New Approaches in Simulation-driven Optimization

Y Tenne*
Department of Mechanical and Mechatronic Engineering, Ariel University, Israel

*Corresponding author email: y.tenne@ariel.ac.il

Abstract. This paper presents a work regarding the integration of discriminant functions (classifiers) with search algorithms to tackle the problem of failed simulation runs. The discriminant output is used to guide the search towards better solutions while minimizing adverse effects. The search is managed with a trust-region approach for convergence in the presence of prediction inaccuracies. Numerical evaluations based on engineering problem show that the approach yielded better final results in the mean and median statistics when compared to reference algorithms.

1. Introduction
Numerical simulations are employed in a variety of product design and optimization scenarios. Their application yields an optimization problem which is commonly referred to as a black-box problem since the simulation provides the output values without an analytic inputs-outputs relation being known. Furthermore, each simulation execution may require significant computational resources hence a tight limit is placed on the number of simulations performed.
An additional issue is that the simulation may not produce an output value for some input vectors, namely invalid vectors, thereby wasting computer resources and adversely impacting the search. Some studies have reported encountering such vectors and suggested various heuristic solutions with varied degree of effectiveness. As such this study presents an alternate approach based on discriminant functions whose estimates are used to both find better solutions and avoid invalid vectors. Numerical results with an engineering problem show that the approach obtained good final results. The layout of the paper is as follows: Section 2 gives the background details, Section 3 presents an implementation, and then Section 4 gives a numerical evaluation, while lastly Section 5 concludes the paper.

2. Background
Numerical-simulation based search algorithms often use metamodels (or surrogates), such as Kriging/polynomials/neural networks, to approximate the simulation and to obtain output values with less computational effort [6-8]. Since the metamodel is not likely to be accurate some methodology is required to monitor and improve it during the search [9-11], and one such approach is implemented in this study as described Section 3.
With regards to simulation failures previous solutions in the literature included ignoring such vectors or attaching them a fictitious value which indicates a poorly performing design [4-6,11]. Such approaches ignore the data collected during the search or can lead to poor metamodel accuracy (due to the fictitious values) and would therefore make the search less effective.

3. Improving the Search
To improve the search in the presence of invalid vectors discriminant analysis functions were used so that they predict if a vector is invalid or prior to its evaluation. The functions are trained by using...
previously evaluated vectors and are therefore updated during the search. In this manner data about invalid vectors is retained and the metamodel is not affected.
In the presented implementation the search begins by generating a Latin hypercube sample [7] to obtain spatially distributed vectors. The vectors are then sent to the simulation and recorded. In the main loop which follows a discriminant function is generated based on all the recorded vectors and a Kriging metamodel is generated based only on the valid recorded vectors. The metamodel model is given by

\[ m(x) = \beta + \kappa(x) \]  

where the location correction \( \kappa(x) \) is a Gaussian process with a zero mean and a covariance

\[ \text{Cov}[\kappa(x_1), \kappa(x_2)] = \sigma^2 c(\theta, x_1, x_2) \]  

\[ c(\theta, x_1, x_2) = \prod_{i=1}^{d} \exp(-\theta_i (x_{1,i} - x_{2,i})^2) \]  

The overall metamodel expression is then

\[ m(x) = \hat{\beta} + r(x)^TR^{-1}(f - 1\hat{\beta}) \]  

where \( \hat{\beta} \) is an estimated coefficient, \( R \) is the vector correlations matrix, \( f \) is the observations vector, \( 1 \) is a unity vector, while \( r(x)^T \) are the correlations new and existing vectors correlations.

After a metmodel has been generated a discriminant function is selected among three variants:
- \( k \) nearest-neighbours (\( k \)-NN): A vector is assigned to the group of the closest \( k \) vector. In the tests \( k=3 \) was used.
- Linear discriminant analysis (LDA): A vector is assigned to a group based on its projection on a plane which is based on the variance between vectors and the variance between classes.
- Support vector machines (SVM): A vector is assigned a group based on its projection on separating plane in a higher-dimensional.

Selection was based on training-testing evaluation of the recorded vectors with an 80-20 division ratio.

In the step which follows a constrained region search was performed [11,13-15] to handle metamodel inaccuracy. A new optimum was searched in the constrained region

\[ \tau = \|x - x_b\|_2 \]  

which is centered around the current best vector \( x_b \). The search was conducted with a differential evolution (DE) algorithm [16] and followed by a gradient sequential programming local search to improve the solutions. The objective values generated during the search were based on the following relation

\[ \hat{m}(x) = \begin{cases} m(x) & \text{if the discriminant function predicts } x \text{ is valid} \\ p & \text{if it predicts } x \text{ is invalid} \end{cases} \]  

where \( m(x) \) is the metamodel output and \( p \) is a penalty value. This way the penalty was passed only for vectors estimated to be invalid but otherwise the metamodel output was passed. In this way the data about invalid vectors was retained but metamodel was not affected.

The new vector found was then evaluated with the true (expensive) function. If the new vector was better than the current best then the constrained region was doubled. Otherwise its was halved (if there were enough vectors in the region) or a new vector was evaluated in the region to improve the metamodel. The search was stopped based on region size or maximum number of simulations runs. Figure 1 describes the complete algorithm.
4. Numerical Tests
For evaluations the algorithm described was applied to an engineering problem of airfoil shape optimization since it uses a simulation and has invalid vectors. The problem requires finding an airfoil which maximizes the lift to drag ratio while having a thickness $t$ of at least 10% of the chord length. As such the objective function used was

$$ f = -\frac{c_l}{c_d} + \Omega, \quad \Omega = \begin{cases} \frac{0.1}{t} \cdot \frac{c_l}{c_d} & \text{if } t < 0.1 \\ 0 & \text{otherwise} \end{cases} $$

(7)

where $c_l, c_d$ are the forces (lift and drag) coefficients and $\Omega$ is a penalty the thickness requirement. The functional airfoil description was [17]

$$ y = y_b + \sum_{i=1}^{N} \theta_i \tau_i(x), \quad \tau_i(x) = \sin\left(\pi x \log_{10}(i/(k+1))\right), \quad i = 1 \ldots k $$

(8)

where $y_b$ is a base airfoil (NACA series 0012), $\theta_i \in [0,1]$ are variables, $\tau_i(x)$ are the shape terms, and $k$ is the number of terms. Aerodynamic forces were calculated with the Xfoil numerical subsonic airfoil simulation [18]. Figure 2 shows the problem parameters.
In terms of invalid vectors - numerical tests were performed from which it was observed that their prevalence was mainly affected by the attack angle parameter shown in Figure 3. To ensure there will be numerous invalid vectors during the search the flight settings were set to $\alpha = 20^\circ, 30^\circ, 40^\circ$ and with a flight speed of Mach=0.75 (75% of the speed of sound) with an altitude of 32 Kft. The search algorithm itself was implemented in the Octave scientific programming language.

The algorithm was also tested against two reference algorithms: i) Ratle's algorithm which uses an evolutionary algorithm and a Kriging metamodel with regular updates [19], and ii) the algorithm by Emmerich et al. which uses a covariance matrix adaption evolutionary strategies algorithm and the expected improvement approach [6]. The number of simulation runs was limited to 200 and the initial sample contained 20 vectors, while 30 repeats were done per scenario. Table 1 summarises the obtained statistics.

### Table 1. Statistics.

| Attack angle [deg] | Statistic | Implemented | Ratle | Emmerich et al. |
|--------------------|-----------|-------------|-------|-----------------|
| 20                 | Mean      | -9.031      | -6.802| -8.041          |
|                    | SD        | 1.214       | 0.661 | 2.197           |
|                    | Median    | -11.105     | -6.732| -8.009          |
|                    | Best      | -10.002     | -7.801| -10.050         |
|                    | Worst     | -7.109      | -5.413| -5.417          |
| 30                 | Mean      | -4.309      | -2.912| -3.976          |
|                    | SD        | 0.036       | 0.032 | 0.049           |
|                    | Median    | -4.103      | -2.946| -3.959          |
|                    | Best      | -5.311      | -3.181| -5.092          |
|                    | Worst     | -2.959      | -2.817| -2.891          |
| 40                 | Mean      | -2.928      | -2.489| -2.815          |
|                    | SD        | 0.032       | 0.046 | 0.041           |
|                    | Median    | -2.912      | -2.509| -2.811          |
|                    | Best      | -2.956      | -2.671| -2.918          |
|                    | Worst     | -2.791      | -2.125| -2.725          |

SD: Standard Deviation. The best mean and median are emphasized. Smaller results are better.

It follows that the implemented algorithm obtained better final results in comparison to the reference algorithms which points to the effectiveness of using the discriminant functions in problems which contain invalid vectors.

### 5. Conclusion

Engineering problems often use simulations to obtain function values. In such cases invalid vectors (where the simulation fails) may be encountered. To effectively handle this issue a framework was presented which uses discriminant functions to efficiently guide the search away from invalid vectors. Numerical results based on a simulation-based problem show that this approach was effective and
obtained better mean and median statistics when compared to reference algorithms. It also avoided issues such as data discardment or metamodel alterations which are associated with other approaches.

References
[1] Tenne Y and Goh C K 2010 Computational Intelligence in Expensive Optimization Problems, vol. 2 of Evolutionary Learning and Optimization (Berlin: Springer).
[2] Dinh X and Anh K K 2017 Radial basis function neural network based adaptive fast nonsingular terminal sliding mode controller for piezo positioning stage, International Journal of Control, Automation, and Systems, vol. 15, no. 6, p. 2892–2905.
[3] Grarton S and Vicente L N 2014 A surrogate management framework using rigorous trust-region steps, Optimization Methods and Software, vol. 29, no. 1, p. 10–23.
[4] Rasheed K and Hirsh H and Gelsey A 1997 A genetic algorithm for continuous design space search, Artificial Intelligence in Engineering, vol. 11, p. 295–305.
[5] Büche D and Schraudolph N N and Koumoutsakos P 2005 Accelerating evolutionary algorithms with Gaussian process fitness function models,” IEEE Transactions on Systems, Man, and Cybernetics–Part C, vol. 35, no. 2, p. 183–194.
[6] Emmerich M T M and Giotis A and Özdemir M and Bäck T and Giannakoglou K C 2002 Proc 7th International Conference on Parallel Problem Solving from Nature–PPSN VII (Springer Berlin), p 361–370.
[7] Talgorn B and Le Digabel S and Kokkolaras M 2015 Statistical surrogate formulations for simulation-based design optimization Journal of Mechanical Design, vol. 137, no. 2, Paper number MD-14-1128 (Online).
[8] Jin Y 2011 Surrogate-assisted evolutionary computation: Recent advances and future challenges Swarm and Evolutionary Computation, vol. 1, p. 61–70.
[9] Regis R G and Shoemaker C A 2013 A quasi-multistart framework for global optimization of expensive functions using response surface models, Journal of Global Optimization, vol. 56, p. 1719—1753.
[10] Tenne Y 2013 An optimization algorithm employing multiple metamodels and optimizers, International Journal of Automation and Computing, vol. 10, no. 3, p. 227–241.
[11] Conn A R and Scheinberg K Vicente L N 2009 Introduction to Derivative-Free Optimization (MPS-Siam Series on Optimization, Philadelphia, Pennsylvania: SIAM).
[12] Yang X S 2019, Introduction to Algorithms for Data Mining and Machine Learning, Elsevier.
[13] Regis R G and Shoemaker C A 2008, ORBIT: Optimization by radial basis function interpolation in trust-regions, vol. 30, no. 6, p. 3197–3219.
[14] Muller J and Shoemaker C A 2014 Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive blackbox global optimization problems, Journal of Global Optimization, vol. 60, no. 2, p. 123–144.
[15] Wortmann T and Costa A and Nannicini G and Schroeper T 2015, Advantages of surrogate models for architectural design optimization, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol. 29, no. 4, p. 471–481.
[16] Ernesto M and Neri F 2010 A Memehtic Differential Evolution Approach in Noisy Optimization, Journal of Soft Computing, vol. 2, p. 111-135.
[17] Hicks R M and Henne P A 1978 Wing design by numerical optimization, Journal of Aircraft, vol. 15, no. 7, p. 407–412.
[18] Drela M and H. Youngren 2001, XFOIL 6.9 User Primer. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA.
[19] Ratle A 1999 Proc IEEE Congress on Evolutionary Computation–CEC 1999, (New Jersey IEEE) p. 2078–2085.