Approximation Methods in Optimization

Overview

Several different approximation methods are utilized in the field of optimization. Here we consider two types of approximations:

Approximating the behavior of objective functions

In order to reduce the number of required objective function evaluations, so-called metamodels (such as neural networks, support vector machines [4], design and analysis of computer experiments [3], etc) may be used to approximate the behavior of computationally costly objective functions, and thus aid to decide where the next function evaluation should be made. Here we present an algorithm utilizing some ideas borrowed from the metamodeling community in solving a single-objective optimization problem.

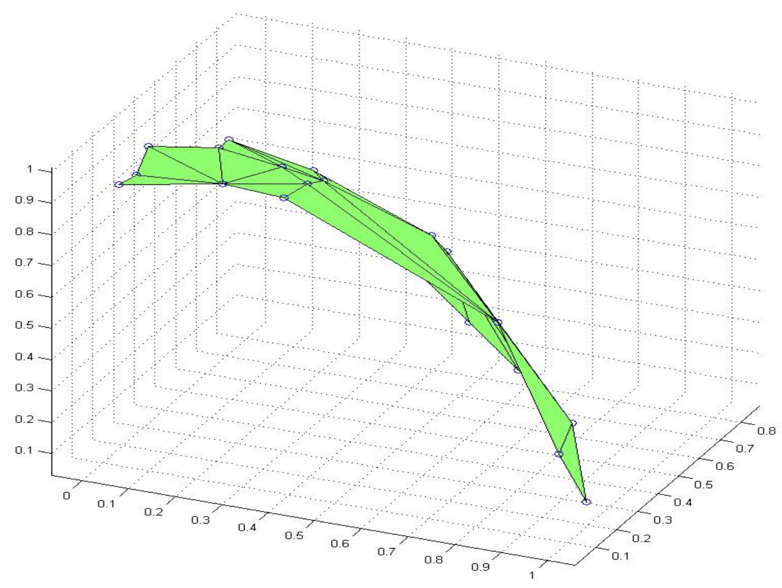
Approximating the Pareto front

The Pareto front of a multiobjective optimization problem can be approximated with methods tailor-made that purpose, by evolutionary multiobjective optimization algorithms, or it can be approximated with meta models. Here a tailor-made approach and a metamodel based approach are described.

Even though these three are presented here together, they are not in any way similar. As an example of the differences, let us stress that in a sence evolutionary algorithms stop, where the two other methods start. Evolutionary algorithms produce nondominated points and the two other methods use a set of Pareto optimal points to approximate the rest of the Pareto front. The approximation of the Pareto front can be viewed for example as a result of the optimization process, as it is done in classical evolutionary multiobjective optimization algorithms, or as a part of the multiplecriteria decision making process.

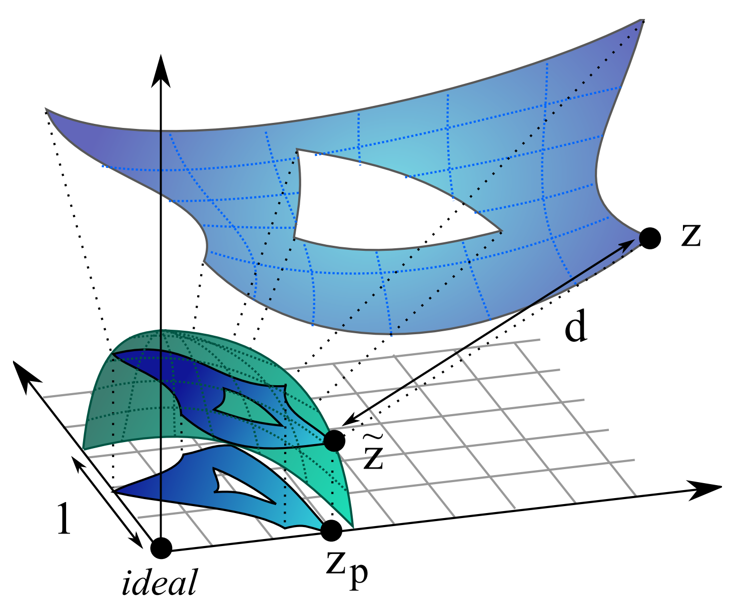
Approximating the Pareto Front with Simplices

The Pareto front can be approximated with simplices that are generated by the initial Pareto optimal points. The simplices that describe the Pareto front in the best way are then selected by using known dominance structures in the objective space. The major benefits of this model are accurate prescription of the Pareto front with few points and mutual non-dominance of the constructed approximation. This method has been developed by Markus Hartikainen and Professor Kaisa Miettinen and it is to be presented in the MCDM2009 conference in China, Chengdu in June 2009. Below is a picture of the approximation of the Pareto front of the DTLZ2 [2] multiobjective optimization problem. This approximation was initialized with 20 random points on the front.



Approximating the Pareto Front with Metamodels

The main idea is to construct a distance function approximation from a part of an $ideal$-centered 1-radius sphere to $\mathbb{R}$. To do this, we assume that we have a given set of Pareto optimal objective vectors. First, we calculate for each given Pareto optimal vector the distance to the sphere. Then we project all the given Pareto optimal vectors into the sphere by normalizing the distance to 1. After that, we have a projected vector set in the sphere and for each projected vector we have a real value distance, and this set of pairs is a training data set. Now, we are ready to fit a metamodel to the training data and that model we use as an approximation of the Pareto front. See more from the below picture, where a Pareto front of a three objective optimization problem is illustrated.



$$ideal = (\min_{x \in S} f_1(x), \ldots, \min_{x \in S} f_n(x))^\mathsf{T}$$

$z \in$ *Pareto front*

$$d = ||z - ideal|| - 1 \geq -1$$

$$\tilde{z} = \frac{z - ideal}{||z - ideal||}$$

$$z_p = (\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_{n-1})^\mathsf{T} \in \mathbb{R}^{n-1}$$

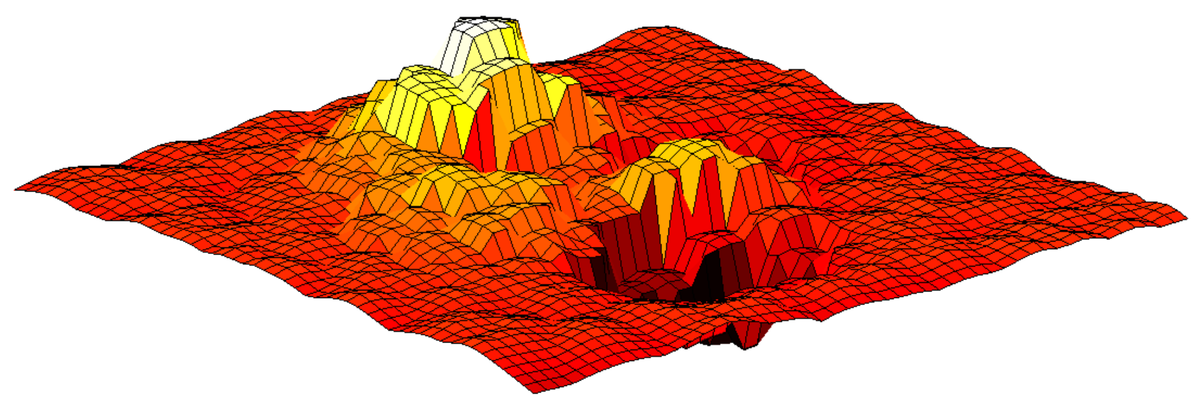
The modeling method and technique is still under construction and it is a part of the dissertation of Tomi Haanpää. Professor Kaisa Miettinen and senior assistant Jussi Hakanen are supervising the work.

Improving Efficiency with Metamodels

In this approach based on Differential Evolution (DE) we do not explicitly construct the surrogate, but we use the information of previously sampled points to filter away some of the trial points so that expensive function evaluations are avoided in uninteresting areas of the search space. The algorithm [1] is simple and rather efficient:

1. Initialize the population (randomly or using some space filling pseudo random sequence to guarantee a sufficient diversity of solutions).
2. While stopping criterion is not met:
   (a) For each parent in the population:
      i. generate the user specified number (one or more) of trial points (using mutation and crossover operators of DE),
      ii. predict the objective function values of all the trial points, select the best value, and set this value as a trial point value,
      iii. filtering: if the trial point value is better than the parent's objective value, evaluate the trial point using the real objective function, and set this value as a trial point value,
      iv. add the current trial point to the child population.
   (b) Select the members for the next population by choosing in a pairwise manner the better member from the parent and the child populations, using the respective parent and child pairs in comparison

Predicted values are computed using means of interpolation, accompanied with mechanism to incorporate information about the reliability of prediction. Benefit of this approach is rather efficient method with simple implementation. In comparison to more complex approaches, such as Efficient Global Optimization, our approach offers smaller computational overhead of the algorithm itself, but with the penalty of inferior efficiency (in terms of required objective function evaluations).



Researchers

Professor Kaisa Miettinen
kaisa.miettinen@jyu.fi

Researcher Timo Aittokoski
timo.p.aittokoski@jyu.fi

PhD student Markus Hartikainen
markus.e.hartikainen@jyu.fi

PhD student Tomi Haanpää
tomi.v.haanpaa@jyu.fi

For further information you are encouraged to freely contact us.

Collaborators

- Professor Kathrin Klamroth, University of Wuppertal

References

References

[1] T. Aittokoski. Efficient evolutionary optimization algorithm: Filtered differential evolution. In *Reports of the Department of Mathematical Information Technology Series B. Scientific Computing, No. B 20/2008*. University of Jyväskylä, 2008.

[2] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, volume 1, pages 825–830, May 2002.

[3] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[4] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

UNIVERSITY OF JYVÄSKYLÄ

Industrial Optimization Group
Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland