Queueing Systems
https://doi.org/10.1007/s11134-022-09767-6

A data-driven approach to deriving closed-form approximations for queueing problems using genetic algorithms

Rob van der Mei[1] · Sandjai Bhulai[2]

Received: 31 January 2022 / Accepted: 28 February 2022

1 Introduction

The most powerful results in queueing theory are *closed-form* expressions for key performance metrics (e.g., waiting times, sojourn times, number of customers), because they *explicitly* show how the performance depends on the system parameters. Unfortunately, most queueing models prohibit the derivation of exact closed-form expressions. Faced by this, it is common practice to use numerical techniques (e.g., numerical algorithms, approximation methods, and simulations) or to develop exact expressions in asymptotic regimes (e.g., heavy-load, heavy-tails). Despite the fact that tremendous progress has been made in the development of efficient numerical techniques, by definition they provide limited insight into how the performance metrics depend on the system parameters. In this paper, we propose a new view on attacking queueing models by presenting a *data-driven approach* to develop *closed-form approximations* for key performance metrics based on the use of genetic algorithms (GAs), using the concept of symbolic regression (SR).

SR is a regression method that searches the space of algebraic expressions to find one that 'best' fits a given data set, both in terms of accuracy and simplicity. Within the SR framework, an *individual* represents a specific formula, which is expressed as a *tree*. Like any other GA, SR forms an initial population of individuals. Next, the GA iteratively generates a new *offspring* of individuals (a new generation) by *crossing* and/or *mutating* already existing individuals. The idea is that over time, the population's accuracy improves due to evolving the well-performing individuals (i.e., survival of the fittest). See Fig. 1 for an illustration.

✉ Rob van der Mei
mei@cwi.nl

Sandjai Bhulai
s.bhulai@vu.nl

1 CWI, Stochastics group, Amsterdam, The Netherlands

2 Department of Mathematics, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

Published online: 28 March 2022

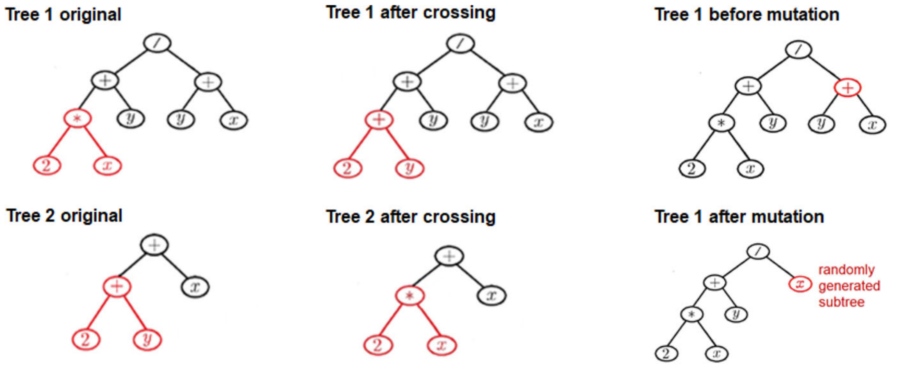

Fig. 1 Illustration of SR: The formulas $\frac{2x+y}{y+x}$ and $(2+y)+x$ expressed as a tree (left), the trees after crossing (middle), and tree 1 *before* and *after* mutation (right)

2 Methodology

A common way to derive closed-form approximations for queueing models is to (1) make simplifying assumptions, (2) derive closed-form approximations based on these assumptions, and (3) do a numerical analysis (and/or derive error bounds) on the accuracy of the approximation. In contrast, we propose the following stepwise approach:
Step 1 (Create data set): Select a (finite) number of model instances, with a good spread over the parameter space, and evaluate the corresponding performance metrics (e.g., by a numerical algorithm or simulations). This results in what is called the *data set*, which is the basis for the next steps.
Step 2 (Choose parameters for the genetic algorithm): Specify the set of operators that can be used to create expressions via symbolic regression; for example, binary operators like $+$, $-$, $*$, $/$, and unary operators such as $\cdot^{\alpha}$, $exp(\cdot)$ and $log(\cdot)$. Also, specify the set of *features* (i.e., the potential symbols in the expressions to be generated), and limits to the tree structure (e.g., the maximal tree depth or the size of the offspring).
Step 3 (Choose the fitness function): This function quantifies approximation errors (e.g., MSE, (w)RMSE, (w)MAPE), and provides a means to assess the accuracy of a given symbolic expression, where the data set serves as the benchmark.
Step 4 (Creation of symbolic expressions via GAs): Starting with an initial set of expressions (represented in trees), the GA creates and evaluates symbolic expressions iteratively via *mutation* and *cross-over* operators, creating new expressions out of existing ones. These new expressions are then evaluated according to the fitness function (defined in Step 3), until a convergence criterion is met.

Remark 1 Consider a GI/G/1 queue, with general interarrival times A and service times B and load $\rho := \lambda/\mu < 1$, with $\lambda = 1/E[A]$ and $\mu = 1/E[B]$, and where we want to derive a two-moment approximation for the mean delay $E[W]$. Then a natural choice of the feature set is $\{E[A], E[A^2], E[B], E[B^2]\}$.

Remark 2 The proposed SR-based approach is still in its infancy, but initial results are promising. In [4], it was shown that the approach can be used to obtain symbolic expressions for MDP value functions, and [2] gives an example where the approach is also useful to approximate optimal control policies in MDPs.

Remark 3 Conceptually, one could argue that machine learning methods could achieve the same result. Simple models such as linear regression could add the system parameters as feature. However, such a method retains the linear structure that one starts with. Neural networks, however, usually lead to a lot of training weights, which require a lot of data. Our methodology does not suffer from these drawbacks. A similar remark holds for Gaussian process regression. The advantage is that uncertainty estimates are built-in, but come at the cost of interpretability of the results and the computation tractability.

3 Discussion

This approach is promising, but raises many research questions, outlined below.

First, which set of model instances do we need to evaluate to create the data set? It seems good to have some level of 'orthogonality' on the data set, such that the selected instances have a good spread over the parameter space (cf. [5] for sampling methods).

Second, the evaluation methods used to create the data, by definition, have some level of inaccuracy. There is stochasticity in the data generation, errors in the fitting, and statistical errors from the finiteness of the data set. How to dissect these errors to understand how they impact the closed-form expressions generated?

Third, many choices have to be made to determine the degree of freedom given to the algorithm. What are good sets of operators, features, and initial expressions? What is the maximum size of the SR-trees (depth, number of branches per node)?

Fourth, many queueing models prohibit exact expressions, but allow for closed-form asymptotics, e.g., in heavy-traffic (cf., e.g., [3, 6–9]). These results may be used to speed up the convergence of the SR-algorithm, e.g., by adding symbols to the GA.

Lastly, for polling models, pseudo-conservation laws (PCLs) give closed-form expressions for a *weighted* sum of the mean waiting times [1]. How to include these?

References

1. Boxma, O.J., Groenendijk, W.P.: Pseudo-conservation laws in cyclic-service systems. J. Appl. Probab. **24**, 949–964 (1987)
2. Hristov, A., Bhulai, S., van der Mei R.D.: Deriving explicit control policies for Markov Decision Processes using Symbolic Regression. In: Proceedings of ValueTools (Tsukuba, Japan) (2020)
3. Kingman, J.F.C.: The single server queue in heavy traffic. Math. Proc. Camb. Philos. Soc. **57**, 902–904 (1961)
4. Onderwater, M., Bhulai, S., van der Mei, R.D.: Value function discovery in MDPs with evolutionary algorithms. IEEE Trans. Syst. Man Cybern. **46**, 1190–1201 (2015)
5. Taherdoost, H.: Sampling methods in research methodology; how to choose a sampling technique for research. Int. J. Res. Manag. **5**, 18–27 (2016)
6. van der Mei, R.D.: Polling systems in heavy traffic: higher moments of the delay. Queueing Syst. **31**, 365–394 (1999)
7. van der Mei, R.D.: Distribution of the delay in polling systems in heavy traffic. Perform Eval. **38**, 133–148 (1999)
8. van der Mei, R.D.: Polling systems with switch-over times under heavy load: moments of the delay. Queueing Syst. **36**, 381–404 (2000)
9. Verloop, I.M., Nunez-Queija, R., Ayesta, U.: Heavy-traffic analysis of a multiple-phase network with discriminatory processor sharing. Oper. Res. **59**, 648–660 (2011)