Benchmark problems for constrained global optimization with high-dimensional black-box functions

Donald R. Jones
donjon@umich.edu

May 20, 2025

1 Introduction

In this paper, I introduce two test problems for constrained global optimization with high-dimensional black-box functions. The first problem is identical to the "MOPTA08 Jones Benchmark" problem that I presented in the MOPTA 2008 conference [1]. This problem is based on a full-vehicle mass optimization project from my previous work at General Motors. The design variables specify the thickness of 124 vehicle parts, and the objective is to choose these variables so as to minimize the mass of the vehicle subject to meeting constraints from crash-worthiness, noise and vibration, and durability. The second problem is based on a model of a supersonic business jet first presented in [2] and further developed in [3]. The design variables describe the airfoil, the wing structure, and the propulsion system; the goal is to choose these variables so as to minimize the mass of the airplane subject to achieving a given range and satisfying structural stress constraints.

Both test problems take the following standard form:

$$\begin{aligned}
\text{minimize} \quad & f(\mathbf{x}) && && (1)\\
\text{subject to} \quad & g_i(\mathbf{x}) \leq 0, && i = 1, \ldots, n_g && (2)\\
& h_j(\mathbf{x}) = 0, && j = 1, \ldots, n_h && (3)\\
& \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U && && (4)
\end{aligned}$$

The automotive problem has a linear objective, 68 nonlinear inequality constraints, and no equality constraints. The business jet problem has two versions. The first is based on the "Individual Discipline Feasible" (IDF) formulation of the multidisciplinary optimization (MDO) problem and has 41 variables, 78 inequality constraints, and 10 equality constraints. The second version is based on the "Multidisciplinary Design Feasible" (MDF) formulation and has 31 variables, 78 inequality constraints, and no equality constraints.

MATLAB source code is provided for both problems; the source code for the business jet problem is adapted from code graciously provided by Pascal Etman.

2 Automotive test problem

The automotive test problem is based on an actual full-vehicle mass optimization that was performed at General Motors; the objective and constraints for the test problem come from kriging response surfaces fit to the input-output data collected during the entire project. The input variables describe the gauges of certain stamped metal parts of the vehicle; these gauge variables are normalized to the unit interval. The objective is the mass of the parts being optimized, which actually is a linear function of the variables. Thus, for this problem, all of the computational complexity lies in the nonlinear constraints, as the objective is a simple linear function. The nonlinear constraints capture various performance requirements that the vehicle must meet.

Most multidisciplinary optimization problems involve several separate simulations. In the case of the automotive problem, there were separate simulations for front crash, side crash, rear crash, noise and vibration, and durability; these simulations provided the performance metrics used in the nonlinear constraints. The outputs from the front crash simulation depend mainly on the gauges in the front of the vehicle, forward of the B-pillar; gauges in the rear of the car have essentially no impact on the front crash performance. This made it possible for engineers to specify a subset of variables that are "relevant" for front crash (and similarly for all the other simulations).

The same kind of phenomenon is common in aircraft MDO problems. In particular, the simulation for lift and drag will depend primarily on the shape of the wings and will be unaffected by gauges of support members inside the wing. These gauges, however, will be relevant for the structural simulation to predict critical stresses and wing deflection under load.

In short, while the objective and constraints in real-world problems are often "black-box" in the sense that we do not have a closed-form formula or know properties such as convexity, in many cases it is possible to specify, *a priori*, which variables are relevant for different outputs. We will provide such information for both benchmark problems presented here. This extra information on relevant variables, and the availability of MATLAB source code, are the only differences between the automotive problem described here and the original "MOPTA08 Jones Benchmark."

The number of relevant variables for a given output can be much smaller than the total number of variables in the optimization problem. Optimization methods based on surrogate modeling can take advantage of this fact by fitting the surrogate models using only the relevant variables and possibly also by reducing the size of the initial design of experiments (DOE).

The provided MATLAB code for the test problems comes in a directory called `Benchmarks`, and one should put this directory and all its subdirectories in the MATLAB path. One gets all the information needed to do optimization with the automotive problem with the following command:

```
[automotive,frel,grel,hrel,xl,xu,xopt,x0] = automotive_benchmark();
```

The output `automotive` is a function handle that is called as follows:

```
[f,g,h] = automotive(x);
```

Here `x` is a $1 \times d$ array of input variables, where $d =$ `length(xl)` $= 124$. The output `f` is the scalar objective; `g` is a $1 \times n_g$ array of inequality constraint values, where $n_g =$ `size(grel,1)` $= 68$; and `h` is a $1 \times n_h$ array of equality constraint values, where $n_h =$ `size(hrel,1)`. For the automotive problem, there are no equality constraints, so we have `hrel=[]` and $n_h = 0$.

The output `frel` is a $1 \times d$ logical vector, where `frel(k)=true` if input k is relevant for the objective (i.e., it has an impact on the objective), and `frel(k)=false` if input k is *not* relevant for the objective. Similarly, `grel` is an $n_g \times d$ logical vector, where `grel(i,k)=true` if input k is relevant for inequality constraint `g(i)`. Finally, `hrel` is an $n_h \times d$ logical vector, where `hrel(i,k)=true` if input k is relevant for equality constraint `h(i)`. As mentioned earlier, for the automotive problem there are *no* equality constraints; we therefore have `hrel=[]`, $n_h = 0$, and the call to `automotive` returns `h=[]`.

The outputs `xl` and `xu` are $1 \times d$ vectors giving the lower and upper bounds on the variables. The output `xopt` is a $1 \times d$ vector giving the best known solution, presumably the global minimum. Finally, the output `x0` is a $1 \times d$ vector giving a feasible baseline design, which can be used as a starting point or as part of the initial design of experiments.

In the directory `Benchmarks` there is a file `example.m` that provides an example of solving the automotive problem using MATLAB's "surrogateopt" optimizer. This optimizer is based on surrogate models and is available in the global optimization toolbox. The performance of surrogateopt is sensitive to the setting of the random seed number. When using a seed of 43, it obtains a solution of 231.703 after about 1200 function evaluations (see Figure 1). The best known solution for this problem has an objective of 221.085. Thus, MATLAB's surrogateopt falls quite short of the optimum. I think the goal for new algorithm development should be to get closer to the global minimum in 1000 or fewer function evaluations. An important secondary goal is to make the performance of the algorithm insensitive to hyperparameters such as the random seed number or other algorithmic settings.

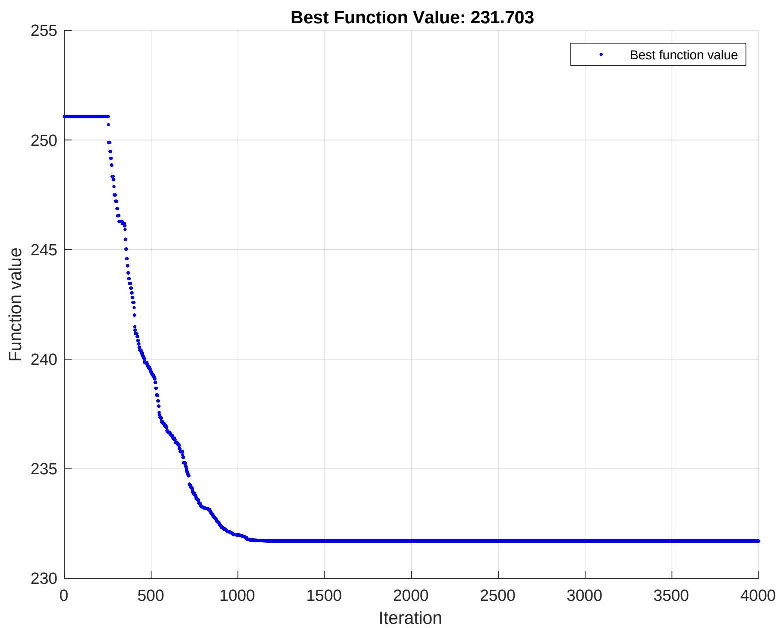

Figure 1: Iteration history of MATLAB's "surrogateopt" optimizer on the automotive problem.

3 Supersonic business jet test problem

The supersonic business jet problem is based on a model first presented in [2] and further developed in [3]. The requirements and constraints for the problem come from the 1995/96 AIAA student aircraft design competition. The aircraft model is divided into subsystems for structures, aerodynamics, propulsion, and range. There are 8 design variables that are shared by all subsystems and 23 other design variables that are specific to one subsystem or another, for a total of 31 variables. In addition to these 31 variables, there are 10 coupling variables that link the subsystems. Each coupling variable is the input to one subsystem and the output of another. Figure 2 illustrates the different modules and the data dependencies.

The system has 78 inequality constraints. These include: the requirement to meet the desired range; limits on engine scale factor, throttle setting, and pressure drag; 30 structural stress con-

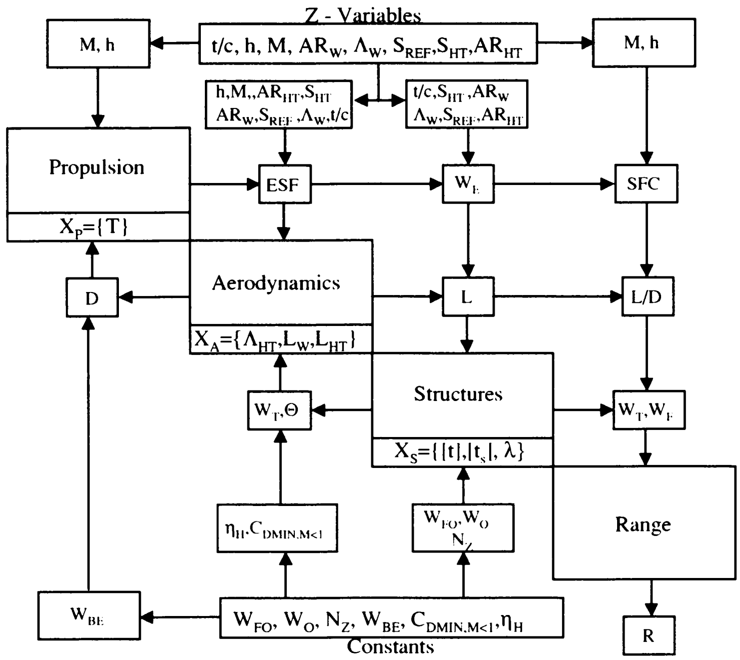

Figure 2: Subsystems and data dependencies for the supersonic business jet model.

straints; 12 restrictions on wingbox thicknesses to ensure physical dimensions remain realistic; and various move limits. When using the individual discipline feasible (IDF) formulation, there are 10 additional equality constraints necessary to make sure the value of a coupling variable used as input to some subsystem equals the value output from another.

For this problem, the stress constraints are very poorly scaled, ranging over several orders of magnitude. Such bad scaling can create difficulties when fitting surrogate models using kriging or radial basis functions. To alleviate this problem, I provide the option to scale all of the constraints using a sigmoid function; in particular, if we have the constraint $g(x) \leq 0$, then I replace this with the constraint $\tilde{g}(x) \leq 0$ where:

$$\tilde{g}(x) = \frac{2}{1 + \exp(-g(x)/c)} - 1 \tag{5}$$

Figure 3 illustrates this mapping. As you can see from the figure, the sigmoid scaling maps negative values of g/c to negative values of $\tilde{g}$, and maps positive values of g/c to positive values of $\tilde{g}$. Hence, the constraint $\tilde{g}(x) \leq 0$ is equivalent to the constraint $g(x) \leq 0$. The advantage of the scaled constraints is that the scaled values all lie in the region $[-1, +1]$ and so are more easily fit using typical surrogate models. The constant $c > 0$ in equation (5) has been set so that $g(x)/c$ is in the interval $[-2, +2]$ about 80% of the time; in this interval, the sigmoid function is fairly linear. In this way, the sigmoid scaling provides little distortion for most of the constraint values and only flattens the extremely positive and negative values. A different value of c is used for each inequality and equality constraint.

One gets all the information needed to do optimization with the business jet problem with the following command:

```
[bjet,frel,grel,hrel,xl,xu,xopt] = businessjet_benchmark(UseEq,Scale);
```

As you can see, the statement is very similar to the one used for the automotive benchmark, and

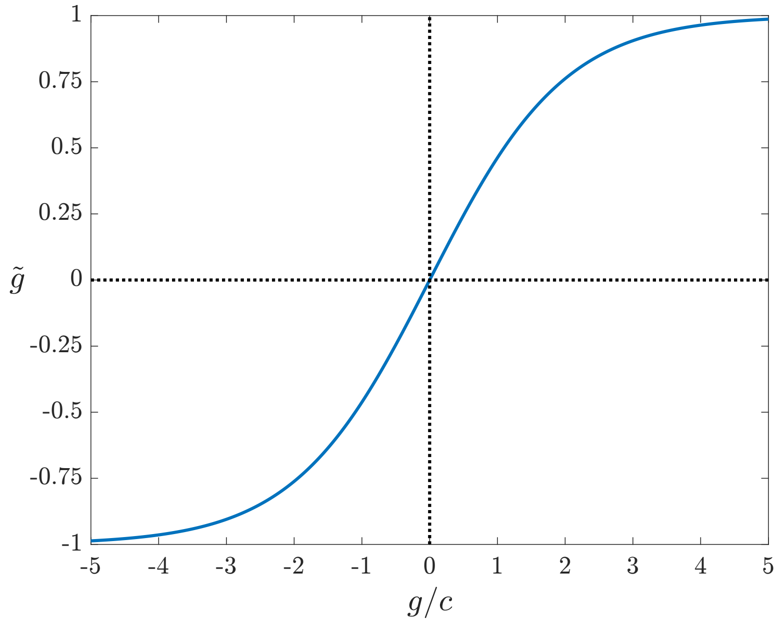

Figure 3: Sigmoid scaling of the constraint functions.

the meaning of all the output variables is the same. The output `bjet` is a function handle that is called as follows:

```
[f,g,h] = bjet(x);
```

For the business jet problem, there are two new inputs, `UseEq` and `Scale`. If one sets `UseEq=true`, then the outputs correspond to the IDF formulation of the business jet problem which *has* equality constraints; otherwise, the outputs correspond to the MDF formulation of the business jet problem which has *no* equality constraints. The input `Scale` determines whether sigmoid scaling is used; if `Scale=true`, then the inequality and equality constraints will be sigmoid scaled; otherwise, the constraints will be on the original scale.

The MDF version of the business jet problem has a feature which needs to be taken into account when trying to solve it. In order to eliminate the coupling variables, we use functional iteration to solve for them. While this usually works, occasionally the iterations fail to converge, and when this happens the call to `bjet` returns `NaN` (not a number) for the objective and all the inequality constraints. In the literature, this situation is sometimes referred to as a "hidden constraint." The idea is that there really is a constraint that the problem functions should be computable, but this constraint is not explicitly provided but rather is " hidden"; we only know when the objective and constraints have or have not been successfully computed. The algorithm used to solve the MDF formulation of the business jet problem needs to be able to handle such hidden constraints.

The file `example.m` provides two examples of solving the business jet problem. The first example solves the IDF formulation with sigmoid scaling using the local optimizer `fmincon` from a given starting point saved in the file `Benchmarks/businessjet/x0.mat`. From the provided starting point, local optimizer converges to the global minimum. But if you experiment with different random starting points, you will find that the optimization very often either fails to find a feasible point or converges to a suboptimal local minimum.

The second example solves the MDF formulation with sigmoid scaling using `surrogateopt`. As mentioned earlier, this optimizer is sensitive to the random seed number, and for this illustration we again use the seed of 43. Using this seed, the optimizer converges to an objective value of 33.47

in about 3000 function evaluations (see Figure 4). Since the true global minimum has an objective value of 32.635, `surrogateopt` falls a little short; using other random number seeds can result in substantially worse solutions, with objective values above 40. I think the goal for new algorithm development should be to get closer to the global minimum (with $f < 33$) in 1000 or fewer function evaluations. As always, an important secondary goal is to make the performance of the algorithm insensitive to hyperparameters such as the random seed number or other algorithmic settings.

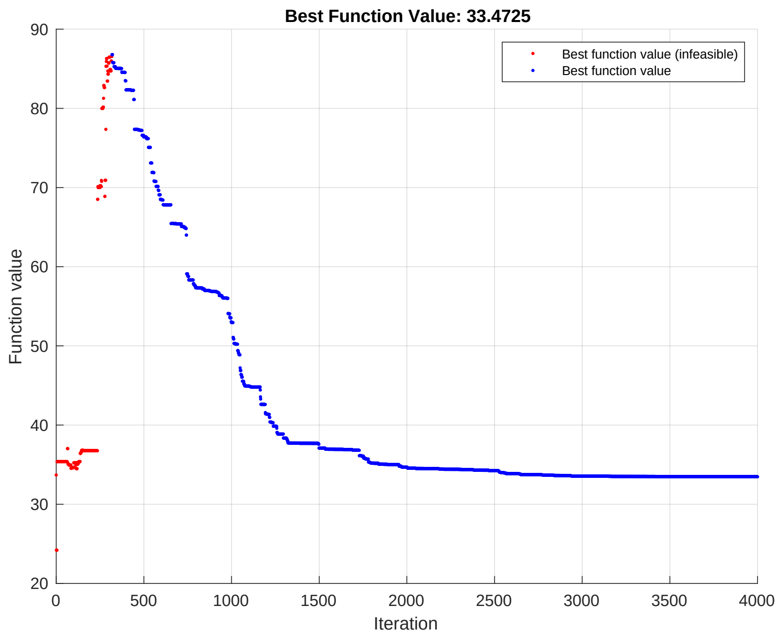

Figure 4: Iteration history of MATLAB's "surrogateopt" optimizer on the business jet problem

.

I encourage my colleagues in the field of global optimization to try their hand at these test problems. The publication of the MOPTA08 Jones Benchmark over 15 years ago stimulated quite a few authors to develop methods that can handle the challenge of high-dimensional, constrained global optimization of black-box functions. I hope that the benchmarks presented here will serve to further push the field forward and provide a basis for comparing algorithms. If you publish a paper in which you use these test problems, please cite the source as [4].

References

[1] D. R. Jones, "MOPTA 2008 benchmark." https://datashare.ed.ac.uk/handle/10283/8960.

[2] J. S. Agte, J. Sobieszczanski-Sobieski, and R. R. Sandusky, "Supersonic business jet design through bi-level integrated system synthesis," *SAE Transactions*, vol. 108, pp. 1356–1364, 1999.

[3] S. Tosserams, M. Kokkolaras, L. F. P. Etman, and J. E. Rooda, "A Nonhierarchical Formulation of Analytical Target Cascading," *Journal of Mechanical Design*, vol. 132, p. 051002, 04 2010.

[4] D. R. Jones, "Benchmark problems for constrained global optimization with high-dimensional black-box functions." https://github.com/donaldratnerjones/2025JonesBenchmarks.