Multi-Objective Evolutionary Design of Composite Data-Driven Models

Iana S. Polonskaia
Natural Systems Simulation Lab
ITMO University
Saint-Petersburg, Russia
ispolonskaia@itmo.ru

Nikolay O. Nikitin
Natural Systems Simulation Lab
ITMO University
Saint-Petersburg, Russia
nnikitin@itmo.ru

Ilia Revin
Natural Systems Simulation Lab
ITMO University
Saint-Petersburg, Russia

Pavel Vychuzhanin
Natural Systems Simulation Lab
ITMO University
Saint-Petersburg, Russia

Anna V. Kalyuzhnaya
Natural Systems Simulation Lab
ITMO University
Saint-Petersburg, Russia

Abstract—In this paper, a multi-objective approach for the design of composite data-driven mathematical models is proposed. It allows automating the identification of graph-based heterogeneous pipelines that consist of different blocks: machine learning models, data preprocessing blocks, etc. The implemented approach is based on a parameter-free genetic algorithm (GA) for model design called GPComp@Free. It is developed to be part of automated machine learning solutions and to increase the efficiency of the modeling pipeline automation. A set of experiments was conducted to verify the correctness and efficiency of the proposed approach and substantiate the selected solutions. The experimental results confirm that a multi-objective approach to the model design allows us to achieve better diversity and quality of obtained models. The implemented approach is available as a part of the open-source AutoML framework FEDOT.

Index Terms—AutoML, evolutionary algorithms, multi-objective optimization, model design, composite models

I. INTRODUCTION

The design of data-driven mathematical models is an actual research direction in modern data science [1]. The internal structure of the model depends on the type of the learning algorithm, so complex data-driven models can consist of several semi-independent blocks - this approach is usually referred to as ensembling [2]. There are several techniques to build complex models: for example, blending allows creating single-level ensembles of machine learning (ML) models, and stacking allows creating multi-level ones. Other approaches are based on the representation of a model structure (or even the whole modeling pipeline) as a directed acyclic graph (DAG).

The selection of the most suitable model design is the core problem in automated machine learning (AutoML) [3]. Until recently, successful machine learning pipelines have been designing manually by experts. However, this method requires performing many routine operations and repeating experiments which is generally a computationally expensive procedure that does not always lead to success.

The aim of the model design search is to achieve the appropriate values of different criteria (e.g. quality of the modeling) and satisfy the constraints (e.g. time-related ones). In many cases, it is impossible to maximize all criteria simultaneously, which leads to a multi-objective optimization problem. The usage of multi-objective approaches in AutoML solutions is a quite promising direction that can lead to better suitability of the obtained ML pipelines and even makes it possible to co-design the models and infrastructure [4].

In this paper, a flexible and effective multi-objective evolutionary approach (EA) to the composite model design is proposed and implemented as a part of the automated modeling framework FEDOT. We modified the standard evolutionary scheme with a self-configuration heuristic for the hyper-parameters (such as population size, crossover and mutation probabilities, maximal model structure graph size). The performed experiments demonstrate that the implemented approach named GPComp@Free provides an ability to design effective data-driven models from scratch.

II. RELATED WORK

There is a variety of approaches that can be used to identify the optimal design of a data-driven model. For instance, AutoML solutions can be based on random search [5], Bayesian optimization [6], reinforcement learning (RL) [7], Monte Carlo tree search [8], gradient-based approaches [9]. However, most of them are less flexible than EAs to the model design (implemented e.g. in [10]). Their conceptual advantage is open-endedness [11], which allows "growing" of the model with the most suitable structure instead of the direct search of appropriate solutions in some restricted range of structural parameters.

Furthermore, evolutionary algorithms (EA) are much more flexible due to their simplicity and clarity and can be easily extended with new evolutionary operations. Thus, in [12] the authors demonstrate a tournament selection modification with memory (which contains the best genotypes) based on the association of genotypes with their age, and displacement of the tournament selection to choose the "younger" genotypes. This

1

modification allows us to obtain similar results in convolution neural network architecture search problems faster than when using well-established reinforcement learning (RL) algorithms. Another case when the EA competes with RL approaches is presented in [13]. Authors demonstrate the advantages of evolution strategies in comparison with popular RL techniques (such as Q-learning and policy gradients) on test problems from the OpenAI Gym library.

Multi-objective optimization is implemented in several existing AutoML solutions, e.g. TPOT [14] and autoxgboostmc [15]. However, the diversity of the obtained solutions is quite limited in these tools (because the custom graph-based model structure is not supported). Also, multi-objective optimization is widely used as a part of Neural Architecture Search approaches (NAS). For example, in [16] the second objective function is usually based on structural complexity. Besides the model structure design, the multi-objective hyperparameters tuning can be applied to ML [17] and domain-specific models.

Various criteria can be involved in the multi-objective model design. Besides the basic characteristics like quality and complexity, there are additional objectives that can be used, e.g. fairness, interpretability, robustness, and the sparsity of models (as described in [15]). However, in [15] multi-objective optimization is applied to a single machine learning model and graph-based structures are not discussed.

It can be concluded that the existing methods in this field are quite specialised, which leads us to the aim of developing a more universal and effective multi-objective approach that can be used to create an effective design of the various data-driven models with graph-based structure.

III. Problem statement

There is a lot of different modeling tasks that can be solved using machine learning: classification, clustering, regression, time series forecasting, ranking. However, all of them can be formulated as results of evaluation of the model M.

In a common case, the structure of the model M can be represented as a graph that consists of several atomic models. An atomic model is a machine learning model that has a logically indivisible structure in terms of a particular problem. In fact, an atomic model can have an arbitrary internal structure (e.g. an ensemble obtained by gradient boosting). It is a quite flexible representation that allows us to describe different types of models beyond machine learning: hybrid [18], equation-based [19], etc. In this case, M can be formalized as a composite model - a data-driven model with an explicit internal structure, where several atomic models can be identified.

Since the data flow is directed from the input X of the model to the output X with the general statement of numerical simulation problem, the internal structure of the model M can be represented as a directed acyclic graph (DAG) M^G and defined as follows:

$$Y = \mathcal{H}(M|X), M^G = \langle V_i, E_j \rangle = \langle A_i, \{H_{A_i}\}_k, E_j \rangle, \quad (1)$$

where $\mathcal{H}$ is a modelling operator for the model M with input data X and Y as model results. For ML models, the data set X is usually split into subsamples X_{train} and X_{test}. In the graph model M^G, the vertices V have complex structure and can be represented as tuple $\langle A_i, \{H_{A_i}\}_k \rangle$ with different data-driven models A_i and its hyper-parameters $\{H_{A_i}\}_k$. Directed edges E represent the data flow from income vertices (the model's input) to outcome vertices (the model's output).

The application of the evolutionary optimization approaches leads to the following interpretation: M^G structure can be considered as the genotype and phenotype is expressed through integral characteristics of realization M^G. In the frame of this paper, the integral characteristics fill the objective vector function F.

As can be seen from Sec. II, there are different objective functions that can be used in the design of the composite model: modeling quality for a specific data set, structural complexity, computation time, interpretability, robustness, number of features required to build the model (related to the sparsity). Also, specific infrastructure-related objectives can be involved as part of the co-design and scheduling tasks [20]. In this paper, we decided to consider the quality, structural complexity, and computational performance objectives.

In this case, the formulation of the optimization task can be defined as follows:

$$M_{opt}^G = \underset{M^G}{\operatorname{argmin}} F(A_i, \{H_{A_i}\}_k, E_j), \quad (2)$$

$$F(M^G) = (Q(M^G), S(M^G), P(M^G)),$$
$$F_{hyp} = HV(F(M^G)), \quad (3)$$

where F is a vector objective function, Q is a quality-based criterion for the prediction obtained by the model M^G, S is the structural complexity criteria (tightly connected with the interpretability of the model), P is the computational performance criterion of the model. Here we miss arguments X and Y for compactness of formulas and because these conditions remain constant while optimization proceeds. Additional measure F_{hyp} that helps to join components of vector objective function is Pareto hypervolume operator HV.

There are different interpretations of the quality of the multi-objective design that can be used. In the paper, we used the following set of criteria: the root-mean-seared-error (RMSE) as Q criterion for the regression problems, the negative area under the ROC curve ((-1)*(ROC AUC)) as Q criterion for the classification problems, fitting time as the computational complexity criterion P, graph size G_s as a structural complexity criterion S, and hypervolume [21] as a quality measure for the Pareto frontiers.

The possible alternative to the 'real' multi-objective approach is the application of penalty-based optimization algorithm. In this case, the Eq. 3 can be formulated as follows:

$$F(M^G) = Q(M^G) - S(M^G)w_1 - P(M^G)w_2, \quad (4)$$

where w_1 and w_1 are weights for the complexity penalties. However, this implementation can affect the diversity during

the optimization and lead to insufficient efficiency of the model design.

The described statement of the composite model problem is summarized in Figure 1.

In this paper, the following hypotheses can be stated:

Hypothesis 1: The use of multi-objective optimization as a part of composite models identification methods allow us to obtain the models with better quality than single-objective approaches due to the more effective preservation of the diversity.

Hypothesis 2: The selection of the appropriate genetic operators makes it possible to increase the quality of the obtained solutions for different cases.

Hypothesis 3: The hyper-parameters of the multi-objective evolutionary algorithm can be self-configured during the evolution using heuristic techniques without the computationally expensive meta-optimization methods.

IV. MULTI-OBJECTIVE DESIGN OF COMPOSITE MODELS

A. Evolutionary optimization of the models structure

There are many extensively used conventional optimization techniques, but evolutionary algorithms deserve particular attention due to their range of advantages. They have a lower probability of getting stuck in a local optima than most optimization algorithms because they perform a global search in the solution space. In addition, such algorithms can be easily parallelized. This property is especially important for applied engineering problems since the time for finding the solution is usually restricted. Moreover, evolutionary algorithms are easily interpreted for almost any problem, therefore the user of an evolutionary-based framework always has an opportunity to adapt it to another optimization problem (e.g. to the problem of the neural architecture search).

The proposed implementation of composite models structure optimizer is based on genetic programming (GP) algorithms [22]. Unlike many other algorithms, genetic programming allows us to use a high-level problem statement. It finds a composite model (or several candidate models in the multi-objective case) for a certain machine learning problem (classification, regression and time series forecasting problems are currently supported) from a pre-defined set of available models.

The algorithm initializes the population of chains with random structure at the first step. Next, a few models in the population are selected as the parents using a preferred selection method. Each new model (offspring) is constructed from two parents by the application of a crossover operator, and some of them are impacted by a transformation called a mutation. Each operator has a performing probability which can be static or dynamic depending on the chosen evolutionary scheme. Once the child model structure is produced, it is then fitted with a training sample, evaluated with a test sample, and added to the new population. These manipulations are performed during several generations until the termination criteria are satisfied.

In experiments we used subtree crossover which replaces random subtree in one parent to random subtree from another and one-point crossover which performs in the same way, but subtrees is selected from common structural parts between two trees. Three mutation's options: simple, subtree and reduce (first option changes models in random nodes, second changes random subtree to randomly generated and last removes random subtree). In [23] we described in detail the composite model representation in the genetic programming algorithm GPComp and how the custom evolutionary operators are used. Thus, in this paper, we focus on the aspects of multi-objective evolutionary optimization technique, describe implemented evolutionary schemes, and demonstrate its effectiveness on certain data sets for machine learning algorithms benchmarking.

The composite model optimization algorithm GPComp includes two of the most common evolutionary schemes: steady-state (also named $(\mu + \lambda)$) and generational. In the first scheme, the new population is generated by using a selection operator which is applied to the union of the offspring and the previous population. In the generational scheme the offspring completely replaces the parent population. Our previous experiments (also in [23]) demonstrate the clear domination of steady-state scheme over the generational. Thus, we didn't apply the generational scheme in this paper.

B. GPComp@Free: algorithm for parameters-less multi-objective evolution of composite models

Although the AutoML concept focuses on reducing (or excluding altogether) human participation in the optimization process, the majority of the state-of-the-art tools raise an additional meta-optimization problem related to the identification of their most-effective hyperparameters. It is well-known that the EA has been solving many real-world optimization problems successfully, but their performance can depend considerably on the particular hyperparameters values used. Hyperparameters self-configuration procedures allow us to achieve complete automation. Our parameter-free multi-objective genetic scheme (GPComp@Free) based on an adaptive EA (described in [24]) allows configuring most of EA hyperparameters through the algorithm execution unlike aforementioned GPComp which suppous hand-tuning of hyperparameters. The proposed algorithm uses a steady-state evolutionary scheme, but μ (population size) changes during evolution like the Fibonacci sequence and λ always equals to the previous item of the sequence with respect to μ. Crossover and mutation rates (c_{rate}, m_{rate}) are also changed depending on the diversity of the population. The scheme of this approach is presented in Figure 2. The approach assumes two optimization objectives are used: solution quality (main goal) and chain complexity. We used two conditions for the hyperparameters changes: if none of the objective function values is improved then μ is increased to the next number in the sequence and, if both objectives (quality and complexity) are improved in the offspring population (with respect to their parents) then μ is decreased to the previous number in the
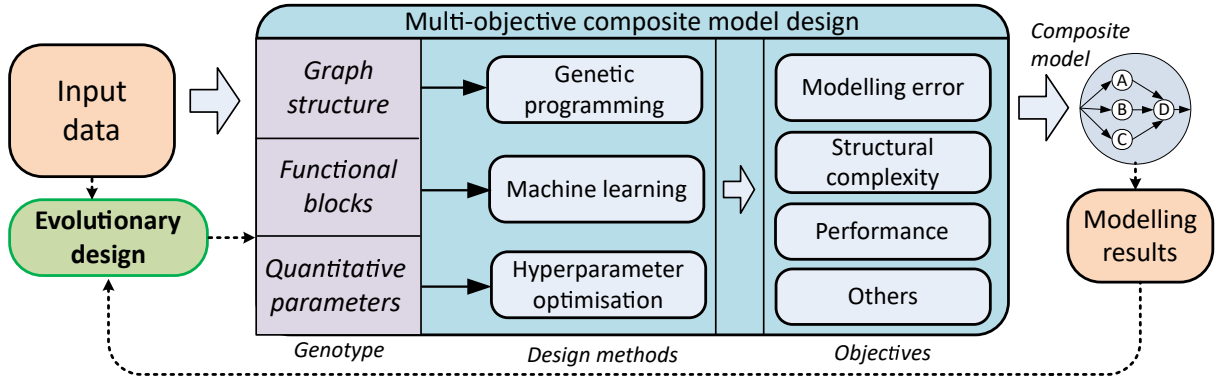
Fig. 1: The concept of the multi-objective evolutionary design of the composite machine learning model. The internal structure of the model, design approaches, possible objectives and their interactions are presented as separate blocks to highlight the main issues of the optimization.

sequence and crossover and mutation rates change according to the rule described in the AdaptedEvoParams procedure (Alg. 1).
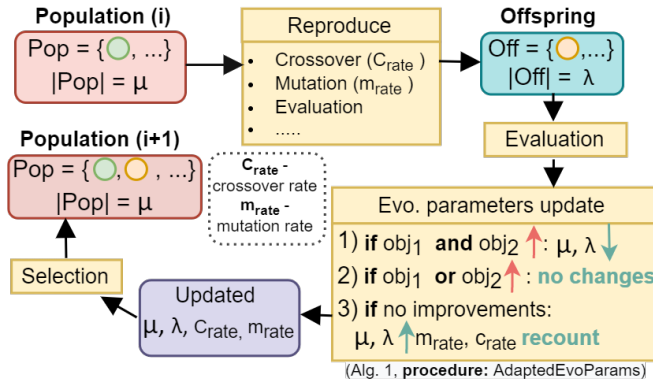

Fig. 2: The description of one iteration of the parameter-free evolutionary algorithm. Population size (μ), offspring size (λ), crossover and mutation rates (c_{rate}, m_{rate}) adapt during the algorithm execution.

Also, we implemented the heuristics adaptation of chains' depth which allows avoiding over-complicated individuals. The algorithm with depth adaptation starts from low-depth chains and if there are no possibilities to improve objectives (stagnation during several generations) then depth is increased (see Alg. 1, DepthAdaptation procedure). The details of the parameter-free optimizer implementation (further in the paper it will be called GPComp@Free) are described in Alg. 1. The version without depth adaptation will be called "GP-Comp@Free with fixed depth" in the paper. Both algorithms are implemented using DEAP [25].

The described approach is implemented as a part of the automated modeling framework FEDOT [18]. It supports both single- and multi-objective optimization of the ML pipelines that includes the composite models. The main feature of the framework is the complex management of interactions between various computing elements of the pipelines. The

TABLE I: Key features of the selected data sets

Data set	$N_{samples}$	$N_{features}$	Task type	Imbalance
Churn	5000	20	binary clf.	0.51
Dis	3772	29	binary clf.	0.93
Hill_Valley	1212	100	binary clf.	9.81
Elusage	55	2	regr.	-

framework can be used to automate the creation of mathematical models for various problems, different types of data, and models. It is available under the open-source license in https://github.com/nccr-itmo/FEDOT.

V. EXPERIMENTAL STUDIES

A. Setup of experiments

To analyze the correctness and effectiveness of the proposed approach, a set of experiments was conducted using several data sets from the Penn Machine Learning Benchmarks repository [26]. These data sets cover a broad range of applications, and combinations of categorical, ordinal, and continuous features. There are no missing values in these data sets. Selected data sets were split on training and test sets in the ratio of 70/30.

Key features of the selected data sets are shown in I. Feature 'Imbalance' show a value of imbalance metric, where zero means that the data set is perfectly balanced and the higher the value, the more imbalanced the data set.

There were three complex experiments prepared:

Experiment 1: Comparison of the various model design approaches. The single-objective genetic algorithm (GPComp), a single-objective genetic algorithm with penalty-based fitness (described in Eq. 4), and the multi-objective algorithm GP-Comp@Free were used. The aim of the experiment is to ensure that the multi-objective approach can find models with equal or better prediction quality than the single-objective approach (see Hypothesis 1).

Experiment 2: Comparison of different implementations of the GPComp@Free algorithm. Two variants of the algorithm are based on different strategies inspired by well-known algorithms SPEA2 [27] and NSGA-II [28]. The aim of the

Algorithm 1 GPComp@Free (parameter-free implementation of the multi-objective GP algorithm for the model design)

1: **procedure** PARAMETERFREECOMPOSER(models, data, objectiveFunctions, timeConstraint)
2:     Input: $objectiveFunctions = \{qualityObjective, complexityObjective\}$
3:     Output: array with nondominated models (Pareto frontier)
4:     $stagnationThreshold \leftarrow 5$ ▷ Allowed number of populations without improvement before max depth update
5:     $maxDepth \leftarrow 2$ ▷ Inital values of chain depth
6:     $SequenceFunction \leftarrow FibonacchiIterator$ ▷ Heuristic for the population size
7:     $popSize \leftarrow SequenceFunction.StateByIndex(index = 2)$ ▷ Default size equals to second sequence item
8:     $crossRate, mutRate \leftarrow 0.5, 0.5$
9:     $pop \leftarrow$ INITPOPULATION($models, popSize$)
10:     $pop \leftarrow$ EVALUATE($pop, objectiveFunctions, data$)
11:     $currentStd, maxStd \leftarrow$ QUALITYSTDCALCULATION(pop)   ▷ Standard deviation for prediction quality objective
12:     **while** $time() < timeConstraint$ **do**
13:         $offspringSize \leftarrow SequenceFunction.PreviousState(popSize)$
14:         $pareto, selectedParents \leftarrow$ UPDATEPARETO(pop), MULTIOBJSELECTION($pop, offspringSize$)
15:         $offspring \leftarrow$ REPRODUCT($selectedParents, offspringSize, crossRate, mutRate, maxDepth, models$)
16:         $pop \leftarrow$ EVALUATE($offspring, objectiveFunctions, data$)
17:         $stagnationCnt, maxDepth \leftarrow$ DEPTHADAPTATION($stagnationCnt, pareto, stagnationThreshold, maxDepth$)
18:         $mutRate, crossRate, popSize \leftarrow$ ADAPTEDEVOPARAMS($offspring, pareto, popSize, mutRate, crossRate$)
19:         $pop \leftarrow$ MULTIOBJSELECTION($pop \cup offspring, popSize$)
20:         $currentStd \leftarrow$ QUALITYSTDCALCULATION(pop)
21:         $maxStd \leftarrow$ UPDATEMAXSTD($currentStd$)
22:     **return** $pareto$
23: **procedure** DEPTHADAPTATION(stagnationCnt, pareto, stagnationThreshold, maxDepth)
24:     $stagnationCnt \leftarrow$ UPDATESTAGNATIONCNT($pareto$)   ▷ Number of generations without Pareto frontier's changes
25:     **if** $stagnationCnt == stagnationThreshold$ **then**
26:         $maxDepth \leftarrow maxDepth + 1$ ▷ Increase the maximum allowed depth of the composite model structure graph
27:     **return** $stagnationCnt, maxDepth$
28: **procedure** ADAPTEDEVOPARAMS(offspring, pareto, popSize, mutRate, crossRate)
29:     **if** NOOBJECTIVESIMPROVEMENTS(offspring, pareto) **then**
30:         $popSize \leftarrow SequenceFunction.NextState(popSize)$
31:         $mutRate, crossRate \leftarrow 1 - (currentStd/maxStd), currentStd/maxStd$
32:     **else if** AREBOTHOBJECTIVESIMPROVED(offspring, pareto) **then**
33:         $popSize \leftarrow SequenceFunction.PreviousState(popSize)$
34:     **return** $mutRate, crossRate, popSize$

---

experiment is to analyze the possibility of choosing a single effective selection strategy for different model design tasks (see Hypothesis 2).

Experiment 3: Comparison of the parameter-free GP-Comp@Free approach against different multi-objective EAs (based on the simple steady-state algorithm and a partial algorithm with a fixed maximum depth of the model structure graph). The aim of the experiment is to verify the effectiveness of the proposed approach against existing multi-objective schemes (see Hypothesis 3).

Every experiment consists of several stages: the evaluation of each version of the evolutionary algorithm (repeated ten times to obtain a stable result); evaluation of the prediction quality (with test sample) and computational time (measured during the evaluations and verified using an empirical performance model [4]); analysis of Pareto frontiers and hypervolume values. The solid area around line on the graph shows the deviation in quality metric for each of the proposed algorithms.

For the non-parameter-free algorithms, the population size was set to 20 and the maximum number of generations was set to 30.

B. Experimental results

Experiment 1: The first experiment is aimed to compare the multi-objective (MO) approach to the single-objective (SO) competitors. From the analysis of the quality metrics for each of the algorithms presented in Fig. 3, we can conclude that starting with the 10th generation the MO algorithm consistently outperforms the SO approach and is slightly inferior to the SO approach with a penalty function. However, later in the runs, with the growth of the number of generations, the MO approach surpasses all others. An important fact that is worth mentioning is that the results of the MO algorithm are much more robust and stable in time compared to the alternative approaches. Summing up the above, it can be argued that with a sufficient number of generations, a MO algorithm is
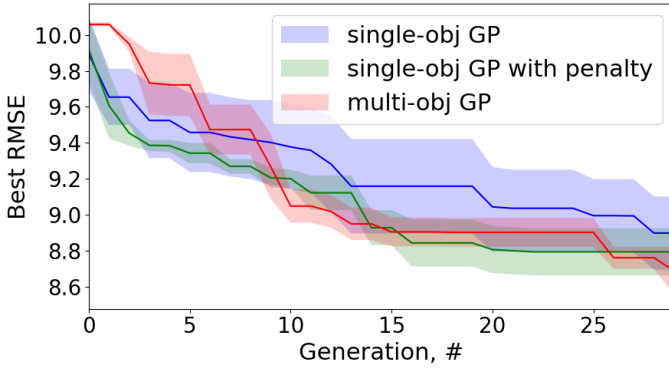
Fig. 3: Comparison of RMSE objective-function value dynamics during the optimization for different algorithms: single and multi-objective (from Experiment 1). The results are obtained with the 228_elusage data set.

more preferable. Partial results for the 228_elusage data set are presented in Fig. 3.

It can be seen that the multi-objective approach allows achieving a slightly better quality of the prediction. The values of the graph size G_s and graph depth G_d from Table II are also lower for the multi-objective case that confirms its practical applicability and allows us to continue the set of experiments and confirms Hypothesis 1.

Experiment 2: In the next experiment, we compared two multi-objective selection strategies. As can be seen in Figure 4, the value of the first objective function (in this case, ROC AUC) for the algorithm with the SPEA2-based selection is higher for each data set. It should also be noted that based on the results of ten runs for each of the algorithms, it can be concluded that SPEA2-based selection is more robust and reliable. The analysis of the number of points in Pareto frontiers (described in Table II) confirms that the SPEA2-based selection is more suitable for further experiments since the Pareto frontiers obtained for this variant of the algorithm are more diverse and contain more points, which confirms the Hypothesis 2.

Experiment 3: The final experiment was devoted to the comparison of different multi-objective approaches. The main idea of this experiment is to find out whether it is possible to get higher values of the objectives using self-configuration of the evolutionary algorithm hyperparameters during optimization. As an alternative to this method, an approach was used in which the values of hyperparameters were set before the optimization began. The results presented in Fig. 5 for the Hill Valley data set indicate that starting from a certain generation (in our case, from the fifth), the approach using self-configured hyperparameters of the evolutionary algorithm begins to consistently outperform the alternative approach. It should also be noted that as the number of generations increases, the self-configured algorithm provides more robust and stable solutions. The results of the experiment for all data sets that are shown in the Table II confirm Hypothesis 3.

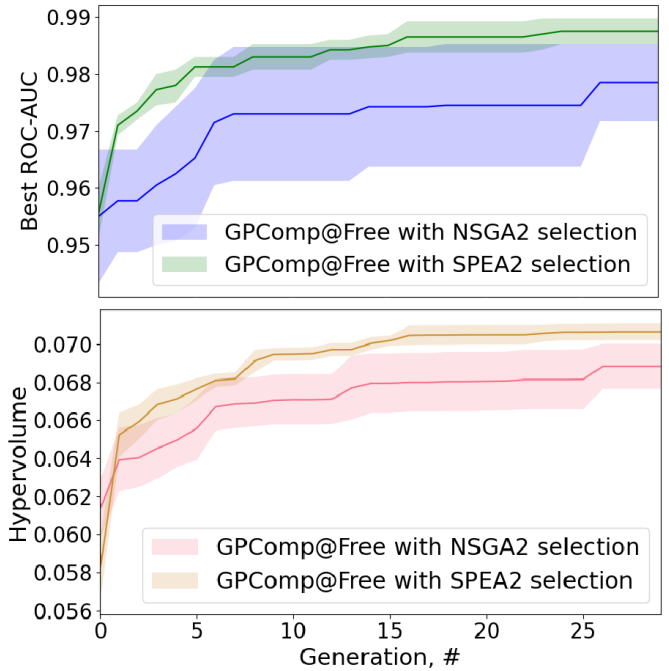Additional comparison of best Pareto frontiers for each

Fig. 4: Comparison of objective function values convergence (top) and hypervolume values convergence (bottom) during the optimization for different selection operators (from Experiment 2). The results are obtained with the Dis data set.

approach in the experiment with the Dis data set and detailed representation of certain models is presented in Figure 6. It can be seen that all points in the frontier for GPComp@Free are dominating the points for the steady-state GP algorithm, which confirms its efficiency.

Detailed results of experiments: The summary of the results for all experiments and data sets is presented in Table II. It also includes the comparison of GPComp@Free with state-of-the-art AutoML approach TPOT (also based on GP concept) and single-model baseline Random Forest with default parameters which demonstrates its greatest efficiency. Best algorithm for each experiment and its solution characteristics highlighted with bold.

As can be seen in this table, the results obtained during the experiments demonstrate the advantage of multi-objective optimization algorithms over other used approaches. The only exception is a single case from the first experiment, where the maximum value of the quality metric was obtained using a single-objective algorithm. However, it should be noted that the difference in the obtained values of the quality metric for the single-objective and multi-objective algorithms is less than 0.01. At the same time, a much more simple model structure was designed in the case of a multi-objective approach. In the second experiment, the SPEA2 selection showed higher results for each of the three selected data sets. Thus, this algorithm was chosen for the next experiments. Results obtained during the third experiment confirm that the parameter-free optimization approach GPComp@Free allows achieving

TABLE II: Results of the experiments for different versions of the single-objective algorithm and the multi-objective algorithm. The algorithms with the complexity penalty term (PT) and the fixed depth heuristic (FD) were involved in the comparison. ROC AUC, root mean squared error (RMSE) and hypervolume (F_{hyp}) were used to evaluate the solutions. G_s and G_d represent the size and the depth of the model structure graph. The classification and regression benchmarks from PMLB repository were used as test cases. For all experiments, the deviation in the obtained quality metric was less then 5 percent.

| Data set	Option	Quality (ROC AUC or RMSE)	G_s and G_d for best	F_{hyp}	N_f
Exp. #1 - Comparison of the single-obj and multi-obj approaches					
Churn	**Single-objective**	**0.932**	**15;3**	**0.142**	**20**
	Single-objective PT	0.921	6;3	0.139	20
	Multi-objective	0.924	7;3	0.14	5
Dis	Single-objective	0.928	13;3	0.067	4
	Single-objective PT	0.921	6;3	0.067	5
	Multi-objective	**0.958**	**7;3**	**0.069**	**10**
Hill Valley	Single-objective	0.995	12;3	0.032	3
	Single-objective PT	0.921	5;2	0.030	5
	Multi-objective	**0.999**	**4;2**	**0.033**	**6**
Elusage (regr.)	Single-objective	8.477	10;4	0.235	20
	Single-objective PT	8.367	12;4	0.237	20
	Multi-objective	**8.364**	**8;3**	**0.240**	**4**
Exp. #2 - comparison of the multi-obj selections types					
Churn	NSGA selection	0.924	5;3	0.140	4
	SPEA2 selection	**0.929**	**5;3**	**0.140**	**10**
Dis	NSGA selection	0.994	6;3	0.067	5
	SPEA2 selection	**0.995**	**6;3**	**0.069**	**5**
Hill Valley	NSGA selection	0.982	7;3	0.032	4
	SPEA2 selection	**0.986**	**6;3**	**0.033**	**6**
Elusage (regr.)	NSGA selection	8.413	13;3	0.236	4
	SPEA2 selection	**8.314**	**6;3**	**0.240**	**5**
Exp. #3 - Comparison of multi-obj algorithms					
Churn	**Paremeter-free with FD**	**0.929**	**8;3**	**0.435**	**10**
	GPComp@Free	0.928	6;3	0.430	8
	Steady-state with FD	0.923	7;3	0.427	7
	Steady-state	0.928	6;3	0.427	8
Dis	Parameter-free with FD	0.992	8;3	0.761	8
	GPComp@Free	**0.994**	**6;3**	**0.771**	**7**
	Steady-state with FD	0.991	6;3	0.756	8
	Steady-state	0.99	5;2	0.758	6
Hill Valley	Parameter-free with FD	0.972	8;4	0.032	4
	GPComp@Free	**0.974**	**5;3**	**0.032**	**5**
	Steady-state with FD	0.960	6;3	0.030	4
	Steady-state	0.973	5;3	0.031	5
Elusage (regr.)	**Parameter-free with FD**	**8.583**	**7;3**	**0.223**	**5**
	GPComp@Free	8.762	10;3	0.219	7
	Steady-state with FD	8.825	7;3	0.198	6
	Steady-state	9.211	6;3	0.137	4
Comparasion with state-of-art and baseline					
Churn	TPOT	0.908	2;1	-	-
	Random forest	0.887	1;1	-	-
	GPComp@Free	**0.924**	**7;3**	**0.14**	**5**
Dis	TPOT	0.932	3;1	-	-
	Random forest	0.918	1;1	-	-
	GPComp@Free	**0.994**	**6;3**	**0.771**	**7**
Hill Valley	TPOT	0.999	3;1	-	-
	Random forest	0.638	1;1	-	-
	GPComp@Free	**0.999**	**4;2**	**0.033**	**6**
Elusage (regr.)	TPOT	11.682	3;1	-	-
	Random forest	13.417	1;1	-	-
	GPComp@Free	**8.364**	**8;3**	**0.240**	**4**
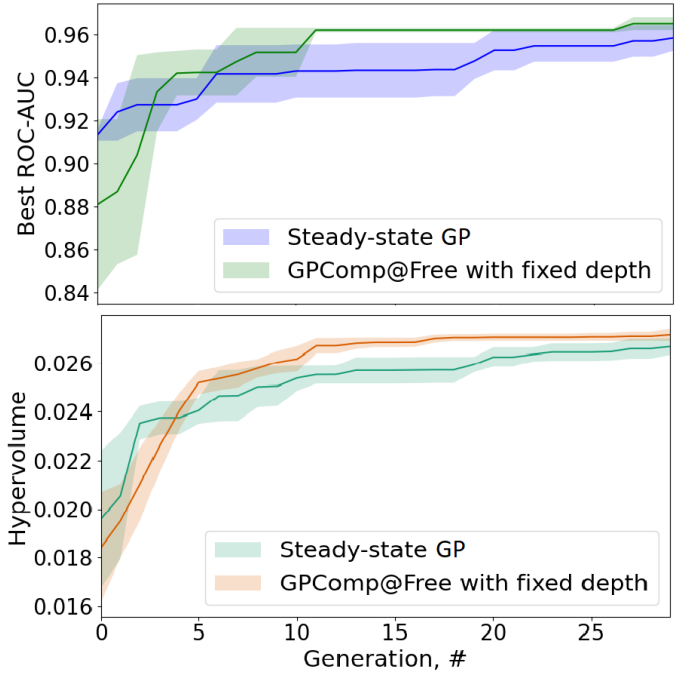
Fig. 5: Comparison of objective function and hypervolume convergence during the optimization of the different types of multi-objective approaches (from Experiment 3). The two variants of the steady-state GP and two variants of the GP-Comp@Free are presented. The results are obtained with the Hill Valley data set.
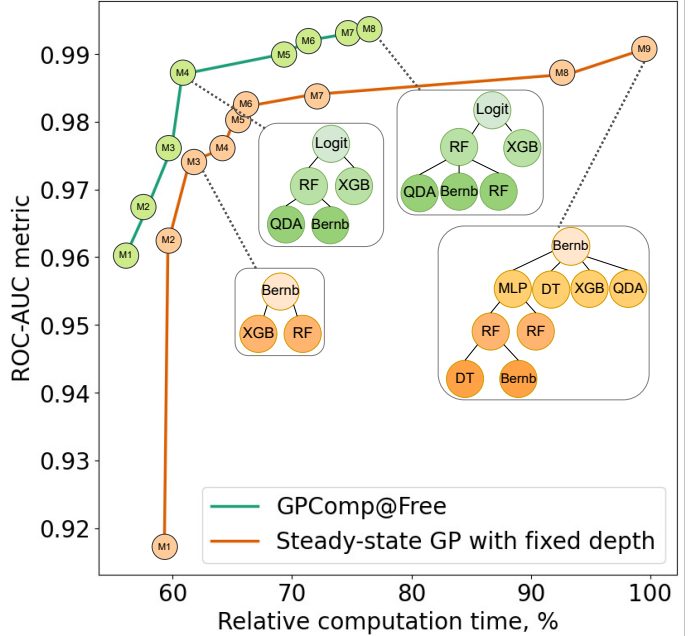

Fig. 6: Pareto frontiers for multi-objective optimization with different algorithms for the Hill valley data set. The internal structure of the obtained composite models is presented as graphs with names of the used ML models in nodes.

7

better results compared to the other approaches. The version of the algorithm with the non-fixed depth of the final composite model did not decrease the values of the quality metric and hypervolume. It allows us to obtain more simple composite models (from both structural and computational points of view) without loss of prediction quality.

VI. Conclusion and discussions

There are a lot of different decisions that should be made to build an effective, robust, and flexible algorithm for the design of the composite models or modeling pipelines. In the paper, we propose the self-tuning approach that can be effectively used without additional meta-optimization, which is critically important for the application in AutoML-related solutions since it allows achieving a better degree of automation.

The results of the experiments confirm that the proposed evolutionary approach GPComp@Free allows achieving a stable advantage against different single-objective and multi-objective implementations. It can be seen that the composite models found with a parameter-free approach are more effective and less complex according to the various benchmarks.

The proposed approach is not limited to the problem of the composite models' design or specific AutoML framework and can be applied as a part of various automated modeling solutions (e.g. NAS, equation discovery, etc).

Code and data availability

The implemented approach is available as a part of the open-source FEDOT framework[1]. Data and scripts that were used to conduct the experiments in the paper can also be obtained from the open repository[2].

References

[1] J. Caldwell and Y. M. Ram, *Mathematical modelling: concepts and case studies*. Springer Science & Business Media, 2013, vol. 6.

[2] B. Pavlyshenko, "Using stacking approaches for machine learning models," in *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. IEEE, 2018, pp. 255–258.

[3] M. Feurer and F. Hutter, "Towards further automation in automl," in *ICML AutoML workshop*, 2018, p. 13.

[4] A. V. Kalyuzhnaya, N. O. Nikitin, A. Hvatov, M. Maslyaev, M. Yachmenkov, and A. Boukhanovsky, "Towards generative design of computationally efficient mathematical models with evolutionary learning," *Entropy*, vol. 23, no. 1, p. 28, 2021.

[5] E. LeDell and S. Poirier, "H2o automl: Scalable automatic machine learning," in *7th ICML workshop on automated machine learning*, 2020.

[6] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, "Auto-sklearn: efficient and robust automated machine learning," in *Automated Machine Learning*. Springer, Cham, 2019, pp. 113–134.

[7] Y. Heffetz, R. Vainshtein, G. Katz, and L. Rokach, "Deepline: Automl tool for pipelines generation using deep reinforcement learning and hierarchical actions filtering," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2103–2113.

[8] H. Rakotoarison, M. Schoenauer, and M. Sebag, "Automated machine learning with monte-carlo tree search," *arXiv preprint arXiv:1906.00170*, 2019.

[9] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[10] F. Qi, Z. Xia, G. Tang, H. Yang, Y. Song, G. Qian, X. An, C. Lin, G. Shi *et al.*, "A graph-based evolutionary algorithm for automated machine learning," *Software Engineering Review*, vol. 1, no. 2 (2020), pp. 10–37 686, 2020.

[11] N. Packard, M. A. Bedau, A. Channon, T. Ikegami, S. Rasmussen, K. Stanley, and T. Taylor, "Open-ended evolution and open-endedness: Editorial introduction to the open-ended evolution i special issue," 2019.

[12] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4780–4789.

[13] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[14] R. S. Olson and J. H. Moore, "Tpot: A tree-based pipeline optimization tool for automating machine learning," in *Workshop on automatic machine learning*. PMLR, 2016, pp. 66–74.

[15] F. Pfisterer, S. Coors, J. Thomas, and B. Bischl, "Multi-objective automatic machine learning with autoxgboostmc," *arXiv preprint arXiv:1908.10796*, 2019.

[16] Z. Lu, I. Whalen, V. Boddeti, Y. Dhebar, K. Deb, E. Goodman, and W. Banzhaf, "Nsga-net: neural architecture search using multi-objective genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019, pp. 419–427.

[17] M. Binder, J. Moosbauer, J. Thomas, and B. Bischl, "Multi-objective hyperparameter tuning and feature selection using filter ensembles," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 2020, pp. 471–479.

[18] A. V. Kalyuzhnaya, N. O. Nikitin, P. Vychuzhanin, A. Hvatov, and A. Boukhanovsky, "Automatic evolutionary learning of composite models with knowledge enrichment," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, 2020, pp. 43–44.

[19] A. Hvatov and M. Maslyaev, "The data-driven physical-based equations discovery using evolutionary approach," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 129–130. [Online]. Available: https://doi.org/10.1145/3377929.3389943

[20] A. M. Chirkin, A. S. Belloum, S. V. Kovalchuk, M. X. Makkes, M. A. Melnik, A. A. Visheratin, and D. A. Nasonov, "Execution time estimation for workflow scheduling," *Future generation computer systems*, vol. 75, pp. 376–387, 2017.

[21] Y. Cao, B. J. Smucker, and T. J. Robinson, "On using the hypervolume indicator to compare pareto fronts: Applications to multi-criteria optimal experimental design," *Journal of Statistical Planning and Inference*, vol. 160, pp. 60–74, 2015.

[22] S. Winkler, M. Affenzeller, and S. Wagner, "New methods for the identification of nonlinear model structures based upon genetic programming techniques," *Systems Science*, vol. 31, pp. 5–13, 2005.

[23] N. O. Nikitin, I. S. Polonskaia, P. Vychuzhanin, I. V. Barabanova, and A. V. Kalyuzhnaya, "Structural evolutionary learning for composite classification models," *Procedia Computer Science*, vol. 178, pp. 414–423, 2020.

[24] B. Evans, B. Xue, and M. Zhang, "An adaptive and near parameter-free evolutionary computation approach towards true automation in automl," in *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.

[25] F.-M. De Rainville, F.-A. Fortin, M.-A. Gardner, M. Parizeau, and C. Gagné, "Deap: enabling nimbler evolutions," *ACM SIGEVOlution*, vol. 6, no. 2, pp. 17–26, 2014.

[26] R. S. Olson, W. La Cava, P. Orzechowski, R. J. Urbanowicz, and J. H. Moore, "Pmlb: a large benchmark suite for machine learning evaluation and comparison," *BioData mining*, vol. 10, no. 1, pp. 1–13, 2017.

[27] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[28] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

---

[1] https://github.com/nccr-itmo/FEDOT

[2] https://github.com/ITMO-NSS-team/FEDOT-benchmarks