Improving High-dimensional Simulation-driven Optimization

YOEL TENNE
Ariel University
Department of Mechanical and Mechatronic Engineering
Ariel
ISRAEL
y.tenne@ariel.ac.il

Abstract: Computer simulations are being extensively used as a partial substitute for real-world experiments. Such simulations are often computationally intensive and hence metamodels are used to approximate them and to yield estimated output values more economically. While this setup can work well in low dimensional problems it can struggle in high-dimensional ones due to poor metamodel prediction accuracy. As such this study examines the application of dimensionality-reduction procedures during the search so that a simplified problems is formulated which is easier to solve and which could yield a better solution of the original one. An extensive performance analysis with both mathematical test functions and an engineering application shows the effectiveness of the proposed approach.

Key–Words: optimization, simulations, dimensionality reduction, computational intelligence

Received: August 23, 2019 . Revised: February 1, 2020. Accepted: February 21, 2020. Published: March 6, 2020.

1 Introduction

Computer simulations are used in engineering and science to yield significant cost reductions by partially replacing costly and time consuming real-world experiments. Using such simulations transforms the design process, namely, that of finding a better product/system configuration, into an optimization problem with three distinct features: a) the simulation is a 'black-box' function, namely, it assigns input vectors (candidate designs) a corresponding merit value but an analytic expression for this mapping is often unavailable, b) typically the simulation is computationally expensive so only a small number of designs can be evaluated, and c) the optimization landscape is often nonconvex and discontinuous thereby further complicating the optimization. Figure 1 shows the layout of such problems.

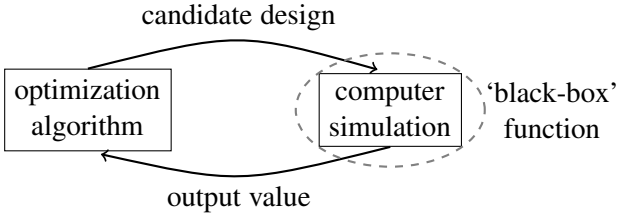

Figure 1: The layout of a simulation-driven optimization problem.

Metamodels, also termed in the literature as surrogates, have been used to circumvent these issues by approximating the latter inputs–outputs relation and providing estimated outputs more economically [1–3]. While this approach can be effective in low-dimensional settings it can falter in high-dimensional ones due to the 'needle-in-a-haystack' effect and the difficulty of generating an accurate approximation in a large search space based on a small sample.

Accordingly high-dimensional simulation-driven problems pose an optimization challenge and as such this paper studies the integration of procedures which transform the original problem into a lower-dimensional one which is simpler to solve. To ensure search progress in the presence of metamodel inaccuracy a trust-region mechanism is used to dynamically adjust the degree of dimensionality reduction. An extensive performance analysis based on a set of high-dimensional problems shows the effectiveness of the proposed approach. The remainder of this paper is organized as follows: Section 2 provides the background information, Sections 3 and 4 describe the algorithm and the dimensionality reduction methods, respectively, while Section 5 gives a detailed performance analysis. Lastly, Section 6 concludes the paper.

2 Background

It has been shown that an effective optimization requires both global and local search stages. To achieve this a common approach is to combine explorative methods, such as evolutionary algorithms (EAs) , with

localized gradient methods which yields a hybrid algorithm [4, 5].

With regards to metamodels multiple variants have been proposed and effectively applied [6, 7]. However, a metamodel accuracy may severely degrade if the sample is too small or the search space is large and this may not not only impair the search but may even introduce false optima which can lead the optimization algorithm to a poor final solution [8]. As such, metamodel-assisted optimization algorithm must account for this inherent inaccuracy and manage it to ensure convergence to a correct solution [9]. An established approach to achieve this is with the trust-region framework in which the metamodel is assumed to be valid in a confined region (the trust-region) and the optimization algorithm seeks the best metamodel vector in the trust-region (TR) only. The TR is then updated based on the success of the optimization trial step.

Given the effectiveness of the hybrid algorithms and the TR framework several studies have explored different formulations. Examples include using RBF [10] or quadratic interpolants for noisy functions [11], and multi-model approaches [5]. The combination of hybrid search and metamodels allows to handle the challenges mentioned, namely, the lack of an analytic expression, high evaluation cost, and a complicated function landscape.

3 A baseline algorithm

A baseline hybrid algorithm is used which combines an explorative EA search with a localized SQP search, and operates in four steps: a) sampling an initial set, b) training a metamodel, c) performing a trust-region search, and d) updating the interpolation set and the trust-region. The initial sampling is made with an optimized Latin hypercube method to achieve an adequate space-filling sample [12]. The metamodel used is Kriging whose details are given in Appendix A, while the real-coded evolutionary algorithm (EA) uses a population size of 100, stochastic universal selection (SUS), intermediate recombination with probability 0.75, mutation with probability 0.1, 10% elitism and a limit of 50 generations. These values were chosen after experimentation with a range of test functions. Candidate solutions obtained by the EA are then further refined with a localized SQP search. After resultant candidate vector is evaluated with the true objective function and is compared to the current best solution obtained and accordingly the following updates take place:

- If the predicted solution is indeed better than the current best then the TR is centred at the new vector and is enlarged.
- If the predicted solution isn't better than the current best and there are sufficient vectors in the TR then the TR is contracted, where the extra check for the number of vectors in the trust-region is done to avoid rapid contraction of the TR and thereby premature convergence [13]
- If the predicted solution isn't better but there is an insufficient number of sampled vectors in the TR then a new sample vector is added to improve the local prediction accuracy. This vector is chosen to be remote from existing interior ones and is obtained by a max-min distance criterion (effectively generating a LHD sample in the TR and selecting the vector with largest max-min distance) [14].

The TR expansion/contraction factor is 2 and at least five interior points are needed to allow contraction. To conclude the description Algorithm 1 gives the pseudocode of the procedures described.

```
sample and evaluate an initial set;
while number of max simulation runs not reached do
    train a metamodel with the sampled vectors;
    define a TR around current best and search for a new optimum;
    evaluate found optimum and update:
        if new optimum is better than current then expand the TR
        else add a new vector in TR
        else if sufficient vectors in TR then contract the TR
        cache all new vectors sampled
end
Output: best solution found
```

Algorithm 1: The baseline optimization algorithm.

4 Dimensionality Reduction

To improve the search effectiveness the above algorithm was modified to include dimensionality reduction methods. Such methods require the output

(lower) dimension to be specified but in practice the optimal value of the latter in unknown. Accordingly in this study a dynamic approach has been implemented in which the lower dimension is adjusted during the search. Specifically the number of successful trial steps within the last n is monitored, where a successful step is such that the predicted solution was indeed better than previously best known. If no iteration was successful within the above interval then the output dimension is reduced by a factor r. If at least 50% of the recent n iterations were successful then the output dimension is increased by a factor r. The way, the dimensionality reduction is used to assist search progress when it appears to stagnate, but the dimension is increased once progress has been restored.

Two alternative dimensionality reduction methods were studied:

- Variable/Subset selection: The goal is to identify a subset of important variables out of the original full set so that a lower-dimensional problem can be formulated [15]. Since the number of possible subsets is $\frac{d!}{p!(d-p)!}$ (d being the dimension and p the subset size) implies that the computational cost of an exhaustive search is prohibitive. Accordingly a search based on a binary GA was implemented such that the best subset is identified based on a cross-validation error assessment, that is, by comparing the predictions based on the high-dimensional set and the lower-dimensional subsets. The binary GA used for the subset selection used SUS selection, a shuffle crossover (shuffling chromosomes between pairs of parents) with probability 0.75, a discrete mutation with probability 0.1, a population size of 100, elitism of 10% elitism, and a generation limit of 50. Figure 2(a) shows that the time complexity of the subset selection was mainly affected by sample size and not by problem dimension due to the prescribed limit of GA generations.

- Mapping: Here the original high-dimensional data is mapped into a lower-dimensional one such that certain aspects in the data are preserved. One such method is the *Sammon mapping* [16] which uses the *stress function*

$$C = \frac{1}{\sum_{i<j} d_{i,j}} \sum_{i<j} \frac{(d_{i,j} - \hat{d}_{i,j})^2}{d_{i,j}} \qquad (1)$$

where $d_{i,j}$ is the distance between the ith and jth high-dimensional (original) vectors the $\hat{d}_{i,j}$ is the distance between the lower dimensional projections. The method was shown to be both effective and efficient [16]. Figure 3(a) shows an example how a Sammon mapping from three to two dimensions preserves the proximity relations in the data, namely, points which are adjacent in high-dimension are mapped to adjacent low-dimensional points and vice-versa. Figure 3(b) shows a metamodel trained by using data of a Rastrigin-5D function based on a mapping to two dimensions. The metamodel preserves the multimodality of the original high-dimensional landscape. With respect to performance, Figure 4(a) shows the mapping time is mainly affected by dimension and is practically unaffected by sample size. Figure 4(b) shows a typical variation of the reduced dimension during the optimization search, and similarly to the example in the previous section, the operating dimension is adjusted based on the search progress with a monotonous decrease in later stages of the search when progress is slower.

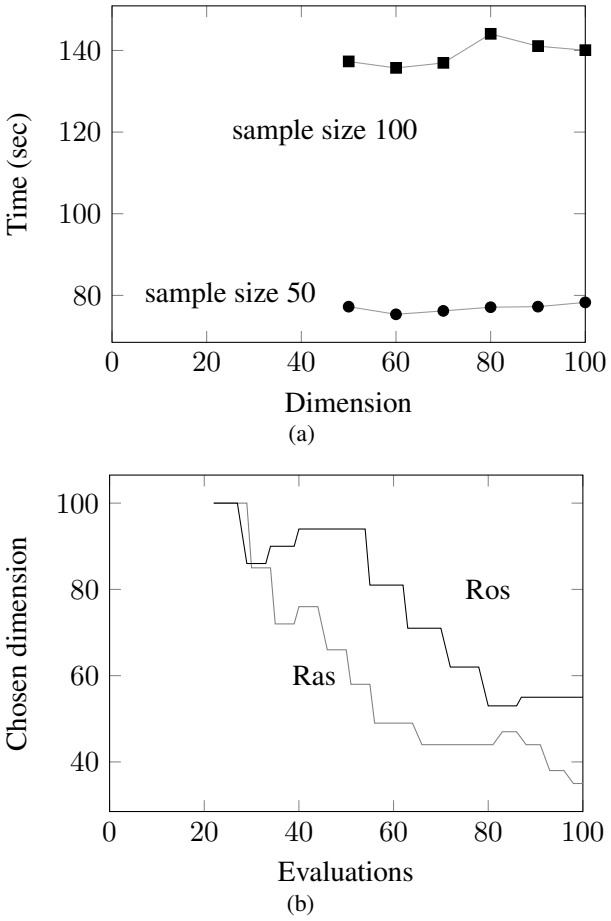

Figure 2: Performance examples of the subset selection approach. (a) Time dependency on dimension and sample size for a single data set. (b) Variation of operating dimension during the search (for 100D Rastrigin and Rosenbrock functions).

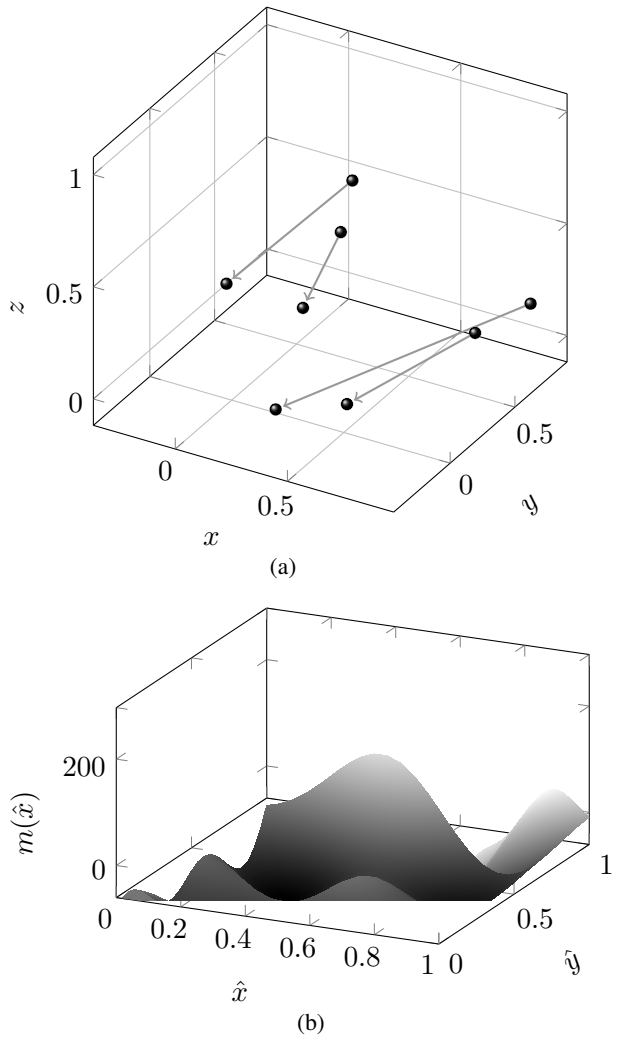

Figure 3: (a) A Sammon mapping from 3D to 2D preserves the proximity relations (adjacent points in 3D are mapped to adjacent ones in 2D and vice-versa). (b) A Kriging metamodel based on data projected from 5D to 2D preserves the landscape multimodality.

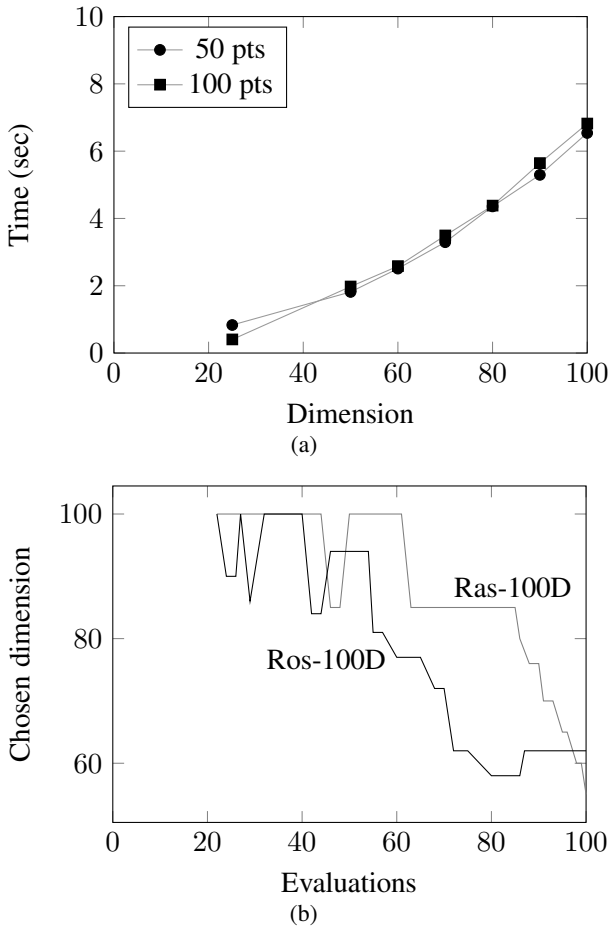

Figure 4: Performance examples for the Sammon topological mapping approach. (a) The time dependency on dimension and sample size for a single data set. (b) The variation of the chosen dimension during the optimization search with a 100D Rastrigin and Rosenbrock functions.

A key element in the mapping-based approach is how to accurately project vectors from high-to-low and low-to-high dimensions, and for this two alternatives were considered: a) Nearest-neighbour interpolation (NNI): A low-dimensional vector is found in the TR step and is associated with a high-dimensional one, and the nearest-neighbour to the latter is evaluated with the the true expensive function. b) Kriging interpolation: An interpolant is trained with the high-dimensional vectors as input and low-dimensional as responses for a high-to-low mapping, and vice versa for low-to-high mapping. To improve the mapping accuracy in both methods the training cache was supplemented with a LHD sample of 100 high-dimensional vectors (without evaluating them) and their corresponding low-dimensional vectors. The accuracy of the two methods was compared by mapping the full set of evaluated vectors from high to low dimension, holding-out one high-dimensional vector and retraining the mapping, and lastly mapping the hold-out low-dimensional vector back to high-dimension. The procedure was repeated for 20 vectors and over 10 re-runs. Figure 5 shows the accuracy comparison from which it follows that the Kriging-based approach achieved an accuracy which was an order of magnitude or more better than that of the NN one and hence it was used in the numerical tests.

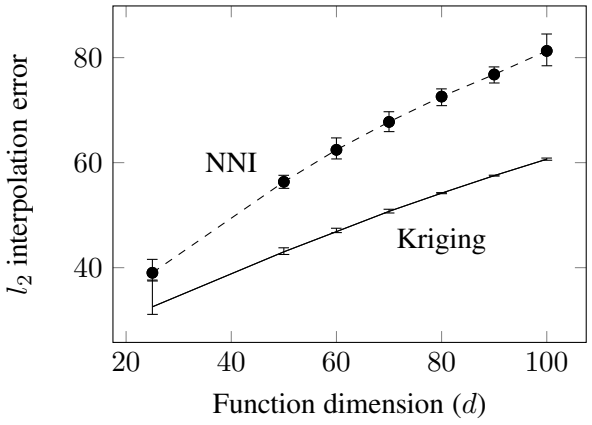

Figure 5: Comparison of l_2 error in low-to-high projected vectors.

5 Performance Analysis

5.1 Parameter sensitivity analysis

As detailed in Section 4 the dimensionality reduction procedure relies on two user defined parameters: a) the number of iterations between successive dimension updates (n), and b) the dimension change factor (r). Analysis of these parameters was performed based on a Design of Experiments approach [3] such that each parameter was tested in both low (L) and high (H) settings in all possible combinations. For n the low and high settings are 5 and 10, respectively, and for r 1.1 and 1.5, respectively. Tests were performed with the Rastrigin and Rosenbrock function in dimension 50. Table 1 gives the mean objective function values obtained (repeated trials) with each design and its ranking. It follows that the (LL) design was the best performing one, that is, updating the dimension every $n = 5$ iterations and reduction factor of $r = 1.1$.

5.2 Mathematical test functions

To evaluate the effectiveness of the proposed framework it was applied to five established test functions: Ackley, Rastrigin, Rosenbrock, Schwefel 2.13 and Weierstrass [17], as detailed in Table 2, in dimension 50 and 100. In all cases the limit of function evaluations was 100 and to obtain statistically significant results 30 runs were repeated per function.

Table 3 gives the test results from which it follows:

- The variable selection (VS) variant performed best in two cases (Schwefel 100 and Weierstrass 100) and second best in five tests (Ackley 50, Rastrigin 50 and 100, Schwefel 50 and Weierstrass 50). This indicates that it performed better in the higher dimensional settings. In the two cases it performed best its was slightly better than the baseline method.

- The Sammon mapping (SM) variant performed best in seven cases (Ackley 50, Rastrigin 50, Rosenbrock 50, Weierstrass 50, Ackley 100, Rastrigin 100 and Rosenbrock 100), which indicates that it performed well over a range of functions and dimensions. In six cases the SM variant performed significantly better than the baseline algorithm and the VS variant.

- The VS variant typically had a higher standard-deviation compared to the SM one which indicates that it was more affected by the sample.

- The baseline framework without any dimensionality reduction outperformed the two variants only in the Schwefel 50 case.

Overall the SM variant was the best performing variant followed by the VS one. The former was less affected by the sample and appeared to handle non-separable functions well, presumably since it projects the entire high-dimensional set while VS does not. It also performed well in 6 cases where it obtained final solutions much better than the other methods. In contrast VS worked well with the difficult functions (the 100D Schwefel and Weierstrass). Lastly, both variants improved the search effectiveness with respect to the baseline framework without dimensionality reduction.

5.3 Airfoil Shape Optimization

The tests also included an engineering problem of shape optimization in which the goal is to find an airfoil shape which would maximize the ratio of lift coefficient c_l to drag (friction) coefficient c_d at some specified flight conditions [18]. To ensure structural integrity the airfoil maximum thickness had to be no less than a threshold value $t^\star$= 0.1. The flight conditions set were typical for a commercial jet aircraft: an altitude of 30 kft, a cruising speed of 70% of the speed of sound, and an angle of attack of 2°. A minimization problem formulation was chosen and accordingly the objective function used was

$$f = -\frac{c_l}{c_d} + |\frac{c_l}{c_d}| \cdot \frac{\max\{t^\star - t\,,\,0\}}{t^\star} \qquad (2)$$

so that a penalty was added for airfoils which violated the thickness constraint.

Lift and drag coefficients were obtained with the Xfoil analysis code [19] and each airfoil evaluation

Table 1: Parameter sensitivity analysis

	Rastrigin				Rosenbrock				
Design	VS	Rank	SM	Rank	VS	Rank	SM	Rank	Total
LL	5.492e+02	1	2.624e+03	4	2.420e+02	2	4.798e+01	1	**08**
LH	5.629e+02	2	1.234e+03	2	5.267e+02	4	7.383e+02	3	11
HL	5.644e+02	3	2.333e+03	3	3.527e+01	1	5.312e+01	2	09
HH	6.164e+02	4	1.227e+03	1	4.689e+02	3	1.149e+03	4	12

VS: variable selection, SM: Sammon mapping. A lower rank is better.

Table 2: Mathematical test functions.

Fun.	Definition	Domain
Ack	$-20\exp\left(-0.2\sqrt{\frac{x_i^2}{d}}\right)$ $-\exp\left(\frac{\sum_{i=1}^{d}\cos(2\pi x_i)}{d}\right)+20+e$	$[-32,32]^d$
Gri	$\sum_{i=1}^{d}\{x_i^2/4000\}-\prod_{i=1}^{d}\{\cos\left(x_i/\sqrt{i}\right)\}+1$	$[-100,100]^d$
Ras	$\sum_{i=1}^{d}\left\{x_i^2-10\cos(2\pi x_i)+10\right\}$	$[-5,5]^d$
Ros	$\sum_{i=1}^{d-1}\left\{100(x_i^2-x_{i+1})^2+(x_i-1)^2\right\}$	$[-10,10]^d$
Sch	$\sum_{i=1}^{d}\{\sum_{j=1}^{d}[(a_{i,j}\sin(\alpha_j)+b_{i,j}\cos(\alpha_j))-(a_{i,j}\sin(x_j)+b_{i,j}\cos(x_j))]\}^2$	$[-\pi,\pi]^d$
Wei	$\sum_{i=1}^{d}\{\sum_{k=0}^{20}0.5^k\cos(2\pi 3^k(x_i+0.5))\}-d\sum_{k=0}^{20}0.5^k\cos(\pi 3^k)$	$[-0.5,0.5]^d$

Ack:Ackley, Gri:Griewank, Ras:Rastrigin
Ros:Rosenbrock, Sch:Schwefel 3.12, Wei:Weierstrass

required 10–30 seconds on a desktop computer. Airfoils were defined by the Hicks-Henne method [20] which combines a baseline airfoil shape with shape functions [21]

$$b_i(x)=\left[\sin\left(\pi x^{\frac{\log(0.5)}{\log(t_1(x))}}\right)\right]^{t_2} \tag{3}$$

where

$$t_1(x)=i/(n-1) \tag{4}$$

determines the location of the bump peak and $t_2=3$ determines the bump width. Accordingly the definition of the upper and lower airfoil curves was

$$y=y_b+\sum_{i=1}^{l}\alpha_i b_i(x) \tag{5}$$

where y_b is the baseline upper/lower curve (NACA0012 symmetric airfoil) and $\alpha_i \in [-0.01,0.01]$ are weights which are the variables whose optimal values need to be found. Figure 6 shows the parameterization layout.

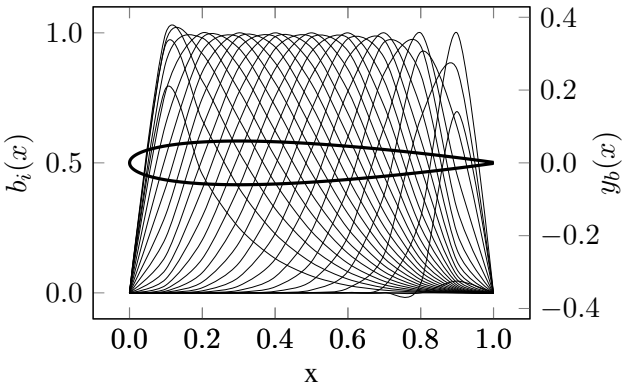

Figure 6: The baseline NACA0012 airfoil $y_b(x)$ and the Hicks-Henne basis functions $b_i(x)$.

Tests were performed in dimension 50 and 100, namely, using 50 or 100 shape functions, while the test setup followed that of Section 5.2. Table 4 provides the resultant test statistics from it follows that the VS variant outperformed both the SM variant and the baseline algorithm. These results are attributed to the intrinsic structure of the problem: a good performing airfoil can be obtained by using a relatively small number of shape functions, namely, by using a small subset of the original variables.

6 Conclusion

High-dimensional and computationally expensive simulation-driven problems can be challenging to solve due to the enormous search space and limited number of function evaluations. In these settings classical algorithms may not perform well and accordingly this paper has explored the incorporation of dimensionality reduction methods into the optimization search so that simplified problems can solved instead. The variants studied were based on variable selection or on Sammon mapping. An extensive performance

Table 3: Test statistics–Mathematical test functions

		$d = 50$			$d = 100$		
Function	Statistic	VS	SM	Baseline	VS	SM	Baseline
	Min	1.503e+01	3.374e+00	1.696e+01	1.642e+01	4.116e+00	1.476e+01
	Max	1.693e+01	3.469e+00	1.855e+01	1.849e+01	4.904e+00	2.089e+01
Ackley	Mean	1.585e+01	**3.423e+00**	1.745e+01	1.750e+01	**4.390e+00**	1.708e+01
	SD	8.203e-01	4.756e-02	7.415e-01	8.377e-01	3.502e-01	2.275e+00
	Median	1.560e+01	3.425e+00	1.714e+01	1.752e+01	4.270e+00	1.656e+01
	Min	4.817e+02	1.424e+02	4.890e+02	1.037e+03	2.115e+02	1.032e+03
	Max	6.345e+02	4.788e+02	6.895e+02	1.239e+03	5.509e+02	1.471e+03
Rastrigin	Mean	5.492e+02	**2.420e+02**	5.772e+02	1.172e+03	**4.135e+02**	1.260e+03
	SD	6.109e+01	1.594e+02	7.985e+01	8.078e+01	1.457e+02	1.809e+02
	Median	5.377e+02	1.734e+02	5.800e+02	1.211e+03	4.458e+02	1.233e+03
	Min	1.435e+03	4.698e+01	8.471e+02	3.626e+03	1.029e+02	3.083e+03
	Max	4.086e+03	4.952e+01	1.274e+03	1.975e+04	1.291e+02	5.277e+03
Rosenbrock	Mean	2.624e+03	**4.798e+01**	1.025e+03	9.008e+03	**1.133e+02**	3.888e+03
	SD	9.518e+02	1.225e+00	1.700e+02	6.730e+03	1.224e+01	9.382e+02
	Median	2.470e+03	4.770e+01	1.028e+03	6.894e+03	1.106e+02	3.558e+03
	Min	2.891e+06	6.737e+06	2.539e+06	1.760e+07	3.162e+07	1.683e+07
	Max	5.272e+06	8.258e+06	4.207e+06	2.398e+07	4.426e+07	2.323e+07
Schwefel	Mean	**4.103e+06**	7.634e+06	3.021e+06	**2.039e+07**	3.795e+07	2.041e+07
	SD	1.092e+06	7.966e+05	7.974e+05	2.551e+06	5.225e+06	2.303e+06
	Median	3.948e+06	7.906e+06	2.669e+06	2.004e+07	3.847e+07	2.068e+07
	Min	-5.374e+01	-6.041e+01	-4.834e+01	-1.209e+02	-9.540e+01	-1.070e+02
	Max	-4.901e+01	-5.047e+01	-4.534e+01	-9.452e+01	-8.522e+01	-9.422e+01
Weierstrass	Mean	-5.151e+01	**-5.402e+01**	-4.728e+01	**-1.022e+02**	-8.969e+01	-1.006e+02
	SD	1.962e+00	4.588e+00	1.331e+00	1.067e+01	3.765e+00	5.975e+00
	Median	-5.089e+01	-5.261e+01	-4.771e+01	-9.858e+01	-8.878e+01	-9.747e+01

VS: Variable selection, SM:Sammon mapping

Table 4: Test statistics–Airfoil problem

		$d = 50$			$d = 100$		
Function	Statistic	VS	SM	Baseline	VS	SM	Baseline
	Min	-1.809e+02	-1.788e+02	-1.725e+02	-1.675e+02	-1.331e+02	-1.403e+02
	Max	-1.188e+02	-1.233e+02	-1.149e+02	-1.263e+02	-7.038e+01	-1.380e+02
airfoil	Mean	**-1.488e+02**	-1.477e+02	-1.473e+02	**-1.466e+02**	-9.372e+01	-1.392e+02
	SD	2.485e+01	2.005e+01	2.415e+01	1.550e+01	2.780e+01	1.643e+00
	Median	-1.389e+02	-1.433e+02	-1.509e+02	-1.496e+02	-8.571e+01	-1.392e+02

VS: Variable selection, SM:Sammon mapping

analysis based both on mathematical test functions and an engineering problem showed the effectiveness of the proposed approaches.

A The Kriging Metamodel

The metamodel combines two components: a 'drift' function, which is a global coarse approximation of the true function, and a local correction based on the correlation between the interpolation vectors. Given a set of evaluated vectors, $\vec{x}_i \in \mathbb{R}^d$, $i = 1 \dots n$, the metamodel is trained such that it exactly interpolates the observed values, that is, $m(\vec{x}_i) = f(\vec{x}_i)$, where $m(\vec{x})$ and $f(\vec{x})$ are the metamodel and true objective function, respectively. Using a constant drift function yields the Kriging metamodel as

$$m(\vec{x}) = 1\,, \tag{6}$$

with the drift function β and local correction $\kappa(\vec{x})$. The latter is defined by a stationary Gaussian process with a zero mean and covariance

$$Cov[\kappa(\vec{x})\kappa(\vec{y})] = \sigma^2 c(\vec{\theta}\,, \vec{x}, \vec{y})\,, \tag{7}$$

where $c(\vec{\theta}\,, \vec{x}, \vec{y})$ is a user-prescribed correlation function. A common choice for the latter is the Gaussian correlation function [22], defined as

$$c(\vec{\theta}\,, \vec{x}, \vec{y}) = \prod_{i=1}^{d} \exp\left(-\theta_i \left(x_i - y_i\right)^2\right)\,, \tag{8}$$

and combining it with the constant drift function transforms the metamodel in (6) into

$$m(\vec{x}) = \hat{\beta} + \vec{r}(\vec{x})^T R^{-1}(\vec{f} - \vec{1}\hat{\beta})\,. \tag{9}$$

Here, $\hat{\beta}$ is the estimated drift coefficient, R is the symmetric matrix of correlations between all interpolation vectors, $\vec{f}$ is the vector of objective values, and $\vec{1}$ is a vector with all elements equal to 1. $\vec{r}^T$ is the correlation vector between a new vector $\vec{x}$ and the sample vectors, namely,

$$\vec{r}^T = [c(\vec{\theta}, \vec{x}\,, \vec{x}_1), \dots\,, c(\vec{\theta}\,, \vec{x}\,, \vec{x}_n)]\,. \tag{10}$$

The estimated drift coefficient $\hat{\beta}$ and variance $\hat{\sigma}^2$, which are required in Equation (9), are calculated as

$$\hat{\beta} = \left(\vec{1}^T R^{-1} \vec{1}\right)^{-1} \vec{1}^T R^{-1} \vec{f}, \tag{11a}$$

$$\hat{\sigma}^2 = \frac{1}{n}\left[(\vec{f} - \vec{1}\hat{\beta})^T R^{-1}(\vec{f} - \vec{1}\hat{\beta})\right]\,. \tag{11b}$$

Fully defining the metamodel requires the correlation parameters $\vec{\theta}$ which are commonly taken as the maximizers of the metamodel likelihood. This is achieved by minimizing the expression [22]

$$\psi(\boldsymbol{\theta}) = |R|^{1/n}\hat{\sigma}^2\,, \tag{12}$$

which is a function only of $\vec{\theta}$ and the sample data [22, 23].

References:

[1] Y. Tenne and C. K. Goh, Eds., *Computational Intelligence in Expensive Optimization Problems*, ser. Evolutionary Learning and Optimization. Berlin: Springer, 2010, vol. 2. [Online]. Available: http://www.springerlink.com/content/v81864

[2] F. A. C. Viana, R. T. Haftka, and L. T. Watson, "Efficient global optimization algorithm assisted by multiple surrogate technique," *Journal of Global Optimization*, vol. 56, no. 2, pp. 669–689, 2013.

[3] J. Muller and C. A. Shoemaker, "Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems," *Journal of Global Optimization*, vol. 60, no. 2, pp. 123–144, 2014.

[4] Y. Tenne and S. W. Armfield, "A framework for memetic optimization using variable global and local surrogate models," *Journal of Soft Computing*, vol. 13, no. 8, pp. 781–793, 2009.

[5] Z. Zhou, Y.-S. Ong, M.-H. Lim, and B. Lee, "Memetic algorithms using multi-surrogates for computationally expensive optimization problems," *Journal of Soft Computing*, vol. 11, no. 10, pp. 957–971, 2007.

[6] B. Talgorn, S. Le Digabel, and M. Kokkolaras, "Statistical surrogate formulations for simulation-based design optimization," *Journal of Mechanical Design*, vol. 137, no. 2, 2015, paper number MD-14-1128 (Online).

[7] Y. Tenne, "An optimization algorithm employing multiple metamodels and optimizers," *International Journal of Automation and Computing*, vol. 10, no. 3, pp. 227–241, 2013.

[8] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on evolutionary computation*, vol. 6, no. 5, pp. 481–494, 2002.

[9] A. R. Conn and S. Le Digabel, "Use of quadratic models with mesh-adaptive direct search for constrained black-box optimization," *Optimization Methods and Software*, vol. 28, no. 1, pp. 139–158, 2013.

[10] R. G. Regis, "Particle swarm with radial basis function surrogates for expensive black-box optimization," *Journal of Computational Science*, vol. 5, no. 1, pp. 12–23, 2014.

[11] Y. Tenne and S. W. Armfield, "A memetic algorithm using a trust-region derivative-free optimization with quadratic modelling for optimization of expensive and noisy black-box functions," in *Evolutionary Computation in Dynamic and Uncertain Environments*, ser. Studies in Computational Intelligence, S. Yang, Y.-S. Ong, and Y. Jin, Eds. Berlin: Springer-Verlag, 2007, vol. 51, pp. 389–415.

[12] B. G. M. Husslage, G. Rennen, E. R. van Dam, and D. den Hertog, "Space-filling latin hypercube designs for computer experiments," *Optimization and Engineering*, vol. 12, no. 4, pp. 611–630, 2011.

[13] S. Grarton and L. N. Vicente, "A surrogate management framework using rigorous trust-region steps," *Optimization Methods and Software*, vol. 29, no. 1, pp. 10–23, 2014.

[14] R. V. Joseph and Y. Hung, "Orthogonal-maximin latin hypercube designs," *Statistica Sinica*, vol. 18, pp. 171–186, 2008.

[15] S. Arlot, "A survey of cross-validation procedures for model selection," *Statistics Survey*, vol. 4, pp. 40–79, 2010.

[16] A. Flexer, "On the use of self-organizing maps for clustering and visualization," *Intelligent Data Analysis*, vol. 5, no. 5, pp. 373–384, 2001.

[17] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Nanyang Technological University, Singapore and Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India, Technical Report KanGAL 2005005, 2005.

[18] A. Filippone, *Flight Performance of Fixed and Rotary Wing Aircraft*, 1st ed. Amsterdam; Boston; Heidleberg: Elsevier, 2006.

[19] M. Drela and H. Youngren, *XFOIL 6.9 User Primer*, 1st ed., Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2001.

[20] H.-Y. Wu, S. Yang, F. Liu, and H.-M. Tsai, "Comparison of three geometric representations of airfoils for aerodynamic optimization," in *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2003, AIAA 2003-4095.

[21] Y. Tenne, "An adaptive-topology ensemble algorithm for engineering optimization problems," *Optimization and Engineering*, vol. 16, no. 2, pp. 303–334, 2015.

[22] A. I. J. Forrester and A. J. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Science*, vol. 45, no. 1–3, pp. 50–79, 2008.

[23] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, pp. 61–70, 2011.