

doi: 10.17586/2226-1494-2023-23-1-97-104

Multiobjective evolutionary discovery of equation-based analytical models for dynamical systems

Mikhail A. Maslyayev¹, Alexander A. Hvatov²✉

^{1,2} ITMO University, Saint Petersburg, 197101, Russian Federation

¹ mikemaslyayev@itmo.ru, <https://orcid.org/0000-0001-5595-0802>

² alex_hvatov@itmo.ru✉, <https://orcid.org/0000-0002-5222-583X>

Abstract

In this article, an approach to modeling dynamical systems in case of unknown governing physical laws has been introduced. The systems of differential equations obtained by means of a data-driven algorithm are taken as the desired models. In this case, the problem of predicting the state of the process is solved by integrating the resulting differential equations. In contrast to classical data-driven approaches to dynamical systems representation, based on the general machine learning methods, the proposed approach is based on the principles, comparable to the analytical equation-based modeling. Models in forms of systems of differential equations, composed as combinations of elementary functions and operation with the structure, were determined by adapted multi-objective evolutionary optimization algorithm. Time-series describing the state of each element of the dynamic system are used as input data for the algorithm. To ensure the correct operation of the algorithm on data characterizing real-world processes, noise reduction mechanisms are introduced in the algorithm. The use of multicriteria optimization, held in the space of complexity and quality criteria for individual equations of the differential equation system, makes it possible to improve the diversity of proposed candidate solutions and, therefore, to improve the convergence of the algorithm to a model that best represents the dynamics of the process. The output of the algorithm is a set of Pareto-optimal solutions of the optimization problem where each individual of the set corresponds to one system of differential equations. In the course of the work, a library of data-driven modeling of dynamic systems based on differential equation systems was created. The behavior of the algorithm was studied on a synthetic validation dataset describing the state of the hunter-prey dynamic system given by the Lotka-Volterra equations. Finally, a toolset based on the solution of the generated equations was integrated into the algorithm for predicting future system states. The method is applicable to data-driven modeling of arbitrary dynamical systems (e.g. hydrometeorological systems) in cases where the processes can be described using differential equations. Models generated by the algorithm can be used as components of more complex composite models, or in an ensemble of methods as an interpretable component.

Keywords

differential equation discovery, evolutionary optimization, multi-objective optimization, differential equations system, symbolic regression

Acknowledgements

This research is financially supported by the Russian Scientific Foundation, Agreement No. 21-71-00128.

For citation: Maslyayev M.A., Hvatov A.A. Multiobjective evolutionary discovery of equation-based analytical models for dynamical systems. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 1, pp. 97–104. doi: 10.17586/2226-1494-2023-23-1-97-104

УДК 004.89

Определение аналитических моделей динамических систем в форме дифференциальных уравнений на основе многокритериальной эволюционной оптимизации

Михаил Александрович Масляев¹, Александр Александрович Хватов²✉

^{1,2} Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

¹ mikemaslyaev@itmo.ru, <https://orcid.org/0000-0001-5595-0802>

² alex_hvatov@itmo.ru ✉, <https://orcid.org/0000-0002-5222-583X>

Аннотация

Предмет исследования. В работе предложен метод моделирования динамических систем при условии, что управляющие процессом физические законы неизвестны. В качестве искомым моделей приняты полученные при помощи управляемого данными алгоритма системы дифференциальных уравнений. В результате решается задача прогнозирования состояния процесса при помощи интегрирования результирующих дифференциальных уравнений. В отличие от классических подходов к воспроизведению динамических систем на основе данных, основанных на общих принципах машинного обучения, предложенный алгоритм позволяет сформировать модели процессов, сопоставимые с аналитическими. **Метод.** В качестве модели процесса приняты системы дифференциальных уравнений, представленные через комбинации элементарных функций и операторов, определенные при помощи адаптированного эволюционного алгоритма многокритериальной оптимизации. В качестве входных данных для алгоритма использованы временные ряды, описывающие состояние каждого элемента динамической системы. Для обеспечения работы алгоритма на данных, характеризующих реальные процессы, в алгоритм включены механизмы компенсации шума. Использование многокритериальной оптимизации, проводимой в пространстве критериев сложности и качества отдельных уравнений системы дифференциальных уравнений, позволило улучшить разнообразие предлагаемых кандидатных решений. Также получена высокая сходимость алгоритма, что обеспечило поиск модели, наилучшим образом показывающей динамику процесса. Результатом работы алгоритма является множество Парето-оптимальных решений оптимизационной задачи, каждое из которых соответствует одной системе дифференциальных уравнений.

Основные результаты. В ходе работы создана библиотека управляемого данными моделирования динамических систем на основе систем дифференциальных уравнений. Поведение алгоритма исследовано на синтетическом валидационном наборе данных, описывающем состояние динамической системы «охотник-жертва», заданной уравнениями Лотки–Вольтерра. Предложен интегрированный в алгоритм механизм прогнозирования состояний системы, основанный на решении сформированных уравнений. **Практическая значимость.** Метод применим к управляемому данными моделированию произвольных динамических систем (например, гидрометеорологических) в случаях, когда процессы могут быть описаны при помощи дифференциальных уравнений. Сформированные алгоритмом модели можно использовать в качестве компонент более сложных композитных моделей, или в ансамбле методов как интерпретируемую составляющую.

Ключевые слова

определение дифференциальных уравнений, эволюционная оптимизация, многокритериальная оптимизация, динамические системы, символьная регрессия

Благодарности

Исследование поддержано Российским научным фондом, грант № 21-71-00128.

Ссылка для цитирования: Масляев М.А., Хватов А.А. Определение аналитических моделей динамических систем в форме дифференциальных уравнений на основе многокритериальной эволюционной оптимизации // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 1. С. 97–104 (на англ. яз.). doi: 10.17586/2226-1494-2023-23-1-97-104

Introduction

Systems of Ordinary or Partial Differential Equations (ODEs and PDEs) are powerful tools that can describe complex dynamics of structures involving multiple variables. While many tools can be used for creating mathematical models for processes, such as classical machine learning models, or unconventional ones, like Bayesian networks [1], they tend to have strict limitations to their applications. In cases of many real-world systems, in addition to the issues above, these models are often abstracted from the intrinsic physical principles guiding the system. The classical approach to deriving systems of differential equations necessitates the use of mathematical analysis in combination with an in-depth understanding of the process. The data-driven approach to system discovery involves the creation of an individual differential equation

for each dependent variable that can be measured from a system.

The forms of the discovered models containing systems of differential equations are selected due to the prevalence of differential equation in physical systems. For example, flow of viscous fluid is governed by Navier-Stokes equations that are a system of partial differential equations. Dynamics and interactions between electric and magnetic components of the electromagnetic field are described with Maxwell's equations which are a system of PDE as well. Many simpler systems, such as rotation of the spherical pendulum, can be defined with system of ordinary differential equations.

Apart from the descriptive possibilities, provided by models in forms of systems of differential equations, obtained systems can be solved to predict further states of the process. While the toolkit for the automatic solution of

systems of ordinary/partial differential equations is out of scope of this study, several studies have been conducted towards implementing equation-solving module into the frameworks of differential equations discovery, as in work [2]. With this ability to solve model equations, the system dynamics can be propagated into the future.

Analysis of existing approaches

Creating models for dynamical systems governed by differential equations has recently drawn interest. The first perspective to the task involves developing substitutes for the equations in forms of propagation operators that map the state of the system forward in time like in [3] or [4]. Dynamic Mode Decomposition (DMD) involves approximation of the system dynamics with a finite-dimensional linear operator. While that can be useful for multiple real-world applications where the propagator is linear, many other cases involve non-linear dynamics that cannot be fully explained with DMD approach.

A number of data-driven solutions to the problem of explaining dynamical system with explicitly derived governing equations have been developed. Here, we will inspect methods that are applicable not only for problems of discovering Ordinary Differential Equations (ODE) and systems of ODEs, but also for tasks of partial differential equations discovery. The first problem has sufficient solution in forms of Multilayer Stochastic Models (MSMs) proposed by Kondrashov, Chekroun and Ghil in [5]. However, due to the non-Markovian approach, the approach is not extendable to the problems of partial differential equations.

The earliest advances were made with the symbolic regression [6]. Governing equations are viewed as computational tree graphs where leaves are inputs, and on the other levels various operators are located. The search of the equation can be done with the typical graph-targeted evolutionary optimization algorithms. More contemporary approaches are represented by sparse regression based models developed in many works, including Kaheman et al. in [7] and Berg & Nystrom in [8], and with artificial neural networks (ANN) representation of the dynamical system. While there are multiple approaches to discover differential equations with artificial neural networks, notable ones include PINN [9], PDE-Net, developed by Long et al. [10], and physics-informed neural networks by Raissi et al. [11].

Partial differential equation search with sparse regression uses LASSO operator that is applied to approximate time derivative with a library of candidate terms. That library has to contain all possible equation terms, and the usage of sparsity operator allows selection of only a few active feature terms. The main issues of this approach can be linked with its rigidity: the term library has to be extensive enough to contain all possible terms including all non-linear functions that can be present in equations. While many of the presented approaches can be applied to the systems of differential equations, their possibilities are limited by description of time dynamics of a vector variable, like in paper [12].

The algorithm described in this article is based on the multiobjective evolutionary optimization approach,

where the model obtained is evaluated by several metrics describing quality and complexity of the equations of the system. Thus, the algorithm can provide the parsimonious model that is not overly complex but can sufficiently simulate the dynamics of the process. However, the problem of selecting that parsimonious model from the discovered Pareto frontier is the problem for another study. This paper is dedicated to the problem of discovering the optimal set of candidate equations for the further expert conclusions and applications.

Equation discovery problem

To describe some unstudied process, which involves multiple (n) dependent variables, we desire to derive a system of differential equations. Let us denote these variables in general problem statement as $\mathbf{u} = (u_1(t, \mathbf{x}), u_2(t, \mathbf{x}), \dots, u_n(t, \mathbf{x}))$. They are defined in the spatial domain Ω , represented by coordinates \mathbf{x} , and dependent from time t . In case of a system of ordinary differential equations, the variables can be assumed to be only time-dependent (i.e., $u_1(t), u_2(t), \dots, u_n(t)$).

For the equations search process, the algorithm requires sets of observations arranged on a rectangular grid. For the equation search process, the algorithm demands arrays of calculated derivatives. While in some cases these derivatives can be obtained directly, using measurement techniques, in others they necessitate a preprocessing phase where the derivatives are calculated numerically from the input data variables. While the numerical techniques of derivatives estimation are numerous [13], the most efficient approaches are finite-difference differentiation and analytical differentiation of variable-approximating polynomials. In many cases, additional smoothing is required to reduce magnitudes of noise in the data. Here, the algorithm employs Gaussian smoothing in the spatial domain, or replacement of the initial data fields with their artificial neural network approximation.

$$S(\mathbf{u}) = \begin{cases} L_1(\mathbf{u}) = 0 \\ \dots \\ L_k(\mathbf{u}) = 0 \end{cases},$$

where $S(\mathbf{u})$ is the system of differential equations that involves variables \mathbf{u} comprised of individual equations $L_1(\mathbf{u})$ to $L_k(\mathbf{u})$.

The search for the optimal structures of equations in the system is done with the multi-objective optimization implemented with the Many-Objective Optimization Evolutionary Algorithm Based on Dominance and Decomposition (MOEA/DD), introduced in [14].

The search is performed in the criteria space of complexities $C(L_j'\mathbf{u})$ and modeling errors $Q(L_j'\mathbf{u})$ for each individual equation in a system. Therefore, the problem can be reformulated to

$$\begin{aligned} \text{minimize } \mathbf{F}(S(\mathbf{u})) &= f_1(S(\mathbf{u})), \dots, f_m(S(\mathbf{u})) = \\ &= (C(L_1'\mathbf{u}), Q(L_1'\mathbf{u}), \dots, C(L_n'\mathbf{u}), Q(L_n'\mathbf{u})). \end{aligned}$$

Here the constraints are introduced in the equations construction logic rather than explicitly specified during the optimization problem statement.

The complexity metric $C(L'_j \mathbf{u})$ is defined as a number of “active” tokens in the equation, i.e. ones presented in terms of non-zero coefficients.

The problem of selecting the most appropriate metric for evaluating the properties of process representation for the equation has been studied in work [2]. The best metrics for modeling quality are L_2 norm of matrices of differential operator residuals represented by:

$$Q(L'_j \mathbf{u}) = \|L'_j \mathbf{u}\|_2$$

or the norm of matrices of differences between the input variable fields u_j and the solutions \tilde{u}_j of corresponding equations

$$Q(L'_j \mathbf{u}) = \|u_j - \tilde{u}_j\|_2.$$

Due to the necessity to conduct optimization, having a limited number of candidate solutions, the implemented approach uses concept of domination for the proposed solutions to the problem of searching for systems of equations. It is said that candidate system $S_1(\mathbf{u})$ dominates candidate system $S_2(\mathbf{u})$ if for all optimized criteria f_i : $f_i(S_1(\mathbf{u})) \leq f_i(S_2(\mathbf{u}))$ and for a single criterion f_i : $f_i(S_1(\mathbf{u})) < f_i(S_2(\mathbf{u}))$. A solution is called Pareto-optimal if no other solutions dominate it. The objective of the implemented algorithm is to obtain a set of candidate solutions where each solution is Pareto-optimal. In addition to the Pareto-optimal set, other non-dominated sets can be introduced by induction: n -th non-dominated level is comprised by solution that is not dominated by any solutions, except the ones on the $n-1$ th, or lower levels.

Approach description

In this section, we briefly describe the main diversions of our approach from the original algorithm [14] and case-specific solutions employed during the system of differential equations derivation, such as evolutionary operators. Following the optimization objectives stated in the previous section, the algorithm performs a simultaneous search of system equations and parameters which define the equations structures. The structure of an equation can be decomposed into a set of equation terms and a set of their real-valued coefficients a_i as in:

$$L'_j \mathbf{u} = \sum_i a_i \prod_j t_{ij}.$$

The terms of constructed equations are represented with a tokens product $\prod_j t_{ij}$, $t_{ij} \in \mathbf{T}$, elementary building blocks containing arbitrary user-defined functions. This approach enables the discovery of non-linear equations with compound structures that can be represented as a sum of product terms. During search of differential equations, various derivatives (e.g. $\frac{\partial^n u_i}{\partial x_j^n}$) are included into the pool \mathbf{T} .

Other case-specific functions or external variables can be included as tokens into the token pool to be available for the algorithm during equation search. For example, suppose a study objective is to discover the equation for the temperature dynamics in a medium. In that case, the velocity field of the medium can be considered an external variable.

To create a system that can model the studied process, it is possible to assume that each equation in the system must represent the spatial or temporal dynamics of at least one variable. By describing a variable dynamic, we understand that the equation contains corresponding derivatives of the variable. During the evolutionary search, evolutionary operators affecting the structures of the equations have to preserve the descriptive properties of such terms.

Evolutionary algorithm details

To start the evolutionary optimization, the algorithm has to construct the initial population $\mathbf{P} = S_1(\mathbf{u}), \dots, S_2(\mathbf{u})$ of randomly generated candidate systems of differential equations. As mentioned above, a system equation has to represent a corresponding variable’s dynamic. Therefore, during the initialization, a variable is assigned to each equation as its “main” one. Without loss of generality, we can assume that the i -th equation describes i -th variable.

To emphasize the duality of the system discovery, an individual encoding must represent both equations and meta-parameters of the equations. The chromosome of an individual contains computational graphs of the equations as “equation genes” and values of the parameters that define the creation of the equation. Equation graphs take the form of tree graphs, where the leaves are elementary functions stored in tokens, and intermediate nodes are product operators that form equation terms from factor tokens. The graph root comprises the summation operator which combines separate terms into the equation. The scheme of the equation system encoding is presented in Fig. 1.

A regularization tool has to be created to regulate the complexity of the equations proposed by the algorithm. Its main objective is to exclude terms with low significance and explanatory power in the resulting model. Selection of the terms can be made with sparse regression, operating with the LASSO operator:

$$\|\mathbf{F}_k \beta - F_{\text{target},k}\|_2^2 + \lambda \|\beta\|_1 - \min_{\beta}$$

As the predicted value of the operator, a random equation term representing an “equation variable”, i.e., containing its derivative, is selected. LASSO operator can obtain a vector of term weights β with values of the terms in the left-hand side of the equation, evaluated on the space-time grid, normalized and combined into matrix \mathbf{F}_k , with vector of right-hand part values of $F_{\text{target},k}$. In the operator statement $\|\cdot\|_p$, the i -th norm of the matrix is designated.

The sparsity constant parameter λ determines the penalty of optimized functional with respect to the values of weights in β , prioritizing setting zero coefficients to the less significant predictors. The algorithm can control the equation complexity by regulating the value of the sparsity constant. Higher values of λ promote equations with fewer numbers of terms, while lower values tend to lead to more complex equations. Due to the significance of the sparsity parameter for the equation definition, it is included for each equation in the system in the encoding of the individual.

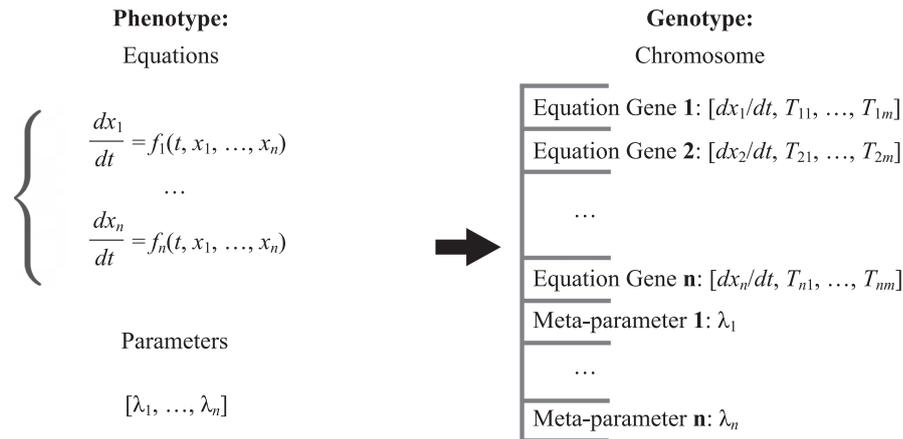


Fig. 1. Scheme of the encoding for a system of ODEs for arbitrary variables x_j with sparsity constants λ_j as meta-parameters. T_{nm} denotes the m -th term of the n -th equation of the system, while f_n is an arbitrary right-hand side function of the n -th equation of the system

The coefficients of the equation are computed with linear regression where active terms from the left-hand part are combined into a matrix of predictors, and values of the term on the right part are used as a predicted value.

Evolutionary operators

The general idea of evolutionary operators affecting the population to obtain the set of optimal systems of equations is borrowed from the single-objective algorithm of equation discovery proposed in [15]. The alterations of an individual equation can be done with operators of mutations and crossover. The operators are applied to individuals of the population following the guidelines presented in the paper and describing the base algorithm [16].

The process of the evolution is held iteratively, for a specified number of iterations and over sectors, defined by the weight vectors introduced into the space of optimization criteria to decompose the problem into smaller sub-problems. As in the original version, the algorithm constructs a set of weight vectors $\mathbf{W} = \mathbf{w}_1, \dots, \mathbf{w}_N$ from a unit simplex, one for each candidate solution in \mathbf{P} .

After the weights are defined, each individual of the population \mathbf{P} is assigned to a random sector of the criteria space. That enables a more even coverage of the search space due to the property that the individuals converge in the directions of weights.

The selection of the individuals for the crossover operators is held in a manner that respects problem decomposition. In the base scenario, the parents are selected from the neighboring sectors to the one associated with the processed weight vector. However, to increase the algorithm exploratory properties, which are vital in the problem of equation construction, with a relatively small probability, the parents are selected from other, non-adjacent sectors. The selected candidates are added to the parent pool, and the crossover is held among them.

The crossover operator affects both systems of equations and corresponding vectors of meta-parameters.

The interactions between equations of the systems comply with variable description requirements. For each modeled variable, the corresponding equations of the parent

systems are affected by crossover. Two main types of operators are used here: term-wise exchange and complete equation swapping.

The first type of equation-level crossover operator involves an exchange of terms between parent equations. All initial terms of the equation are divided into three groups. First group includes terms present in the same form in both parents. Second group includes terms present in both parents, but in this case the parameters of their tokens are different. Third group contains unique ones between parents terms. The first group is not affected by crossover at all. The crossover between parents in the second group is parametric-only: the same tokens exchange the parameter values from a specified proportion.

After the creation of offspring individuals, they are affected by mutation operators. Their purpose is twofold. They are increasing the exploratory properties of the algorithm and preventing the generation of repeating individuals which is mandatory for the implemented multi-objective optimization approach. The main idea of the mutation operator is the random change of a term into a new, unique one. The first type of operator changes a factor representing a token into a new, randomly generated one; or changes token parameters (e.g. frequency of a sine token) with an increment taken from normal distribution $N(0, \sigma)$ with pre-defined variance σ^2 . The second type involves a replacement of a term with a newly generated one. When the offspring creation procedures are conducted, the Pareto levels are updated with respect to the newly created solution. The population update algorithm considers the decomposition of the problems with a set of weight vectors and their domination. The approach in which the evolutionary operators are applied during the search is presented in Fig. 2.

Validation

Several validation experiments have been conducted to assess the proposed approach performance in discovering systems of equations that govern the dynamical system. The most demonstrative approach to check the behavior of our algorithm employs synthetic data obtained from the solution of known equations.

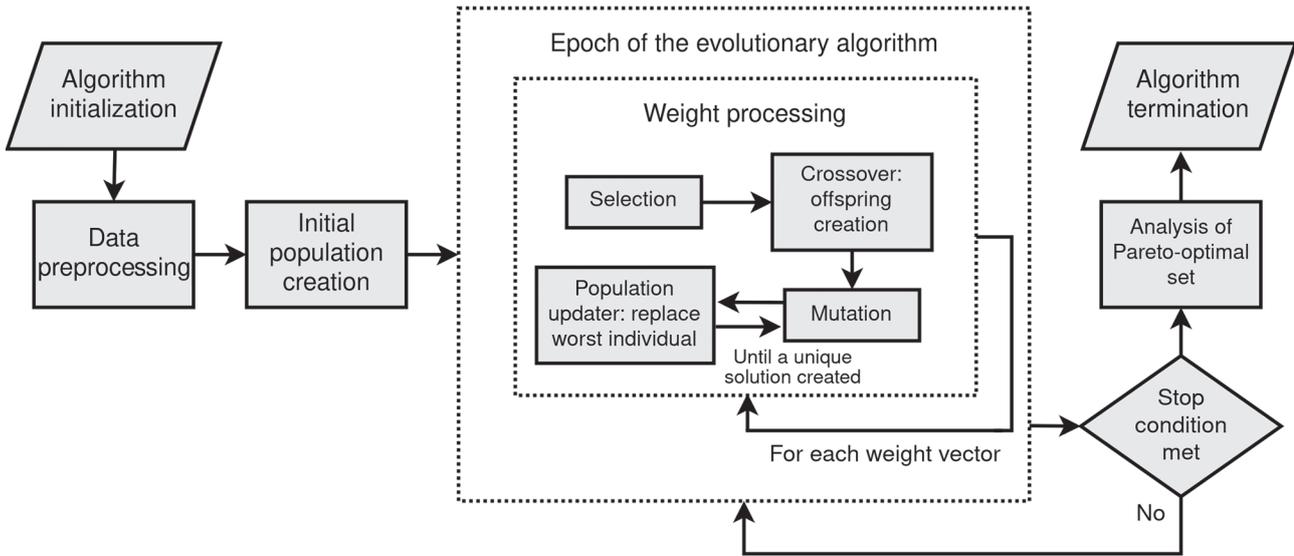


Fig. 2. Generalized scheme of the main search sequence of the algorithm

A hunter-prey model described by Lotka-Volterra equations:

$$S(\mathbf{u}) = \begin{cases} \dot{u} = \alpha u - \beta uv \\ \dot{v} = \delta uv - \gamma u \end{cases}$$

was selected as the dynamical system to be described. The model represents simplified dynamics of two species: $u = u(t)$ depicts “prey”, while $v = v(t)$ represents the “hunter” species. Usual time-derivative convention $\dot{u} = \frac{du}{dt}$ is implemented. Constants α , β , δ , and γ determine the dynamics of the system.

The solutions of the equations were numerically obtained using Runge-Kutta methods. The solutions for $u(t)$ and $v(t)$ are demonstrated in Fig. 3.

The Pareto-optimal set of equations obtained from the algorithm typically has forms similar to the one presented in Fig. 4. Here, the algorithm output is reformulated with

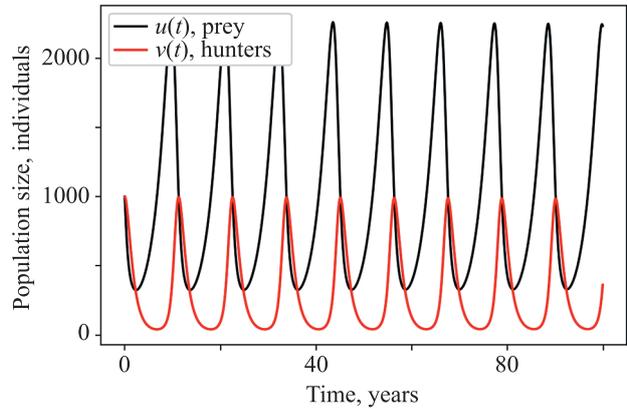


Fig. 3. Visualization of the solution of Lotka-Volterra equations

the combination of optimized metrics: instead of evaluating complexity or approximation errors of individual equations

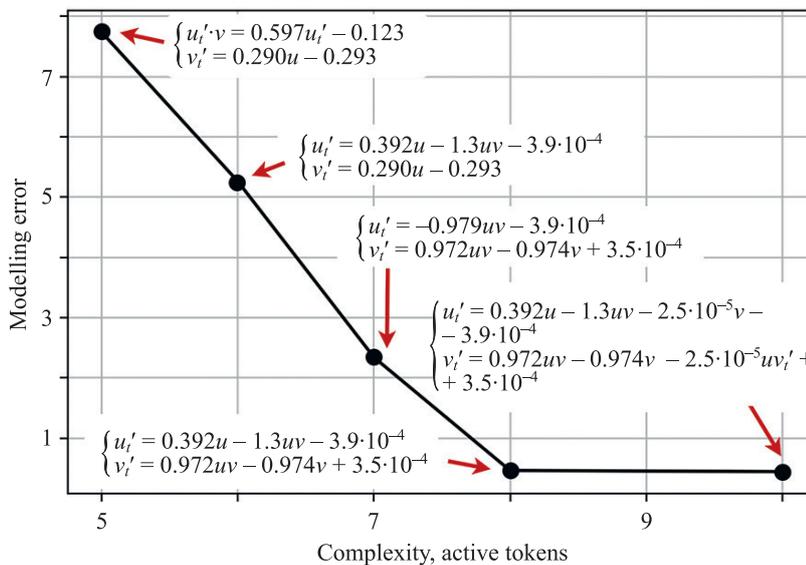


Fig. 4. Pareto frontier of systems of equations obtained by the algorithm

they are viewed for the system integrally. The allowable interval for complexity controlling parameters λ used in sparsity operators is between 10^{-8} and 10^{-2} . Next, an additional family of trigonometric tokens was introduced into the pool to create a diversity of created terms.

Ten independent runs with ten multi-objective optimization evolutionary algorithm iterations and ten more with 25 iterations were performed, and the obtained Pareto-optimal sets were compared. Due to the relatively simple structure of the initial system of equations, a successful convergence to the similar (in terms of obtaining sets with similar equations) was achieved in every case. While this test cannot be considered a comprehensive study of the algorithm properties, it can be viewed as proof that the algorithm can operate and discover the equations.

Conclusion

In this article, we proposed a robust extension of the single differential equation discovery approach to the problems of creating models for systems of differential equations. The multi-objective approach enables the

creation of a diverse set of models. With the analysis of complexity-quality tradeoff, an expert should be able to select the parsimonious model for the process description. The approach has high levels of versatility that are untypical and novel among equation discovery algorithms. It can obtain both ordinary and partial differential equations with arbitrary structures.

The main drawback of the developed approach has high computational cost which can be especially noticeable in multidimensional data (i.e., systems of partial differential equations) or data with high noise levels where high numbers of iterations are required for the algorithm convergence. Therefore, improving the algorithm computational performance can be the priority for further development. Also, developing sufficient tools for using the derived equations for the process state prediction is another goal of the next research.

The numerical solution data and the Python code that partially reproduce the experiments are available at the GitHub repository of the ITMO University¹.

¹ Available at: <https://github.com/ITMO-NSS-team/EPDE> (accessed: 07.12.2022).

References

1. Bubnova A.V., Deeva I., Kalyuzhnaya A.V. MiXBN: library for learning Bayesian networks from mixed data. *Procedia Computer Science*, 2021, vol. 193, pp. 494–503. <https://doi.org/10.1016/j.procs.2021.10.051>
2. Maslyaev M., Hvatov A. Solver-based fitness function for the data-driven evolutionary discovery of partial differential equations. *Proc. of the 2022 IEEE Congress on Evolutionary Computation (CEC)*, 2022. <https://doi.org/10.1109/cec55065.2022.9870370>
3. Brunton S.L., Brunton B.W., Proctor J.L., Kaiser E., Kutz J.N. Chaos as an intermittently forced linear system. *Nature Communications*, 2017, vol. 8, no. 1, pp. 19. <https://doi.org/10.1038/s41467-017-00030-8>
4. Schmid P.J., Sesterhenn J. Dynamic mode decomposition of numerical and experimental data. *Proc. of the 61st Annual Meeting of the APS Division of Fluid Dynamics*. American Physical Society, November 2008.
5. Kondrashov D., Chekroun M.D., Ghil M. Data-driven non-Markovian closure models. *Physica D: Nonlinear Phenomena*, 2015, vol. 297, pp. 33–55. <https://doi.org/10.1016/j.physd.2014.12.005>
6. Schmidt M., Lipson H. Distilling free-form natural laws from experimental data. *Science*, 2009, vol. 324, no. 5923, pp. 81–85. <https://doi.org/10.1126/science.1165893>
7. Kaheman K., Kutz J.N., Brunton S.L. SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2020, vol. 476, no. 2242, pp. 20200279. <https://doi.org/10.1098/rspa.2020.0279>
8. Berg J., Nyström K. Data-driven discovery of PDEs in complex datasets. *Journal of Computational Physics*, 2019, vol. 384, pp. 239–252. <https://doi.org/10.1016/j.jcp.2019.01.036>
9. Han G., Zahr M.J., Wang J.-X. Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 2022, vol. 390, pp. 114502. <https://doi.org/10.1016/j.cma.2021.114502>
10. Long Z., Lu Y., Ma X., Dong B. PDE-Net: learning PDEs from data. *Proceedings of Machine Learning Research*, 2018, vol. 80, pp. 3208–3216.
11. Raissi M. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 2018, vol. 19, pp. 1–24.
12. Zhang J., Ma W. Data-driven discovery of governing equations for fluid dynamics based on molecular simulation. *Journal of Fluid*

Литература

1. Bubnova A.V., Deeva I., Kalyuzhnaya A.V. MiXBN: library for learning Bayesian networks from mixed data // *Procedia Computer Science*. 2021. V. 193. P. 494–503. <https://doi.org/10.1016/j.procs.2021.10.051>
2. Maslyaev M., Hvatov A. Solver-based fitness function for the data-driven evolutionary discovery of partial differential equations // *Proc. of the 2022 IEEE Congress on Evolutionary Computation (CEC)*. 2022. <https://doi.org/10.1109/cec55065.2022.9870370>
3. Brunton S.L., Brunton B.W., Proctor J.L., Kaiser E., Kutz J.N. Chaos as an intermittently forced linear system // *Nature Communications*. 2017. V. 8. N 1. P. 19. <https://doi.org/10.1038/s41467-017-00030-8>
4. Schmid P.J., Sesterhenn J. Dynamic mode decomposition of numerical and experimental data // *Proc. of the 61st Annual Meeting of the APS Division of Fluid Dynamics*. American Physical Society, November 2008.
5. Kondrashov D., Chekroun M.D., Ghil M. Data-driven non-Markovian closure models // *Physica D: Nonlinear Phenomena*. 2015. V. 297. P. 33–55. <https://doi.org/10.1016/j.physd.2014.12.005>
6. Schmidt M., Lipson H. Distilling free-form natural laws from experimental data // *Science*. 2009. V. 324. N 5923. P. 81–85. <https://doi.org/10.1126/science.1165893>
7. Kaheman K., Kutz J.N., Brunton S.L. SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics // *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2020. V. 476. N 2242. P. 20200279. <https://doi.org/10.1098/rspa.2020.0279>
8. Berg J., Nyström K. Data-driven discovery of PDEs in complex datasets // *Journal of Computational Physics*. 2019. V. 384. P. 239–252. <https://doi.org/10.1016/j.jcp.2019.01.036>
9. Han G., Zahr M.J., Wang J.-X. Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems // *Computer Methods in Applied Mechanics and Engineering*. 2022. V. 390. P. 114502. <https://doi.org/10.1016/j.cma.2021.114502>
10. Long Z., Lu Y., Ma X., Dong B. PDE-Net: learning PDEs from data // *Proceedings of Machine Learning Research*. 2018. V. 80. P. 3208–3216.
11. Raissi M. Deep hidden physics models: Deep learning of nonlinear partial differential equations // *Journal of Machine Learning Research*. 2018. V. 19. P. 1–24.
12. Zhang J., Ma W. Data-driven discovery of governing equations for fluid dynamics based on molecular simulation // *Journal of Fluid Mechanics*. 2020. V. 892. P. A5. <https://doi.org/10.1017/jfm.2020.184>

- Mechanics*, 2020, vol. 892, pp. A5. <https://doi.org/10.1017/jfm.2020.184>
13. Van Breugel F., Kutz J.N., Brunton B.W. Numerical differentiation of noisy data: A unifying multi-objective optimization framework // *IEEE Access*, 2020, vol. 8, pp. 196865–196877. <https://doi.org/10.1109/access.2020.3034077>
 14. Maslyaev M., Hvatov A., Kalyuzhnaya A. Partial differential equations discovery with EPDE framework: Application for real and synthetic data. *Journal of Computational Science*, 2021, vol. 53, pp. 101345. <https://doi.org/10.1016/j.jocs.2021.101345>
 15. Li K., Deb K., Zhang Q., Kwong S. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 2015, vol. 19, no. 5, pp. 694–716. <https://doi.org/10.1109/TEVC.2014.2373386>
 16. Das I., Dennis J.E. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 1998, vol. 8, no. 3, pp. 631–657. <https://doi.org/10.1137/s1052623496307510>

Authors

Mikhail A. Maslyaev — Junior Researcher, ITMO University, Saint Petersburg, 197101, Russian Federation, <https://orcid.org/0000-0001-5595-0802>, mikemaslyaev@itmo.ru

Alexander A. Hvatov — PhD (Physics & Mathematics), Head of Laboratory, ITMO University, Saint Petersburg, 197101, Russian Federation, <https://orcid.org/0000-0002-5222-583X>, alex_hvatov@itmo.ru

Received 11.10.2022

Approved after reviewing 07.12.2022

Accepted 15.01.2023

Авторы

Масляев Михаил Александрович — младший научный сотрудник, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, <https://orcid.org/0000-0001-5595-0802>, mikemaslyaev@itmo.ru

Хватов Александр Александрович — кандидат физико-математических наук, заведующий лабораторией, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, <https://orcid.org/0000-0002-5222-583X>, alex_hvatov@itmo.ru

Статья поступила в редакцию 11.10.2022

Одобрена после рецензирования 07.12.2022

Принята к печати 15.01.2023



Работа доступна по лицензии
Creative Commons
«Attribution-NonCommercial»