

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



A Simulated Annealing Based Optimization Algorithm

Yoel Tenne

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/66455>

Abstract

In modern engineering finding an optimal design is formulated as an optimization problem which involves evaluating a computationally expensive black-box function. To alleviate these difficulties, such problems are often solved by using a metamodel, which approximates the computer simulation and provides predicted values at a much lower computational cost. While metamodels can significantly improve the efficiency of the design process, they also introduce several challenges, such as a high evaluation cost, the need to effectively search the metamodel landscape and to locate good solutions, and the selection of which metamodel is most suitable to the problem being solved. To address these challenges, this chapter proposes an algorithm that uses a hybrid simulated annealing and SQP search to effectively search the metamodel. It also uses ensembles that combine prediction of several metamodels to improve the overall prediction accuracy. To further improve the ensemble accuracy, it adapts the ensemble topology during the search. Finally, to ensure convergence to a valid optimum in the presence of metamodel inaccuracies, the proposed algorithm operates within a trust-region framework. An extensive performance analysis based on both mathematical test functions and an engineering application shows the effectiveness of the proposed algorithm.

Keywords: simulated annealing, metamodeling, ensembles, trust-region, expensive optimization problems

1. Introduction

With the advent of high performance computing, intricate computer simulations can now model real-world physics with high accuracy. This, in turn, transformed the engineering design process into a simulation-driven process in which candidate designs are evaluated by

a computer simulation instead of a laboratory experiment. In this set-up, the design process is formulated as an optimization problem with several unique features [1]:

- The computer simulation acts as the objective function, since it assigns objective values to candidate designs. However, the simulation is often a legacy code or commercial software whose inner workings are inaccessible to the user, and so there is no analytic expression that defines how candidate designs are mapped to objective values. Such a *black-box function* precludes the use of optimization algorithms that require an analytic function.
- Each simulation run is *computationally expensive*, that is, it requires a lengthy run time, and this severely restricts the number of candidate designs that can be evaluated.
- Both the real-world physics being modelled and the numerical simulation process may result in an objective function, which has a complicated landscape, and this further complicates the optimization process.

An optimization strategy that has proven effective in such problems is that of *metamodel-assisted* optimization, namely, where a metamodel approximates the true expensive function and provides predicted objective values at a lower computational cost [1]. However, the integration of metamodels and ensembles into the optimization search introduces several challenges:

- Locating a good solution requires effectively searching the metamodel, which can have a complicated landscape with multiple local solutions, and hence can be a difficult task.
- Since function evaluations are expensive, only a small number of evaluated vectors will be available and hence the metamodel will be inaccurate. In severe cases, the optimization search can converge to a false optimum, namely, which was artificially created by the metamodel's inaccurate approximation of the true expensive function.
- A variety of metamodels have been proposed, for example, artificial neural networks (ANNs), Kriging and radial basis functions (RBFs) [2, 3], but the optimal variant is problem-dependant, and is typically not known *a priori*. In an attempt to address this issue, *ensembles* employ several metamodels concurrently and aggregate their individual predictions into a single one [4, 5]. However, the effectiveness of ensembles depends on their topology, namely, which metamodels they incorporate, but the optimal topology is again typically unknown *a priori*, and may be impractical to identify by numerical experiments due to the high cost of each simulation run.

To address this issue, this chapter proposes an optimization algorithm that uses a hybrid-simulated annealing (SA) search followed by a local refinement of solutions based on an SQP search. In this manner, this set-up achieves both an effective global and local search, which assists in locating good solutions. To address the issue of inaccurate metamodel predictions, the proposed algorithm operates within a trust region (TR) approach that manages the metamodel and ensures convergence to a valid optimum. Finally, to further improve the prediction accuracy the proposed algorithm uses ensembles and selects the most suitable topology during the search.

Accordingly, this chapter presents several contributions: (a) a hybrid SA-SQP metamodel-assisted search, (b) integration within a TR framework, and (c) continuous selection of the ensemble topology during the search. An extensive performance analysis based on both mathematical test functions and an engineering problem of airfoil shape optimization shows the effectiveness of the proposed algorithm.

The remainder of this chapter is organized as follows: Section 2 provides the background information, Section 3 describes the proposed algorithm, then Section 4 provides an extensive performance evaluation and discussion, and finally Section 5 concludes this chapter.

2. Background

2.1. Optimization techniques

As mentioned in Section 1, simulation-driven problems often include a challenging objective function, such as having multiple local optima or lacking an analytic expression. In such settings, classical gradient-based optimization algorithms can perform poorly, and therefore researchers have explored various alternative approaches [1]. One class of such algorithms is the *nature-inspired metaheuristics*, which include evolutionary algorithms (EAs), particle swarm optimization (PSO), and SA. The latter was inspired by the physics of the annealing process in metals: initially a metal has a high temperature and so particles have a high probability of moving to a higher energy state. As the metal cools in the annealing process, particles are more likely to move to a lower energy level than to a higher one. The process is completed once the system has reached the lowest possible energy level, typically its temperature of equilibrium with the environment. In the realm of global optimization, these mechanics have been translated into a heuristic search, which starts with an initial vector, namely, a candidate solution. During the search, the current vector is perturbed so that new vectors in its vicinity are obtained. These vectors are evaluated and replace the original vector if: (a) they are better, or (b) they are worse and with probability p , which is analogous to the energy state changes. As p decreases, the search is transformed from being explorative to being more localized. Two main parameters of the SA algorithm are the *annealing schedule*, namely, the duration of the search process, which is determined by the manner that the temperature is decreased, and the selection probability function, which defines the dynamic threshold for accepting a worse solution. Algorithm 1 gives a pseudocode of a baseline SA algorithm, while Section 3 gives the specific parameter settings of the SA implementation used in this study.

The underlying mechanism of the SA algorithm was originally proposed by Metropolis et al. [6], while the more common annealing-inspired formulation was later proposed by Černý [7] and Kirkpatrick et al. [8]. Since then the algorithm has been widely used in the literature, and some recent examples include [9] in finance, [10] in machine learning, [11] in chemical engineering and [12] in production line machine scheduling.

2.2. Expensive black-box problems

Computationally expensive optimization problems are common across engineering and science. Typically in such problems, candidate designs are parameterized as vectors of design variables and sent to the simulation for evaluation, as shown in **Figure 1**.

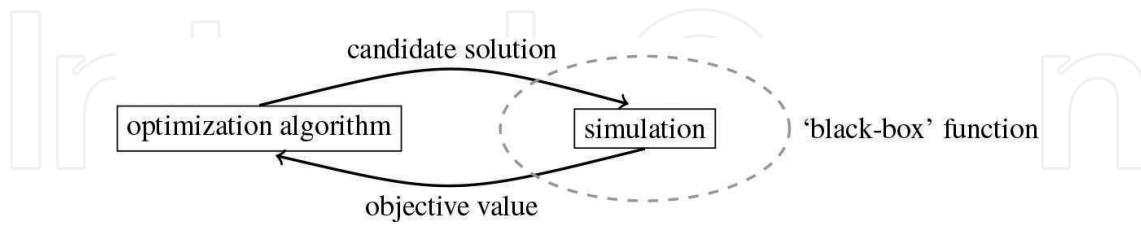


Figure 1. The layout of an expensive black-box optimization problem. The optimization algorithm generates candidate solutions, and these are evaluated by the simulation, which acts as a ‘black-box’ function, to obtain their corresponding objective values.

As mentioned in Section 1, metamodels are often used in such settings to alleviate the high computational cost of the simulation runs [2, 3]. However, the integration of metamodels into the search introduces two main challenges:

- **Prediction uncertainty:** Due to the high cost of the simulation runs only a small number of designs can be evaluated, which in turn degrades the prediction accuracy of the metamodel and leads to optimization with uncertainty regarding the validity of the predicted objective values [13]. To address this, the metamodel needs to be updated during the search to ensure that its accuracy is sufficient to drive the search to a correct final result. To accomplish this, the proposed algorithm is structured based on the TR approach [14]. In this way, the algorithm performs a sequence of *trial steps* that are constrained to the TR, namely, the region where the metamodel is assumed to be accurate. Based on the success of the trial step, namely, if a new optimum has been found, the TR and the set of vectors are updated. Section 3 presents a detailed description of the TR approach implemented in this study.
- **Metamodel suitability:** Various metamodel variants have been proposed, but the optimal variant is problem dependant and is typically unknown *a priori* [15]. To address this, ensembles employ multiple metamodels concurrently and combine their predictions into a single one to obtain a more accurate prediction [4, 5, 16]. **Figure 2** shows an example based on the Rosenbrock function.

The ensemble *topology*, namely, which metamodel variants are incorporated, is typically determined *a priori* and is unchanged during the search. However, the topology directly affects the prediction accuracy, and hence an unsuitable topology can degrade the search effectiveness. As an example, RBFs, RBFN and Kriging metamodels (described in Appendix 1) were used to generate several ensemble topologies. The same testing and training samples were used with all the topologies (sized 30 and 20 vectors, respectively), such that each ensemble was trained with training sample and its prediction accuracy was tested on

the testing sample. The prediction accuracy was measured both with the root mean squared error (RMSE),

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\hat{f}(x_i) - f(x_i) \right)^2} \tag{1}$$

and the R^2 indicator,

$$R^2 = \frac{S_t - S_r}{S_t}, \tag{2a}$$

where

$$S_t = \sum (f(x_i) - \overline{f(x_i)})^2, \quad S_r = \sum (f(x_i) - \hat{f}(x_i))^2, \tag{2b}$$

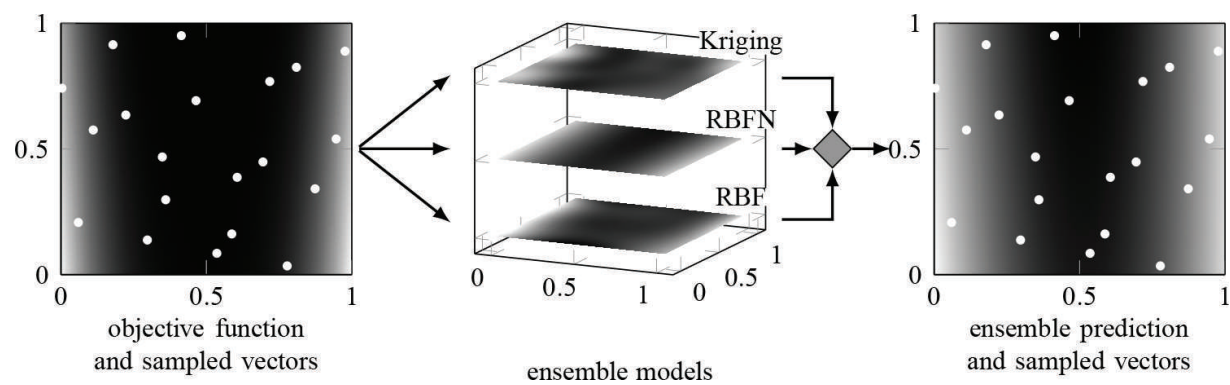


Figure 2. An ensemble topology consisting of RBF, RBFN and Kriging metamodels. The overall prediction is the aggregation of the individual predictions.

Ensemble topology						
	R+RN		R+K		RN+K	
Function	RMSE	R^2	RMSE	R^2	RMSE	R^2
Ackley-5D	1.172e+00	-1.830e-02	1.630e+00	-2.613e-02	1.176e+00	-9.521e-01
Rastrigin-10D	3.149e-02	-2.407e-01	2.471e-02	-1.812e-01	2.907e-02	-3.381e-01
Rosenbrock-20D	6.101e-04	-3.566e-01	4.968e-04	-4.541e-01	3.859e-04	-1.410e-01
Schwefel 2.13-30D	1.003e-03	-9.573e-01	2.234e-04	-7.182e-02	3.212e-04	-1.321e-01

Note: R: RBF, RN: RBF network, K: Kriging.

Table 1. Statistics for prediction accuracies of different topologies.

where $f(x)$, and $\hat{f}(x)$ are the true and the ensemble predicted values, respectively, and x_i , $i = 1 \dots n$ are the testing vectors. **Table 1** presents the test results, from which it follows that the prediction accuracy varied with the topology and that no single topology was the overall best.

Addressing this issue, the algorithm proposed in this study selects the most suitable ensemble topology during the search, as explained in the following section.

3. Proposed algorithm

This section describes the algorithm proposed in this study, which is designed to address the issues described in Sections 1 and 2. The proposed algorithm operates in five main steps, as follows:

Step 1. Initialization: The algorithm begins by generating an initial sample of vectors based on the optimal Latin hypercube design (OLHD) method [17]. The method ensures that the resultant sample is space-filling, namely, adequately covers the search space, which in turn improves the prediction accuracy of the metamodels. The OLHD method exploits patterns of point locations for optimal Latin hypercube designs based on a variation of the maximum distance criterion to produce near-optimal designs efficiently. After generating the sample, the vectors are evaluated with the true expensive function.

Step 2. Ensemble selection:

Step 2.1. Following the cross-validation (CV) procedure [18], the vectors that have been evaluated so far are split into a training set and a testing set. In turn, each candidate metamodel variant is trained with the training set and is tested on the testing set. The prediction accuracy is measured with the root mean squared error (RMSE), which is calculated as

$$\varepsilon_j = \sqrt{\frac{1}{l} \sum_{i=1}^l \left(m_j(x_i) - f(x_i) \right)^2}, \quad (3)$$

where x_i , $i = 1 \dots l$, are the vectors in the testing set, $f(x_i)$ is the exact and already known function value at x_i , and $m_j(x)$ is the prediction of the j th metamodel variant that has been trained with the training set.

Step 2.2. The evaluated vectors are again split into a training and a testing set. For each candidate ensemble topology, the following steps are performed:

- Each metamodel variant that is active in the ensemble is assigned an *ensemble weight*, which is calculated as

$$w_i = \frac{\varepsilon_i^{-1}}{\sum_{j=1}^{n_m} \varepsilon_j^{-1}}, \quad (4)$$

and the overall ensemble prediction is then given by

$$\psi(x) = \sum_{i=1}^{n_m} w_i m_i(x), \quad (5)$$

where n_m is the number of participating metamodels in the ensemble and $m_i(x)$ is a metamodel that has been trained with the new training sample.

- The prediction accuracy of the ensemble is estimated based on its RMSE on the testing set

$$\varepsilon = \sqrt{\frac{1}{k} \sum_{i=1}^k \left(\psi(x_i) - f(x_i) \right)^2}, \quad (6)$$

where $x_i, i = 1 \dots k$, are the vectors in the testing set.

Step 2.3. After repeating the process with all the candidate topologies, the one having the lowest RMSE, as defined in Eq. (6), is selected for the current optimization iteration. A new ensemble is then trained based on the selected topology and on all the evaluated vectors. This ensemble will be used in the following step.

It is emphasized that during the above process, the true expensive function is not used, and hence the process requires negligible computational resources.

For the specific implementation in this chapter, three well-established metamodels were considered for the ensembles: RBF, RBFN and Kriging that are all described in Appendix 1, while **Table 2** presents the candidate ensemble topologies. Also, the split ratio between the training and testing vectors was calibrated by numerical experiments, as described in Appendix 2.

Step 3. *Optimization trial step*: The proposed algorithm now seeks an optimum based on the ensemble prediction in the bounded TR around the current best solution (x_b), namely

$$\mathfrak{S} = \{x : \|x - x_b\|_2 \leq \Delta\}, \quad (7)$$

where Δ is the TR radius. The search is performed by using a hybrid approach [19], which is composed of an initial search performed by a SA algorithm, and followed by a localized refinement of the solution with a sequential quadratic programming (SQP) algorithm. The main settings of the SA algorithm were based on existing studies [20–22], as follows:

- Initial temperature: $T_{\max} = 1000 \cdot d$ where d is the function dimension. This was done to increase the annealing schedule for higher dimensional problems, which require a more extensive search.
- Final temperature (stopping condition): $T \leq 10^{-8}$.
- Temperature decrease function: A 5% decrease, namely

$$T(t) = 0.95 \cdot T(t-1) \quad (8)$$

Metamodels participating in the ensemble			
Index	RBF	RBFN	Kriging
1	•		
2		•	
3			•
4	•	•	
5	•		•
6		•	•
7	•	•	•

Table 2. Candidate ensemble topologies.

where t is the current time-step counter. This way the temperature initially decreases at a slow rate, which assists the exploration search, but the reduction is much faster in later stages when the search is localized.

- Acceptance probability function: A decaying exponent, namely

$$p(T) = \exp\left(-\frac{f(x_n)-f(x_c)}{T(t)}\right)$$

(9)

where x_n is the new vector being examined and x_c is the current vector.

During the entire process, objective values are obtained only from the ensemble at a negligible computational cost, and hence the SA and SQP were able to evaluate a large number of candidate vectors.

Step 4. *TR updates*: The best vector found in the previous step (x^*) is evaluated with the true expensive function, and the following updates are performed [14]:

- If $f(x^*) < f(x_b)$: The trial step was successful since the vector found was indeed better than the current best one. This indicates that the ensemble prediction is accurate, and accordingly the TR radius is doubled.
- If $f(x^*) \geq f(x_b)$ and there are sufficient vectors in the TR: The search was unsuccessful since the solution found is not better than the current best one. However, since there are enough vectors in the TR to train the metamodels the failure is attributed to the TR being too large, and accordingly the TR radius is halved.
- If $f(x^*) \geq f(x_c)$ and there are insufficient vectors in the TR: As above but now the failure is attributed to having too few vectors in the TR to train the metamodels with. Therefore, a new vector is sampled in the TR, as explained below.

The implementation described above differs from the classical TR framework in two main aspects:

- The TR is contracted only if the number of vectors in the TR is above a threshold n , which is done to avoid premature convergence. The threshold parameter was calibrated through numerical experiments, as described in Appendix 2.
- In Step 4, a new vector may be sampled in the TR to improve the prediction accuracy. This vector should be located in a region that is sparse with vectors, namely away from existing TR vectors. To accomplish this efficiently, the proposed algorithm generates a Latin hypercube design (LHD) sample of vectors in the TR and selects the one having the largest minimal distance (max-min criterion) from the existing TR vectors.

To conclude the description, Algorithm 2 gives the pseudocode of the proposed algorithm.

4. Performance analysis

4.1. Benchmark tests based on mathematical test functions

To evaluate its effectiveness, the proposed algorithm was applied to the established set of mathematical functions [23]: Ackley, Griewank, Rastrigin, Rosenbrock, Schwefel 2.13 and Weierstrass in dimensions 5–40, as listed in **Table 3**. These test functions represent challenging features such as high multimodality, deceptive landscapes and noise, and are therefore adequate for the testing purpose.

For a comprehensive evaluation, the proposed algorithm was benchmarked against the following four reference algorithms:

- V1: A variant of the proposed algorithm, which is identical to it in operation, except that it used a *single metamodel* (RBF), but no ensembles.
- V2: A variant of the proposed algorithm, which is identical to it in operation, except that it used a *fixed ensemble*, which consisted of the RBF, RBFN and Kriging metamodels, but without any topology selection.

Function	Definition, $f(x)=$	Domain
Ackley	$-20 \exp(-0.2 \sqrt{\sum_{i=1}^d x_i^2 / d}) - \exp(\sum_{i=1}^d \cos(2\pi x_i) / d) + 20 + e$	$[-32, 32]^d$
Griewank	$\sum_{i=1}^d \{x_i^2 / 4000\} - \prod_{i=1}^d \{\cos(x_i / \sqrt{i})\} + 1$	$[-100, 100]^d$
Rastrigin	$\sum_{i=1}^d \{x_i^2 - 10 \cos(2\pi x_i) + 10\}$	$[-5, 5]^d$
Rosenbrock	$\sum_{i=1}^{d-1} \{100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\}$	$[-10, 10]^d$
Schwefel 2.13	$\sum_{i=1}^d \left\{ \sum_{j=1}^d [(a_{i,j} \sin(\alpha_j) + b_{i,j} \cos(\alpha_j)) - (a_{i,j} \sin(x_j) + b_{i,j} \cos(x_j))] \right\}^2$	$[-\pi, \pi]^d$
Weierstrass	$\sum_{i=1}^d \left\{ \sum_{k=0}^{20} 0.5^k \cos(2\pi 3^k (x_i + 0.5)) \right\} - d \sum_{k=0}^{20} 0.5^k \cos(\pi 3^k)$	$[-0.5, 0.5]^d$

Table 3. Mathematical test functions.

- Evolutionary algorithm with periodic sampling (EA-PS): The algorithm leverages on the concepts in references [24, 25]. It uses a Kriging metamodel and a real-coded evolutionary algorithm (EA). The accuracy of the metamodel is maintained by periodically evaluating a small subset of the EA population with the true objective function, and using these sampled vectors to update the metamodel.
- *Expected improvement with covariance matrix adaptation evolutionary strategy* (CMA-ES) (EI-CMA-ES) [26]: The algorithm combines a covariance matrix adaptation evolutionary strategy (CMA-ES) algorithm with a Kriging metamodel, and uses the expected improvement (EI) framework to select new vectors for evaluation based both on the response and uncertainty in the metamodel prediction. This way the algorithm balances between a local search around the current best solution, and an explorative search in less-visited regions of the search space.

This testing set-up was used for several reasons: (i) any gains brought by using ensembles are highlighted through the comparisons to the V1 and V2 algorithms and (ii) the performance of the proposed algorithm is benchmarked against representative metamodel-assisted algorithms from the literature, namely, the EA-PS and EI-CMA-ES algorithms. Each algorithm-objective function combination was tested over 30 runs to support a valid statistical analysis, and the number of evaluations of the true objective function was limited to 200, to represent the tight optimization budget in expensive problems.

Tables 4 and **5** give the resultant test statistics of mean, standard deviation (SD), median, minimum (best) and maximum (worst) objective value for each algorithm-objective function combination. It also gives the statistic α that indicates the significance level at which the results of the proposed algorithm were better than those of the competing algorithms, measured at either the 0.01 or 0.05 levels, while an empty entry indicates no such statistically significant advantage. The α statistic was obtained by using the Mann-Whitney non-parametric test [27].

Test results show that the proposed algorithm performed well, as it obtained the best mean and median statistics in 8 out of 12 cases. Its results also had a statistically significant advantage over the other algorithms in 30 out of 48 cases. The performance advantage of the proposed algorithm was particularly pronounced in the high-dimensional cases, where it obtained the best mean and median in five out of six test functions. In terms of repeatability of performance, its SD was often slightly higher but was competitive with the best SD in each test case.

Overall, the proposed algorithm consistently outperformed the V1 and V2 variants, which shows that selecting the ensemble topology during the search improved the search effectiveness, both when compared to using a fixed metamodel or a fixed ensemble topology. Also, the proposed algorithm consistently outperformed the two reference algorithms from the literature, which shows that it compares well with existing approaches.

The analysis of the experiments also examined how the ensemble topology was updated, to examine if either a single or multiple topologies were predominantly selected. Accordingly, **Figure 3** shows representative plots of the ensemble topologies selected during a run with the Ackley-10D function and another with the Rosenbrock-20D function. It follows that in both

		Proposed	V1	V2	EA-PS	EI-CMA-ES
Ackley-10	Mean	7.576e+00	1.415e+01	1.320e+01	5.241e+00	1.796e+01
	SD	8.218e+00	4.522e+00	7.843e+00	5.590e-01	1.529e+00
	Median	2.274e+00	1.547e+01	1.855e+01	5.408e+00	1.797e+01
	Min(best)	8.677e-02	2.362e+00	3.381e+00	4.098e+00	1.443e+01
	Max(worst)	1.779e+01	1.778e+01	2.046e+01	6.010e+00	1.988e+01
	α			0.01		0.01
Griewank-10	Mean	1.253e-01	1.920e-01	8.625e-01	9.579e-01	9.338e-01
	SD	1.759e-01	1.643e-01	1.401e-01	1.076e-01	2.435e-01
	Median	7.507e-02	1.267e-01	8.834e-01	9.862e-01	1.007e+00
	Min(best)	9.163e-03	3.485e-02	5.304e-01	7.146e-01	2.441e-01
	Max(worst)	6.194e-01	5.401e-01	1.022e+00	1.046e+00	1.050e+00
	α			0.01	0.01	0.01
Rastrigin-5	Mean	6.259e+00	9.031e+00	7.828e+00	7.631e+00	2.131e+01
	SD	3.695e+00	7.483e+00	8.286e+00	4.811e+00	4.890e+00
	Median	5.844e+00	7.273e+00	4.246e+00	7.226e+00	2.139e+01
	Min(best)	1.950e+00	1.005e+00	3.224e+00	1.621e+00	1.353e+01
	Max(worst)	1.189e+01	2.653e+01	3.046e+01	1.456e+01	3.006e+01
	α					0.01
Rosenbrock-5	Mean	1.443e+01	3.247e+01	1.358e+02	2.074e+02	3.701e+02
	SD	3.839e+01	7.480e+01	2.784e+02	1.640e+02	2.320e+02
	Median	2.508e+00	3.522e+00	6.636e+00	1.796e+02	3.498e+02
	Min(best)	2.465e-02	1.730e+00	4.085e+00	1.368e+01	7.677e+01
	Max(worst)	1.236e+02	2.388e+02	8.787e+02	5.617e+02	6.719e+02
	α			0.01	0.01	0.01
Schwefel-5	Mean	5.379e+02	3.809e+02	3.663e+02	5.598e+02	3.333e+02
	SD	8.894e+02	9.297e+02	6.567e+02	4.995e+02	3.227e+02
	Median	7.371e+01	2.749e+00	1.715e+02	4.804e+02	2.050e+02
	Min(best)	2.508e-02	5.862e-02	4.835e+01	5.685e+01	3.426e+01
	Max(worst)	2.202e+03	2.955e+03	2.215e+03	1.817e+03	1.080e+03
	α					
Weierstrass-10	Mean	6.620e+00	8.936e+00	8.168e+00	3.706e+00	5.909e+00
	SD	1.970e+00	1.731e+00	2.042e+00	5.593e-01	2.777e+00
	Median	6.395e+00	9.023e+00	8.494e+00	3.702e+00	5.805e+00
	Min(best)	3.786e+00	6.875e+00	5.234e+00	2.787e+00	1.657e+00
	Max(worst)	1.052e+01	1.236e+01	1.108e+01	4.627e+00	9.409e+00
	α		0.01			

Table 4. Test statistics: test functions—low dimension.

		Proposed	V1	V2	EA-PS	EI-CMA-ES
Ackley-20	Mean	6.446e+00	8.795e+00	1.947e+01	6.814e+00	1.863e+01
	SD	5.712e+00	6.302e+00	2.836e-01	2.461e-01	1.921e+00
	Median	3.885e+00	5.524e+00	1.944e+01	6.744e+00	1.934e+01
	Min(best)	2.683e+00	3.574e+00	1.909e+01	6.468e+00	1.493e+01
	Max(worst)	1.802e+01	1.804e+01	1.998e+01	7.203e+00	2.044e+01
	α		0.05	0.01	0.05	0.01
Griewank-40	Mean	1.045e+00	1.270e+00	8.192e+00	1.461e+00	1.102e+00
	SD	2.942e-02	1.185e-01	1.151e+00	6.031e-02	3.032e-02
	Median	1.040e+00	1.246e+00	8.068e+00	1.454e+00	1.096e+00
	Min(best)	1.013e+00	1.120e+00	6.567e+00	1.387e+00	1.071e+00
	Max(worst)	1.114e+00	1.475e+00	1.035e+01	1.595e+00	1.157e+00
	α		0.01	0.01	0.01	0.01
Rastrigin-20	Mean	6.490e+01	6.501e+01	1.492e+02	1.223e+02	2.105e+02
	SD	3.753e+01	1.703e+01	2.754e+01	1.219e+01	3.914e+01
	Median	4.808e+01	6.683e+01	1.449e+02	1.230e+02	2.296e+02
	Min(best)	3.924e+01	4.203e+01	1.144e+02	1.046e+02	1.395e+02
	Max(worst)	1.568e+02	8.807e+01	1.983e+02	1.429e+02	2.507e+02
	α			0.01	0.01	0.01
Rosenbrock-20	Mean	5.653e+02	1.005e+03	9.013e+03	8.435e+02	3.967e+03
	SD	2.067e+02	5.794e+02	5.142e+03	3.012e+02	9.406e+02
	Median	5.807e+02	8.295e+02	8.112e+03	7.782e+02	3.685e+03
	Min(best)	2.042e+02	5.281e+02	4.165e+03	4.676e+02	3.141e+03
	Max(worst)	8.756e+02	2.497e+03	2.271e+04	1.439e+03	6.144e+03
	α		0.01	0.01	0.05	0.01
Schwefel-40	Mean	7.503e+05	8.786e+05	2.322e+06	1.774e+06	1.667e+06
	SD	2.173e+05	2.506e+05	5.317e+05	2.509e+05	6.520e+05
	Median	7.074e+05	8.420e+05	2.369e+06	1.744e+06	1.528e+06
	Min(best)	4.989e+05	5.815e+05	1.666e+06	1.415e+06	8.933e+05
	Max(worst)	1.126e+06	1.348e+06	3.186e+06	2.104e+06	2.871e+06
	α			0.01	0.01	0.01
Weierstrass-40	Mean	2.730e+01	4.048e+01	5.105e+01	2.343e+01	3.598e+01
	SD	4.414e+00	4.106e+00	2.138e+00	1.265e+00	1.463e+01
	Median	2.445e+01	4.100e+01	5.135e+01	2.304e+01	2.597e+01
	Min(best)	2.365e+01	3.351e+01	4.817e+01	2.214e+01	2.100e+01
	Max(worst)	3.419e+01	4.685e+01	5.333e+01	2.567e+01	5.817e+01
	α		0.01	0.01		

Table 5. Test statistics: test functions—high dimension.

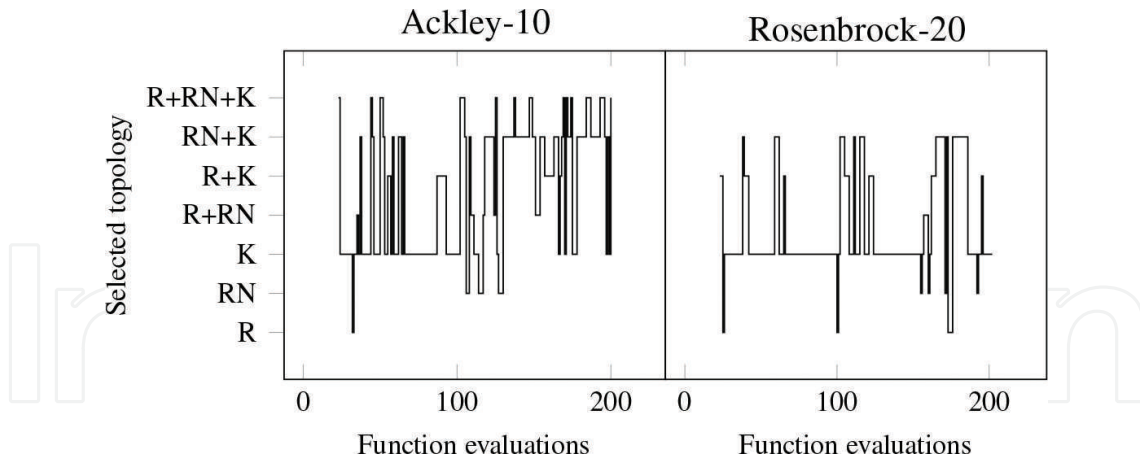


Figure 3. Ensemble topologies selected during two test runs. Abbreviations: K: Kriging, R: RBF, RN: RBF network.

cases, different topologies were selected, which indicate that no single topology was the overall optimal, and further justifies the proposed approach.

4.2. Engineering test problem

The proposed algorithm was also applied to a representative simulation-driven engineering problem, where the goal is to find an airfoil shape, which maximizes the lift L and minimizes the drag (aerodynamic friction) D at some prescribed flight conditions. In practise, the design requirements for airfoils are specified in terms of the non-dimensional *lift and drag coefficients*, c_l and c_d , respectively, defined as

$$c_l = \frac{L}{\frac{1}{2}\rho V^2 S} \quad (10a)$$

$$c_d = \frac{D}{\frac{1}{2}\rho V^2 S} \quad (10b)$$

where L and D are the lift and drag forces, respectively, ρ is the air density, V is the aircraft speed, and S is the reference area, such as the wing area. The relevant flight conditions are the aircraft altitude, speed and angle of attack (AOA) that is the angle between the aircraft velocity and the airfoil *chord line*. **Figure 4** gives a schematic layout of the airfoil problem.

Candidate airfoils were represented with the Hicks-Henne method [28], such that an airfoil profile was defined as

$$y = y_b + \sum_{i=1}^h \alpha_i b_i(x), \quad (11)$$

where y_b is a baseline profile taken here to be the NACA0012 symmetric profile, and b_i are the shape basis functions [29].

$$b_{i(x)} = \left[\sin \left(\pi x^{\frac{\log(0.5)}{\log(i/(h+1))}} \right) \right]^4, \quad (12)$$

where $\alpha_i \in [-0.01, 0.01]$ are the variables, as shown in **Figure 4**. Ten basis functions were used for the upper and lower airfoil profiles, respectively, which resulted in a total of 20 variables per airfoil. Also, for structural integrity the thickness of an airfoil between 0.2 and 0.8 of its chord line was required to be greater than a critical thickness $t^* = 0.1$. The lift and drag coefficients of candidate airfoils were obtained by using *XFOIL*, an aerodynamics simulation code for subsonic isolated airfoils [30]. Each airfoil evaluation required up to 30 s on a desktop computer, so evaluations were not prohibitively expensive and the tests could be completed within a reasonable time.

Based on the above discussion, the objective function used was

$$f = -\frac{c_l}{c_d} + p, \quad p = \begin{cases} \frac{t^*}{t} \cdot \left| \frac{c_l}{c_d} \right| & \text{if } t < t^* \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where p is the penalty for violation of the thickness constraint. The flight conditions were an altitude of 30,000 ft, a speed of Mach 0.7, namely 70% of the speed of sound, and an AOA of 2° .

Tests were performed along the set-up of Section 4.1, and **Table 6** gives the resultant test statistics. The trends are consistent with those of the test functions, and the proposed algorithm

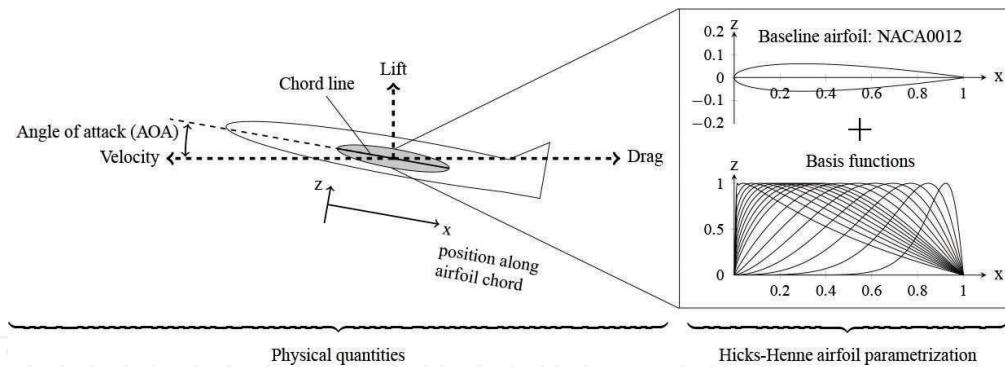


Figure 4. The layout of the airfoil problem: main components (left) and parameterization (right).

	Proposed	V1	V2	EA-PS	EI-CMA-ES
Mean	-3.376e+00	-3.279e+00	-3.368e+00	-3.231e+00	-3.278e+00
SD	1.242e-01	6.683e-02	1.008e-01	7.164e-02	9.597e-02
Median	-3.355e+00	-3.274e+00	-3.352e+00	-3.227e+00	-3.267e+00
Min(best)	-3.624e+00	-3.393e+00	-3.533e+00	-3.335e+00	-3.395e+00
Max(worst)	-3.158e+00	-3.161e+00	-3.214e+00	-3.134e+00	-3.098e+00
α		0.01		0.01	0.05

Table 6. Test statistics for the airfoil problem.

outperformed the other candidate algorithms also here. It obtained the best mean and median statistics, and had a competitively low SD.

5. Conclusion

While computer simulations can improve the efficiency of the engineering design process, they also introduce new optimization challenges. Metamodels aim to alleviate the challenge of a high evaluation cost by providing computationally cheaper approximations of the true expensive function.

While metamodels can significantly improve the search effectiveness, they also introduce various challenges, such as identifying an optimal combination of metamodel variants, and effectively searching the metamodel landscape. To address these issues, this chapter has proposed a hybrid algorithm that uses SA to perform a global search, and it then refines the solutions with an SQP local search. To further enhance its effectiveness, the proposed algorithm uses ensembles of metamodels and selects the most suitable ensemble topology during the search. Lastly, to ensure convergence to an optimum of the true expensive function in the light of the inherent metamodel inaccuracies, the proposed algorithm operates within a TR approach such that the optimization is performed through a series of trial steps.

In an extensive performance analysis, the proposed algorithm was benchmarked against two implementations without selection of the ensemble topology, and two reference algorithms from the literature, which also do not use topology adaption. Analysis of the results shows that the proposed algorithm consistently outperformed the other algorithms: it achieved better results in 30 out of 48 cases with mathematical test functions, and also performed well with a simulation-driven problem. Its performance advantage was evident from the superior mean and median statistics that was obtained across the tests and was particularly pronounced in the high-dimensional problems.

The analysis also showed that during the optimization search the optimal topology continuously varied during the search, and that no single topology was the overall optimal. This observation further supports the approach proposed of selecting the ensemble topology during the search.

Overall, the solid performance of the proposed algorithm shows the merit of the hybrid SA+SQP algorithm proposed. It also suggests that the proposed algorithm could be applied to problems from a variety of academic domains, such as scheduling, systems engineering and model calibration, to name a few.

Appendix 1: Candidate metamodels

This appendix describes the metamodels used, as follows:

- Kriging: A statistical metamodel that combines a global 'drift' function and a local adjustment based on the correlation between the sample vectors. The metamodel replicates the

observed responses precisely (Lagrangian interpolation). For a constant drift function, the metamodel becomes

$$m(\mathbf{x}) = \beta + \kappa(\mathbf{x}), \quad (14)$$

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

$$\text{Cov}[\kappa(\mathbf{x})\kappa(\mathbf{y})] = \sigma^2 c(\theta, \mathbf{x}, \mathbf{y}), \quad (15)$$

where $c(\theta, \mathbf{x}, \mathbf{y})$ is a user-prescribed correlation function. With the Gaussian correlation function [3]

$$c(\theta, \mathbf{x}, \mathbf{y}) = \prod_{i=1}^d \exp\left(-\theta(x_i - y_i)^2\right), \quad (16)$$

the above metamodel becomes

$$m(\mathbf{x}) = \hat{\beta} + \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1} (\mathbf{f} - \mathbf{1}\hat{\beta}) \quad (17)$$

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

$$\mathbf{r}^T = [c(\theta, \mathbf{x}, \mathbf{x}_1), \dots, c(\theta, \mathbf{x}, \mathbf{x}_n)] \quad (18)$$

The estimated drift coefficient $\hat{\beta}$ and variance $\hat{\sigma}^2$ are calculated as

$$\hat{\beta} = (\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1})^{-1} \mathbf{1}^T \mathbf{R}^{-1} \mathbf{f}, \quad (19a)$$

$$\hat{\sigma}^2 = \frac{1}{n} [(\mathbf{f} - \mathbf{1}\hat{\beta})^T \mathbf{R}^{-1} (\mathbf{f} - \mathbf{1}\hat{\beta})] \quad (19b)$$

For an isotropic (single correlation parameter) Kriging metamodel the optimal value of the parameter is obtained by maximizing the metamodel likelihood [31]

$$\theta^* : \max\{|\mathbf{R}|^{1/n} \hat{\sigma}^2\} \quad (20)$$

- *Radial basis functions* (RBF): The metamodel approximates the true objective function with a set of basis functions, namely

$$m(\mathbf{x}) = \alpha_i \sum_{i=1}^n \phi_i(\mathbf{x}) + c \quad (21)$$

where $\phi_i(\mathbf{x})$ are basis functions of the form

$$\phi_i(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{x}_i\|_2), \quad (22)$$

where \mathbf{x}_i is a sampled vector and c is a constant. The coefficients α_i and c are determined from the interpolation conditions

$$m(\mathbf{x}_i) = f(\mathbf{x}_i), i = 1 \dots n, \quad (23a)$$

$$\sum_{i=1}^n \alpha_i = 0. \quad (23b)$$

In this study, the widely used Gaussian basis function [32] was used:

$$\phi_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|_2^2}{\tau}\right), \quad (24)$$

where τ controls the width of the Gaussians, and is determined by cross-validation [33, 34].

- *Radial basis function network (RBFN)*: A variant of the RBF approach but in which the number of basis functions is smaller than the sample size, which can improve the prediction accuracy in certain scenarios. The metamodel is given by

$$m(\mathbf{x}) = \sum_{j=1}^{\hat{n}} \alpha_j \phi_j(\mathbf{x}), \quad (25)$$

where the coefficients α_i are determined from the least-squares interpolation conditions

$$\Phi^T \Phi \alpha = \Phi^T \mathbf{f} \quad (26)$$

where

$$\Phi_{i,j} = \phi_j(\mathbf{x}_i) \quad (27)$$

and $\mathbf{x}_i, i = 1 \dots n$, are the sample vectors, \mathbf{f} is the vector of corresponding objective function values, and $\phi_j(\mathbf{x}), j = 1 \dots \hat{n}$ are the basis functions, which in this study were taken as the Gaussian functions described above. The basic function centres \mathbf{x}_j are obtained by clustering the sampled vectors and using the resultant cluster centres.

Appendix 2: Parameter sensitivity analysis

As described in Section 3, the proposed algorithm relies on two main parameters: (i) the minimum number (n) of TR vectors needed to allow a TR contraction, and (ii) the split ratio (s) between the training and testing subsets, as used in estimating the accuracy of candidate metamodel and ensembles.

To calibrate these parameters, a set of numerical experiments were performed where different parameter settings were used, and for each setting the algorithm was tested with the Rastrigin-10D, Rosenbrock-10D, Rastrigin-20D and Rosenbrock-20D functions. The parameter settings examined were:

- n : $0.1d$, $0.5d$, d , where d is the dimension of the objective function.
- s : 80–20, 60–40, 40–60, in percent.

Table 7 gives the resultant test statistics of mean objective value, rank per objective function and the overall rank based on each setting where a lower score is better. From these results it follows:

(a) Statistics: different TR vectors threshold (n)						
$n = 0.1d$			$n = 0.5d$		$n = d$	
Function	Mean	Rank	Mean	Rank	Mean	Rank
Ras-10	4.598e+01	02	4.827e+01	03	3.530e+01	01
Ros-10	1.233e+02	02	4.393e+01	01	1.317e+02	03
Ras-20	9.256e+01	03	8.656e+01	02	7.997e+01	01
Ros-20	4.503e+02	01	5.361e+02	02	8.518e+02	03
Overall		08		08		08
(b) Statistics: different split ratios (s)						
80–20			60–40		40–60	
Function	Mean	Rank	Mean	Rank	Mean	Rank
Ras-10	4.598e+01	02	3.981e+01	01	5.414e+01	03
Ros-10	1.233e+02	02	1.266e+02	03	4.929e+01	01
Ras-20	9.256e+01	03	8.899e+01	02	7.970e+01	01
Ros-20	4.503e+02	01	5.578e+02	03	5.318e+02	02
Overall		08		09		07

Note: ratios are in percent.

Table 7. Parameter sensitivity analysis results.

- n : Performance was similar across the different settings and accordingly the intermediate setting of $n = 0.5d$ was selected.
- s : The best performing split ratio was 40–60 between the training and testing subsets.

The above settings were then used during the numerical experiments described in Section 4.

Author details

Yoel Tenne

Address all correspondence to: y.tenne@ariel.ac.il

Department of Mechanical and Mechatronic Engineering, Ariel University, Ariel, Israel

References

- [1] Tenne Y, Goh CK, editors. Computational intelligence in expensive optimization problems. Vol. 2 of Evolutionary Learning and Optimization. Berlin: Springer; 2010.
- [2] Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker KP. Surrogate-based analysis and optimization. *Progress in Aerospace Science*. 2005;41:1–28.
- [3] Forrester AIJ, Keane AJ. Recent advances in surrogate-based optimization. *Progress in Aerospace Science*. 2008;45(1–3):50–79.
- [4] Acar E. Various approaches for constructing an ensemble of metamodels using local measures. *Structural and Multidisciplinary Optimization*. 2010;42(6):879–896.
- [5] Zhou X, Ma Y, Cheng Z, Liu L, Wang J. Ensemble of metamodels with recursive arithmetic average. In: Luo Q, editor. *Proceedings of the International Conference on Industrial Mechatronics and Automation–ICIMA 2010*. Piscataway, NJ: IEEE; 2010. pp. 178–182.
- [6] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Edward T. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*. 1953;21(6):1087–1092.
- [7] Černý V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*. 1985;45(1):41–51.
- [8] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science*. 1983;220(4598):471–480.
- [9] Xiang Y, Gubian S, Suomela B, Hoeng J. Generalized simulated annealing for global optimization: The gensa package. *The R Journal*. 2013;5(1):13–28.

- [10] Belloni A, Liang H, Tengyuan Narayanan, Rakhlin A. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. *The Journal of Machine Learning Research*. 2015;40:240–265.
- [11] Rodríguez DA, Oteiza PP, Brignole NB. Simulated annealing optimization for hydrocarbon pipeline networks. *Journal of Industrial Engineering and Chemical Research*. 2013;52(25):8579–8588.
- [12] Sivasankaran P, Sornakumar T, Panneerselvam R. Design and comparison of simulated annealing algorithm and grasp to minimize makespan in single machine scheduling with unrelated parallel machines. *Intelligent Information Management*. 2010;2:406–416.
- [13] Jin Y, Olhofer M, Sendhoff B. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*. 2002;6(5):481–494.
- [14] Conn AR, Scheinberg K, Toint PL. On the convergence of derivative-free methods for unconstrained optimization. In: Iserles A, Buhmann MD, editors. *Approximation Theory and Optimization: Tributes to M.J.D. Powell*. Cambridge, New York: Cambridge University Press; 1997. pp. 83–108.
- [15] Simpson TW, Poplinski JD, Koch PN, Allen JK. Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*. 2001;17:129–150.
- [16] Jin Y, Sendhoff B. Reducing fitness evaluations using clustering techniques and neural network ensembles. In: Deb K, et al., editors. *Proceedings of the Genetic and Evolutionary Computation Conference—GECCO 2004*. Berlin, Heidelberg: Springer-Verlag; 2004. pp. 688–699.
- [17] Viana FAC, Venter G, Balabanov V. An algorithm for fast optimal Latin hypercube design of experiments. *International Journal of Numerical Methods in Engineering*. 2009;82(2):135–156.
- [18] Burnham KP, Anderson DR. *Model Selection and Inference: A Practical Information-Theoretic Approach*. New York: Springer; 2002.
- [19] Mininno E, Neri F. A memetic differential evolution approach in noisy optimization. *Journal of Soft Computing*. 2010;2:111–135.
- [20] Yang B, Guang L, Säntti T, Plosila J. Parameter-optimized simulated annealing for application mapping on networks-on-chip. In: Hamadi Y, Schoenauer M, editors. *Learning and Intelligent Optimization: 6th International Conference, LION 6, Paris, France, January 16–20, 2012, Revised Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012. pp. 307–322.
- [21] Frausto-Solis J, Alonso-Pecina F. Analytically tuned parameters of simulated annealing for the timetabling problem. *WSEAS Transactions on Advances in Engineering Education*. 2008;5(5):272–281.
- [22] Park MW, Kim YD. A systematic procedure for setting parameters in simulated annealing algorithms. *Computers and Operations Research*. 1998;25(3):207–217.

- [23] Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A, et al. 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; 2005.KanGAL2005005.
- [24] Ratle A. Optimal sampling strategies for learning a fitness model. In: The 1999 IEEE Congress on Evolutionary Computation–CEC 1999. Piscataway, New Jersey: IEEE; 1999. pp. 2078–2085.
- [25] de Jong KA. Evolutionary Computation: A Unified Approach. Cambridge, Massachusetts: MIT Press; 2006.
- [26] Büche D, Schraudolph NN, Koumoutsakos P. Accelerating evolutionary algorithms with Gaussian process fitness function models. IEEE Transactions on Systems, Man, and Cybernetics–Part C. 2005;35(2):183–194.
- [27] Sheskin DJ. Handbook of Parametric and Nonparametric Statistical Procedures. 4th ed. Boca Raton, Florida: Chapman and Hall; 2007.
- [28] Hicks RM, Henne PA. Wing design by numerical optimization. Journal of Aircraft. 1978;15(7):407–412.
- [29] Wu HY, Yang S, Liu F, Tsai HM. 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. pp. 1–11. AIAA 2003-4095.
- [30] Drela M, Youngren H. XFOIL 6.9 User Primer. Cambridge, MA: Massachusetts Institute of Technology; 2001.
- [31] Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and analysis of computer experiments. Statistical Science. 1989;4(4):409–435.
- [32] Powell MJD. Radial basis function methods for interpolation of functions of many variables. In: Lipitakis EA, editor. Proceedings of the 5th Hellenic-European Conference on Computer Mathematics and its Applications (HERCMA-01). Athens, Hellas: LEA Press; 2001. pp. 2–24.
- [33] Golberg MA, Chen CS, Karur SR. Improved multiquadric approximation for partial differential equations. Engineering Analysis with Boundary Elements. 1996;18:9–17.
- [34] Benoudjit N, Archambeau C, Lendasse A, Lee J, Verleysen M. Width optimization of the gaussian kernels in radial basis function networks. In: Verleysen M, editor. Proceedings of the 10th European Symposium on Artificial Neural Networks–ESANN 2002. Evere D-Side; 2002. pp. 425–432.

