delaunay algorithm was used to generate the triangulation. At the end, the statistical heuristic removed connections with large distances to dispose of incontinuities.

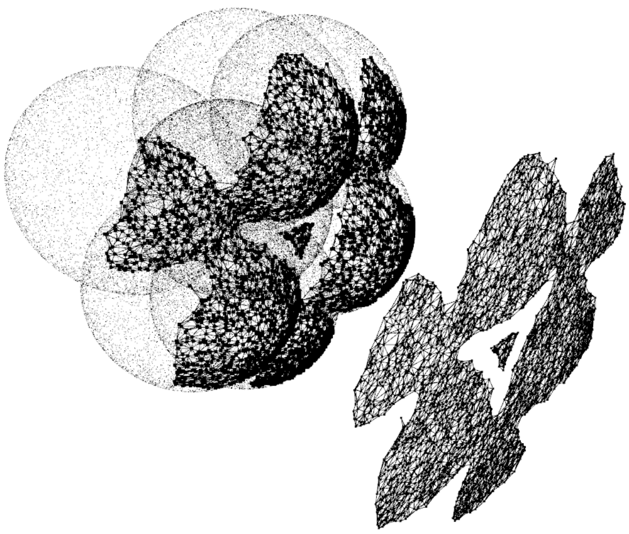


Figure 11: This images shows a three-dimensional domain constructed by seven overlapping uniform sphere distributions and its triangulated Pareto frontier. The same algorithm as in figure 10 was used.