Knowledge-Based Dynamic Systems Modeling: A Case Study on Modeling River Water Quality

Namyong Park
Carnegie Mellon University
namyongp@cs.cmu.edu

MinHyeok Kim
LG Electronics
min.fourleaf@gmail.com

Nguyen Xuan Hoai
AI Academy Vietnam
nxhoai@aiacademy.edu.vn

R.I. (Bob) McKay
Australian National University
rimanucs@gmail.com

Dong-Kyun Kim
K-water Research Institute
dkkim1004@gmail.com

***Abstract*—Modeling real-world phenomena is a focus of many science and engineering efforts, from ecological modeling to financial forecasting. Building an accurate model for complex and dynamic systems improves understanding of underlying processes and leads to resource efficiency. Knowledge-driven modeling builds a model based on human expertise, yet is often suboptimal. At the opposite extreme, data-driven modeling learns a model directly from data, requiring extensive data and potentially generating overfitting. We focus on an intermediate approach, model revision, in which prior knowledge and data are combined to achieve the best of both worlds. We propose a genetic model revision framework based on tree-adjoining grammar (TAG) guided genetic programming (GP), using the TAG formalism and GP operators in an effective mechanism making data-driven revisions while incorporating prior knowledge. Our framework is designed to address the high computational cost of evolutionary modeling of complex systems. Via a case study on the challenging problem of river water quality modeling, we show that the framework efficiently learns an interpretable model, with higher modeling accuracy than existing methods.**

***Index Terms*—dynamic system modeling, prior knowledge incorporation, river water quality modeling, evolutionary algorithm**

I. Introduction

Modeling real-world phenomena is the goal of numerous science and engineering endeavors, such as ecological modeling [1], financial forecasting [2], user modeling [3], disease prediction [4], popularity estimation [5], [6], and drug discovery [7]. An accurate model of these systems can enable better understanding of underlying mechanisms and more effective use of resources. Real-world systems are typically dynamic and complex, with multiple observed and latent variables that change over time, and affect each other in complex and often nonlinear ways. As an example, consider the task of forecasting river water quality. Addressing this problem requires an understanding of processes such as plankton dynamics and hydrological mechanisms, and modeling how they influence the system dynamics as a whole.

Existing approaches can be grouped into three classes. The first is *knowledge-driven modeling*. The structure of knowledge-driven models and their parameters are determined by domain experts, based on their prior knowledge and using observational data to calibrate the model parameters. In knowledge-driven modeling, the state of dynamic systems can be modeled by differential equations. While knowledge-driven models perform reasonably well when the modelled system is simple, they take time to construct, and generally perform less well with increasing system complexity.

The second is *data-driven modeling*: learning a model purely from data, with no need for prior knowledge. Highly accurate models can often be obtained. Modeling complex systems requires plentiful data, but the high cost of measurement [8] means this is often unavailable. Sadly, learning a model from limited data often leads to overfitting. Also, some of the popular methods in this class (e.g., neural networks) generate black box models, lacking explanatory power.

The third class combines knowledge- and data-driven modeling to gain the best of both worlds. *Model calibration* is one widely used approach: the initial model structure is specified by domain knowledge, and then model parameters are optimized using data. However, model calibration updates only the model parameters, not the model structure. If this is oversimplified, the accuracy of the optimized model will be compromised, and the calibrated parameter values will be unrealistic. *Model revision* is a more interesting and effective approach: prior knowledge specifies the initial model structure and parameter values, but both are updated iteratively, guided by the prior knowledge, to obtain a better fit to the data. This approach of revising and improving existing models closely resembles the traditional scientific discovery process [9].

Genetic programming (GP) [10] provides an effective framework for model revision. GP has been successfully applied to real-world problems in various fields [11]–[13], and has the theoretical advantage that the output is interpretable, unlike blackbox models. Among GP's methods, symbolic regression (SR), which aims to discover a function that fits the training data, is the most relevant to process modeling. Standard SR is a form of data-driven modeling, as it sets no restrictions on the model structure. It thus suffers from a lack of guidance in the optimization process.

A number of newer GP methodologies, such as grammar guided GP (GGGP) [14] and tree-adjoining grammar (TAG) guided GP (TAG3P) [15], support constraining or biasing the structure of learnt models [16], [17]. We base our framework on TAG3P, which is a powerful tool for incorporating domain knowledge while exploring the complex search spaces required for modeling real-world processes.

We propose TAG3P-based genetic model revision (GMR), in which the TAG formalism and GP operators provide an effective mechanism to perform data-driven model revisions based on prior knowledge. We show how to represent dynamic processes in TAG, and how to extend the TAG3P framework to incorporate different types of prior knowledge into the optimization process. An important challenge in applying GP to complex systems is the high computational cost of the search and fitness evaluation in GP systems. Our framework achieves efficient and effective optimization by reducing redundancy and speeding up operations. In our case study, GMR allows us to accurately model water quality in a river ecosystem, a complex dynamic system with extensive geographic coverage, which has previously been much less studied than relatively simple lake ecosystems due to its far higher complexity.

In summary, our contributions are as follows:

- *Framework.* We present a GMR framework for dynamic systems modeling, which improves a knowledge-based model in a data-driven manner, guided by prior knowledge.
- *River Modeling.* This is the first work to apply model revision to modeling a river system. Previous work on river modeling used model calibration alone.
- *Effectiveness.* Our framework achieves the best forecasting accuracy in river modeling among a variety of methods.
- *Efficiency.* We present techniques to cut down the computational cost of GP systems, achieving $215\times$ speedup.

Reproducibility: Code and data are available at https://www.cs.cmu.edu/~namyongp/gmr.

II. River Water Quality Modeling

Rivers are precious freshwater resources for households, farming, and industry. Due to intensive use and increasing development, the eutrophication (over-enrichment with nutrients) of rivers has become a serious global problem. Algal blooms are one of the most problematic and widespread consequences. For improved river management, it is crucial to have an accurate model of the water quality.

$$
\begin{aligned}
dB_{Phy}/dt &= B_{Phy}\cdot(\mu_{Phy}-\gamma_{Phy})-B_{Zoo}\cdot\varphi \quad (1)\\
\mu_{Phy} &= C_{UA}\cdot f(V_{lgt})\cdot g(V_n,V_p,V_{si})\cdot h(V_{tmp})\\
\gamma_{Phy} &= C_{BRA}\\
\varphi &= C_{MFR}\cdot\lambda_{Phy}\\
\lambda_{Phy} &= (B_{Phy}-C_{Fmin})/(C_{FS}+B_{Phy}-C_{Fmin})\\
f(V_{lgt}) &= (V_{lgt}/C_{BL})\cdot e^{1-(V_{lgt}/C_{BL})}\\
g(V_n,V_p,V_{si}) &= \min\left(V_n/(C_N+V_n),V_p/(C_P+V_p),V_{si}/(C_{SI}+V_{si})\right)\\
h(V_{tmp}) &= \max(e^{-C_{PT}(V_{tmp}-C_{BTP1})^2},e^{-C_{PT}(V_{tmp}-C_{BTP2})^2})\\
dB_{Zoo}/dt &= B_{Zoo}\cdot(\mu_{Zoo}-\gamma_{Zoo}-\delta_{Zoo}) \quad (2)\\
\mu_{Zoo} &= C_{UZ}\cdot\lambda_{Phy}\\
\gamma_{Zoo} &= C_{BRZ}+C_{BMT}\cdot\varphi\\
\delta_{Zoo} &= C_{DZ}
\end{aligned}
$$

River water quality modeling aims to predict phytoplankton biomass, a proxy for eutrophication. Based on the knowledge of a freshwater ecologist, we designed the biological processes (1) and (2), modeling the change of phytoplankton biomass over time by capturing the interplay between phytoplankton (B_{Phy}) and zooplankton (B_{Zoo}).

The phytoplankton dynamics model (dB_{Phy}/dt) incorporates the photosynthetic productivity (μ_{Phy}), metabolic degradation (γ_{Phy}), and grazing pressure of zooplankton (φ). The photosynthetic productivity depends on multiplicative influences from variables, such as light intensity (V_{lgt}), nutrient (nitrogen, phosphorus, and silica) concentrations (V_n, V_p, V_{si}), and water temperature (V_{tmp}). These functions build on earlier studies on modeling algal dynamics including [18], [19]. Further, considering the effect of summer cyanobacteria and winter diatom blooms, we extend the process with two additional parameters reflecting optimal temperatures (C_{BTP1}, C_{BTP2}). The zooplankton dynamics model (dB_{Zoo}/dt), adapted from [19], incorporates the growth (μ_{Zoo}), respiration (γ_{Zoo}), and death (δ_{Zoo}) rates of zooplankton.

The parameters of these biological processes fall into two classes: *constant parameters* (starting with C) have constant values representing physiological rates (e.g., growth or feeding rate), while *variable parameters* (starting with V) correspond to external conditions and forces, changing over time. In evaluating (1) and (2), variable parameters are imported from the observed data at the evaluation time t. More details on this are given in [20]. The goal of model revision for our task can be summarised as:

> Given biological processes (1) and (2), make relevant changes to the structure and constant parameter values, guided by prior knowledge, such that the estimated phytoplankton biomass (B_{Phy}) is close to the observed values.

III. Methods

In this section, we present our genetic model revision (GMR) framework. There are three major challenges in applying model revision to the modeling of dynamical systems.

1) **Representation of dynamic processes.** How can we represent dynamic processes for successful model revision?
2) **Knowledge-based model revision.** How can we effectively revise a model while incorporating prior knowledge?
3) **Efficiency and effectiveness.** How can we perform efficient and effective model revision for complex dynamic systems?

Our GMR framework addresses these challenges as follows.

1) **Using TAG for representing dynamic processes** allows us to succinctly express dynamic processes and their revisions.
2) **Making revision via TAG-guided GP and expressing prior knowledge using the TAG formalism** leads to an accurate model based on prior knowledge.
3) **Reducing redundancy and speeding up operations, together with local search**, enable fast and effective model revision.

We describe these ideas in detail in Sections III-A to III-C.

A. Representing Dynamic Processes Using TAG

1) Preliminaries on TAG: The TAG grammar formalism [21] generates a tree by composing two types of *elementary trees*, α-trees (or *initial trees*) and β-trees (or *auxiliary trees*); β-trees must have a special frontier node (the foot node) with the same label as the root node (marked with a $*$), as illustrated in Figures 1 and 2.

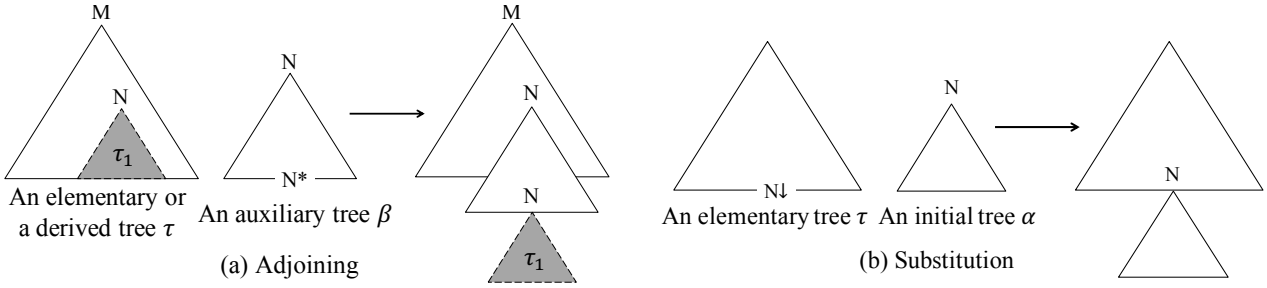

Fig. 1: Illustrations of tree composition operations used by tree-adjoining grammar (TAG): (a) adjoining and (b) substitution.

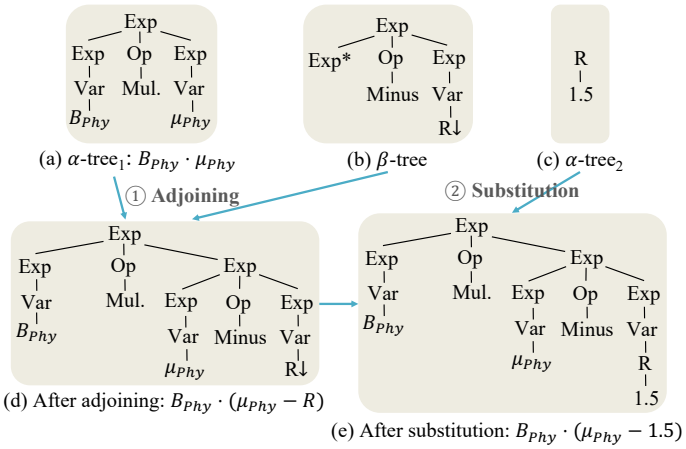

Fig. 2: (a)–(c): Example α- and β- trees representing a dynamic process and potential revisions. (d), (e): Resulting trees after adjoining and substitution (see text for details).

Adjoining and *substitution* are the two composition operations TAG uses to construct a *derived* tree (Figure 1). Given a tree τ (which can be either an elementary or a derived tree), and an auxiliary tree β, adjoining builds a new derived tree. Assume that the root of β is labeled as N, and that the tree τ has an interior node n labeled as N. The steps for adjoining β into τ are as follows (see Figure 1(a) for an illustration):

1) The sub-tree τ_1 rooted at node n is disconnected from τ;
2) The tree β is attached at the place where the node n was;
3) τ_1 is attached to the foot node (marked with $*$) of the tree β.

Substitution creates a derived tree from an elementary tree τ and a tree α which is (derived from) an initial tree. As in Figure 1(b), substitution selects a non-terminal on the frontier of τ (marked as $\downarrow$) matching the root of α, and replaces it with α. A tree derived from an initial tree and lacking frontier non-terminals is a *completed* tree.

An in-depth description of TAG appears in [15], [21].

2) TAG-Based Dynamic Process Representation: Consider this equation, a simplified form of (1), to see how TAG can represent dynamic processes and potential revisions:

$$dB_{Phy}/dt = B_{Phy} \cdot \mu_{Phy} \tag{3}$$

(3) can be represented by an α-tree shown in Figure 2(a) where "Mul." denotes multiplication. Figure 2(b) shows a β-tree representing one potential extension where an expression (denoted by "Exp") is extended by deducting a random variable (denoted by "R") from it. Then adjoining the β-tree in Figure 2(b) into the rightmost "Exp" node of the α-tree in Figure 2(a) yields the tree shown in Figure 2(d), which corresponds to $B_{Phy} \cdot (\mu_{Phy} - R)$. Another α-tree in Figure 2(c) encodes a potential value for variable R. By substituting it into the frontier node R (marked with $\downarrow$) shown in Figure 2(d), we obtain a revised process $B_{Phy} \cdot (\mu_{Phy} - 1.5)$.

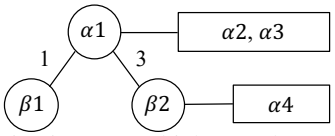

Fig. 3: TAG derivation tree which encodes a revised differential equation. Nodes $\beta 1$ and $\beta 2$ are β-trees that are adjoined into the specified address (the number on the link) of the root. Rectangles are α-trees that are substituted into the linked tree.

A completed tree (e.g., Figure 2(e)) corresponds to a revised process. The history of adjunctions and substitutions is encoded as an object tree called the *derivation tree*. In other words, we encode successive model revisions and the revised process as a derivation tree in TAG. Specifically, the TAG derivation tree used in GMR (Figure 3) is a tree of objects where links between objects indicate adjunction at the specified address, and each node has α-trees to be substituted into the open nodes in the elementary tree labeled by the node.

B. Genetic Model Revision Framework

1) Framework Overview: Figure 4 shows an overview of the genetic model revision (GMR) framework. It builds upon tree-adjoining grammar guided genetic programming (TAG3P) [15]. TAG3P is a population-based optimization algorithm that evolves a population of random (often unfit) initial programs (which are differential equations in our setting) into fitter ones for a given task, over multiple generations. TAG3P differs from standard GP in that it is a grammar-guided GP system where search space exploration is guided by TAG. In Figure 4, the loop marked in red corresponds to one generation. At each generation, genetic model revision is performed on the current population by applying genetic operators to produce a revised population of potentially fitter individuals. Note that two types of prior knowledge (shown in rounded boxes) given to the framework govern the entire search process, from population initialization to iterative model revision, until a final model is obtained.

α- and β-trees. In GMR, one α-tree represents a manually-designed minimal process as in Figure 2(a). We define β-trees to represent potential revisions as in Figure 2(b).

Genetic Operators. Genetic operators revise individuals to obtain others. We introduce two representative operators.

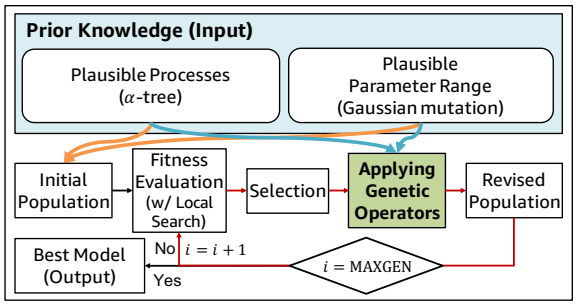

Fig. 4: An overview of the model revision framework. Prior knowledge guides the entire model revision process.

(i) Crossover. Two individuals are chosen by a selection mechanism. Then their subtrees, which are compatible with each other, are randomly selected and swapped.

(ii) Subtree Mutation. A subtree x is randomly selected, and is replaced with a new subtree, which is of similar size to x, and compatible with x (to produce a valid individual).

A detailed review of TAG3P can be found in [15].

2) Incorporating Prior Knowledge: While optimizing both the structure and parameters of dynamic processes is more effective than optimizing parameters alone, it involves exploring a huge search space. For effective optimization, and to learn an accurate model with physically plausible parameters, our framework incorporates two types of prior knowledge.

Prior Knowledge of Plausible Processes. In an effort to explain real-world phenomena, experts develop models based on domain knowledge and experience. We harness this prior knowledge of dynamic processes, specifically what variables are known to be involved and how they interact with each other. For example, the temporal dynamics of phytoplankton (dB_{Phy}/dt) in (1) is expressed as a function of zooplankton biomass (B_{Zoo}) (and other related parameters) since zooplankton grazing pressure is known to be a major regulator of phytoplankton in a river ecosystem. In our framework, this type of knowledge, expressed as a differential equation, is encoded as an α-tree as shown in Figure 2(a). These input processes act as a significant knowledge transfer at the starting point of model revision. With classic GP systems, by contrast, we start from random models.

Prior Knowledge about Model Parameters. From previous research and experience on dynamical systems, domain experts often have the information on the plausible distributions of the model parameters. Even if we obtain a highly accurate model, if its parameters are not within a realistic range, that model is not considered a good representation of underlying processes. In GMR, the domain knowledge of model parameters is summarized as the expected value and allowed range of parameter values. For effective search, ranges need to be chosen to cover most practically feasible values. We assume that naturally occurring values follow a truncated Gaussian distribution centered around the expected value.

To optimize model parameters, we apply a genetic operator, *Gaussian mutation*, which locates all constant parameters in an individual, and updates them to new values sampled from their associated Gaussian distribution. In the beginning, parameters are set to the expected value. When Gaussian mutation is applied to a parameter, a new value is generated, and it becomes the new mean of the Gaussian distribution for that parameter. If the sampled value lies outside of the given range, the boundary value is used instead.

3) Applying GMR *to Real-World Problems:*

River Water Quality Modeling. To apply GMR to river modeling, we represent the two differential equations in (1) and (2) as a single α-tree. This can be done by first representing each equation in separate trees, and then combining them into one α-tree under a new, common root node. Then this combined α-tree is evolved in the same manner as in simpler cases, and decomposed into multiple equations when performing fitness evaluation.

Each expression in (1) and (2) can be revised by adjoining and substitution operations as discussed in Section III-A. Specifically, we generate β-trees for revisions, corresponding to the combinations of $+, -, \times, \div, \log, \exp$ operators with the following variables chosen based on domain knowledge: $V_{cd}, V_{ph}, V_{alk}, V_{tmp}, V_{sd}, V_{do}, R$.

Application to Other Problems. GMR provides general mechanisms to represent and revise process equations guided by prior knowledge, so is readily applicable to diverse problem settings. The only problem-dependent component is representing domain-specific knowledge as discussed in Section III-B2; this itself is a general technique applicable to various problems.

C. Improving the Efficiency and Effectiveness

For efficient and effective optimization, we apply two orthogonal speedup techniques, together with local search.

Runtime Compilation. A tree representing temporal processes needs to be evaluated multiple times over some time period, and each such evaluation can be done by recursively evaluating subtrees, providing the model parameter values appropriately at each step. Instead, we use runtime compilation, which enables more efficient evaluation than repeated tree parsing: a program encoded in the tree is converted into the corresponding source code, compiled at runtime, and dynamically loaded to be used for fitness evaluation.

Tree Caching. We cache the results of tree evaluation, and reuse them when we need to reevaluate the same trees. By using additional memory to store evaluation results, we avoid redundant computations. Note that the effectiveness of caching depends on the hit rate. GMR improves the hit rate by algebraically simplifying the trees before they are evaluated.

Local Search. Local search aims to improve the search effectiveness by making incremental, local revisions to an individual. Specifically, we perform stochastic hill-climbing local search, where a tree resulting from crossover and mutation goes through a series of local search, applying node insertion and deletion with equal probability, and adopting the change if it improves the fitness.

IV. Experiments

In evaluating the GMR framework, we address the following questions in our river modeling case study:

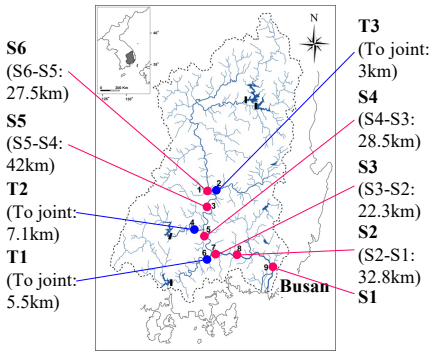

Fig. 5: The Nakdong River basin in South Korea.

Q1. Prediction Accuracy: How accurately does the GMR framework forecast water quality?

Q2. Efficiency: How much do the speedup techniques improve efficiency?

A. Dataset and Modeling Task Description

The Nakdong River catchment in South Korea is one of the largest water-quality monitoring networks supporting long-term ecological research. Our dataset is a collection of measurements for 13 years (1996–2008) at nine stations located in the catchment; (Figure 5): six (S1–S6) are sited on the main channel, while three (T1–T3) are on major tributaries. The dataset contains five types of variables: geographical (e.g., catchment area), hydrological (e.g., flow rate), meteorological (e.g., irradiance), physicochemical (e.g., water temperature), and biological (e.g., chlorophyll a). Most were measured daily, except for nutrient concentrations and chlorophyll a, which were measured weekly (at S1) or bi-weekly (at others). For those variables measured with a longer interval, we performed linear interpolation to obtain values between measurements. Given these measurements, our goal is to forecast the algal biomass at the lowest station (S1) due to its geographical importance. More details of how we model the river systems (e.g., the hydrological process) are provided in [20].

B. Comparators

For evaluation, we use representative comparators in the three classes of methods for modeling dynamic systems.

1) Knowledge-Driven Modeling: (a) MANUAL. This is the biological process in (1) and (2), designed by domain experts.

2) Data-Driven Modeling: (a) RNN (Recurrent Neural Network). We use long short-term memory (LSTM), predicting the phytoplankton biomass at S1 at the next time step from observed variables at the current time. We experiment with two variants: RNN-S1 uses variables observed at station S1 alone; RNN-ALL uses variables observed at all nine stations. (b) ARIMAX is widely used for time series forecasting. As with RNN, we consider two variants differing in variables used (denoted as ARIMAX-S1 and ARIMAX-ALL).

3) Model Calibration: Given the biological process in (1) and (2), model calibration methods optimize the values of process parameters without revising the form of equations. We use the following widely-used approaches: (a) GA (genetic algorithm) (b) MC (Monte Carlo) (c) LHS (Latin hypercube sampling) (d) MLE (maximum likelihood estimation) (e) MCMC (Markov chain Monte Carlo) (f) SA (simulated annealing) (g) SCE-UA (shuffled complex evolution [22]).

We provide experimental settings of all methods in [20].

TABLE I: GMR achieves the best forecasting accuracy (28% and 27% more accurate than the second best method in terms of RMSE and MAE, respectively), among a variety of methods. Best results are underlined.

Method Class	Method	Training (96–05)		Test (06–08)	
		RMSE	MAE	RMSE	MAE
Knowledge-driven	MANUAL	2.79e+9	2.15e+8	2.23e+6	7.93e+5
Data-driven	RNN-S1	19.605	11.533	23.057	16.833
	RNN-ALL	21.326	13.166	23.009	16.276
	ARIMAX-S1	12.710	5.012	37.770	25.504
	ARIMAX-ALL	12.365	5.775	260.468	71.471
Model calibration	GA	26.329	14.693	20.308	13.291
	MC	26.581	14.426	19.259	12.675
	LHS	26.812	14.536	18.287	12.064
	MLE	26.033	14.408	19.513	13.242
	MCMC	26.514	14.554	18.661	12.480
	SA	26.463	14.585	18.740	12.532
	SCE-UA	25.995	14.353	19.876	13.275
Model revision	**GMR**	**19.635**	**12.048**	**13.203**	**8.762**

C. Q1. Prediction Accuracy

We split the data into two periods, 1996–2005 for training and 2006–2008 for testing, and report the forecasting accuracy in Table I, in terms of the best RMSE and MAE where best models denote those with the smallest test RMSE.

MANUAL performed significantly worse than other approaches, although it is designed with domain knowledge of the biological process and a careful selection of parameter values. Model calibration approaches, such as GA, LHS, and SA, obtained a much better result than MANUAL, indicating the benefits of tuning model parameters. However, model calibration methods were outperformed by model revision as they can update only the model parameters, but not the model structure. In both criteria, GMR achieved the best testing performance, with 28% and 27% smaller RMSE and MAE, respectively, than the second best method LHS.

For data driven models, we used two types of input variables. Both variants of RNN and ARIMAX (denoted by S1 and ALL) performed worse than model calibration and model revision methods. While RNN achieved much smaller training RMSE (~6.7) than others as training continued, it suffered from overfitting and its test RMSE increased to ~44.0. Note that for RNN and ARIMAX, using additional input variables observed at stations other than S1 did not improve the performance. In fact, ARIMAX-ALL performed worse than ARIMAX-S1. As measuring stations are located over a wide area (see Figure 5), using measurements from distant stations simply as additional input features was not helpful for predicting at S1. Also, as is typically the case with ecological

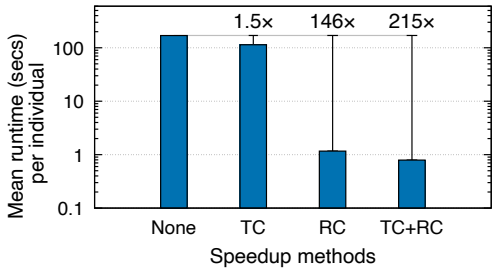

Fig. 6: Mean runtime (seconds) per individual by speedup techniques. TC: Tree Caching, RC: Runtime Compilation. Applying all leads to 215× speedup.

data, the dataset is not large enough (2,435 data points for training) for learning complex processes in a purely data-driven manner. In contrast, by using prior knowledge, GMR can learn an effective model from a small dataset.

D. Q2. Efficiency

We evaluate the effectiveness of two orthogonal methods for speedup, i.e., tree caching (TC), and runtime compilation (RC). Figure 6 shows the mean runtime (seconds) per individual obtained with different speedup methods, which indicates that these techniques effectively reduce computational costs, achieving 215× speedup when the two methods are applied together, compared to when no speedup technique was used.

V. Related Work

Scientific Discovery from Data. Interest in computational methods for scientific discovery from data is growing. Combining prior knowledge and data science [8] is an emerging paradigm showing promising results. One direction is to guide the learning algorithm, e.g., via theory-guided constrained optimization [8], [23]. GMR also falls in this category, using TAG formalism for knowledge-guided navigation of the search space. Model calibration [8] is another such approach: data are used to optimize the parameters of a knowledge-based model. GMR outperforms various model calibration approaches.

Modeling River Water Quality. Neural networks (NN) and genetic algorithms (GA) have recently been used for river modeling. NN-based methods performed data-driven modeling using multilayer perceptron [24] and recurrent neural networks [25]. By performing model calibration, GA-based methods [1], [26] successfully improved the accuracy of a knowledge-based river model. However, previous approaches fail to jointly optimize the structure and parameters of a model. Our work provides a more effective way for modeling complex river processes by using knowledge-based model revision.

VI. Conclusion

Model revision is effective in modeling real-world phenomena: domain expertise and data are used to model complex dynamical systems. Our framework, based on TAG3P, revises models guided by prior knowledge. The case study of river modeling shows its effectiveness. In future work, we will explore new mechanisms to incorporate domain knowledge (new search operators and language biases), and apply it to other domains, such as financial forecasting.

References

[1] D. Kim, B. McKay, H. Shin, Y. Lee, and X. H. Nguyen, "Ecological application of evolutionary computation: Improving water quality forecasts for the nakdong river, korea," in *CEC*, 2010, pp. 1–8.

[2] C.-J. Lu, T.-S. Lee, and C.-C. Chiu, "Financial time series forecasting using independent component analysis and support vector regression," *Decision Support Systems*, vol. 47, no. 2, pp. 115–125, 2009.

[3] G. I. Webb, M. J. Pazzani, and D. Billsus, "Machine learning for user modeling," *User Model. User-Adapt. Interact.*, vol. 11, no. 1-2, 2001.

[4] N. Park, E. Kang, M. Park, H. Lee, H.-G. Kang, H.-J. Yoon, and U. Kang, "Predicting acute kidney injury in cancer patients using heterogeneous and irregular data," *PLOS ONE*, vol. 13, 07 2018.

[5] N. Park, A. Kan, X. L. Dong, T. Zhao, and C. Faloutsos, "Estimating node importance in knowledge graphs using graph neural networks," in *KDD*. ACM, 2019, pp. 596–606.

[6] ——, "Multiimport: Inferring node importance in a knowledge graph from multiple input signals," in *KDD*. ACM, 2020, pp. 503–512.

[7] O. Becker, S. Shacham, Y. Marantz, and S. Noiman, "Modeling the 3d structure of gpcrs: advances and application to drug discovery." *Current opinion in drug discovery & development*, vol. 6, no. 3, 2003.

[8] A. Karpatne, G. Atluri, J. H. Faghmous, M. S. Steinbach, A. Banerjee, A. R. Ganguly, S. Shekhar, N. F. Samatova, and V. Kumar, "Theory-guided data science: A new paradigm for scientific discovery from data," *TKDE*, vol. 29, no. 10, pp. 2318–2331, 2017.

[9] S. Džeroski, P. Langley, and L. Todorovski, "Computational discovery of scientific knowledge." Springer, 2007, pp. 1–14.

[10] J. R. Koza, *Genetic programming - on the programming of computers by means of natural selection*, ser. Complex adaptive systems. MIT Press, 1993.

[11] R. Vyas, P. Goel, and S. S. Tambe, "Genetic programming applications in chemical sciences and engineering," in *Handbook of Genetic Programming Applications*, 2015, pp. 99–140.

[12] M. Affenzeller, S. M. Winkler, S. Wagner, and A. Beham, *Genetic Algorithms and Genetic Programming - Modern Concepts and Practical Applications*. CRC Press, 2009.

[13] N. P. Dao, X. H. Nguyen, R. I. B. McKay, C. Siriteanu, Q. U. Nguyen, and N. Park, "Evolving the best known approximation to the Q function," in *GECCO*, 2012, pp. 807–814.

[14] P. A. Whigham, "Grammatical bias for evolutionary learning," Ph.D. dissertation, New South Wales, Australia, Australia, 1996, aAI0597571.

[15] X. H. Nguyen, "A flexible representation for genetic programming: lessons from natural language processing," Ph.D. dissertation, University of New South Wales, Australian Defence Force Academy, 2004.

[16] D. J. Montana, "Strongly typed genetic programming," *Evolutionary Computation*, vol. 3, no. 2, pp. 199–230, 1995.

[17] M. O'Neill and C. Ryan, "Grammatical evolution," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 4, pp. 349–358, 2001.

[18] K.-J. Cho and J.-K. Shin, "Growth and nutrient kinetics of some algal species iso-lated from the naktong river," *Algae*, vol. 13, no. 2, 1998.

[19] P. Hongping and M. Jianyi, "Study on the algal dynamic model for west lake, hangzhou," *Ecological Modelling*, vol. 148, no. 1, pp. 67–77, 2002.

[20] N. Park, M. Kim, N. X. Hoai, R. I. B. McKay, and D. Kim, "Knowledge-guided dynamic systems modeling: A case study on modeling river water quality," *CoRR*, vol. abs/2103.00792, 2021.

[21] A. K. Joshi and Y. Schabes, "Tree-adjoining grammars," in *Handbook of formal languages*. Springer, 1997, pp. 69–123.

[22] Q. Duan, S. Sorooshian, and V. K. Gupta, "Optimal use of the sce-ua global optimization method for calibrating watershed models," *Journal of hydrology*, vol. 158, no. 3-4, pp. 265–284, 1994.

[23] A. Karpatne, W. Watkins, J. S. Read, and V. Kumar, "Physics-guided neural networks (PGNN): an application in lake temperature modeling," *CoRR*, vol. abs/1710.11431, 2017.

[24] K. P. Singh, A. Basant, A. Malik, and G. Jain, "Artificial neural network modeling of the river water quality–a case study," *Ecological Modelling*, vol. 220, no. 6, pp. 888–895, 2009.

[25] K. Kim, D.-K. Kim, J. Noh, and M. Kim, "Stable forecasting of environmental time series via long short term memory recurrent neural network," *IEEE Access*, vol. 6, pp. 75 216–75 228, 2018.

[26] M. Kim, N. Park, R. B. McKay, H. Shin, Y.-G. Lee, K.-S. Jeong, and D.-K. Kim, "Improvement of complex and refractory ecological models: Riverine water quality modelling using evolutionary computation," *Ecological Modelling*, vol. 291, pp. 205–217, 2014.