

Multipoint Approximation Method for Large Scale Design Optimization Problems Under Uncertainty

Vassili V. Toropov*

*Professor of Aerospace Engineering, School of Engineering and Materials Science
Queen Mary University of London, London, UK
Email: v.v.toropov@qmul.ac.uk*

*Senior Research Fellow, Institute of Information Technology, Mathematics and Mechanics
Lobachevsky University of Nizhny Novgorod, Russia*

Yury M. Korolev

*Research Fellow, Institute of Mathematics and Image Computing
University of Lübeck, Germany
Email: yury.korolev@mic.uni-luebeck.de*

Victor P. Gergel, Konstantin A. Barkalov

*Institute of Information Technology, Mathematics and Mechanics
Lobachevsky University of Nizhny Novgorod, Russia*

Summary

TBD

Keywords: TBD

Introduction

In industrial design optimization problems, random inaccuracies of the production process may lead to the designer's inability to fully control the design variables. Therefore, there can be a discrepancy between the products *as designed* and *as produced* that results in actual performance being different from the performance expected at the design stage. Possible discrepancy between the design and the manufactured product becomes even more important if the latter violates some critical constraints, e.g. safety regulations.

In addition, the system's performance may depend on parameters of random nature that are beyond the designer's control, e.g. the operation conditions. Therefore, accounting for both of these types of uncertainties is crucial for achieving robust and reliable performance.

Optimisation under uncertainty deals with responses $F(\mathbf{x}, \mathbf{y})$ that depend on deterministic design variables \mathbf{x} and random 'environmental' variables \mathbf{y} . Uncertainty in the design variables can also be modelled by introducing additional random variables, e.g. as additive or multiplicative noise. In this case, the responses $F(\mathbf{x}, \mathbf{y})$ receive as input the sum $\mathbf{x} + \mathbf{u}$ (if the noise is additive),

where \mathbf{u} is a random variable with a known probability distribution, yielding responses depending on \mathbf{x} , \mathbf{u} and \mathbf{y} . Randomness in \mathbf{u} and \mathbf{y} induces randomness in the responses $F(\mathbf{x} + \mathbf{u}, \mathbf{y})$. To convert them into deterministic functions, that can be optimised, a mapping R , called the *risk measure*¹ or *robustness measure*,² is applied to the responses. The result is then a deterministic function of the design variables: $\tilde{F}(\mathbf{x}) = R(F(\mathbf{x} + \mathbf{u}, \mathbf{y}))$. The choice of a particular risk measure R reflects the designer's attitude towards the risk of sub-optimal performance or constraint violation. For example, R can be a combination of the mean and standard deviation of $F(\mathbf{x} + \mathbf{u}, \mathbf{y})$ with respect to the joint probability distribution of \mathbf{u} and \mathbf{y} or a quantile of the distribution of $F(\mathbf{x} + \mathbf{u}, \mathbf{y})$, i.e. such a value that the random variable of $F(\mathbf{x} + \mathbf{u}, \mathbf{y})$ exceeds it with a given, typically small, probability.

Computations of the risk measures typically involve evaluation of high-dimensional integrals, which becomes a challenging task if the number of design variables and/or environmental variables is large. The problem becomes even more difficult if the original responses are computationally expensive, which is the case, for example, in computational fluid dynamics (CFD), where one function evaluation may take hours of even days. Therefore,

direct computation of the risk measures using original responses becomes infeasible and therefore approximations of responses, also referred to as metamodels, need to be used.

In problems with a large (in the order of hundreds) number of design variables, the multipoint approximation method (MAM³⁻⁵) proved to be efficient, e.g. in turbomachinery applications.⁶⁻⁸ This method is an iterative optimization technique based on mid-range approximations built in trust regions. A trust region is a sub-domain of the design space in which a set of design points, produced according to a small-scale design of experiments (DoE), are evaluated. These and a subset of previously evaluated design points are used to build metamodels of the objective and constraint functions that are considered to be valid within a current trust region. The trust region will then translate and change size as optimization progresses. The trust region strategy has gone through several stages of development to account for the presence of numerical noise in the response function values^{9,10} and occasional simulation failures.¹¹ The mid-range approximations used in the trust regions, as originally suggested in Ref.³ for structural optimization problems, are intrinsically linear functions (i.e. nonlinear functions that can be led to a linear form by a simple transformation) for individual sub-structures, and assembly of them for the whole structure. This was enhanced by the use of gradient-assisted metamodels,⁵ use of simplified numerical models that is also termed a multi-fidelity approach,¹² and the use of analytical models derived by Genetic Programming.¹³ One of the recent developments¹⁴ involved the use of approximation assemblies, i.e. a two stage approximation building process that is conceptually similar to the original one used in Ref.³ but is free from the limitation that lower level approximations are linked to individual substructures.

This paper presents a new development in the Multipoint Approximation Method that makes it capable of handling problems with uncertainty in design variables as well as in additional ‘environmental’ variables. The approach relies on approximations built in the combined space of design variables and environmental variables, and subsequent application of a risk measure and optimization with respect to the deterministic design variables, all within the iterative trust-region-based framework of MAM.

The Multipoint Approximation Method

It is useful to start with a brief description of the deterministic version of MAM. A typical formulation of a constrained optimization problem that MAM works with is as follows:

$$\begin{aligned} \min_{a_i \leq x_i \leq b_i} F_0(\mathbf{x}) \\ \text{s.t. } F_j(\mathbf{x}) \leq 1, j = 1, \dots, M, \end{aligned} \quad (1)$$

where \mathbf{x} is the vector of design variables, \mathbf{a} and \mathbf{b} are the lower and upper bounds for the design variables, respectively, $F_0(\mathbf{x})$ is the objective function, and $F_j(\mathbf{x})$

are the constraints. The numbers of design variables and constraints are n and M , respectively. MAM attempts to solve this problem by using approximations of the objective function and constraints in a series of trust regions. The trust region strategy seeks to zoom in on the region where the constrained minimum is achieved. It aims at finding a trust region that is sufficiently small for the approximations to be of sufficiently good quality to improve the design, and that contains the point of the constrained minimum as its interior point. The main loop of the MAM is organized as follows.

Algorithm 1 (deterministic MAM).

1. Initialization: choose a starting point \mathbf{x}^0 and initial trust region $[\mathbf{a}^0, \mathbf{b}^0]$ such that $\mathbf{x}^0 \in [\mathbf{a}^0, \mathbf{b}^0]$.
2. At the k -th iteration the current approximation to the constrained minimum is \mathbf{x}^k , the current trust region is $[\mathbf{a}^k, \mathbf{b}^k] \subset [\mathbf{a}^0, \mathbf{b}^0]$.

- (a) Design of Experiments (DoE): a set of points $\mathbf{x}_k^i \in [\mathbf{a}^k, \mathbf{b}^k]$ is chosen to be used for approximation building. Responses are evaluated at the DoE points and approximations are built using the obtained values. Currently, the pool of approximation methods available in MAM consists of a metamodel assemblies¹⁵ and the moving least-squares metamodels¹⁶⁻¹⁹. Other metamodel types could be used as well.

Denote the approximate objective function and constraints by $\tilde{F}_0(\mathbf{x})$ and $\tilde{F}_j(\mathbf{x})$, respectively.

- (b) The original optimization problem (1) is replaced by the following problem:

$$\begin{aligned} \min_{a_i^k \leq x_i^k \leq b_i^k} \tilde{F}_0(\mathbf{x}) \\ \text{s.t. } \tilde{F}_j(\mathbf{x}) \leq 1, j = 1, \dots, M, \end{aligned} \quad (2)$$

The approximate problem (2) is solved using Sequential Quadratic Programming (SQP) and the solution of this problem determines the centre of the next trust region.

- (c) The size of the next trust region is determined depending on the quality of approximations at the previous iteration, on the history of the points \mathbf{x}^k , and on the size of the current trust region¹⁰.
- (d) The termination criterion is checked (it is a part of the trust region strategy and depends on the position of the point \mathbf{x}^{k+1} in the current trust region, the size of the current trust region and the quality of approximations). If the termination criterion is satisfied, the algorithm proceeds to step 3. Otherwise, it returns to step 2.

3. Optimization terminates. The obtained approximation to the solution of the problem (1) is \mathbf{x}^{k+1} .

Optimization Under Uncertainty

The presence of uncertainties in the responses requires certain reformulations in the problem statement (1). Informally, the optimization problem with uncertainty can be written as follows:

$$\begin{aligned} \min_{a_i \leq x_i \leq b_i} F_0(\mathbf{x} + \mathbf{u}, \mathbf{y}) \\ \text{s.t. } F_j(\mathbf{x} + \mathbf{u}, \mathbf{y}) \leq 1, j = 1, \dots, M, \end{aligned} \quad (3)$$

where the random variable \mathbf{u} represents (additive) noise in the design variables and \mathbf{y} is the environmental variable, representing input to the responses that cannot be influenced by the designer. This problem statement is not precise since the random responses $F_0(\mathbf{x} + \mathbf{u}, \mathbf{y})$, $F_j(\mathbf{x} + \mathbf{u}, \mathbf{y})$ cannot be directly optimized. Therefore, a risk measure R needs to be applied to the objective and the constraints:

$$\begin{aligned} \min_{a_i \leq x_i \leq b_i} R_{\mathbf{u}, \mathbf{y}}(F_0)(\mathbf{x}) \\ \text{s.t. } R_{\mathbf{u}, \mathbf{y}}(F_j)(\mathbf{x}) \leq 1, j = 1, \dots, M, \end{aligned} \quad (4)$$

where the subscript in $R_{\mathbf{u}, \mathbf{y}}$ indicates that the variables \mathbf{u} and \mathbf{y} ‘collapse’ when the risk measure is applied and the result only depends on \mathbf{x} . The choices of risk measure will be discussed in Section 4. Here the changes in the Algorithm 1 that are necessary to solve the problem (4) are discussed.

First, it needs to be decided on whether to build metamodels for the risk measure $R_{\mathbf{u}, \mathbf{y}}(F_j)(\mathbf{x})$ or for the original responses $F_j(\mathbf{x} + \mathbf{u}, \mathbf{y})$. The first option may seem attractive due to the simplicity of its implementation in Algorithm 1 but the deciding argument has to be that of the computational cost associated with the evaluation of either function. On the one hand, one evaluation of the function $R_{\mathbf{u}, \mathbf{y}}(F_j)(\mathbf{x})$ involves computing one or several integrals of the original response $F_j(\mathbf{x} + \mathbf{u}, \mathbf{y})$ over the distribution of \mathbf{u} and \mathbf{y} , which requires many evaluations of the original response. On the other hand, the dimensionality of the argument of $R_{\mathbf{u}, \mathbf{y}}(F_j)(\mathbf{x})$ is less than that of the argument of $F_j(\mathbf{x} + \mathbf{u}, \mathbf{y})$, therefore, in principle, fewer function evaluations are needed to approximate the former than the latter. In practice, this may be a matter of a trade-off between the two abovementioned effects that depends on the proportion between the numbers of design variables n and the environmental variables n_e and on the complexity of the integration method and metamodeling method used. If the number of design variables is significantly larger than the number of environmental variables, which is often the case and which is assumed in this paper, building metamodels for the original responses $F_j(\mathbf{x} + \mathbf{u}, \mathbf{y})$ is preferred.

This choice requires some changes in the MAM algorithm, since metamodels are now built in a larger space than the one where optimisation is performed in. The changes are summarised below.

Algorithm 2 (MAM under uncertainty).

1. Initialization: choose a starting point $\mathbf{x}_0 \in \mathbb{R}^n$. Choose an initial trust region $[\mathbf{a}^0, \mathbf{b}^0]$ in the combined space of design variables and environmental variables. Ensure

that the initial guess \mathbf{x}_0 belongs to the projection of the initial trust region onto the subspace of design variables. The projection of the initial trust region onto the subspace of environmental variables must cover the region where the realisations of the random variable \mathbf{y} are likely to be. The coordinates of the trust region corresponding to the environmental variables will not be changed in the course of optimisation. Typically, the trust region along the coordinates corresponding to the i -th environmental variable is centred at the mean of the i -th environmental variable and has the width of six of its standard deviations.

2. At the k -th iteration the current approximation to the constrained minimum is \mathbf{x}^k , the current trust region is $[\mathbf{a}^k, \mathbf{b}^k]$.

- (a) Design of Experiments (DoE): a set of points $(\mathbf{x}_i^k, \mathbf{y}_i^k) \in [\mathbf{a}^k, \mathbf{b}^k]$ is chosen that will be used to build approximations. Responses are evaluated at the DoE points and approximations are built using the obtained values. Denote the approximate objective function and constraints by $\tilde{F}_0(\mathbf{x}, \mathbf{y})$ and $\tilde{F}_j(\mathbf{x}, \mathbf{y})$, respectively.
- (b) The original optimization problem is replaced by the following problem:

$$\begin{aligned} \min_{a_i \leq x_i \leq b_i} R_{\mathbf{u}, \mathbf{y}}(\tilde{F}_0)(\mathbf{x}) \\ \text{s.t. } R_{\mathbf{u}, \mathbf{y}}(\tilde{F}_j)(\mathbf{x}) \leq 1, j = 1, \dots, M, \end{aligned} \quad (5)$$

The risk measures are calculated using the metamodels obtained in the previous step. Integrals that arise during this process are calculated with respect to the joint distribution of \mathbf{u} and \mathbf{y} . This distribution is assumed to be fixed. The approximate problem (5) is solved using Sequential Quadratic Programming (SQP) and the centre of the next trust region \mathbf{x}^{k+1} is determined as the solution of this problem. The optimisation runs in a subspace of the space where metamodels are built: metamodels are defined in the combined space of design variables and environmental variables, whilst optimisation runs in the subspace of the design variables.

- (c) The trust region is updated. The size of the trust region along the dimensions corresponding to the environmental variables is unchanged. The size of the next trust region along the dimensions corresponding to the design variables is determined depending on the quality of approximations at the previous iteration, on the history of the points \mathbf{x}^k , and on the size of the current trust region according to the trust region strategy explained in Ref.⁹
- (d) The termination criterion is checked. If the termination criterion is satisfied, the algorithm proceeds to step 3. Otherwise, it returns to step 2

3. Optimization terminates. The obtained approximation to the optimum is \mathbf{x}^{k+1} .

Summarising, the key changes introduced to account for uncertainties are (i) metamodels built in the combined space of design variables and environmental variables; (ii) optimisation performed in a subspace of the space where metamodels are defined; (iii) additive noise in the design variables; (iv) application of risk measures to transform random responses into deterministic functions.

Risk Measures

As it has been already said, a risk measure is a mapping that transforms a random response $F(\mathbf{x} + \mathbf{u}, \mathbf{y})$ into a deterministic function that can be optimized. Next we provide some examples of risk measures^{2,3}.

- $R(F) = \mu(F)$ where $\mu(F)$ is the mean of F . This choice corresponds to optimising average performance of the system. In typical engineering applications, however, this is not sufficient and more conservative risk measures need to be chosen.
- $R(F) = \sup(F)$. This is the most conservative case, in which one is concerned with the worst case performance of the system. The expression $\sup(F)$ stands for the least of all values that are greater than the random response F with probability 1. If the support of the probability density function (PDF) of F is not compact (i.e. F can take arbitrary large values with non-zero probability) then $\sup(F)$ will be infinite.
- $R(F) = \mu(F) + k\sigma(F)$, where $\sigma(F)$ is the standard deviation of F and $k > 0$ is a constant. This risk measure is sometimes referred to as the safety margin. Optimisation using this risk measure is referred to as robust design optimisation^{22,23,24} and reflects the paradigms of a '3 σ -design', a '6 σ -design', etc.
- $R(F) = q_F(\alpha)$, where $q_F(\alpha)$ is the value of the quantile function, or simply the quantile, corresponding to a given probability $\alpha \in (0, 1)$. This choice corresponds to the paradigm of reliability-based design optimization^{25,23,26}. In this approach, one wants to keep the probability of the system failing (i.e. violating the constraint or underperforming, depending on the context) below a given threshold. The value $1 - \alpha$ is the desired probability of failure. Although rather intuitive, this choice has known computational issues which are discussed by Rockafellar and Royset²⁷.

For this study, we chose as risk measure the safety margin $R(F) = \mu(F) + k\sigma(F)$, aiming at covering both robust design optimisation and reliability design optimisation. Let us briefly discuss its computation.

To compute the safety margin risk measure, one needs to compute two integrals:

$$\begin{aligned}\mu(F) &= E[F] = \int F(\mathbf{x} + \mathbf{u}, \mathbf{y}) pdf(\mathbf{u}, \mathbf{y}) d\mathbf{u} d\mathbf{y} \\ \sigma^2(F) &= E[(F - E[F])^2] = \\ &= \int (F(\mathbf{x} + \mathbf{u}, \mathbf{y}) - E[F])^2 pdf(\mathbf{u}, \mathbf{y}) d\mathbf{u} d\mathbf{y}\end{aligned}$$

The dimension of these integrals is $n + n_e$. To compute them efficiently, methods of high-dimensional integration must be employed. Here, quasi-Monte-Carlo integration³⁰ based on Niederreiter sequences³¹ is used.

Numerical Example

The example considered in this study is a classical engineering optimisation problem known as the scalable cantilevered beam³⁴. The engineering object to be optimised is shown in Fig. 1 taken from Ref. 34.

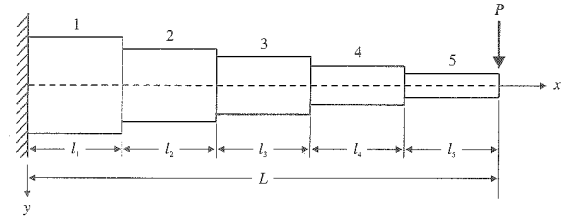


Figure 1: The cantilevered beam

The design variables are the widths b_i and heights h_i of the segments. The number of segments N can be chosen arbitrary. The total length of the beam is 500 cm, the lengths of the segments are $l_i = 500/N$ cm. There are N geometric constraints (the aspect ratios of each block (i.e. heights divided by widths) should not exceed 20) and N constraints on the stress, calculated at the left end of each segment (stresses should not exceed $\bar{\sigma} = 14000 \text{ N/cm}^2$). There is also a constraint on the displacement at the tip, which should not exceed 2.5 cm. The load is $P = 50000 \text{ N}$, the Young's modulus is $E = 2 \cdot 10^7 \text{ N/cm}^2$.

The deflection y_i at the right hand of the i -th segment is given by the following recursive formulas:

$$\begin{aligned}y_0 &= y'_0 = 0, \\ y'_i &= \frac{P \cdot l_i}{E \cdot I_i} \left[L + \frac{l_i}{2} - \sum_{j=1}^i l_j \right] + y'_{i-1}, \\ y_i &= \frac{P \cdot l_i^2}{2E \cdot I_i} \left[L - \sum_{j=1}^i l_j + \frac{2l_i}{3} \right] + y'_{i-1} l_i + y_{i-1}.\end{aligned}$$

The moment of inertia of a segment is $I_i = \frac{b_i h_i^3}{12}$ and the bending moment at the left end is $M_i = P[L + l_i - \sum_{j=1}^i l_j]$. The maximum bending stress in the segment i is then given by the following formula:

$$\sigma_i = \frac{M_i h_i}{2I_i}$$

We are looking for a design with smallest volume $V = \sum_{i=1}^N b_i h_i l_i$. The widths b_i vary from 1.0 to 10.0 cm and the heights h_i from 5.0 to 100.0 cm. The optimisation problem is formulated as follows:

$$\begin{aligned} & \min_{\mathbf{b}, \mathbf{h}} V(\mathbf{b}, \mathbf{h}) \\ & \text{s.t. } 1.0 \leq b_i \leq 10.0, \\ & \quad 5.0 \leq h_i \leq 100.0, \\ & \quad y_N \leq 2/5, \\ & \quad \sigma_i \leq \bar{\sigma} = 14000, \\ & \quad \frac{h_i}{b_i} \leq 20. \end{aligned}$$

With $N = 50$ segments (corresponding to 100 design variables) the SQP solution of this problem is $V = 63704.598 \text{ cm}^3$. MAM obtained the solution $V = 63935.360 \text{ cm}^3$, using 1500 function evaluations (as compared to almost 8000 evaluations used by SQP). The number of points in a trust region used to build approximations was 150. The solution obtained by MAM is very close to the reference solution obtained by SQP, except for the last design variable (the height of the last segment), which indicates that the problem is insensitive to this variable near the optimum, making it hard for the metamodels to capture this dependence. Both SQP and MAM solutions are, however, feasible and differ only slightly in the value of the objective function.

Now let us add uncertainty to the problem, adding Gaussian noise with zero mean and variance 0.1 to each design variable. Using the $\mu(F) + k\sigma(F)$ risk measure, we obtain the following problem statement:

$$\begin{aligned} & \min_{\mathbf{b}, \mathbf{h}} \mu(V(\mathbf{b}, \mathbf{h}) + k\sigma V(\mathbf{b}, \mathbf{h})) \\ & \text{s.t. } 1.0 \leq b_i \leq 10.0, \\ & \quad 5.0 \leq h_i \leq 100.0, \\ & \quad \mu(y_N) + k\sigma(y_N) \leq 2.5, \\ & \quad \mu(\sigma_i) + k\sigma(\sigma_i) \leq \bar{\sigma} = 14000, \\ & \quad \mu\left(\frac{h_i}{b_i}\right) + k\sigma\left(\frac{h_i}{b_i}\right) \leq 20. \end{aligned}$$

The notation σ was used to denote the variance of a response, but it also stands for the maximum bending stresses σ_i , resulting in somewhat confusing notation $\sigma(\sigma_i)$. Hopefully, this explanation eliminates the confusion. Optimising $\mu(F) + k\sigma(F)$ with $k = 3.0$ using SQP on original functions, we obtain the value of the deterministic objective (volume of the structure) at the robust optimum $V = 67598.049 \text{ cm}^3$. To compute $\mu(F)$ and $\sigma(F)$ at each iteration, 1000 points were used. Multiplied by the number of evaluations of the robust objective $\mu(F) + k\sigma(F)$ (approximately 8000), we get a rough estimate of $8 \cdot 10^6$ of the total number of points that SQP used to find this solution.

Solving the same problem using the new version of MAM suitable for problems with uncertainty, we obtain the value of the deterministic objective (volume of the structure) at the robust optimum $V = 67799.335 \text{ cm}^3$. To obtain this result, MAM again used 150 points per

trust region to build metamodels, which resulted in a total of 1500 function evaluations, which is orders of magnitude smaller than the number of points required if no metamodels are used.

The $\mu(F) + k\sigma(F)$ risk measure uses quantities that are estimated in the middle of the distribution — the mean and the standard deviation. On the one hand, this has the advantage that samples drawn directly from the distribution of the random variables \mathbf{u} and \mathbf{y} are well suited to approximately compute this risk measure. On the other hand, probabilistic interpretation of the obtained results in this case is only possible under the normality assumption for the distribution of F ; if the actual distribution of F has heavier tails than the normal one, the actual probability of failure will be higher than the one the system was designed for.

Conclusion

Recent developments in the Multipoint Approximation Method made it capable of solving large scale problems with uncertainty in parameters modelled as additional ('environmental') random variables and uncertainty in the design variables modelled as additive noise. The approach taken combines metamodels built in the combined space of design variables and environmental variables, application of risk measures that map random responses to deterministic functions, and optimisation in a subspace of the space where metamodels are defined, all within the iterative and trust-region-based framework of MAM. Performance is demonstrated on a benchmark example of structural optimisation known as the scalable cantilevered beam.

Acknowledgement

The authors are grateful for the support provided by the Russian Science Foundation, project No. 16-11-10150.

References

- [1] Rockafellar, R. and Uryasev, S. Optimization of conditional value-at-risk. *The Journal of Risk* **2**(3), 21–41 (2000).
- [2] Schillings, C. and Schulz, V. On the influence of robustness measures on shape optimization with stochastic uncertainties. *Optimization and Engineering* **16**(2), 347–386 (2015).
- [3] Toropov, V. Simulation approach to structural optimization. *Structural Optimization* **1**, 37–46 (1989).
- [4] Toropov, V. Multipoint approximation method in optimization problems with expensive function values. In *Computational Systems Analysis*, Sydow, A., editor, 207–212. Elsevier, (1992).
- [5] Toropov, V., Filatov, A., and Polynkin, A. Multiparameter structural optimization using fem and multipoint explicit approximations. *Structural Optimization* **6**, 7–14 (1993).

- [6] Shahpar, S., Polynkin, A., and Toropov, V. Large scale optimization of transonic axial compressor rotor blades. In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and 4th AIAA Multidisciplinary Design Optimization Specialist Conference*, (2008).
- [7] Polynkin, A., Toropov, V., and Shahpar, S. Design optimization of aircraft engine components. In *Proc. of 7th ASMO UK / ISSMO Conf. on Engineering Design Optimization. Process and Product Improvement*, (2008).
- [8] Polynkin, A., Toropov, V., and Shahpar, S. Multidisciplinary optimization of turbomachinery based on metamodel built by genetic programming. In *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (2010).
- [9] van Keulen, F., Toropov, V., and Markine, V. Recent refinements in the multi-point approximation method in conjunction with adaptive mesh refinement. In *Proc. ASME Design Engineering Technical Conferences and Computers in Engineering Conf.*, McCarthy, J., editor, 12 (ASME, Irvine CA, 1996).
- [10] Toropov, V., van Keulen, F., Markine, V., and de Boer, H. Refinements in the multi-point approximation method to reduce the effects of noisy responses. In *6th AIAA/NASA/ISSMO Symp. on Multidisciplinary Analysis and Optimization*, 941–951 (AIAA, Bellevue WA, 1996).
- [11] Toropov, V., Markine, V., and Holden, C. Use of mid-range approximations for optimization problems with functions of domain-dependent calculability. In *3rd ISSMO/UBCAD/UB/AIAA World Congress of Structural and Multidisciplinary Optimization*, (1999).
- [12] Toropov, V. and Markine, V. The use of simplified numerical models as mid-range approximations. In *6th AIAA/NASA/ISSMO Symp. on Multidisciplinary Analysis and Optimization*, (1996).
- [13] Toropov, V. and Alvarez, L. Creation of multipoint approximations using genetic programming. In *Adaptive Computing in Design and Manufacture*, 3rd Int. Conf., Parmee, I., editor, 21–24 (PEDC, Dartington, 1998).
- [14] Polynkin, A. and Toropov, V. Mid-range metamodel assembly building based on linear regression for large scale optimization problems. *Structural and Multidisciplinary Optimization* **45**(4), 515–527 (2012).