# Automatic Exploration of the Property Space of Reservoirs

Mika Ito
Ochanomizu University
Tokyo, Japan
g1720505@is.ocha.ac.jp

Leo Cazenille
Ochanomizu University
Tokyo, Japan
leo.cazenille@gmail.com

Nathanael Aubert-Kato
Ochanomizu University
Tokyo, Japan
naubertkato@is.ocha.ac.jp

## ABSTRACT

Reservoir Computing is an efficient implementation of a recurrent neural network for dealing with temporal/sequential data processing. However, in terms of matching reservoir dynamics to tasks, the precise balance of properties (kernel rank, generalization rank, memory capacity, size) of reservoirs will vary. To provide guidance for the generation of reservoirs, we use NSGA-II and MAP-Elites to explore the balance between those properties. We further provide three generation strategies for reservoirs: (a) the optimization of the properties of the random generator, (b) the direct optimization of general purpose reservoir, and (c) a combination of both approaches. We show that each approach can generate reservoirs with different ranges of characteristics, making them thus appropriate for different categories of tasks.

## CCS CONCEPTS

• **Computing methodologies** → **Machine learning approaches**;

## KEYWORDS

Reservoir Computing, Quality-Diversity, Multi-objective optimization, NSGA-II, MAP-Elites

**ACM Reference Format:**

Mika Ito, Leo Cazenille, and Nathanael Aubert-Kato. 2021. Automatic Exploration of the Property Space of Reservoirs. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459426

## 1 INTRODUCTION

Reservoir Computing (RC) is a computing paradigm initially proposed to take advantage of the complex dynamics of recursive neural networks (RNNs) while reducing the cost of weight training. [8, 9] The main characteristic of that approach is that input weights and the weights of the recurrent connections within the reservoir are not trained, whereas the readout weights are trained with a simple learning algorithm such as linear regression. This simple and fast training process makes it possible to drastically reduce the computational cost of learning compared with standard RNNs. The performance of RC depends on many factors, such as the structure and size of the reservoir, and the readout learning algorithm, which are very difficult to set up optimally. In this paper, we aim to find in an automated way a large range of reservoirs with specific characteristics, providing the user a range of choice appropriate to their goal task. We take inspiration from previous work by Dale *et al.*[2] who used novelty search to explore the characteristic of specific reservoir substrates. However, we seek to improve the performance of randomly generated reservoirs and thus focus on multi-objective approaches, comparing the results of two algorithms: NSGA-II [3] and MAP-Elites [7].

In this study, three approaches were used as design guidelines to realize a high performance reservoir. The first approach (approach 1) is to generate a good reservoir in advance by optimizing the input weight matrix $\mathbf{W}^{\mathbf{in}}$ and the reservoir connection weight matrix $\mathbf{W}$, which are randomly generated according to the following parameters: (1) the size of the reservoir, (2) the leaking rate, and (3) the spectral radius. The second one (approach 2) is direct optimization of $\mathbf{W}^{\mathbf{in}}$ and $\mathbf{W}$. In this case, we use empirically good values for the leaking rate and the spectral radius. Finally, in the third approach (approach 3), we optimize the size of the reservoir, the leaking rate, the spectral radius, $\mathbf{W}^{\mathbf{in}}$, and $\mathbf{W}$ simultaneously. Approach 2 and approach 3 are highly dimensional problems compared to approach 1.

## 2 METHODS

We use Echo State Network (ESN) [6] as the substrate for the reservoir. ESNs provide supervised learning principles for RNNs and are well established state-of-the-art reservoir substrates. The typical update equations are

$$\tilde{\mathbf{x}}(n) = \tanh(\mathbf{W}^{\mathbf{in}}[1; \mathbf{u}(n)] + \mathbf{W}\mathbf{x}(n-1)), \tag{1}$$

$$\mathbf{x}(n) = (1-\alpha)\mathbf{x}(n-1) + \alpha\tilde{\mathbf{x}}(n), \tag{2}$$

where $\mathbf{x}(n)$ is a vector of reservoir neuron activation and $\tilde{\mathbf{x}}(n)$ is its update, all at time step $n$, $\mathbf{W}^{\mathbf{in}}$ and $\mathbf{W}$ are the input and recurrent weight matrices respectively, and $\alpha$ is the leaking rate.

The linear readout layer is defined as

$$\mathbf{y}(n) = \mathbf{W}^{\mathbf{out}}[1; \mathbf{u}(n); \mathbf{x}(n)], \tag{3}$$

where $\mathbf{y}(n)$ is network output and $\mathbf{W}^{\mathbf{out}}$ is the output weight matrix. Training of the readout is typically carried out in a supervised way using linear regression with a teacher signal.

To characterize the performance of reservoirs, we use four metrics: size, memory capacity (MC) [4], kernel rank (KR), and generalization rank (GR) [5]. Size is the number of units in the reservoir. Smaller-sized reservoirs are less sensitive to the reality gap. MC is maximum delay length still allowing the recovery of an input, which can be seen as dynamic short-term memory. KR is a measure of the reservoir's ability to separate distinct input patterns. As many practical tasks are linearly inseparable, reservoirs typically require some nonlinear transformation of the input. GR is a measure of


*GECCO '21, July 10–14, 2021, Lille, France*

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3459426

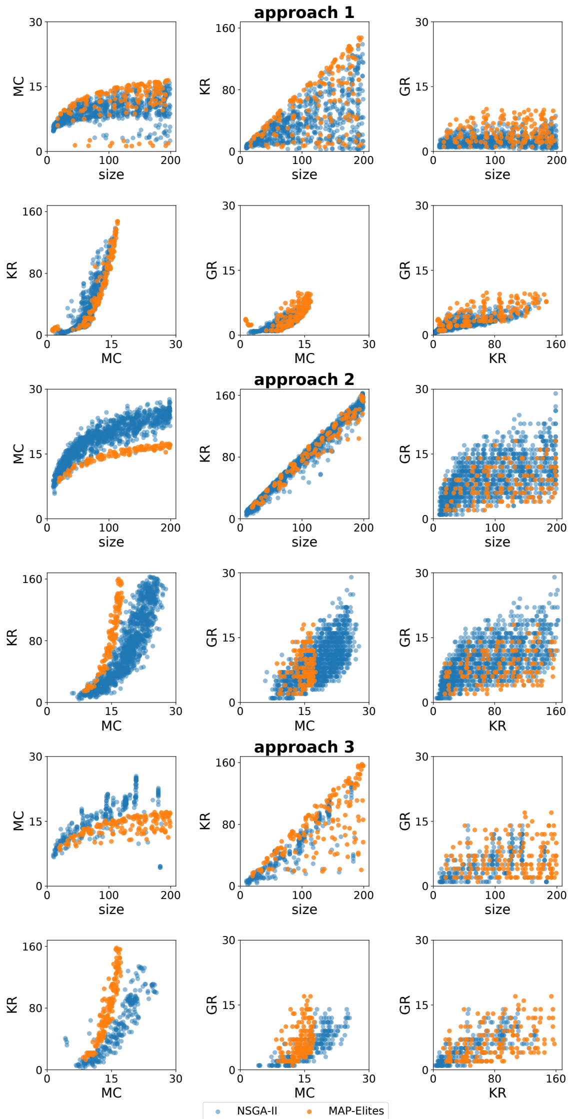


**Figure 1: Example of Pareto Front and final individuals.**

the reservoir's capability to generalize given similar input streams. In general, a low GR [1] symbolizes a robust ability to map similar inputs to similar reservoir states, rather than overfitting noise. Through this research, we look for reservoirs with high MC, high KR, low GR, and a minimum size.

We use these four objective functions with NSGA-II. In the case of MAP-Elites, only MC is treated as fitness, while the other objective functions are considered as features.

## 3 RESULTS

Figure 1 presents the Pareto front obtained by NSGA-II and the final individuals obtained by Map-Elites after 5000 evaluations. Approach 1 shows that there is no significant difference between the two algorithms for low-dimensional problems. We can also see that MAP-Elites struggles to achieve larger values of MC than NSGA-II in approach 2 and 3, but on the other hand, it is suitable for increasing the value of KR with respect to MC and have a slight advantage over NSGA-II in obtaining low GR.

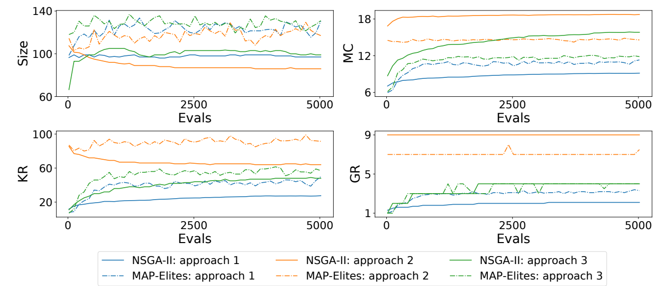


**Figure 2: The evolution of median of objective function values w.r.t evaluation over 5 runs.**

Figure 2 shows the evolution of the median of objectives. In approach 2, high KR and MC are achieved for a small size, but simultaneously, a high GR is obtained. However, the values of MC and GR do not change much, suggesting that we cannot properly optimize the input weight matrix and the reservoir connection weight. In the case of NSGA-II, the value of KR (to be maximized) is decreasing. This is because KR is strongly correlated to the size of the reservoir, and thus reduced while we minimize the size.

## 4 CONCLUSIONS

Focusing on the four properties of the reservoir, we experiment with NSGA-II and MAP-Elites to improve the performance of RC. A balance between properties is essential to match reservoir dynamics to tasks. However, determining the right balance is still challenging. We are able to generate optimal reservoirs with a variety of characteristics through the suggested approaches. It will be possible to select an appropriate reservoir according to the applications and tasks for which the reservoir is used. In the future, we would also like to implement physical RC other than ESN and optimize the properties of them.

## REFERENCES

[1] L Büsing, B Schrauwen, and R Legenstein. 2010. Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons. *Neural computation* 22, 5 (2010), 1272–1311.
[2] M Dale, JF Miller, S Stepney, and MA Trefzer. 2019. A substrate-independent framework to characterize reservoir computers. *Proceedings of the Royal Society A* 475, 2226 (2019), 20180723.
[3] K Deb, A Pratap, S Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.
[4] H Jaeger. 2001. *Short term memory in echo state networks*. Vol. 5. GMD-Forschungszentrum Informationstechnik.
[5] R Legenstein and W Maass. 2007. Edge of chaos and prediction of computational performance for neural circuit models. *Neural networks* 20, 3 (2007), 323–334.
[6] M Lukoševičius. 2012. A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade*. Springer, 659–686.
[7] JB Mouret and J Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).
[8] B Schrauwen, D Verstraeten, and J Van Campenhout. 2007. An overview of reservoir computing: theory, applications and implementations. In *Proceedings of the 15th european symposium on artificial neural networks. p. 471-482 2007*. 471–482.
[9] D Verstraeten, B Schrauwen, M d'Haene, and D Stroobandt. 2007. An experimental unification of reservoir computing methods. *Neural networks* 20, 3 (2007), 391–403.