A software for parameter optimization with Differential Evolution Entirely Parallel method

Konstantin Kozlov[1], Alexander M. Samsonov[1,2] and Maria Samsonova[1]

[1] Mathematical Biology and Bioinformatics Lab, IAMM, Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia
[2] Ioffe Institute, Saint Petersburg, Russia

ABSTRACT

Summary. Differential Evolution Entirely Parallel (DEEP) package is a software for finding unknown real and integer parameters in dynamical models of biological processes by minimizing one or even several objective functions that measure the deviation of model solution from data. Numerical solutions provided by the most efficient global optimization methods are often problem-specific and cannot be easily adapted to other tasks. In contrast, DEEP allows a user to describe both mathematical model and objective function in any programming language, such as *R*, *Octave* or *Python* and others. Being implemented in C, DEEP demonstrates as good performance as the top three methods from CEC-2014 (Competition on evolutionary computation) benchmark and was successfully applied to several biological problems.

Availability. DEEP method is an open source and free software distributed under the terms of GPL licence version 3. The sources are available at http://deepmethod.sourceforge.net/ and binary packages for Fedora GNU/Linux are provided for RPM package manager at https://build.opensuse.org/project/repositories/home:mackoel:compbio.

INTRODUCTION

The estimation of dynamical model parameters minimizing the discrepancy between model solution and the set of observed data is among the most important, widely studied problems in applied mathematics, and is known as an inverse problem of mathematical modeling (*Mendes & Kell, 1998*; *Moles, Mendes & Banga, 2003*). Numerical strategies for solutions of an inverse problems often involve optimization methods. Many global and local, stochastic and deterministic optimization techniques, including the nature-inspired ones, have been developed and implemented in a wide range of free, open source and commercial software packages.

Mathematical modeling being one of the primary tools of computational systems biology provides new insights into the mechanisms that control the biological systems. It becomes very attractive to experimentalists due to predictive abilities of carefully selected models, if any.

Corresponding author
Konstantin Kozlov,
kozlov_kn@spbstu.ru,
mackoel@gmail.com

Researchers benefit from the ability of a model to predict *in silico* the consequences of a biological experiment, which was not used for training. The properties of the model are determined by the structure of the mathematical description and the values of the unknown constants and control parameters that represents the coefficients of underlying biochemical reactions. These unknowns are to be found as a best suited solution to an inverse problem of mathematical modeling, i.e., by the fitting model output to experimental observations. The parameter set is to be reliable, and different types of data are to be considered. Development of reliable and easy-to-use algorithms and programs for solution to the inverse problem remains a challenging task due to diversity and high computational complexity of biomedical applications, as well as the necessity to treat large sets of heterogeneous data.

In systems biology the commonly used global optimization algorithm is the parallel Simulated Annealing (SA) (*Chu, Deng & Reinitz, 1999*). This method requires considerable CPU time, but is capable to eventually find the global extremum and runs efficiently in parallel computations. However, the wide range of methods called Genetic Algorithms (GA) has been developed later and successfully applied to biological problems (*Spirov & Kazansky, 2002*). Modern Evolutionary algorithms such as Evolution Strategies (ESs) or Differential Evolution (DE) can outperform other methods in the estimation of parameters of several biological models (*Fomekong-Nanfack, Kaandorp & Blom, 2007*; *Fomekong-Nanfack, 2009*; *Suleimenov, 2013*). The general challenge in the efficient implementation of the global optimization methods is that they depend on problem-specific assumptions and thus are not able to be easily adapted to another problems. For example, in SA both the final result and computational time depend on the so-called cooling schedule, the success of the GA optimization strongly depends on selected mutation, recombination and selection rules, and the evolutionary algorithms heavily rely on the algorithmic parameters which define the model of evolution. Currently a lot of approaches exist based on metaheuristics for parameters estimation in biology. For example, *enhanced Scatter Search* (*Egea, Martí & Banga, 2010*), implemented in *MEIGOR* (Metaheuristics for systems biology and bioinformatics global optimization) package for *R* statistical language was reported to outperform the state-of-the-art methods (*Egea et al., 2014*). This method can provide high quality solution for integer and real parameters, however, it is computationally expensive.

We developed DEEP, a software that implements the Differential Evolution Entirely Parallel (DEEP) method introduced recently (*Kozlov & Samsonov, 2011*). The rationale behind the design of this programme was to provide an open source software with performance comparable to the competitive packages, as well as to allow a user to implement both mathematical model and comparison of solution with experimental data in any software package or programming language, such as *R*, *Octave*, *Python* or others.

PROBLEM STATEMENT

DEEP method was developed to solve the inverse problem of mathematical modeling. For a given mathematical model with parameters $q \in R^K$, where K is number of parameters, and observable data Y we seek the vector $\hat{q}$:

$$\hat{q} = \operatorname{argmin} F(q, Y) \tag{1}$$

where F is a measure of the deviation of model prediction from observable data. Additional constraints may be imposed:

$$h_j(q) = 0, \quad j = 1, \ldots, NH \tag{2}$$

$$g_m(q) \leq 0, \quad m = 1, \ldots, NG \tag{3}$$

$$q_k^L < q_k < q_k^U, \quad k \in I \subset \{1, \ldots, K\} \tag{4}$$

where NH and NG are numbers of constraints in a form of equality and inequality respectively, and $I \subset \{1, \ldots, K\}$ denotes the set of indices for parameters with box constraints. Several objective functions can be combined:

$$F(q, Y) = \overset{NF}{\underset{i=1}{\odot}} F_i(q, Y) \tag{5}$$

where $\odot$ denotes one of the aggregation methods—summation or maximization.

DIFFERENTIAL EVOLUTION ENTIRELY PARALLEL METHOD

The main algorithm

Differential Evolution (DE) was proposed in *Storn & Price (1995)* as an effective stochastic method for function minimization. DE deals with a set (population) of randomly generated parameter vectors (individuals). The population on each iteration is referred to as generation, moreover, a size of population NP is fixed.

DEEP method (*Kozlov & Samsonov, 2011*) incorporates two enhancements found in literature, as well as some elaborated modifications. These enhancements consist in the "trigonometric mutation" rule proposed in *Fan & Lampinen (2003)* and used to take into account a value of the objective function for each individual at the recombination step, and the adaptive scheme for selection of internal parameters based on the control of the population diversity developed in *Zaharie (2002)*. The motivation to select these enhancements was to make algorithm more suitable for biological problems containing big data sets and not properly defined objective function.

The key concept behind the DEEP method is the age of individual defined as a number of generations, during which the individual survived without any change of its parameter vector. The failure to update the parameter values indicates a convergence to a local minimum. To avoid it we propose to substitute the number of oldest individuals Ψ with the same number of the best ones after the predefined number of iterations Θ.

To enhance the method further we incorporated an optional *Scatter Search* step (*Egea, Martí & Banga, 2010*) that is performed each Ξ iteration, where Ξ is a control parameter. The specified number of best individuals Ψ is used to produce NP offsprings.

To increase the reliability of the DEEP method we implemented a new selection rule, described in detail in *Kozlov et al. (2013)*, in which several different objective functions are considered in order to accept or reject an offspring to new generation. This feature permits to combine different types of experimental observations and the *a priori* information in one and the same fitting procedure.

Algorithm 1.   Differential Evolution Entirely Parallel method

---

INITIALIZATION: The population is initialized randomly.
Iteration = 0
while Stopping criterion is not met **do**
   Iteration = Iteration + 1
   RECOMBINATION:
   if The predefined number of iterations Ξ passed. **then**
      Make Scatter Search Step
   else
      for all individuals in population **do**
         recombine (individual)
      end for
   end if
   EVALUATION:
   objfunc (population)
   SELECTION:
   for all offsprings **do**
      select (offspring)
   end for
   ADAPTATION:
   Update scaling and crossover parameters.
   if The predefined number of iterations Θ passed. **then**
      Substitute Ψ oldest individuals with the best ones.
   end if
   Sort the population members by age and by quality.
end while

The operation of the method is described in the Algorithm 1 insertion. The execution starts with *Initialization* block in which a set of *NP* parameter vectors $v^{g,0}$ of length K is randomly assigned the values satisfying the given constraints $q_k^L < q_k < q_k^U$, $k = 1, \ldots, K$.

The main loop of the algorithm consists of *Recombination*, *Evaluation*, *Selection* and *Adaptation* blocks that are detailed below. Calculations are terminated when the objective function variation becomes less than a predefined value during the several consecutive steps or the maximal number of generations $G_{\max}$ is exceeded. Objective function F for several vectors is calculated in parallel.

The size of population *NP*, the frequencies Θ and Ξ for old population members substitution and scatter search step together with the number Ψ of substituted individuals and maximal number of iterations $G_{\max}$ are the main control parameters of the method.

Constraints handling

Using the DEEP method, one can solve both unconstrained and constrained optimization problems. Upper and lower bounds are to be defined for each parameter for initialization.

Constraints may be imposed in the form of inequalities or equalities for a subset of parameters or their combinations. Constraints in form (2), (3) can be reduced by maximization or summation:

$$H(q,Y) = \overset{NH}{\underset{j=1}{\odot}} H_j(q,Y); \qquad G(q,Y) = \overset{NG}{\underset{m=1}{\odot}} G_m(q,Y),$$

where H_j and G_m is the violation of the corresponding constraint h_j or g_m and $\odot$ denotes one of the aggregation methods—summation or maximization.

$$F(q,Y) = F(q,Y) + H(q,Y) + G(q,Y).$$

However, the box constraints in the form (4), that may be imposed for a subset I of parameters q_k^g, $k \in I$ are to be transformed. To do it let us introduce the new parameters u_k:

$$q_k = \alpha_k + \beta_k \sin(u_k) \quad \text{or} \quad q_k = \alpha_k + \beta_k \tanh(u_k),$$

where

$$\alpha_k = (q_k^U + q_k^L)/2; \quad \beta_k = (q_k^U - q_k^L)/2.$$

Consequently, DEEP is applied to determine unconstrained parameters u_k. The impact of different algorithmic parameters on method convergence was discussed in *Kozlov & Samsonov (2011)*.

Recombination strategy

Algorithm 2. RECOMBINATION

---

proc recombine (individual) =
{
Select other two random individuals.
Build the combined trial vector using (6).
Generate the random index of parameter I.
for all parameters starting from I **do**
  Generate the random number U.
  if ($U <$ probability of crossover) **then**
    take this parameter from first trial vector to offspring
  else if ($U < 1-$ probability of crossover) **then**
    take this parameter from second trial vector to offspring
  else
    Leave this parameter as is in offspring
  end if
end for
}

The recombination step is demonstrated in the Algorithm 2 insertion. Let $v^{i,0}$, $i = 1, \ldots,$ NP denote a set of the randomly generated parameter vectors for the initial generation

$j = 0$, $v^{i,j} = \{v^{i,j}(k)\} = (q_k^i)_{k=1,\ldots,K}$, where K is the number of parameters q and NP is the fixed size of the set. The first trial vector for index g is calculated by:

$$v_1^{g,j+1} = v^{a,j} + S \circ (v^{b,j} - v^{c,j})$$

where v^a, v^b and v^c are different members of the current generation j with randomly chosen indices a, b and c, and S is a vector of scaling constants for each parameter and $\circ$ denotes element-wise multiplication.

The second optional trial vector is calculated using "trigonometric mutation rule" (*Fan & Lampinen, 2003*).

$$v_2^{g,j+1} = \frac{v^{a,j} + v^{b,j} + v^{c,j}}{3} + (\varphi_b - \varphi_a)(v^{a,j} - v^{b,j})$$
$$+ (\varphi_c - \varphi_b)(v^{b,j} - v^{c,j}) + (\varphi_a - \varphi_c)(v^{c,j} - v^{a,j})$$

where $\varphi_\bullet = |F(v^{\bullet,j})|/\varphi^*$, $\bullet = a, b, c$, $\varphi^* = |F(v^{a,j})| + |F(v^{b,j})| + |F(v^{c,j})|$, and $F(x)$ is the main objective function to be minimized.

The combined trial vector in case of binomial recombination type is defined as follows:

$$v^{g,j+1}(k) = \begin{cases} v_1^{g,j+1}(k), & \text{if } r_k < p_k, \\ v_2^{g,j+1}(k), & \text{else if } r_k < 1 - p_k, \\ v^{g,j}(k), & \text{otherwise} \end{cases} \tag{6}$$

where $r_k = U(0,1)$ is a random number uniformly distributed between 0 and 1, p is the vector of crossover probabilities for all parameters. The first trial vector $v_1^{g,j+1}$ is used continuously for all parameters k until the random number becomes bigger than p in case of the exponential type of recombination.

Scatter search step

Let us consider $v^{g,j}$, $g = 1,\ldots,\Psi$ as the best members of current generation j sorted according to the value of the objective function F such that $F(v^{1,j}) < F(v^{2,j}) < \cdots < F(v^{\Psi,j})$ as described in *Egea, Martí & Banga (2010)*. Each vector $v^{b,j}$ is to be combined with the rest of vectors $v^{a,j}$, $\forall a, a \in [1, 2, \ldots, \Psi]$, $a \neq b$. Two new points within the search space are defined:

$$c_1 = v^{b,j} - d(1 + \alpha\beta); \quad c_2 = v^{b,j} + d(1 - \alpha\beta); \quad d = \frac{v^{a,j} - v^{b,j}}{2},$$

where

$$\alpha = \begin{cases} 1 & \text{if } b < a \\ -1 & \text{if } b > a, \end{cases} \quad \beta = \frac{|a - b| - 1}{\Psi - 2}.$$

Then the offspring is created according to the formula:

$$v^{b,j+1} = c_1 + (c_2 - c_1) \circ r,$$

where $r = \{r_k\}$, $r_k = U(0,1)$, $k = 1,\ldots,K$ is a random number uniformly distributed between 0 and 1.

Selection rule

Algorithm 3. SELECTION

---

proc select (individual) =
{
if $(F <$ the value of the parent) **then**
   Accept offspring
else
   for all criteria F_i, h_j, g_m as f **do**
     if $(f <$ the value of the parent) **then**
       Generate the random number U.
       if $(U <$ control parameter for this criterion) **then**
         Accept offspring
       end if
     end if
   end for
end if
}

In order to increase the robustness of the procedure we have implemented the following selection rule for DE, described in detail in *Kozlov et al. (2013)* (see the Algorithm 3 insertion). Briefly, several different objective functions are used to decide if an offspring will be selected for a new generation. Firstly, the main objective function is checked. The offspring replaces its parent if the value of this function for offspring's set of parameters is less than that for the parental one. In the opposite case the additional objective functions are considered. The offspring replaces its parent if the value of any other objective function is better, and a randomly selected value is less than the predefined parameter for this function.

Preserving population diversity

The original DE algorithm was highly dependent on internal parameters as reported by other authors, see, for example (*Gaemperle, Mueller & Koumoutsakos, 2002*). An efficient adaptive scheme for selection of internal parameters S_k and p_k based on the control of the population diversity was proposed in *Zaharie (2002)*. Let us consider the variation for parameter k in the current generation:

$$\text{var}_k = \frac{1}{NP} \sum_{i=1}^{NP} \left(q_k^i - \frac{1}{NP} \sum_{l=1}^{NP} q_k^l \right)^2$$

where $k = 1, \ldots, n$. For the next generation the scaling constant is calculated by

$$S_k = \begin{cases} \sqrt{\dfrac{NP \cdot (\rho_k - 1) + p_k(2 - p_k)}{2 \cdot NP \cdot p_k}} & NP \cdot (\rho_k - 1) + p_k(2 - p_k) \geq 0 \\ S_{\inf} & NP \cdot (\rho_k - 1) + p_k(2 - p_k) < 0 \end{cases}$$

or alternatively the crossover probability is adopted as

$$p_k = \begin{cases} -(NP \cdot S_k^2 - 1) + \sqrt{(NP \cdot S_k^2 - 1)^2 - NP \cdot (1 - \rho_k)} & \rho_k \geq 1 \\ p_{\text{inf}} & \rho_k < 1 \end{cases}$$

where $S_{\text{inf}} = 1/\sqrt{NP}$, $p_{\text{inf}} = 0$, $\rho_k = \gamma \left(\text{var}_k^{\text{previous}} / \text{var}_k \right)$ and γ is a new constant that controls the decrease of the variability of parameters in the course of iteration process.

Mixed integer-real problems

DE operates on floating point parameters, while many real problems contain integer parameters, e.g., indices of some kind. Two algorithms for parameter conversion from real to integer are implemented in DEEP method as described in *Kozlov et al. (2013)*. The first method rounds off a real value to the nearest integer number. The second procedure consists of the following steps:

- The values are sorted in ascending order.
- The index of the parameter in the floating point array becomes the value of the parameter in the integer array.

Parallelization of objective function calculation

Algorithm 4.   OBJECTIVE FUNCTION

---

proc objfunc (population) =
{
Create a Pool of a specified number worker threads.
Create an Asynchronous Queue of tasks Q in the Pool.
for all individuals in population as x **do**
    Push x to queue Q.
end for
Wait all worker threads in the Pool to finish.
}
proc Worker Thread (parameters) =
{
1. Transform parameters from real to integer as needed.
2. Save parameters into temporary file of specified format.
3. Call specified program and supply the temporary file to it.
4. Capture output of the program.
5. Split output with specified delimiters into a list of values.
6. Assign values in the specified order to $F_i, h_j, g_m, \forall i, j, m$.
7. Return Worker Thread to Pool.
}

DEEP can be effectively parallelized due to independent evaluation of each population member. Various models for evolutionary algorithms parallelization have been developed, such as master-slave, island, cellular or hybrid versions (*Tasoulis et al., 2004*).

The approach implemented in DEEP (see the Algorithm 4 insertion) utilizes the multicore architecture of modern CPUs and employs the pool of worker threads with asynchronous queue of tasks to evaluate the individual solutions in parallel. The calculation of objective function for each trial vector using the command supplied by a user is pushed to the asynchronous queue and starts as soon as there is an available thread in the pool. Such approach is similar to "guided" schedule in OpenMP but gives us more flexibility and control. The command output is automatically recognized according to the specified format. All threads started in the current iteration are to be finished before the next one starts.

IMPLEMENTATION

DEEP is implemented in C programming language as console application and employs interfaces from GLIB project (https://developer.gnome.org/glib/), e.g., *Thread Pool API*. The architecture allows a user to utilize any programming language or computer system, such as *R*, *Octave* or *Python* to implement both mathematical model and comparison of solution with experimental data.

Control parameters

All the control parameters are specified in the single input file as a key-value pairs in INI-format supplied to the DEEP executable on the command line. The control parameters are arranged into three groups described below.

Mathematical Model section specifies the parameter number, both the lower and upper parameter bounds, as well as the software itself necessary to run a model. A possibility is provided to indicate parameters that are to be kept unchanged.

Objective Function section defines the aggregation methods for constraints and multiple objectives. The type of function, i.e., main or additional objective, equality or inequality constraint, is denoted by special keyword. Ranks and weights are to be given here.

Method Settings section allows the user to tune the settings, namely, population size, stopping criterion, offspring generation strategy, the number of the oldest individuals to be substituted in the next generation Ψ, the maximal number of working threads and the seed for random number generator. Two options for offspring generation are provided, namely the selection of best individual or "trigonometric mutation." The stopping criterion can limit the convergence rate, absolute or relative value of the objective function, number of generations or the wall clock time. The initial population is by default generated randomly within the limits given; however, it is also possible to define one initial guess and generate the individuals in the specified vicinity of it.

Programming interfaces

The DEEP method can be used as the static or dynamic shared object and embedded in another software package. Application programming interfaces (APIs) can be used to

connect with existing code implementing mathematical model and objective function. This approach is often preferred in academic and industrial applications where the high level modeling system language is not sufficient or the computation time should be reduced.

RESULTS

Method testing on benchmark functions

To evaluate the performance of DEEP we used three simple multimodal test functions of dimension $D = 30$ from the *Competition on Real Parameter Single Objective Optimization 2014* (CEC-2014) test suite (*Liang, Qu & Suganthan, 2014*), namely:

Shifted and Rotated Griewank's Function.

$$H(x) = h\left(M\left(\frac{600(x - o_H)}{100}\right)\right) + 700; \quad h(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

Shifted Rastrigin's Function.

$$R(x) = r\left(\frac{5.12(x - o_r)}{100}\right) + 800; \quad r(x) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$$

Shifted Schwefel's Function.

$$S(x) = s\left(\frac{1000(x - o_s)}{100}\right) + 1000; \quad s(x) = 418.9829 \times D - \sum_{i=1}^{D} g(z_i(x_i)),$$

where $z_i = x_i + 4.209687462275036 x 10^2$, and

$$g(z_i) = \begin{cases} z_i \sin(|z_i|^{1/2}) & \text{if } |z_i| < 500, \\ (500 - \mathrm{mod}(z_i, 500)) * \\ * \sin\left(\sqrt{|500 - \mathrm{mod}(z_i, 500)|}\right) - \\ -\frac{(z_i - 500)^2}{1000D} & \text{if } z_i > 500, \\ (\mathrm{mod}(|z_i|, 500) - 500) * \\ * \sin\left(\sqrt{|\mathrm{mod}(z_i, 500) - 500|}\right) - \\ -\frac{(z_i + 500)^2}{1000D} & \text{if } z_i < -500, \end{cases}$$

and the global optimum is shifted to $o_i = [o_{i1}, o_{i2}, \ldots, o_{iD}]^T$ and rotated using the rotation matrix M_i.

For each function 51 runs were performed with identical settings and with random initial population. The maximal allowed number of functional evaluations was set to 3×10^5. Other DEEP settings were $NP = 200$, $G_{\max} = 1,499$ and $\Psi = 40$. The measured error was the difference between the known optimal value and the obtained solution.

Following the methodology described in *Tanabe & Fukunaga (2014)* we used the Wilcoxon rank-sum test with significance level $p < 0.05$ to compare the evaluation

Table 1 The results of statistical comparison of DEEP with the top three methods from CEC-2014 on 3 functions. The symbols $+$, $-$, $\approx$ indicate that DEEP performed significantly better ($+$), significantly worse ($-$), or not significantly different ($\approx$) compared to another algorithm using the Wilcoxon rank-sum test (significantly, $p < 0.05$). All results are based on 51 runs.

DEEP vs	CMLP	L-SHADE	UMOEAs
$-$ (worse)	0	0	0
$\approx$ (no sig.)	3	3	1
$+$ (better)	0	0	2

results for 51 runs with the results of the top three methods from CEC-2014 (*Liang, Qu & Suganthan, 2014*) taken from CEC-2014 report:

1. Covariance Matrix Learning and Searching Preference (CMLP) (*Chen et al., 2014*),
2. Success-History Based Parameter Adaptation for Differential Evolution (L-SHADE) (*Tanabe & Fukunaga, 2014*),
3. United Multi-Operator Evolutionary Algorithms (UMOEAs) (*Elsayed et al., 2014*).

The number of benchmark functions from three tested ($+$), ($-$), ($\approx$) is presented in Table 1. DEEP demonstrated the same or better performance.

The method test on reduced model of gene regulation

To demonstrate how DEEP works in applications we developed a realistic benchmark problem based on real biological model of gap gene regulatory network (*Kozlov et al., 2015b*). A model provides a dynamical description of gap gene regulatory system, using detailed DNA-based information, as well as spatial TF concentration data at varying time points. The gap gene regulatory network controls segment determination in the early *Drosophila* embryo (*Akam, 1987*; *Jaeger, 2011*; *Surkova et al., 2008*).

The state variables of this model are the concentrations of mRNAs and proteins encoded by four gap genes *hb*, *Kr*, *gt*, and *kni*. The model implements the thermodynamic approach in the form proposed in *He et al. (2010)* to calculate the expression of a target gene at the RNA level. This level is proportional to the gene activation level also called the promoter occupancy, and is determined by concentrations of eight transcription factors Hb, Kr, Gt, Kni, Bcd, Tll, Cad and Hkb:

$$E_i^a(t) = \frac{Z_{ON,i}^a(t)}{Z_{ON,i}^a(t) + Z_{OFF,i}^a(t)} \tag{7}$$

where $Z_{ON,i}^a(t)$ and $Z_{OFF,i}^a(t)$ are statistical weights of the enhancer with the basal transcriptional complex bound and unbound, respectively.

Two sets of the reaction–diffusion differential equations for mRNA $u_i^a(t)$ and protein concentrations $v_i^a(t)$ describe the dynamics of the system (*Reinitz & Sharp, 1995*; *Jaeger et al., 2004*; *Kozlov et al., 2012*):

$$du_i^a/dt = R_u^a E_i^a(t) + D_u^a(n)[(u_{i-1}^a - u_i^a) + (u_{i+1}^a - u_i^a)] - \lambda_u^a u_i^a, \tag{8}$$

$$dv_i^a/dt = R_v^a u_i^a(t - \tau_v^a) + D_v^a(n)[(v_{i-1}^a - v_i^a) + (v_{i+1}^a - v_i^a)] - \lambda_v^a v_i^a, \tag{9}$$

where n is the cleavage cycle number, R_v^a and R_u^a are the maximum synthesis rates, D_v^a, D_u^a (to smooth the resulting model output) are the diffusion coefficients, λ_v^a and λ_u^a are the
decay rates for protein and mRNA of gene a. The model spans the time period of cleavage cycles 13 and 14A (c13 and c14 resp.) and the interval of A-P axis from 35% to 92% (58 nuclei) of embryo length. The number of nuclei along the A-P axis is doubled when going from c13 to c14. The model is fitted to data on gap protein concentrations from the FlyEx database (*Pisarev et al., 2008*) and mRNA concentrations from SuperFly (*Cicin-Sain et al., 2015*).

To fit the model we used the residual sum of squared differences between the model output and data and we used the *weighted Pattern Generation Potential* proposed in *Samee & Sinha (2013)* as the second objective function:

$$\text{RSS}(x,y) = \sum_{\forall g,n,t:\exists y_n^g(t)} (x_n^g(t) - y_n^g(t))^2 \qquad \text{wPGP}(x,y) = \frac{1 + (\text{penalty}(x,y) - \text{reward}(x,y))}{2}$$

where g, n and t are gene, nucleus and time point respectively and

$$\text{reward}(x,y) = \frac{\sum_i y_i * \min(y_i, x_i)}{\sum_i y_i * y_i} \qquad \text{penalty}(x,y) = \frac{\sum_i (y_{\max} - y_i) * \max(x_i - y_i, 0)}{\sum_i (y_{\max} - y_i) * \sum_i (y_{\max} - y_i)}$$

were x_i and y_i are respectively predicted and experimentally observed expression in nucleus i, and $y_{\max}$ is the maximum level of experimentally observed expression. Consequently, the combined objective function is defined by:

$$\begin{aligned}
F(q,Y) = {}& 2 * 10^{-7} * \text{RSS}(v(q), V) + 1.5 * 10^{-7} * \text{RSS}(u(q), U) \\
& + \text{wPGP}(v(q), V) + 0.6 * \text{wPGP}(u(q), U) \\
& + 10^{-8} * \text{Penalty}(q),
\end{aligned} \qquad (10)$$

where $Y = \{V, U\}$ contains data for u and v, and the function Penalty limits the growth of regulatory parameters, and the weights were obtained experimentally.

We simplified the original computationally expensive model (*Kozlov et al., 2015b*) to use it as a benchmark in our calculations as follows. Firstly, we reduced the number of nuclei from 58 to 10 and considered only one target gene with DNA sequence from *kni*. Consequently, the number of parameters was reduced to 34, two of which are of integer type. Biologically feasible box constraints in the form (4) are imposed for 28 parameters. Next, we fitted this reduced model to the coarsened data and used the obtained solution and model parameters as the synthetic data for benchmark. Thus, the exact parameters of benchmark optimization problem are known.

To compare DEEP and MEIGOR (*Egea et al., 2014*) we run both methods in the same conditions and record the final value of the objective function (11), final parameters and the number of functional evaluations. We considered those solutions for which the final functional value is less than 0.005 that corresponds to parameters close to exact known values. The Welch two sample t-test demonstrated that DEEP used less objective function evaluations than MEIGOR with $p < 0.005$ (see Fig. 1).

Real applications

DEEP software was successfully applied to explain the dramatic decrease in gap gene expression in early *Drosophila* embryo caused by a null mutation in *Kr* gene. Figure 2A
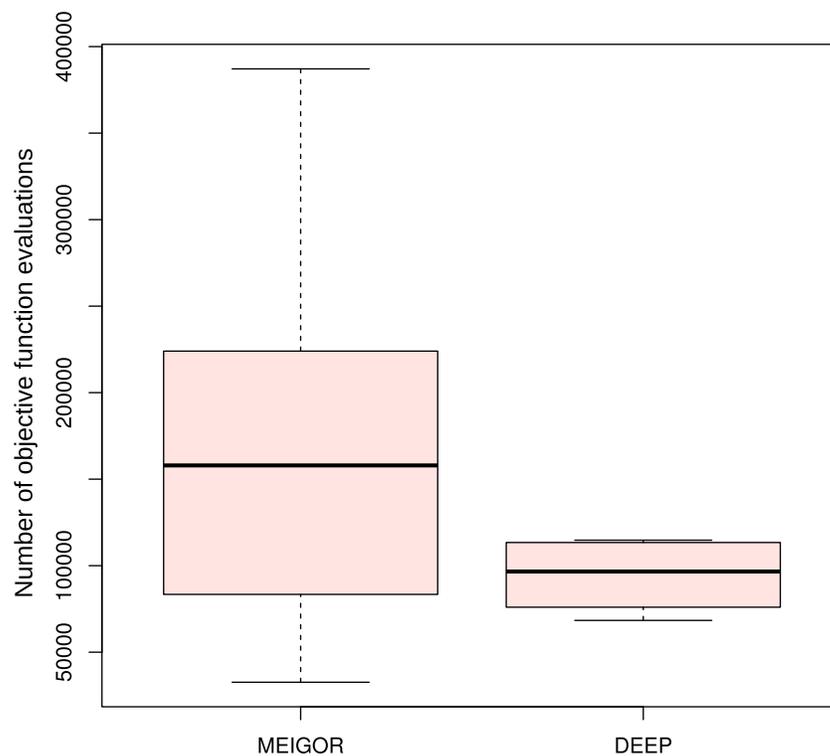
Figure 1 **Comparison of number of objective function evaluations for DEEP and MEIGOR on reduced model of gene regulation.** DEEP used less objective function evaluations than MEIGOR with $p < 0.005$ according to Welch two sample t-test.

presents the topology of regulatory network inferred by fitting the dynamical model with 44 parameters of gap gene expression to the wild type and Kr mutant data simultaneously (*Kozlov et al., 2012*). Other DEEP applications include different problems described in *Ivanisenko et al. (2014)*; *Nuriddinov et al. (2013)*.

Recently, DEEP was used in the online ancestry prediction tool *reAdmix* that can identify the biogeographic origins of highly mixed individuals (*Kozlov et al., 2015a*). *reAdmix* is available at http://chcb.saban-chla.usc.edu/reAdmix/.

Two applications are discussed below in details.

Subgenomic Hepatitis C virus replicon replication model

The hepatitis C virus (HCV) causes hazardous liver diseases leading frequently to cirrhosis and hepatocellular carcinoma. No effective anti-HCV therapy is available up to date. Design of the effective anti-HCV medicine is a challenging task due to the ability of the hepatitis C virus to rapidly acquire drug resistance. The cells containing HCV subgenomic replicon are widely used for experimental studies of the HCV genome replication mechanisms and the *in vitro* testing of the tentative medicine. HCV NS3/4A protease is essential for viral replication and therefore it has been one of the most attractive targets for development of specific antiviral agents for HCV.

We used the new algorithm and software package to determine 18 parameters (kinetic reaction constants) of the mathematical model of the subgenomic Hepatitis C virus
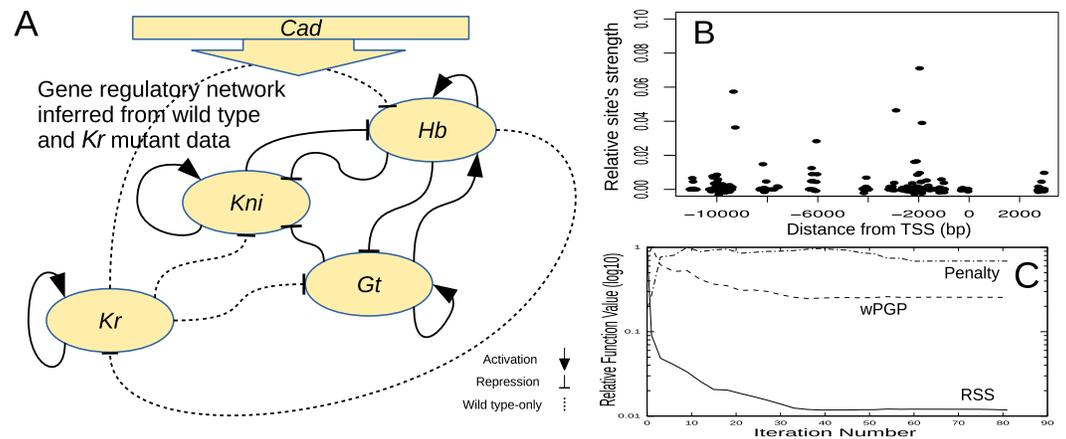
Figure 2 Gene regulatory network, arrows and T-ended curves indicate activation and repressive interactions respectively, dotted lines show interactions present in wild type only (A). Regulatory weights of individual transcription factor binding sites (B). Evolution of three objective functions during parameter fitting (C). See text for details.

(HCV) replicon replication in Huh-7 cells in the presence of the HCV NS3 protease inhibitor, see *Ivanisenko et al. (2013)*.

The experimental data include kinetic curves of the viral RNA suppression at various inhibitor concentrations of the VX-950 and BILN-2061 inhibitors (*Lin et al., 2004*; *Lin et al., 2006*). We seek for the set of parameters that minimizes three criteria. The main criterion (*RSS*) is the residual sum of squared differences between the model output and data. Additional criteria 2 (F_2) and 3 (F_3) penalize the deviation of the time to steady state and the number of viral vesicles at the steady state, respectively.

The combined criterion was defined as follows:

$$F_{\text{combined}} = \text{RSS} + 0.1 \cdot F_2 + 0.1 \cdot F_3 \qquad (11)$$

where the weights were obtained experimentally. The dependence of the best value of the combined criterion (11) in population of individuals on the generation number for 10 runs is plotted in Fig. 3A. The objective function is to be evaluated once for each member of the generation, the size of which was set to 200.

The plot of the criteria in the close vicinity of the optimal values of the two parameters from the set is shown in Figs. 3B and 3C. Despite of the fact that the criteria do not take a minimal values in one and the same point, the algorithm produces reliable approximation of the optimum.

The comparison of the model output and experimental dependencies of the viral RNA suppression rate on inhibitor concentration is shown in Figs. 3D and 3E. It is worth to note that, the model correctly reproduces experimental kinetics of the viral RNA suppression.

The predictive power of the model was estimated using the experimental data on dependencies of the viral RNA suppression rate on the increasing concentration of the SCH-503034 (*Malcolm et al., 2006*) and ITMN-191 (*Seiwert et al., 2008*) inhibitors. These
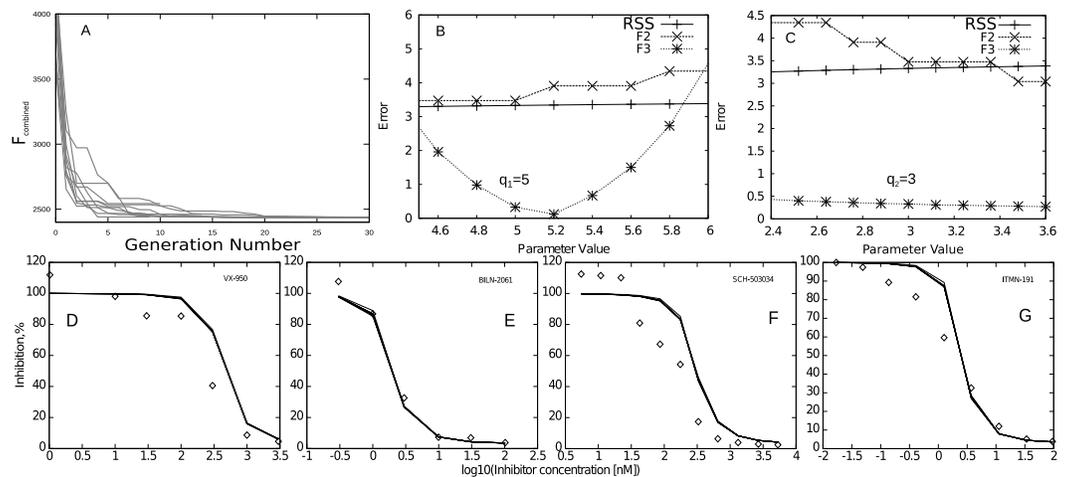
Figure 3 (A) The combined criterion (11) vs. the generation number for 10 runs. 200 function evaluations were performed by the minimization procedure for each generation. (B, C) The criteria graphs are shown in the close vicinity of the optimal values of the four parameters. The values of the parameters found by the algorithm are denoted as x and y. (D, E) The viral RNA suppression in the presence of the NS3 protease inhibitors in different concentrations. The dependence of the viral RNA suppression on the increasing concentration of BILN-2061 (D) and VX-950 (E) inhibitors is shown for the third day post-treatment. A solid line is used to show model output and points correspond to the experimental data (*Lin et al., 2004*; *Lin et al., 2006*). (F, G) The predicted kinetics and the suppression rate of the viral RNA in comparison with data not used for parameter estimation. The dependencies of the suppression rate of the viral RNA on the increasing concentration of the SCH-503034 (F) and ITMN-2061 (G) inhibitors (*Malcolm et al., 2006*; *Seiwert et al., 2008*).

data were not used for parameter estimation. As it can be seen in Figs. 3F and 3G, the model correctly reproduces experimental observations and thus can be used for *in silico* studies.

Sequence-based model of the gap gene regulatory network

Recently, DEEP method was successfully applied to recover 68 parameters of the DNA sequence-based model (7)–(8) of regulatory network of 4 gap genes—*hb*, *Kr*, *gt*, and *kni*—and 8 transcription factors: Hb, Kr, Gt, Kni, Bcd, Tll, Cad and Hkb (*Kozlov et al., 2015b*). The trained model provides a tool to estimate the importance of each TF binding site for the model output (see Fig. 2B). We showed that functionally important sites are not exclusively located in cis-regulatory elements and that sites with low regulatory weight are important for the model output (*Kozlov et al., 2014*).

The evolution of the three objective functions during one optimization run is shown in Fig. 2C. Note that the wPGP and the Penalty functions do not decline monotonically and simultaneously. In a few first steps these functions reach their maximal values while RSS falls sharply, that corresponds to the adaptation of the control parameters of the algorithm and substitution of old parameter sets with good ones. Then wPGP starts to decay, and Penalty fluctuates at high level, while RSS decays approximately at the same rate as wPGP. As Penalty depends only on regulatory parameters, its behaviour at this stage illustrates that it disallows the process to be trapped in some local minimum with extreme values of parameters. During the second half of the optimization process, Penalty

reaches its final low level and stays at it almost constant till convergence while the RSS and wPGP exhibit a modest growth and then converge. This illustrates the ability of DEEP to balance several objective functions. The model output at this stage is not changed much as indicated by RSS though the absolute values of regulatory parameters are fine tuned.

CONCLUSIONS

The parallelization of objective function calculation implemented in DEEP method considerably reduces the computational time. Several members of the current generation are evaluated in parallel, which in our experience with Sequence-based Model of the Gap Gene Regulatory Network, resulted in 24 times speedup on 24 core computational node (Intel Xeon 5670, Joint Supercomputer Center of the Russian Academy of Sciences, Moscow). The calculation of 24 objective functions in parallel threads took approximately the same 20 s as one sequential job, and the optimization runs were able to converge in 14 h after approximately 60,000 functional evaluations.

To sum up, we elaborated both the method and the software, which demonstrated high performance on test functions and biological problems of finding parameters in dynamic models of biological processes by minimizing one or even several objective functions that measure the deviation of model solution from data.

ACKNOWLEDGEMENTS

We are thankful to the Joint Supercomputer Center of the Russian Academy of Sciences, Moscow, for provided computational resources.

Funding

The implementation and testing was supported by RSF grant no. 14-14-00302, the method development was supported by RFBR grant 14-01-00334 and the Programme "5-100-2020" by the Russian Ministry of Science and Education. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
RSF: 14-14-00302.
RFBR: 14-01-00334.
Russian Ministry of Science and Education: 5-100-2020.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Konstantin Kozlov conceived and designed the experiments, performed the experiments, analyzed the data, wrote the paper, prepared figures and/or tables, performed the computation work, reviewed drafts of the paper.

- Alexander M. Samsonov conceived and designed the experiments, analyzed the data, contributed reagents/materials/analysis tools, wrote the paper, reviewed drafts of the paper.
- Maria Samsonova conceived and designed the experiments, wrote the paper, reviewed drafts of the paper.

Data Availability

SourceForge: http://deepmethod.sourceforge.net/

openSUSE: https://build.opensuse.org/project/repositories/home:mackoel:compbio.

REFERENCES

Akam M. 1987. The molecular basis for metameric pattern in the *Drosophila* embryo. *Development* **101**:1–22.

Chen L, Liu H-L, Zheng Z, Xie S. 2014. A evolutionary algorithm based on covariance matrix learning and searching preference for solving CEC 2014 benchmark problems. In: *CEC 2014 special session and competition on single objective real-parameter numerical optimization*, vol. 3. Piscataway: IEEE, 2672–2677.

Chu KW, Deng Y, Reinitz J. 1999. Parallel simulated annealing by mixing of states. *The Journal of Computational Physics* **148**:646–662 DOI 10.1006/jcph.1998.6134.

Cicin-Sain D, Pulido AH, Crombach A, Wotton KR, Jiménez-Guri E, Taly J-F, Roma G, Jaeger J. 2015. SuperFly: a comparative database for quantified spatio-temporal gene expression patterns in early dipteran embryos. *Nucleic Acids Research* **43(D1)**:D751–D755 DOI 10.1093/nar/gku1142.

Egea JA, Henriques D, Cokelaer T, Villaverde AF, MacNamara A, Danciu D-P, Banga JR, Saez-Rodriguez J. 2014. MEIGO: an open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics. *BMC Bioinformatics* **15(1)**:1–9 DOI 10.1186/1471-2105-15-1.

Egea JA, Martí R, Banga JR. 2010. An evolutionary method for complex-process optimization. *Computers & Operations Research* **37(2)**:315–324 DOI 10.1016/j.cor.2009.05.003.

Elsayed SM, Sarker RA, Essam DL, Hamza NM. 2014. Testing united multi-operator evolutionary algorithms on the CEC-2014 real-parameter numerical optimization. In: *CEC 2014 special session and competition on single objective real-parameter numerical optimization*, vol. 3. Piscataway: IEEE, 1650–1657.

Fan H-Y, Lampinen J. 2003. A trigonometric mutation operation to differential evolution. *Journal of Global Optimization* **27**:105–129 DOI 10.1023/A:1024653025686.

Fomekong-Nanfack Y. 2009. Genetic Regulatory Networks Inference: modeling, parameters estimation and model validation. PhD Thesis, University of Amsterdam.

Fomekong-Nanfack Y, Kaandorp J, Blom J. 2007. Efficient parameter estimation for spatio-temporal models of pattern formation: case study of *Drosophila melanogaster*. *Bioinformatics* **23(24)**:3356–3363 DOI 10.1093/bioinformatics/btm433.

Gaemperle R, Mueller SD, Koumoutsakos P. 2002. A parameter study for differential evolution. In: Grmela A, Mastorakis NE, eds. *Advances in intelligent systems, fuzzy systems, evolutionary computation*. WSEAS Press, 293–298.

He X, Samee MAH, Blatti C, Sinha S. 2010. Thermodynamics-based models of transcriptional regulation by enhancers: the roles of synergistic activation, cooperative binding and short-range repression. *PLoS Computational Biology* **6**(9):e1000935 DOI 10.1371/journal.pcbi.1000935.

Ivanisenko N, Mishchenko E, Akberdin I, Demenkov P, Likhoshvai V, Kozlov K, Todorov D, Samsonova M, Samsonov A, Kolchanov N, Ivanisenko V. 2013. Replication of the Subgenomic Hepatitis C virus replicon in the presence of the NS3 protease inhibitors: a stochastic model. *Biophysics* **58**(5):592–606 DOI 10.1134/S0006350913050059.

Ivanisenko NV, Mishchenko EL, Akberdin IR, Demenkov PS, Likhoshvai VA, Kozlov KN, Todorov DI, Gursky VV, Samsonova MG, Samsonov AM, Clausznitzer D, Kaderali L, Kolchanov NA, Ivanisenko VA. 2014. A new stochastic model for Subgenomic Hepatitis C virus replication considers drug resistant mutants. *PLoS ONE* **9**(3):e91502 DOI 10.1371/journal.pone.0091502.

Jaeger J. 2011. The gap gene network. *Cellular and Molecular Life Sciences* **68**:243–274 DOI 10.1007/s00018-010-0536-y.

Jaeger J, Surkova S, Blagov M, Janssens H, Kosman D, Kozlov KN, Manu, Myasnikova E, Vanario-Alonso CE, Samsonova M, Sharp DH, Reinitz J. 2004. Dynamic control of positional information in the early *Drosophila* embryo. *Nature* **430**:368–371 DOI 10.1038/nature02678.

Kozlov K, Chebotarev D, Hassan M, Triska M, Triska P, Flegontov P, Tatarinova T. 2015a. Differential evolution approach to detect recent admixture. *BMC Genomics* **16**(Suppl 8):Article S9 DOI 10.1101/015446.

Kozlov K, Gursky VV, Kulakovskiy IV, Dymova A, Samsonova M. 2015b. Analysis of functional importance of binding sites in the drosophila gap gene network model. *BMC Genomics* **16**(13):1–16 DOI 10.1186/1471-2164-16-S13-S7.

Kozlov K, Gursky V, Kulakovskiy I, Samsonova M. 2014. Sequence-based model of gap gene regulatory network. *BMC Genomics* **15**(Suppl 12):Article S6.

Kozlov K, Ivanisenko N, Ivanisenko V, Kolchanov N, Samsonova M, Samsonov AM. 2013. Enhanced differential evolution entirely parallel method for biomedical applications. In: Malyshkin V, ed. *Lecture notes in computer science*, vol. 7979. New York: Springer, 409–416.

Kozlov K, Samsonov A. 2011. DEEP—differential evolution entirely parallel method for gene regulatory networks. *Journal of Supercomputing* **57**:172–178 DOI 10.1007/s11227-010-0390-6.

Kozlov K, Surkova S, Myasnikova E, Reinitz J, Samsonova M. 2012. Modeling of gap gene expression in *Drosophila* Kruppel mutants. *PLoS Computational Biology* **8**(8):e1002635 DOI 10.1371/journal.pcbi.1002635.

Liang JJ, Qu BY, Suganthan PN. 2014. Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter

numerical optimization. Technical Report 201311. Singapore: Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China And Technical Report, Nanyang Technological University.

Lin C, Lin K, Luong YP, Rao BG, Wei YY, Brennan DL, Fulghum JR, Hsiao HM, Ma S, Maxwell JP, Cottrell KM, Perni RB, Gates CA, Kwong AD. 2004. *In vitro* resistance studies of hepatitis C virus serine protease inhibitors, VX-950 and BILN 2061: structural analysis indicates different resistance mechanisms. *Journal of Biological Chemistry* **279(17)**:17508–17514 DOI 10.1074/jbc.M313020200.

Lin K, Perni RB, Kwong AD, Lin C. 2006. VX-950, a novel hepatitis C virus (HCV) NS3-4A protease inhibitor, exhibits potent antiviral activities in HCv replicon cells. *Antimicrobial Agents and Chemotherapy* **50(5)**:1813–1822 DOI 10.1128/AAC.50.5.1813-1822.2006.

Malcolm BA, Liu R, Lahser F, Agrawal S, Belanger B, Butkiewicz N, Chase R, Gheyas F, Hart A, Hesk D, Ingravallo P, Jiang C, Kong R, Lu J, Pichardo J, Prongay A, Skelton A, Tong X, Venkatraman S, Xia E, Girijavallabhan V, Njoroge FG. 2006. SCH 503034, a mechanism-based inhibitor of hepatitis C virus NS3 protease, suppresses polyprotein maturation and enhances the antiviral activity of alpha interferon in replicon cells. *Antimicrobial Agents and Chemotherapy* **50(3)**:1013–1020 DOI 10.1128/AAC.50.3.1013-1020.2006.

Mendes P, Kell DB. 1998. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics* **14**:869–883 DOI 10.1093/bioinformatics/14.10.869.

Moles CG, Mendes P, Banga JR. 2003. Parameter estimation in biochemical pathways: comparison of global optimization methods. *Genome Research* **13**:2467–2474 DOI 10.1101/gr.1262503.

Nuriddinov M, Kazantsev F, Rozanov A, Kozlov K, Peltek S, Akberdin I, Kolchanov N. 2013. Mathematical modeling of ethanol and lactic acid biosynthesis by theromphilic geobacillus bacteria. *Russian Journal of Genetics: Applied Research* **17(4/1)**:686–704.

Pisarev A, Poustelnikova E, Samsonova M, Reinitz J. 2008. FlyEx, the quantitative atlas on segmentation gene expression at cellular resolution. *Nucleic Acids Research* **37**:D560–D566 DOI 10.1093/nar/gkn717.

Reinitz J, Sharp DH. 1995. Mechanism of *eve* stripe formation. *Mechanisms of Development* **49**:133–158 DOI 10.1016/0925-4773(94)00310-J.

Samee MAH, Sinha S. 2013. Evaluating thermodynamic models of enhancer activity on cellular resolution gene expression data. *Methods* **62**:79–90 DOI 10.1016/j.ymeth.2013.03.005.

Seiwert SD, Andrews SW, Jiang Y, Serebryany V, Tan H, Kossen K, Rajagopalan RPT, Misialek S, Stevens SK, Stoycheva A, Hong J, Lim SR, Qin X, Rieger R, Condroski KR, Zhang H, Do MG, Lemieux C, Hingorani GP, Hartley DP, Josey JA, Pan L, Beigelman L, Blatt LM. 2008. Preclinical characteristics of the HCV NS3/4A protease inhibitor ITMN-191 (R7227). *Antimicrobial Agents and Chemotherapy* **52(12)**:4432–4441 DOI 10.1128/AAC.00699-08.

Spirov AV, Kazansky AB. 2002. Jumping genes-mutators can raise efficacy of evolutionary search. In: *Proceedings of the genetic and evolutionary computation conference GECCO2002*. San Francisco: Morgan Kaufmann Publishers Inc.

Storn R, Price K. 1995. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. Technical Report TR-95-012. Berkeley: ICSI.

Suleimenov Y. 2013. Global parameter estimation for thermodynamic models of transcriptional regulation. *Methods* **62**:99–108 DOI 10.1016/j.ymeth.2013.05.012.

Surkova S, Kosman D, Kozlov K, Manu, Myasnikova E, Samsonova A, Spirov A, Vanario-Alonso CE, Samsonova M, Reinitz J. 2008. Characterization of the *Drosophila* segment determination morphome. *Developmental Biology* **313**(2):844–862 DOI 10.1016/j.ydbio.2007.10.037.

Tanabe R, Fukunaga AS. 2014. Improving the search performance of shade by using linear population size reduction. In: *CEC 2014 special session and competition on single objective real-parameter numerical optimization*, vol. 3. Piscataway: IEEE, 1658–1665.

Tasoulis D, Pavlidis N, Plagianakos V, Vrahatis M. 2004. Parallel differential evolution. In: *Congress on evolutionary computation (CEC 2004)*, vol. 2. Piscataway: IEEE, 2023–2029.

Zaharie D. 2002. Parameter adaptation in differential evolution by controlling the population diversity. In: Petcu D, ed. *Proceedigs of the 4th international workshop on symbolic and numeric algorithms for scientific computing*. Timisoara, Romania, 385–397.