Paul Scherrer Institut

Summer Internship Report

Towards Large-Scale Simulation-Based Multi-Objective Optimization

Author:
Andrew Foster

Supervisor:
Dr. Andreas Adelmann
Yves Ineichen

June 1, 2012

1 Introduction

Large-scale multi-objective optimization problems, particularly those in simulation-based engineering, require finding sets of high-quality solutions in a computationally efficient manner. Two well-established heuristics for sampling the set of Pareto optimal solutions are scalarization methods and evolutionary algorithms. Scalarization (or shooting-based) methods sample the Pareto front by re-parameterizing the problem with objective function weights as in

$$\min(1-\lambda)f_1(x)+\lambda f_2(x) \text{ subject to } \begin{array}{c} g(x)\leq 0 \\ 0\leq\lambda\leq 1 \end{array} \qquad (1)$$

By varying the λ parameter and performing a series of independent (and embarrassingly parallel) optimizations, the trade-off between objectives can be probed. Often, however, the mapping between the relative objective weight, λ, and the arc-length of the Pareto front is heavily non-linear, resulting in poor interpolation performance (See figure 3). Evolutionary multi-objective optimization algorithms, such as NSGA and SPEA, address this shortfall by incorporating global solution comparison mechanisms that prevent solution clustering and ensure a more even sampling of the Pareto optimal set. This, however, involves costly aggregate communication and, combined with the high number of function evaluations required, often hinders scalability and, thus, application to large problems. Here we investigate an alternate algorithm for producing evenly sampled Pareto fronts with improved computational efficiency and excellent parallel scalability.

2 Surrogate Model

While the various optimization strategies mentioned above aim to reduce the number of function evaluations required to reach a minimum, the overall computational burden can also be addressed by reducing the complexity of the objective function. Particularly in cases involving large-scale engineering simulations as in [3], this can be accomplished by substituting a surrogate model that merely interpolates the response of the target functions between predefined sample points. Concretely, after completing a series of simulations using a representative set of design variables during a one-time setup phase, the results are combined with inexpensive interpolation routines to replace the simulation-based target functions of the optimization procedure. Because the optimization of the SwissFEL injector involves a large number of design variables, our surrogate model employs a simple Radial Basis Function Network (RBFN), which demonstrate favorable interpolation properties for scattered samples of smooth functions in high-dimensional spaces. Moreover, RBFN evaluation complexity scales linearly with dimension and number of sample points.

While not implemented in the current incarnation of the optimization framework, RBFN-based surrogates provide opportunities for additional benefits.

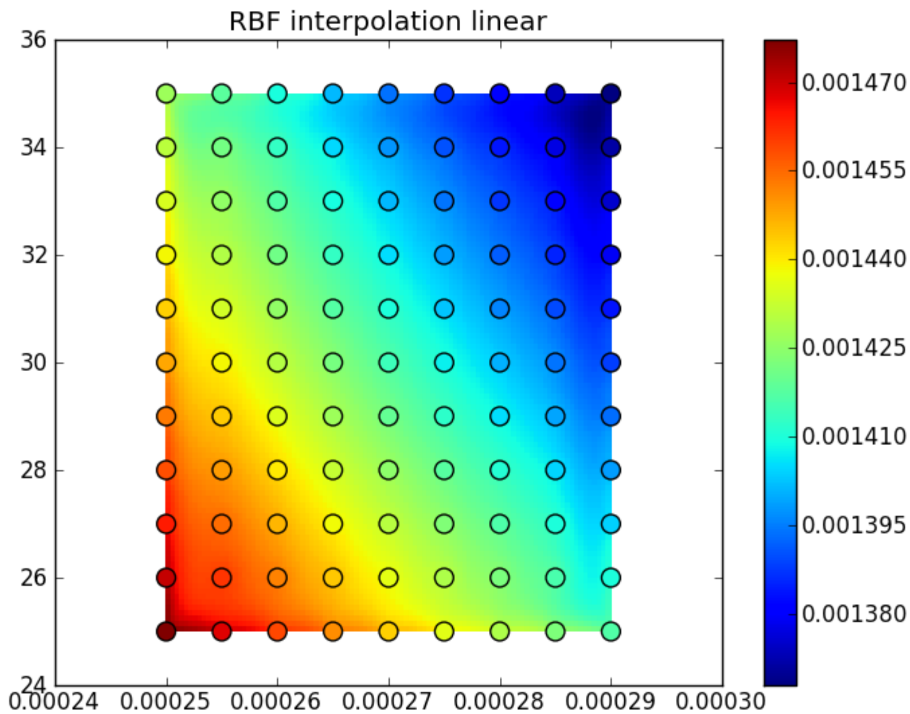

Figure 1: RBFN interpolated values for the RMS beam spread in the s-direction from a grid of simulation-based sample points shown as circles.

Firstly, with a suitable choice of the kernel, the surrogate function is analytically differentiable with respect to the design variables; exploiting this structure would dramatically improve the time-to-solution for the optimization procedure by completely eliminating the need for finite difference-based gradient approximations (which both require a large number of function evaluations and introduce additional error). Secondly, retraining the RBFNs is rather straightforward and inexpensive, thus facilitating the introduction of additional sample points and making them particularly suitable to multi-scale interpolation using data dynamically generated during the optimization process.

3 Homotopy Method

Pereyra proposed in [4] a method for computing the Pareto front of general bi-objective optimization problems with a number of key features. Primarily, it provides a representation of the Pareto front with equispaced discrete samples.

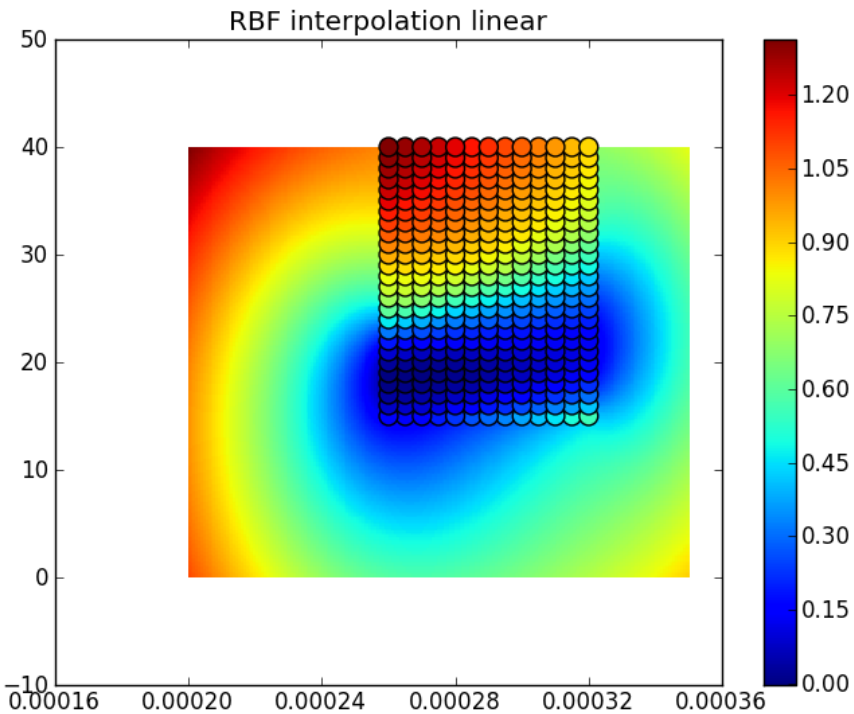

Figure 2: RBFN values for the beam emittance in the x-direction from a dense grid of simulation-based sample points shown as circles.

This is accomplished by adding an equality constraint that guarantees a specific spacing to a scalarized bi-objective problem.

$$||f(x) - f_{prev}||^2 = \gamma^2 \quad (2)$$

where γ is the requested sample spacing. The result is a further constrained optimization problem with (1) and (2) combined . The author then describes a marching homotopy method for generating the Pareto front, by seeding f_{prev} with the image of one of the objective function minimizers (corresponding to a λ value of either zero or one). A self-contained Sequential Quadratic Programming (SQP) package (in this case SNOPT [4]) is used to iteratively solve for the subsequent points until the end of the Pareto front (the minimizer of the other objective function) is reached. We implemented this method and confirmed the quality of results for a subset of the selected problems (see figures 3 and 4).

$$\min \begin{array}{l} (x_0 - 1)^4 + (x_1 - 1)^2 + (x_2 - 1)^2 \\ (x_0 + 1)^2 + (x_1 + 1)^4 + (x_2 + 1)^2 \end{array} \quad (3)$$

$$\min \begin{array}{l} 1 - \exp(-\sum_{i=1}^{3}(x_i - 1/\sqrt{3})^2) \\ 1 - \exp(-\sum_{i=1}^{3}(x_i + 1/\sqrt{3})^2) \end{array} \tag{4}$$

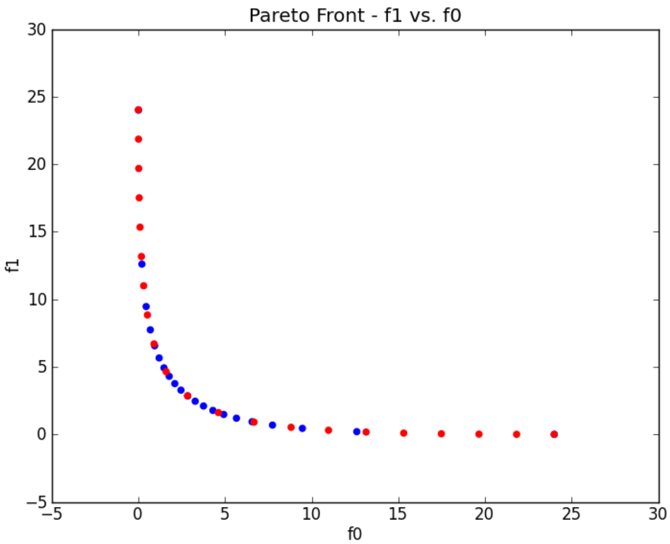

Figure 3: The Pareto front for the bi-objective problem (3). The results from the the classical shooting method are shown in blue, while the homotopy results using the algorithm from [4] are shown in red.

Overall, we can confirm that this method produces more equidistant sampling of the Pareto front and requires far fewer function evaluations than the competing shooting-based methods.

Secondly, the author describes a modified algorithm for parallel computation of the Pareto front (in contrast to the serial marching method used to introduce the concept) by solving the set of loosely coupled minimizations simultaneously and asynchronously updating a global estimate of the front. While this method works best when seeded with a good initial estimate of the front, it is shown that even a simple line connecting the separate objective function minimae will suffice. We, again, implemented this method and confirmed the stated results. See figure 5 for a comparison of the results of the serial marching and the parallel asynchronous algorithms.

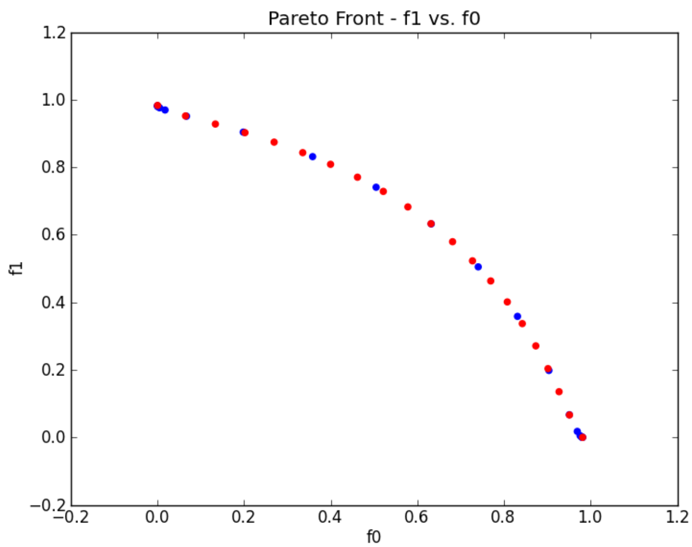

Figure 4: The Pareto front for the bi-objective problem (4). The results from the the classical shooting method are shown in blue, while the homotopy results using the algorithm from [4] are shown in red.

4 Our Contributions

4.1 Alternate Constraint

In [4] and [1], the author describes an issue with proposed method involving the step size in the equispacing constraint, which is computed by

$$\gamma = \alpha \frac{||f_0 - f_{l+1}||}{l} \tag{5}$$

where l is the number of Pareto optimal points in the sample and α is arc-length of the Pareto front. One can imagine a bootstrapping process whereby the total arc-length of the front is estimated and refined by successive applications of this algorithm. A far simpler approach that accomplishes the same end is to adjust the constraint (2) such that equispacing between the up and down-stream neighbors is enforced. Therefore, we propose modifying the algorithm with the following constraint

$$||f_i - f_{i+1}||^2 - ||f_i - f_{i-1}||^2 = 0 \tag{6}$$

With this approach, we bypass the necessity of an a priori accurate arc-length estimate. We have implemented this method and directly compare the results to the marching approach in figure 5.

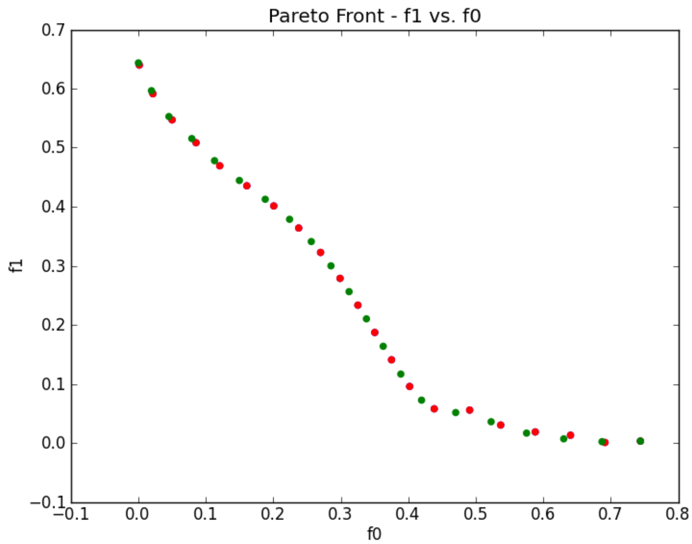

Figure 5: The Pareto front for the bi-objective problem defined by RBFN interpolations from SwissFEL injector simulation data. The results from the marching homotopy method are shown in red, while the homotopy results using the parallel algorithm with the modified constraint (6) are shown in green.

4.2 Locality

Beyond eliminating the need for an explicit a priori arc-length estimate, the constraint (6) has important scalability implications. Specifically, rather than maintaining an estimate of the Pareto front in central storage (as in [4]), each individual optimization sub-problem requires only local communication with neighboring mesh points. Moreover, this communication consists only of asynchronous exchanges of point estimates, meaning that both the number and size of messages required scale linearly with the number of objectives.

4.3 Update Strategies

We also simulated the behavior of a parallel implementation by applying various update strategies to the Pareto front estimation. This amounted to adjusting how many and in what order SQP iterations were applied to the mesh points to mimic the behavior of multiple processors operating simultaneously on various portions of the domain. We found the bi-objective case to be robust against both structured and randomized update strategies indicating the potential for reliable asynchronous, parallel operation. The strongest performance was observed for an even-odd update strategy, wherein mesh nodes were alternately assigned to one of two groups, such that neighboring nodes are not contained in the same group. Alternating updates are applied to each group, allowing for maximum optimization stability.

4.4 Generalization to N-Dimensions

After verifying the performance of the algorithm for various bi-objective problems, we generalized the implementation of the front calculator to multiple dimensions. [1] provides an outline of this procedure, which involves adding a parameter to the scalarization of the objective vector as well as adding analogous equality constraints to ensure equispacing of the resulting front. We deviated from this blue-print by implementing comparable constraints to (6) in the additional dimensions. As discussed above, this allows the front to expand as necessary and localizes the algorithm by restating the constraint in terms of the immediate neighbors. Furthermore, this orthogonalizes the warping of the Pareto front along the mesh-defined axes. Since the parallel homotopy algorithm works by refining an ansatz Pareto front, its application to true multi-objective (three or more) problems involves the generation of suitable N-dimensional regular meshes (vital for the equispacing constraint to be well enforced) as a non-trivial prerequisite. As stated before, this is handled in bi-objective problems by seeding the method with a line connecting the unique minimizers of the individual objectives. This line is then bent into the correct shape by the optimization process. Extending this approach to tri-objective problems, we likewise use the plane defined by the objective function minimizers as the ansatz front. Mapping a regular mesh, however, to a triangle (as in Figure 6) involves a number of subtleties that are currently addressed by compressing one side of a tensor product mesh. Therefore, a more robust and general meshing solution is needed before we can apply the algorithm to higher dimensional problems.

5 Implementation

We have currently implemented a prototype system in the Python programming language, leveraging existing and widely-available open-source libraries wherever possible. This includes the Alglib package for single-objective unconstrained optimization, as well as the NumPy and SciPy packages for constrained optimization via SQP.

6 Benchmark Problems

We have thus far benchmarked the algorithm for the tri-objective using the DTLZ scalable test problems defined in [2]. This test set, however, is inconvenient in that the minimizers of the individual objective functions are non-unique; therefore, the planar ansatz is compressed to a line in objective space. The algorithm does, however, (with one exception) produce correct and uniformly sampled Pareto fronts using fewer than 2,000 function evaluations in most cases. See Figures 7 - 11 for results, which compare favorably with those

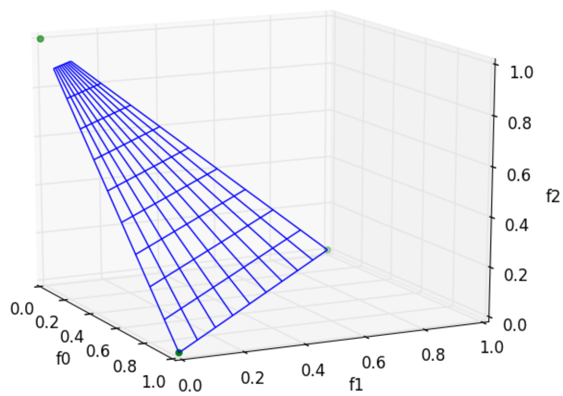

Figure 6: Initial tensor product mesh projected on to the triangular plane defining the ansatz Pareto front.

obtained with other multi-objective optimizers like NSGA and SPEA. We do observe instability in the solution to the DTLZ4 problem (Figure 10), and this is currently a topic of focus.

7 Conclusions

While far from complete, the proposed method provides a promising avenue for further research in multi-objective optimization. The alternative equispacing constraint eliminates the need for an a priori knowledge of Pareto front shape parameters. Furthermore, it requires no global information exchange between optimization sub-problems and opens up the possibility of a distributed approach to high-quality sampling of the Pareto optimal set. The investigation of various update strategies gives some insight into the robustness of the algorithm to asynchronous updates. Finally, the generalization to multiple dimensions allows application of the method to arbitrarily complex problems using a unified framework. While, the alternative constraints permit orthogonalized expansion of the sample mesh in multiple directions, more work needs to be put in to generating suitable multi-dimensional meshes. Finally, more rigorous performance benchmarks and comparisons to existing methods should be undertaken to quantify the quality of the solutions produced as well as the efficiency of the method.

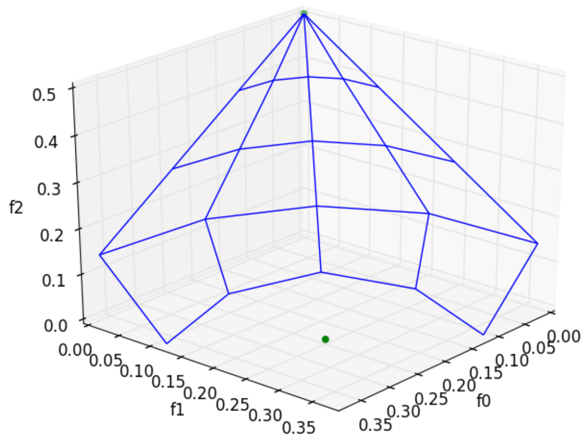

Figure 7: Pareto front representation for DTLZ1 problem.

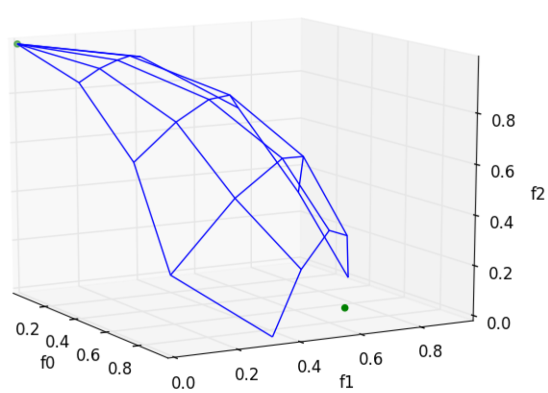

Figure 8: Pareto front representation for DTLZ2 problem.

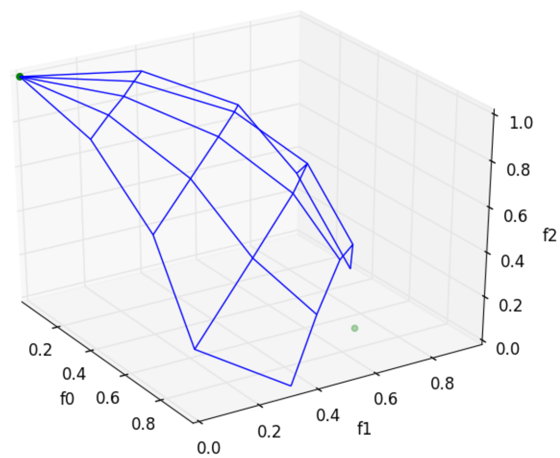

Figure 9: Pareto front representation for DTLZ3 problem.

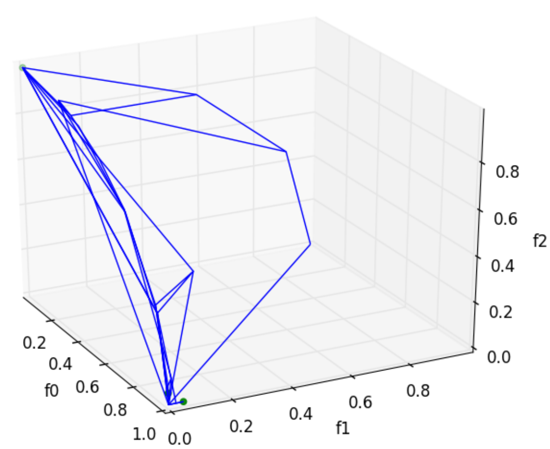

Figure 10: Pareto front representation for DTLZ4 problem.

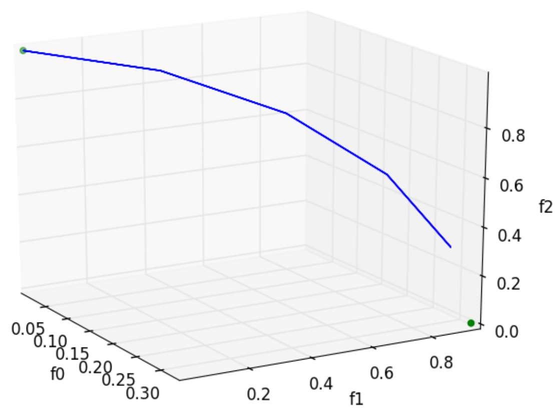

Figure 11: Pareto front representation for DTLZ5 problem.

References

[1] V. Pereyra. 2009. *Fast computation of equispaced Pareto manifolds and Pareto fronts for multiobjective optimization problems.* Mathematics and Computers in Simulation, Volume 79, Issue 6, February 2009, pp. 1935-1947.

[2] K. Deb, L. Thiele, M. Laumanns, E. Zitzler. 2002. *Scalable multi-objective optimization test problems.* Proceedings of the Congress on Evolutionary Computation (CEC-2002), (Honolulu, USA). pp. 825-830.

[3] J. Alonso, P. LeGresley, V. Pereyra. 2009. *Aircraft design optimization.* Mathematics and Computers in Simulation, Volume 79, Issue 6, February 2009, pp. 1948-1958.

[4] V. Pereyra, M. Saunders, J. Castillo. 2011. *Equispaced Pareto front construction for constrained bi-objective optimization.* Mathematical and Computer Modelling, Available online 1 February 2011.