



Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs



Highlights

Fast adaptive sampling with operation time control

Journal of Computational Science xxx (xxxx) xxx

A.S. Algasov, S.A. Guda*, V.I. Kolesnikov, V.V. Ilicheva, A.V. Soldatov

- The proposed adaptive sampling algorithm is either accurate or fast.
- The implementation supports constraints on sampling time and parallel sampling.
- A comparative study of approaches to minimization of approximation error is given.
- The source code for the algorithm MinLpE is available on Github.

Graphical abstract and Research highlights will be displayed in online search result lists, the online contents list and the online article, but **will not appear in the article PDF file or print unless it is mentioned in the journal specific style requirement. They are displayed in the proof pdf for review purpose only.**



Fast adaptive sampling with operation time control

A.S. Algasov^a, S.A. Guda^{a,b,*}, V.I. Kolesnikov^c, V.V. Ilicheva^c, A.V. Soldatov^b

^a Institute for Mathematics, Mechanics and Computer Science in the name of I.I. Vorovich, Southern Federal University, Milchakova 8a, 344090, Rostov-on-Don, Russia

^b The Smart Materials Research Institute, Southern Federal University, Sladkova, 178/24, 344090, Rostov-on-Don, Russia

^c Department of Theoretical Mechanics, Rostov State Transport University, Rostov-on-Don, 344038, Russia

ARTICLE INFO

Keywords:

Surrogate models
Metamodels
Adaptive sampling
Experiment design
Exploitation and exploration
Function approximation

ABSTRACT

The rational choice of sampling points is crucial for supervised machine learning algorithms. Adaptive sampling ensures optimal distribution of points in the multidimensional space and provides higher approximation accuracy with respect to the random sampling. Most approaches to the adaptive sampling maximize the function evaluating the compromise between local exploitation and global exploration. However, existing accurate methods suffer from long sampling time, while fast methods are not accurate. Only a few reported methods are available as open source implementations. We introduce a fast adaptive sampling algorithm with accuracy comparable to that of the best-known adaptive sampling methods and demonstrate a comparative study of different approaches to minimization of approximation errors. The method scales linearly with number of sampling points and supports batch generation of new points. The optimization function is the integral norm of an approximation error, the L_p norm, where the norm parameter p regulates sampling, inclining it either towards local exploitation or, conversely, global exploration. The small L_p norm is achieved by reducing the function approximation error and the size of the region with large variation after adding a new sampling point. Our solution is similar to kriging in terms of the choice between local exploitation and global exploration. The difference is in error estimation which depends on the values of the function in sampling points in contrast to homoscedastic variance estimate of kriging. Fast and accurate adaptive sampling is of interest to supervised machine learning tasks such as training a model interatomic potential, quantitative analysis of spectra, self-driving chemical laboratories, and many others. The source code for the new adaptive sampling algorithm MinLpE is available on Github and is distributed under a LGPL license.

1. Introduction

Approximation methods require a representative training sample. However, the affordable size of a sample is limited for many practical applications. Adding each new point may require a lot of CPU time or financial investments. This is the case for the aircraft design [1], structural reliability analysis [2], mechanical engineering [3], geostatistics [4], environmental analysis [5], various modeling [6] and multiscale simulations [7]. Sequential sampling algorithms significantly reduce the number of necessary points, and therefore the calculation time, compared to conventional (one-shot) sampling, which does not use information about already sampled points.

Several historically different names for sequential (adaptive) sampling algorithms exist today. In the theory of experiment planning, sequential sampling algorithms belong to the field of optimal sequential designs [8] and the constructed model is called the response surface. The task of filling sampling criteria for computationally expensive

optimization problems is very close to adaptive sampling [9,10]. The term “active learning” in machine learning area [11] includes online sampling and artificial synthesis of sampled points that is not trivial if each point is an image [12].

There are many approaches to the adaptive sampling. In this work we focus on those with the distance function which is applicable in the spaces of moderate dimension [13] due to curse of dimensionality. Traditionally the algorithms are presented in the form that maximizes a function evaluating the compromise between local exploitation and global exploration (see, for example, popular reviews [14–16]). Global exploration tries to place new points in unoccupied areas, i.e. it is a space filling design [17]. The following methods approach this result: Latin hypercube sampling [18], maximizing the minimum distance [19, 20], minimizing the maximum size of the area without points [21], Sobol sequences [22], maximizing entropy [23]. Local exploitation

* Corresponding author at: Institute for Mathematics, Mechanics and Computer Science in the name of I.I. Vorovich, Southern Federal University, Milchakova 8a, 344090, Rostov-on-Don, Russia.

E-mail address: gudasergey@yandex.ru (S.A. Guda).

in turn implies a thorough study of the regions with maximal approximation error, which is unknown and can be characterized by either

- a large variance [24] for probabilistic models,
- a large error estimated by a cross-validation [25],
- multiple variances of surrogate models [26],
- a large gradient [27],
- a large Laplacian of the interpolation model [28],
- a large difference in the values of the objective functions at neighboring points [29].

Various strategies balance between local exploitation and global exploration, including a gradual reduction in priority for a global exploration [30] and switching between polar strategies [31]. A comprehensive comparison of the quality of the methods is given in the review [15].

Kriging [32] is one of the most popular interpolation methods. It postulates the probabilistic distribution of an unknown function and analytically estimates the variance. Kriging model is unbiased, hence its approximation error is equal to variance. The latter does not depend on function values, thus the maximization of the approximation error in local exploitation means for kriging the placing new sampling points in less explored regions with large variance, i.e. global exploration. Such an “integrated” approach to the problem of a balance between local exploitation and global exploration is quite attractive. However, the variance in kriging is homoscedastic and simple examples show that kriging does not provide a suitable estimate for the variance of an arbitrary function (see Fig. 1 in [33]). The kriging drawback was fixed in [34] by introducing dependency on function values into the variance formula keeping the same “integrated” balance between local exploitation and global exploration.

We achieved the balance between local exploitation and global exploration by means of the integral norm L_p of the function approximation error. L_p optimization, as we prove, is equivalent to the maximization of the function depending both on the magnitude of the error and the size of the neighborhood where the approximated function changes significantly after adding a new point. Thus, we obtain an “integrated” approach to the choice between local exploitation and global exploration as in the case of kriging, while the error estimation depends on the values of the sampled function.

The article is organized as follows: Section 2, defines the problem and the criterion for automatic balancing between local exploitation and global exploration. Section 3 describes the sampling algorithm. A general scheme is followed by criterion optimization approaches. In Section 4 we describe the software: class structures, user-defined constraints and batch sampling. Finally, we perform numerical experiments, verify the accuracy of the algorithm for the benchmark functions [15] and compare with existing implementations Section 5.

The source code of the algorithm, named as MinLpE, is published at Github development platform under the terms of LGPL license [35].

2. Criterion for optimal sampling

Let the objective function y^* be defined on the set $D \subset \mathbb{R}^n$. Consider a set of l points $\{x\}_{i=1}^l$, $x_i \in D$ with known values of the objective function $y_i = y^*(x_i)$. The training sample $X^l = \{(x_i, y_i)\}_{i=1}^l$ is used for the construction of an approximation $a : D \rightarrow \mathbb{R}$ by a machine learning method μ : $a = \mu(X^l)$.

Adaptive sampling selects each next point x_{l+1} so that the new sample $X^{l+1} = \{(x_i, y_i)\}_{i=1}^{l+1}$ provides the best approximation to the target function y^*

$$\|\mu(X^{l+1}) - y^*\|_{L_p(D)} \rightarrow \min_{x_{l+1}}. \quad (1)$$

Definition 1. We call the ML method μ local if and only if the models trained on X and $X' = X \cup \{x_0\}$ differ only in a small neighborhood $S(x_0)$ of the point x_0 .

An example of the local method is the K nearest neighbor approach.

Theorem 1. Let μ be a local approximation method. Then optimization of the $L_p(D)$ norm (1) over the entire domain D is equivalent to the local optimization of the expression (2) with $L_p(S(x_{l+1}))$ norms for the vicinity of the point x_{l+1} :

$$\|\mu(X^{l+1}) - y^*\|_{L_p(S(x_{l+1}))} - \|\mu(X^l) - y^*\|_{L_p(S(x_{l+1}))} \rightarrow \min_{x_{l+1}}. \quad (2)$$

The proof follows directly from the equality

$$\begin{aligned} \int_D |\mu(X^{l+1}) - y^*|^p &= \int_D |\mu(X^l) - y^*|^p - \int_{S(x_{l+1})} |\mu(X^l) - y^*|^p \\ &\quad + \int_{S(x_{l+1})} |\mu(X^{l+1}) - y^*|^p. \end{aligned}$$

The neighborhood $S(x_{l+1})$ contains x_{l+1} . If μ is an interpolation method, then $\mu(X^{l+1})(x_{l+1}) - y_{l+1} = 0$. Thus, for interpolation method μ (and many other local approximation methods), the first term in (2) is substantially smaller than the second. For simplicity, in the following we will consider optimization of the second term only

$$\int_{S(x_{l+1})} |\mu(X^l) - y^*|^p \rightarrow \max_{x_{l+1}}. \quad (3)$$

Eq. (3) determines the balance between local exploitation and global exploration. On the one hand, the integrand is an approximation error and favors the optimal location of the point x_{l+1} in the region with a large error $|\mu(X^l) - y^*|$, similar to local exploitation strategy. On the other hand, the integral over the neighborhood $S(x_{l+1})$ increases along with the volume of this neighborhood, which forces adaptive sampling to find the regions free from nodal points (large area of the approximation variation $\mu(X^{l+1}) - \mu(X^l)$) i.e. perform global exploration.

Consider, for example, the nearest neighbor method as μ . Then $S(x_{l+1})$ is the Voronoi cell of the point x_{l+1} constructed for the training sample X^{l+1} , $\mu(X^l)(x) - y^*(x) = y_{NN_l(x)} - y^*(x)$, where $NN_l(x)$ is the index of the training sample point closest to x . If we estimate the approximation error $|y_{NN_l(x)} - y^*(x)|$ inside $S(x_{l+1})$ by the maximum of the cross-validation error

$$\max_{k: S(x_k) \cap S(x_{l+1}) \neq \emptyset} |y_{NN_l}(x_k) - y_k|,$$

then the solution to the problem of balance between local exploitation and global exploration will be the choice of the Voronoi vertex x_{l+1} with the maximum value of the expression (4)

$$|S(x_{l+1})| \cdot \max_{k: S(x_k) \cap S(x_{l+1}) \neq \emptyset} |y_{NN_l}(x_k) - y_k|^p \rightarrow \max_{x_{l+1}}. \quad (4)$$

Here $|S(x_{l+1})|$ is the volume of the Voronoi cell. Eq. (4) is the criterion for optimal choice for the next sampling point x_{l+1} when we use nearest neighbor approximation.

3. Optimal point search algorithms

The scheme of the proposed sampling algorithm has a standard form (see Fig. 1). Adaptive sampling starts with generating an initial sample X^l using the Improved Latin Hypercube method [36]. The criterion (3) is optimized in the loop, the value of the target function is calculated at the resulting point $y_{l+1} = y^*(x_{l+1})$ and is added to the sample.

The search for the optimal point x_{l+1} includes the optimization (3) which is an “overhead calculation” of the sampling. In our method, the user may apply constraints on the time for this optimization. Thus, we ensure the independence of the algorithm working time on the size l of the training sample. However, restrictions affect the quality of the approximation.

Since in the proposed method the attention is paid to the execution time, only fast approaches were selected to optimize (3). Maximization of (3) is carried out by iterating over several possible candidates x_{l+1} . In this work we compare the following choices:

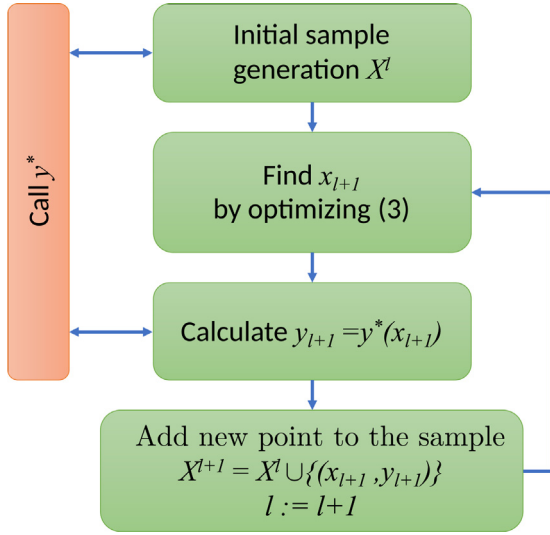


Fig. 1. The sampling algorithm.

1. Voronoi cells vertices for the entire training sample (notation on the graphs: voronoi).
2. Random points in the vicinity of the maximums of the approximation error. Random points are selected in a spherical layer around each possible maximum x_{\max} in the interval $[0.5d_{NN}, 2.5d_{NN}]$, where d_{NN} is the distance to the nearest neighbor of x_{\max} in the training sample (notation on the graphs: neighbors).
3. Random points in the whole definition area of y^* (notation on graphs: random).

The number of candidates selected in options #2 and #3 is proportional to the probability that the point x_{\max} is the actual maximum of an error. We calculate these probabilities assuming that the value of the objective function has a normal distribution, and the estimated error is the standard deviation of this distribution.

The next important point in optimizing (3) is the estimation of the approximation error $|\mu(X^l)(x) - y^*(x)|$. Kriging can estimate the variance together with the prediction of values. It is a regression model based on the Gaussian process and assumes that the target dependence also belongs to the class of Gaussian processes. This imposes certain restrictions on the objective function, which are not always satisfied in practice. Therefore, we use approaches to the error estimation $|\mu(X^l)(x) - y^*(x)|$ based on a variance (and cross-validation), gradient estimation, distance to the neighbors:

- (1) $|\mu(X^l)(x) - y_{NN(x)}|$ (notation on graphs: model-cv);
- (2) $\nabla y_{NN(x)} \cdot \|x - x_{NN(x)}\|$, where ∇y — estimation of y gradient:

$$\nabla y_j = \max_{i: x_i \in S(x_j)} \frac{\|y_i - y_j\|}{\|x_i - x_j\|}$$
 (notation on graphs: grad);
- (3) $\|x - x_{NN(x)}\|$ (notation on graphs: distance);
- (4) the exact value of the error — only for testing (notation on graphs: exact);

In the first model-based approach, before insertion of the each calculated point (x_{l+1}, y_{l+1}) into the sample and refitting the model, we evaluate and save LOO-error $|\mu(X^l)(x_{l+1}) - y_{l+1}|$. Such a trick makes cross-validation unnecessary and reduce sampling time. But the addition of new points to the sample changes the model, that is why we have to recalculate these errors in future.

To save CPU time, the calculation of the integral in (3) was replaced by the approximation error in x_{l+1} divided by the density of points in the neighborhood of x_{l+1} .

4. Implementation description

4.1. Program structure

The abstract adaptive sampling engine consists of the following main parts:

Class Sampler generates initial sample X^l , new point x_{l+1} , and evaluates the quality of the training sample. As the arguments of the class constructor, a user can provide approximation model, criteria for stopping sampling, sampler time constraints and functions for calculating distance in argument and values spaces. By default the sampler uses RBF approximation model, stops calculation when CV error is less than 0.01 and uses time constrain $\max(\min(t1/10, 6) * 10, 0.1)$, where $t1$ is the one point calculation time. To calculate the distance between points x we use scaled L_2 -norm as in the paper [37]. Thus the (random, distance) method is almost the same as the maximin scaled distance approach from the paper [37]. For multi-dimensional function values y , by default, we use L_∞ -norm.

CalculationProgram is a user-defined function that calculates the values of $y^*(x)$. Evaluation of the objective function is a time-consuming step, therefore, parallelization is applied in adaptive sampling. In this regard, *CalculationProgram* is able to call the calculation method in parallel. The user must add unsafe code for a parallel call in a critical section.

Class CalculationOrchestrator is responsible for the calculating order of $y^*(x)$. Depending on the parameter m , the number of points calculated in parallel, *CalculationOrchestrator* simultaneously calculates several values of $y^*(x_{l+i})$ for $i = 1 \dots m$. However, parallel sampling mode may require more points to achieve the necessary quality of the training sample compared to a single-threaded mode.

Class DatasetGenerator links instances of *Sampler* and *CalculationOrchestrator*, according to the single-responsibility principle [38]. It is used to generate a training sample. The pairs $(x, y^*(x))$ are returned by the sampler and stored in *DatasetGenerator*, forming a training sample. At each step, the training sample is verified for compliance with the stop criterion and is returned to the user in case of successful verification.

4.2. Sampling time

We aimed to implement a fast sampling algorithm with the generation time of each point independent on the current sample size. First, the number of points in candidate selection methods during optimization of (3) were restricted. Secondly in the model-cv method the model is re-trained after several steps of the sampling algorithm. The number of steps depends on the time to rebuild the model. Finally, the calculation of the Voronoi cells vertices is performed for the neighborhood of the maximum approximation error instead of the entire sample.

In practice user defines constraint on the execution time of one step of the algorithm. It can be absolute or relative to the average calculation time of the target function. The execution time is measured during sampling for each operation: checking the criterion (3) for one candidate and rebuilding the model in the model-cv method. The number of candidates and the need to rebuild the model are determined from the estimated values and user restrictions. The calculation time of the Voronoi cell vertices for different dimensions and sample sizes were tabulated on a reference computer. The multiplier to reduce the Voronoi calculation time to the tabulated values is measured during sampling.

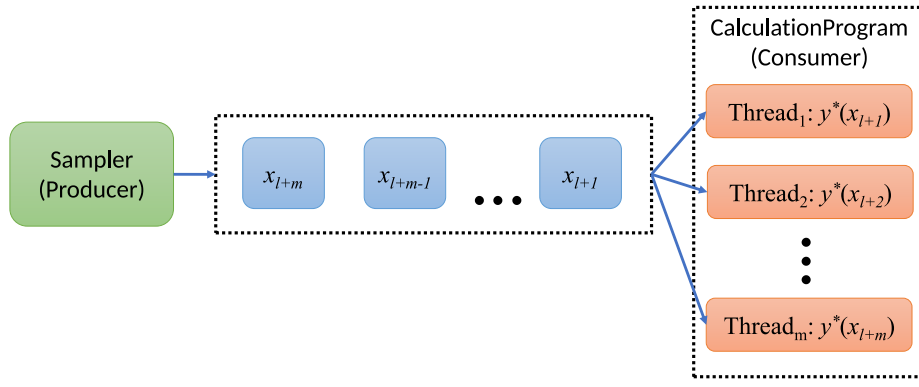


Fig. 2. The parallelization of the sampling algorithm.

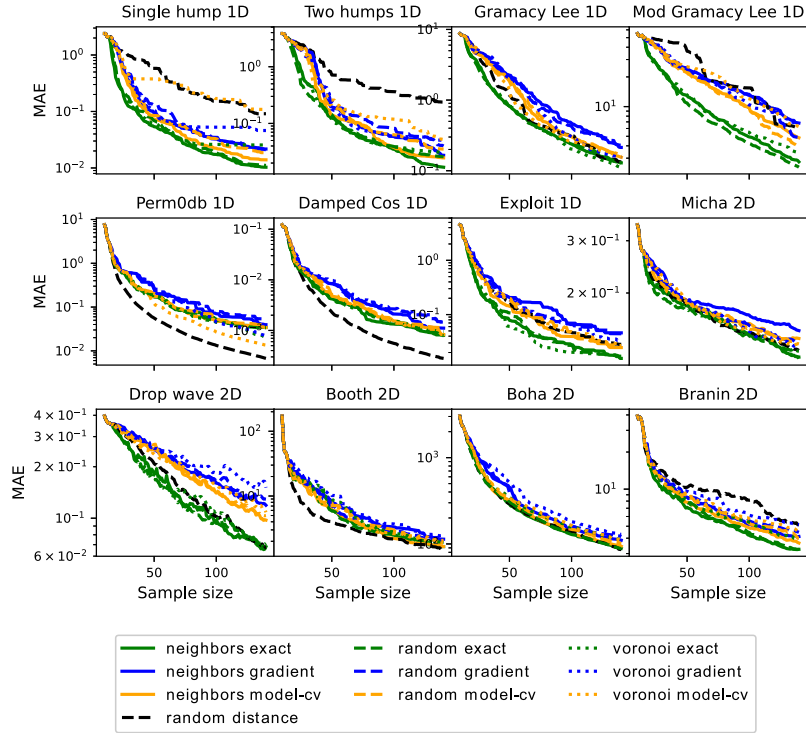


Fig. 3. The dependence of the approximation error on the sample size for various combinations of algorithms from Section 3 on a set of benchmarks [15] (part 1). The time constraint for sampling each point is 0.05 s.

4.3. Batch (parallel) sampling

Batch version of adaptive sampling is recommended when the target function y^* can be calculated at multiple points in a parallel regime. The program finds several points (a batch) and adds to the sample X^l in one iteration:

$$\|\mu(X^{l+m}) - y^*\|_{L_p(D)} \rightarrow \min_{x_{l+1} \dots x_{l+m}}. \quad (5)$$

The common approach to the parallelization is the modification of sampling algorithms to return a given number of optimal points that are sufficiently far apart, for example, tree-like sampling algorithms based on entropy maximization [39], a modified optimization criterion DIRECT [40], a combination of the expected improvement and the space filling criterion [41].

We do not follow this way, because the calculations of $y^*(x_{l+i})$ inside the batch can take different amount of time. This can happen both due to the target function itself, or due to different configuration and load of computing nodes. To ensure optimal load balance on p computing nodes, it is necessary to generate the next sample point when one of

the nodes has finished a calculation. Therefore, we obtain the problem of generating a point x_{l+p} given the training set $X^l = \{(x_i, y_i)\}_{i=1}^l$ and fixed $p-1$ points $x_{l+1}, x_{l+2}, \dots, x_{l+p-1}$, the values of the y^* function in which are currently being calculated.

Taking into account the variety of approaches defined in Section 3 we have chosen a fairly simple and effective method and replaced temporary the exact values of the y^* calculated at points $x_{l+1}, x_{l+2}, \dots, x_{l+p-1}$ by their approximations $\mu(X^l)(x_{l+1}), \mu(X^{l+1})(x_{l+2}), \dots, \mu(X^{l+p-2})(x_{l+p-1})$. Such a trick makes it possible to apply the sequential sampling algorithm as well, and ensures the remote spacing of the selected points.

An important feature of the batch sampling is the use of a lazy thread pool. This parallelization algorithm can be considered as a producer-consumer task (see Fig. 2). The producer *Sampler* creates new points x_{l+i} , and the consumer is *CalculationProgram*. The task of the *CalculationOrchestrator* class is the optimal allocation and utilization of resources. *Sampler* is capable of generating points x_{l+i} indefinitely (excluding the stopping criterion), but computing resources are limited. *CalculationOrchestrator* initially requests as many points as necessary to

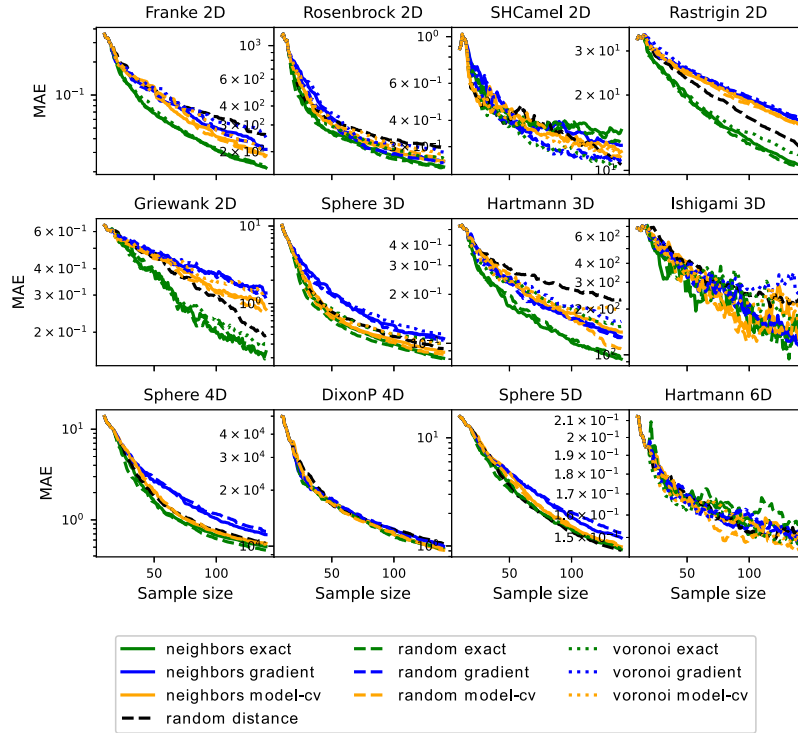


Fig. 4. The dependence of the approximation error on the sample size for various combinations of algorithms from Section 3 on a set of benchmarks [15] (part 2). The time constraint for sampling each point is 0.05 s.

utilize the computing power in the best way. Then it waits for any of the values of $y^*(x_{i+i})$ to be calculated, returns the value to *Sampler*, and then requests a new point x_{i+i+1} for calculation. The last step is repeated until the training sample is ready.

5. Experiment

The [Theorem 1](#) provides a theoretical solution to the problem of the balance between local exploitation and global exploration. However, the practical application of [1](#) suffers from drawbacks of greedy policy. The optimal choice of the approach from Section 3 to optimization (3) depends on the type of benchmark function (see [Figs. 3](#) and [4](#)).

The best approximation is based on exact knowledge of the error (green curves, exact on [Figs. 3](#) and [4](#)). The next in quality is the error estimation by cross-validation (model-cv, orange curves) and the worst is the error estimation by approximate gradient calculation (grad, blue curves). The random-distance method is shown as a black dashed curve on all graphs. It is a global-exploration in its pure form and does not rely on calculated function values. Its advantage in [Fig. 3](#) for Perm0db 1D and Damped Cos 1D is explained by the drawback of a greedy sampling approach of the others algorithms. These are the cases of smooth functions, when the error is approximately constant in the whole space. The optimal sampling for this problem is close to uniform, as obtained in the random-distance method. We notice the following conspicuous cases highlighting the differences between sampling approaches:

1. Function y^* with one or few distinct local extrema. In this case, placing more sample points in the vicinity of the extrema becomes important. Purely global exploration method (random, distance) fails, but the others performs similarly. This case is the most evident for benchmarks: Single hump 1D, Two humps 1D, Branin 2D (see [Fig. 3](#)).
2. Error of approximation is uniform i.e has no distinct localized maxima. Greedy algorithms that maximize (3) are worse than (random, distance). The neighbor method suffers the most from

a uniformly distributed error, even when the exact error estimation is used. Benchmarks with functions of this type are Perm0db 1D, Damped Cos 1D, Booth 2D (for sample size less than 50) (see [Fig. 3](#)).

3. The dataset contains not enough points for a function approximation, that is why the approaches based on exact error estimation are better than others. The characteristic benchmark functions are: Single hump 1D (for sample size less than 40), Two humps 1D (for sample size less than 40), Gramacy Lee 1D (for sample size less than 50), Mod Gramacy Lee 1D, Exploit 1D, Micha 2D (for sample size less than 40), Drop wave 2D, Branin 2D, Franke 2D, Rosenbrock 2D (for sample size less than 60), Rastrigin 2D, Griewank 2D, Hartman 3D (see [Figs. 3](#) and [4](#)).
4. The lack of nodal points is serious and the approximation model is significantly changing during the sampling. All the methods perform the same as in the case of Ishigami 3D and Hartmann 6D (see [Fig. 4](#)).

Note, that first and third cases include common functions. Also benchmarks Drop wave 2D ([Fig. 3](#)), Rastrigin 2D and Griewank 2D ([Fig. 4](#)) share traits of uniform approximation error where greedy approaches fail (case 2) and lack of data points to estimate the error thoroughly (case 3).

In the space of higher dimension, the requirements for the initial sample and the number of candidate points to be tested increase by orders of magnitude when maximizing (3). The accidental detection of one of the extremes causes the neighbor method to assume that it is a function with one localized extremum, and the method searches for new points in its vicinity, instead of a global exploration for other extrema (see [Fig. 6](#)). Even knowledge about the exact value of the error may not lead to the correct maximum (3), as in the case of 5HCamel 2D.

[Fig. 5](#) shows the approximation quality as a function of the sampling step time constraint. Random and Voronoi algorithms, as well as the model-cv error estimation algorithm, suffer most from the lack of analysis time. With the strictest restrictions on the sampling time,

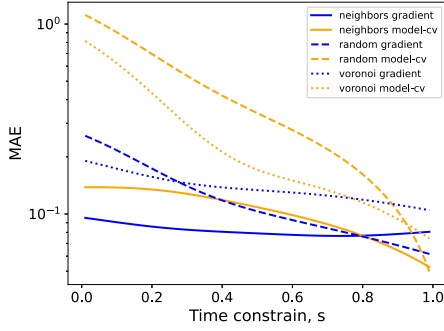


Fig. 5. The dependence of the approximation error on the time constraint of one point sampling for various combinations of algorithms from Section 3 on the Single hump benchmark. The sample size is constant and equal to 140.

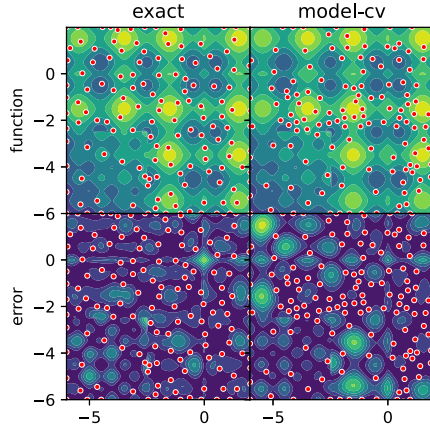


Fig. 6. Samples generated by the (neighbor, exact) and (neighbor, model-cv) methods for the Rastrigin 2D function. The heat maps on the background are built for the true sampled function (top left), approximation (top right), true error (bottom left) and approximated error (bottom right). Sample size is 140, time limit 0.05 s.

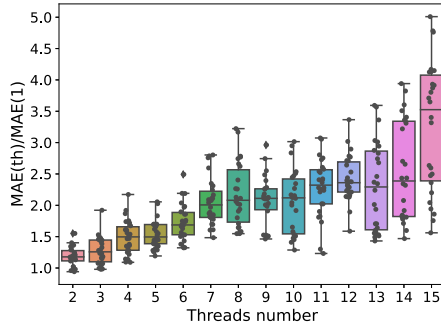


Fig. 7. The dependence of the approximation error on the batch size of the parallel sampling. Method (neighbors, model-cv) is applied to Single hump 1D function. The graph shows the ratio of MAE for parallel sampling to MAE for sequential one for a series of experiments with sample sizes from 25 to 50.

the best combination is (neighbors, gradient). Parallel batch sampling accelerates the calculation and provide results that are similar in quality to sequential. For example, a 10-fold acceleration of calculations decreases the quality of approximation by only 2 times (see Fig. 7).

We made a comparison of the our algorithm to the best reported in [15] (see Fig. 8). The Monte Carlo-intersite-proj-th (MIPT) method [42] is based only on exploration and very close to our’s (random, distance) case. Mixed Adaptive Sampling Algorithm (MASA) [43] combines a local exploitation based on query by committee fluctuation and a global exploration based on distance. Taylor-Expansion based Adaptive Design (TEAD) [44] combines gradient-based exploitation

and distance-based exploration. Expected Improvement for Global Fit (EIGF) [29] uses the kriging variance for exploration and surrogate estimation error for exploitation. The Space-Filling Cross-Validation Tradeoff (SFCVT) [45] and Smart Sampling Algorithm (SSA) [46] combine a cross-validation for exploration and a distance criterion to ensure an exploration character.

The quality of our algorithm is similar to the best values reported in [15] (see Fig. 8) with some deviations. For example, the performance was superior for Two humps 1D, Boha 2D benchmarks and worse for Exploit 1D and Damped Cos 1D benchmarks. Nevertheless, the main advantage of the newly proposed algorithm is the sampling speed.

6. Conclusion

We have developed a fast adaptive sampling algorithm that supports effective parallelization in terms of batch generation of new points. A distinctive feature of the developed algorithm is the possibility to limit the time of “overhead calculations” when generating a next point by user-defined time constraints. The sampling is based on optimization of the L_p integral norm of the approximation error where the norm parameter p regulates the balance between local exploitation and global exploration.

There is no universal sampling strategy equally optimal for all types of functions. We compared several approaches to the optimization of the proposed criterion on a set of benchmark functions [15]. The choice of candidate points, method for an error estimation, number of parallel sampled points in the batch and user imposed time constraint determine the approximation quality. The options available for users to select candidate points in each step are the vertices of Voronoi cells, random sampling in the vicinity of the largest error, and even distribution over the whole space. For all of them, the error can be estimated with the help of cross-validation, the gradient of the function, and by using the exact value of function (option for testing). Based on the benchmark tests we recommend as a first trial to choose candidates in a neighborhood of the approximation error maximum and estimate error by means of cross-validation.

Comparison of our approach with the best from review [15] shows acceptable approximation quality. The main drawback of the proposed method is the simple criteria of the balance between local exploitation and global exploration which results in higher approximation error, comparing to complex methods: MASA, MIPT, SFCVT, LIP. But our method outperforms the others in sampling time. Skillfully managing the number of candidates to maximize error, skipping rebuilding of the approximation model on some sampling steps, our approach keeps the complexity linear with respect to the sample size. It is the main advantage.

There are two important limitations of the proposed method: sampling space should not be high dimensional and the approximation method should be “local” (see Definition 1). Due to the curse of dimensionality, distances to points tend to be equal and we can not determine neighbors reliably [13]. The calculation of the vertices of Voronoi cells becomes complicated in high dimensions. To construct a model sufficient for starting approximation we need a large initial dataset. Despite the fact that time performance of (neighbors, model-cv) remains good, the methods avoiding distance functions are preferable in high dimensional spaces, for example, decomposing the original space problems into sub-spaces, identifying important variables, and reducing design space (see review [47]).

The locality of the approximation method plays crucial role for the proposed adaptive sampling approach. If a new point (x_{l+1}, y_{l+1}) in the constructed sample changes the approximation function not only in the neighborhood of (x_{l+1}, y_{l+1}) , then the condition of Theorem 1 is violated and decreasing of the approximation error in the sampling process is not guaranteed. Kriging, kNN, kernel ridge regression — are local approximation methods, RBF, SVR with RBF kernel are nearly local, but polynomial regression, neural networks, regression trees, SVR with polynomial kernels are not.

The source code of the algorithm, named MinLpE, is open [35] and distributed under LGPL license.

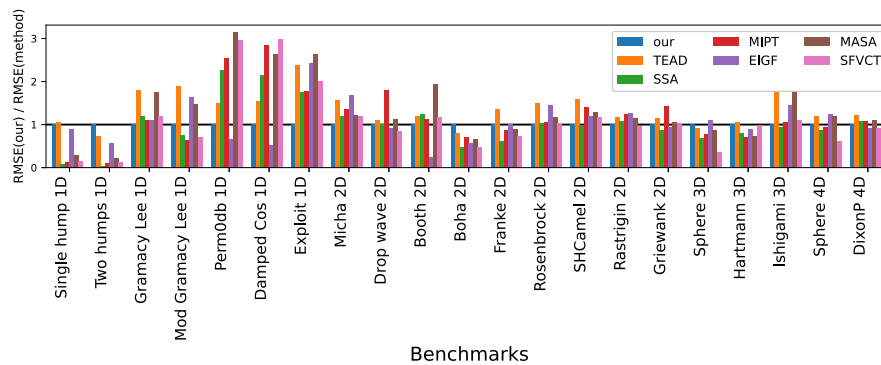


Fig. 8. Comparison with the best adaptive sampling methods from [15] review (higher is better). Method (neighbors, model-cv), sample size 100, time limit 0.1 s. The interpolation in all the methods is done by RBF.

CRediT authorship contribution statement

A.S. Algasov: Software, Investigation, Data curation, Writing – original draft. **S.A. Guda:** Software, Investigation, Data curation, Writing – original draft. **V.I. Kolesnikov:** Supervision, Writing – review & editing. **V.V. Ilicheva:** Writing – original draft, Validation, Visualization. **A.V. Soldatov:** Funding acquisition, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

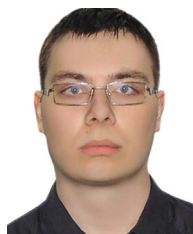
Acknowledgments

AVS acknowledges the Ministry of Science and Higher Education of the Russian Federation for financial support (Agreement No. 075-15-2021-1363) for the work on algorithm implementation. VIK acknowledges the Russian Science Foundation for the financial support (project No 21-79-30007) for the work on experimental section.

References

- [1] Z. Han, Kriging surrogate model and its application to design optimization: A review of recent progress, *Hangkong Xuebao/Acta Aeronaut. Astronaut. Sinica* 37 (11) (2016) 3197–3225, <http://dx.doi.org/10.7527/S1000-6893.2016.0083>.
- [2] Z. Sun, J. Wang, R. Li, C. Tong, LJF: A new kriging based learning function and its application to structural reliability analysis, *Reliab. Eng. Syst. Saf.* 157 (2017) 152–165, <http://dx.doi.org/10.1016/j.res.2016.09.003>.
- [3] D. Han, J.-R. Zheng, A survey of metamodeling techniques in engineering optimization, *Huadong Ligong Daxue Xuebao/J. East China University of Sci. Technol.* 38 (6) (2012) 762–768.
- [4] M.E. Rossi, C.V. Deutsch, *Mineral Resource Estimation*, Springer Netherlands, 2014, <http://dx.doi.org/10.1007/978-1-4020-5717-5>.
- [5] P. Legendre, M. Fortin, *Spatial pattern and ecological analysis*, *Vegetatio* 80 (2) (1989) 107–138, <http://dx.doi.org/10.1007/BF00048036>.
- [6] W. Van Beers, J. Kleijnen, Kriging interpolation in simulation: A survey, 1, 2004, pp. 113–120.
- [7] K.W. Leiter, B.C. Barnes, R. Becker, J. Knap, Accelerated scale-bridging through adaptive surrogate model evaluation, *J. Comput. Sci.* 27 (2018) 91–106, <http://dx.doi.org/10.1016/j.jocs.2018.04.010>.
- [8] H. Chernoff, I. Sequential analysis and optimal design, in: *Sequential Analysis and Optimal Design*, pp. 1–120, <http://dx.doi.org/10.1137/1.9781611970593.ch1>.
- [9] K. Chaiyotha, T. Krityakierne, A comparative study of infill sampling criteria for computationally expensive constrained optimization problems, *Symmetry* 12 (10) (2020) <http://dx.doi.org/10.3390/sym12101631>, URL <https://www.mdpi.com/2073-8994/12/10/1631>.
- [10] S. Rojas-Gonzalez, I. Van Nieuwenhuysse, A survey on kriging-based infill algorithms for multiobjective simulation optimization, *Comput. Oper. Res.* 116 (2020) <http://dx.doi.org/10.1016/j.cor.2019.104869>.
- [11] D. Pereira-Santos, R. Prudêncio, A. de Carvalho, Empirical investigation of active learning strategies, *Neurocomputing* 326–327 (2019) 15–27, <http://dx.doi.org/10.1016/j.neucom.2017.05.105>.
- [12] L. Wang, X. Hu, B. Yuan, J. Lu, Active learning via query synthesis and nearest neighbour search, *Neurocomputing* 147 (1) (2015) 426–434, <http://dx.doi.org/10.1016/j.neucom.2014.06.042>.
- [13] C. Aggarwal, A. Hinneburg, D. Keim, On the surprising behavior of distance metrics in high dimensional space, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1973, 2001, pp. 420–434, http://dx.doi.org/10.1007/3-540-44503-x_27.
- [14] H. Liu, Y.-S. Ong, J. Cai, A survey of adaptive sampling for global metamodelling in support of simulation-based complex engineering design, *Struct. Multidiscip. Optim.* 57 (1) (2018) 393–416, <http://dx.doi.org/10.1007/s00158-017-1739-8>.
- [15] J.N. Fuhg, A. Fau, U. Nackenhorst, State-of-the-art and comparative review of adaptive sampling methods for kriging, *Arch. Comput. Methods Eng.* (2020) 1–59.
- [16] P. Lualdi, R. Sturm, T. Siefkes, Exploration-oriented sampling strategies for global surrogate modeling: A comparison between one-stage and adaptive methods, *J. Comput. Sci.* 60 (2022) 101603, <http://dx.doi.org/10.1016/j.jocs.2022.101603>.
- [17] V. Joseph, Space-filling designs for computer experiments: A review, *Qual. Eng.* 28 (1) (2016) 28–35, <http://dx.doi.org/10.1080/08982112.2015.1100447>.
- [18] D. Chen, S. Xiong, Flexible nested latin hypercube designs for computer experiments, *J. Qual. Technol.* 49 (4) 337–353, <http://dx.doi.org/10.1080/00224065.2017.11918001>.
- [19] S. Ba, W. Myers, W. Brennenman, Optimal sliced latin hypercube designs, *Technometrics* 57 (4) (2015) 479–487, <http://dx.doi.org/10.1080/00401706.2014.957867>.
- [20] V. Joseph, E. Gul, S. Ba, Maximum projection designs for computer experiments, *Biometrika* 102 (2) (2015) 371–380, <http://dx.doi.org/10.1093/biomet/asv002>.
- [21] M. Tan, Minimax designs for finite design regions, *Technometrics* 55 (3) (2013) 346–358, <http://dx.doi.org/10.1080/00401706.2013.804439>.
- [22] I. Sobol', On the distribution of points in a cube and the approximate evaluation of integrals, *USSR Comput. Math Math Phys* 7 (4) 86–112, [http://dx.doi.org/10.1016/0041-5553\(67\)90144-9](http://dx.doi.org/10.1016/0041-5553(67)90144-9).
- [23] C. Sheng, M.H.Y. Tan, L. Zou, Maximum expected entropy transformed latin hypercube designs, *J. Appl. Stat.* 48 (12) (2021) 2152–2177, <http://dx.doi.org/10.1080/02664763.2020.1786674>.
- [24] J. Sacks, W.J. Welch, T.J. Mitchell, H.P. Wynn, Design and Analysis of Computer Experiments, *Statist. Sci.* 4 (4) (1989) 409–423, <http://dx.doi.org/10.1214/ss/1177012413>, URL <http://dx.doi.org/10.1214/ss/1177012413>.
- [25] T. Fushiki, Estimation of prediction error by using K-fold cross-validation, *Stat. Comput.* 21 (2) (2011) 137–146.
- [26] R. Burbidge, J.J. Rowland, R.D. King, Active learning for regression based on query by committee, in: H. Yin, P. Tino, E. Corchado, W. Byrne, X. Yao (Eds.), *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 209–218.
- [27] M. Rumpfkeil, W. Yamazaki, M. Dimitri, A dynamic sampling method for kriging and cokriging surrogate models, in: 49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, <http://dx.doi.org/10.2514/6.2011-883>, URL <https://arc.aiaa.org/doi/abs/10.2514/6.2011-883>.

- [28] T. Mackman, C. Allen, M. Ghosh, K. Badcock, Comparison of adaptive sampling methods for generation of surrogate aerodynamic models, *AIAA J.* 51 (2013) 797–808, <http://dx.doi.org/10.2514/1.J051607>.
- [29] W.I. Notz, C.Q. Lam, Sequential adaptive designs in computer experiments for response surface model fit, 6, 2008, pp. 207–233.
- [30] B. Kim, Y. Lee, D.-H. Choi, Construction of the radial basis function based on a sequential sampling approach using cross-validation, *J. Mech. Sci Technol - J MECH SCI TECHNOL* 23 (2009) 3357–3365, <http://dx.doi.org/10.1007/s12206-009-1014-z>.
- [31] H. Liu, J. Cai, Y. Ong, An adaptive sampling approach for kriging metamodeling by maximizing expected prediction error, *Comput. Chem. Eng.* 106 (2017) <http://dx.doi.org/10.1016/j.compchemeng.2017.05.025>.
- [32] M. Oliver, R. Webster, Kriging: A method of interpolation for geographical information systems, *Int. J. Geograp. Inform. Syst* 4 (3) (1990) 313–332, <http://dx.doi.org/10.1080/02693799008941549>.
- [33] J. Yamamoto, An alternative measure of the reliability of ordinary kriging estimates, *Math. Geol.* 32 (2000) 489–509.
- [34] H. Liu, S. Xu, Y. Ma, X. Chen, X. Wang, An Adaptive Bayesian Sequential Sampling Approach for Global Metamodeling, *J. Mech. Des.* 138 (1) (2015) <http://dx.doi.org/10.1115/1.4031905>, 011404.
- [35] MinLpE – Fast adaptive sampling with operation time control, <https://github.com/gudasergey/minlpe>.
- [36] Q. Xu, Y. Yang, Y. Liu, X. Wang, An improved latin hypercube sampling method to enhance numerical stability considering the correlation of input variables, *IEEE Access* 5 (2017) 15197–15205, <http://dx.doi.org/10.1109/ACCESS.2017.2731992>.
- [37] R. Jin, W. Chen, A. Sudjianto, On Sequential Sampling for Global Metamodeling in Engineering Design, in: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 2: 28th Design Automation Conference, 2002*, pp. 539–548, <http://dx.doi.org/10.1115/DETC2002/DAC-34092>.
- [38] R. Martin, J. Rabaey, A. Chandrakasan, J. Newkirk, B. Nikolić, R. Koss, *Agile Software Development: Principles, Patterns, and Practices*, Alan Apt series, Pearson Education, 2003, URL <https://books.google.ru/books?id=0HYhAQAAIAAJ>.
- [39] R.B. Gramacy, H.K.H. Lee, Adaptive design and analysis of supercomputer experiments, *Technometrics* 51 (2) (2009) 130–145, <http://dx.doi.org/10.1198/TECH.2009.0015>.
- [40] Z. Huang, H. Qiu, M. Zhao, X. Cai, L. Gao, An adaptive SVR-HDMR model for approximating high dimensional problems, *Eng. Comput.* 32 (2015) 643–667, <http://dx.doi.org/10.1108/EC-08-2013-0208>.
- [41] A. Quan, *Batch Sequencing Methods for Computer Experiments* (Ph.D. thesis), The Ohio State University, 2014, URL http://rave.ohiolink.edu/etdc/view?acc_num=osu1401462464.
- [42] K. Crombecq, E. Laermans, T. Dhaene, Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling, *European J. Oper. Res.* 214 (3) (2011) 683–696.
- [43] J. Eason, S. Cremaschi, Adaptive sequential sampling for surrogate model generation with artificial neural networks, *Comput. Chem. Eng.* 68 (2014) 220–232.
- [44] S. Mo, D. Lu, X. Shi, G. Zhang, M. Ye, J. Wu, J. Wu, A Taylor expansion-based adaptive design strategy for global surrogate modeling with applications in groundwater modeling, *Water Resour. Res.* 53 (12) (2017) 10802–10823.
- [45] V. Aute, K. Saleh, O. Abdelaziz, S. Azarm, R. Radermacher, Cross-validation based single response adaptive design of experiments for kriging metamodeling of deterministic computer simulations, *Struct. Multidiscipl. Optim.* 48 (3) (2013) 581–605.
- [46] S.S. Garud, I.A. Karimi, M. Kraft, Smart sampling algorithm for surrogate model development, *Comput. Chem. Eng.* 96 (2017) 103–114.
- [47] S. Shan, G. Wang, Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions, *Struct. Multidiscipl. Optim.* 41 (2010) 219–241, <http://dx.doi.org/10.1007/s00158-009-0420-2>.



Alexander Algasov Ph.D. student, compiler design and data science.

Alexander is a four-year postgraduate student. He is fond of compiler design and works several years on the PascalABC.NET, which is the most popular educational programming system in Russia. The postgraduate study of Alexander was connected with Pyfitit - python implementation of Fitit software to fit X-ray absorption near edge structure (XANES) and other spectra. He applied machine learning for spectra analysis and atomic structure prediction problem.



Sergey Guda Lecturer, machine learning and mathematical physics.

Sergey has a 16 years of research experience. His first works and Ph.D. thesis were related to mathematical physics. But after winning the Yandex prize in «Internet-mathematics 2010» competition, he interested in machine learning too. Last 8 years he teaches students machine learning at the Department of Mathematics, Mechanics and Computer Science, Southern Federal University.



Vladimir Kolesnikov Professor, tribology and composite materials.

Vladimir Ivanovich Kolesnikov is a well-known specialist in the field of technology, physics and mechanics of metal-polymer composite tribosystems. Author of more than 570 scientific papers in the field of tribology. Through his scientific and organizational activities, he made a significant contribution to the development of metal-polymer interfaces production technology and of new methods for controlling their friction properties. Vladimir Kolesnikov is an academician of the Russian Academy of Sciences (Department of Power Engineering, Mechanical Engineering, Mechanics and Control Processes), President of the Rostov State University of Railways.



Vera Ilicheva Associate Professor, mathematical modeling, development of intelligent methods and algorithms for controlling dynamic processes in transport systems.

Research work of Vera Ilicheva connected with Rostov State University of Railways, where she also teach students. Her Ph.D. thesis was devoted to the decision making systems in transport industry.



Alexander Soldatov Professor, nanoscale local atomic, electronic and magnetic structures of advanced smart materials.

Alexander Soldatov is a scientific supervisor of the Smart Materials Research Institute. His broad research interests include theoretical and experimental investigation of local atomic and electronic structures in various types of advances materials, catalysts in-situ study of local structure changes in Li-Ni batteries cathode materials during charge/discharge process various types of solids, biological materials with active metal centers, nanomaterials for bio-medical applications and nanocomposites for xray photodynamic therapy of deep tumor tissues. Alexander is a member of Executive Committee of The International X-ray Absorption Society (IXAS), Review Panel at European X-ray Free Electron Laser (EuXFEL) and at European Synchrotron radiation Facility (ESRF).