Neuro-Evolutionary Direct Policy Search for Multi-Objective Optimal Control

Marta Zaniolo[1], Matteo Giuliani[1], and Andrea Castelletti[1]
[1]Department of Electronic, Information and Bioengineering, Politecnico di Milano, 20133 Milan, Italy

Direct Policy Search (DPS) is emerging as one of the most effective and widely applied Reinforcement Learning methods to design optimal control policies for Multi-Objective Markov Decision Processes (MOMDPs). Traditionally, DPS defines the control policy within a preselected functional class, and searches its optimal parameterization with respect to a given set of objectives. The functional class should be tailored to the problem at hand and its selection is crucial, as it determines the search space within which solutions can be found. In MOMDPs problems, a different objective tradeoff determines a different fitness landscape, requiring a tradeoff-dynamic functional class selection. Yet, in state-of-the-art applications, the policy class is generally selected a priori, and kept constant across the multidimensional objective space. In this work, we present a novel policy search routine called Neuro-Evolutionary Multi-Objective Direct Policy Search (NEMODPS), which extends the DPS problem formulation to conjunctively search the policy functional class and its parameterization in a hyperspace containing policy architectures and coefficients. NEMODPS begins with a population of minimally structured approximating networks and progressively builds more sophisticated architectures by topological and parametrical mutation and crossover, and selection of the fittest individuals with respect to multiple objectives. We tested NEMODPS for the problem of designing the control policy of a multipurpose water system. Numerical results show that the tradeoff-dynamic structural and parametrical policy search of NEMODPS is consistent across multiple runs, and outperforms the solutions designed via traditional DPS with predefined policy topologies.

***Index Terms*—Direct Policy Search, Multi-Objective control, Neural Networks Architecture, Neuroevolution.**

I. Introduction

THE coexistence of multiple heterogeneous conflicting objectives is a major challenge to many complex real world control problems, which are often formalized as Multi-Objective Markov Decision Processes (MOMDPs). In these problems, the optimal solution is an ensemble of Pareto optimal policies covering the space of tradeoffs and compromises across different objectives. In the last decades, Multi-Objective Reinforcement Learning (MORL) established as a solid approach to solve MOMDPs problems, but several open challenges remain in real world applications characterized by large continuous spaces that are too complex for a traditional optimal control formulation (for a review on MORL and open challenges see [1] and references therein). Direct Policy Search (DPS) [2] is emerging as one of the most popular MORL methods for solving complex MOMDPs problems, given its applicability to diverse tasks, scalability, and lack of restrictions in problem and objective formulation [3]. DPS defines the control policy within a given functional parameterization, and explores the policy parameters space by searching for the best solution with respect to a given set of objectives. So far, most of the DPS literature has focused on improving the search method [4], assuming that the subspace defined by the policy parameterization includes the optimal solution. This hypothesis, nevertheless, overlooks the impact that simplifications and mathematical assumptions in the problem formulation and the policy parameterization can have on the representation of the search space [5]. Some DPS works apply a linear or piecewise linear policy parameterization, albeit conditioning the control decision on trivial monodimensional state vectors [6]. A nonlinear multi-input multi-output function, such as an approximating network, provides a more flexible control policy shape [7]. Yet, approximating networks require the specification of a topology, which is crucial to determine the network processing capability and training requirements. The a priori definition of the optimal network topology for a given problem requires a full knowledge of the learning task that is generally unavailable. In practical applications, a topology is hence selected among few options via trials-and-errors, balancing the network approximation capacity, training costs, and overfitting tendency. Crucially, when multiple objectives are considered, the fitness landscape changes depending on the selected tradeoff, and the optimal network topology should be set accordingly. Yet, in state-of-the-art applications of DPS, a single policy class is selected to approximate solutions for every objectives tradeoffs.

This work contributes a novel policy search routine that addresses this challenge by evolving self-adaptive policy architectures responsive to changes in tradeoffs, namely, Neuro-Evolutionary Multi-Objective Direct Policy Search (NEMODPS). NEMODPS builds on a recent Reinforcement Learning branch called Neuro-Evolution (NE) (e.g., [8]), which employs Evolutionary Algorithms to generate optimal networks in terms of topologies and parameters. A well-known NE algorithm is the NeuroEvolution for Augmenting Topology (NEAT, [9]), a Single-Objective (SO) technique which begins with a population of simple networks and progressively builds more sophisticated ones through a complexification process driven by parametrical and topological evolutionary operators. A topological niching scheme protects newly emerged architectures from premature disappearence. Several authors developed NEAT-inspired alternatives to adapt it to a variety of machine learning tasks, mainly for game playing and robotics (see e.g., [10] and references within). Among them, NEAT

Corresponding author: M. Zaniolo (email: marta.zaniolo@polimi.it).

was tailored to problems characterized by highly discontinuous state-action mappings (RBF-NEAT, [11], SNAP-NEAT [12], CA-NEAT [13]), little domain specific knowledge [14], deceptive environments (Novelty Search, [15]), visual tasks (HyperNEAT [16]), dynamic problems with moving optimum (DynNEAT [17], SOMNE [18]), real-time adaptation of control policy (rtNEAT [19], ICONE [20]), and compact policy representation (SUNA [21]).

However, all the above algorithms address SO problems, i.e., optimization problems seeking to satisfy a single criterion or metric, yielding to one single optimal solution. The application of a SO algorithm to a Multi-Objective (MO) problem can be performed via decomposition into several SO sub-tasks, each characterized by a scalarized monodimensional objective function via several methods (e.g., weights, constraints, etc), albeit yielding to significant shortcomings related to computational inefficiency [22] and inadequacy to capture convex portions of the Pareto front [23]. MO methods, conversely, do not suffer from such shortcomings and are widely recognized to be more desirable when tackling a MO problem [3].

Sub-tasks are solved iteratively, each yielding one Pareto-approximate solution, resulting in a factorial growth of computational costs with the number of objectives, and suboptimality in the Pareto Front approximations in its concave regions [23]. An attempt at developing an explicitly MO version of NEAT (MO-neuroevolution [24]) required to sacrifice several crucial NEAT operators, as they are supported by the inherently single-objective niching scheme. In this work, we propose a MO generalization of the niching routine, which allows to preserve all NEAT operators in a MO problem.

We tested NEMODPS on a multi- and a many-objectives (i.e., more than 2 objectives [25]) problem of designing a control policy for a multipurpose water reservoir. Water systems are indeed often characterized by multiple water users such as irrigation, flood protection, navigation, environmental preservation [26]. In these contexts, the multiple objectives cannot be easily aggregated a priori to turn the original problem into a single objective one because of the intricacy of identifying the decision maker (DM) preference structure, potentially yielding to biased decisions [27]. Conversely, we need a posteriori generating techniques to compute the full set of Pareto-optimal solutions exploring the tradeoffs between different objectives prior to eliciting the DM preferences. Stakeholders and DMs can then negotiate the preferred solutions to implement on the basis of the resulting Pareto front (see [28] for a review of negotiation methods). Since this choice is not purely technical but political as it generally results from a negotiation process involving the different stakeholders and water users [29], it is beyond the scope of the algorithm presented here, which aims at the design of the Pareto-optimal set of solutions.

Technically, such MO control problems typically feature a complex decision space, continuous domains, and a noisy input-output mapping. Currently, the state-of-the-art policy architectures for these problems are single-layer, fully connected Artificial Neural Networks [3], [30]–[32].

NEMODPS implementation inherits NEAT basic structure for the dynamic search of efficient policy architectures, and the literature of NEAT refinements for problems presenting large decision spaces and noisy environments. In particular, inspiration came from the Evolutionary Acquisition of Neural Topologies (EANT) algorithm [33] which addresses problems characterized by a large decision space, continuous domains, and a noisy environment by coordinating the search in a dual timescale, optimizing the network's connection weights on a small timescale (exploitation phase), and the network's structure on a larger timescale (exploration phase) in order to give newly created structures time to optimize their parameters. Other recent works dealing with noisy environments and complex decision spaces experimented with the activation functions of neurons. Applications to benchmark classification [34] and regression problems [35] demonstrate how heterogeneous networks characterized by a combination of activation functions can result in improved approximations capabilities, smaller networks with fewer training requirements, and a significantly reduced overfitting tendency when tested on noisy environments. Additionally, the niching routine is generalized for MO problems with a novel strategy, as to support the exploration of multidimensional tradeoffs in a single run of the algorithm.

In this work, we run a benchmark analysis [36] comparing the policies produced by NEMODPS, NEAT, and traditional DPS. Results show that the Pareto-dynamic structural and parametrical policy search of NEMODPS produces reliable policies, highly robust when tested on unseen data. Additionally, we perform a Pareto-dynamic convergence analysis of NEMODPS, and we analyze how the efficient architectures change in response to a change in the objective tradeoff, according to several metrics of structural analysis.

II. Methods

A. Problem formulation

We consider a discrete-time continuous MOMDP defined as a tuple $< \mathcal{X}, \mathcal{U}, \mathcal{T}, \mathbf{G} >$ where $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the continuous state space, $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ is the continuous action space, $\mathcal{T}(x_{t+1}|x_t, u_t)$ is the probabilistic transition function defining the transition density between state x_t and x_{t+1} under action u_t, $\mathbf{G}(x_t, u_t, x_{t+1}) = \left[G^1, \ldots, G^M\right]$ is a M-dimensional reward (or cost) function that specifies the vector of instantaneous rewards (costs) $\mathbf{g}_t = [g_t^1, \ldots, g_t^M]$ for each objective when state x_{t+1} is reached from state x_t by taking action u_t. Action u_t is extracted from a control policy π, $u_t = \pi(x_t, u_t)$, associated with a vector of expected returns $\mathbf{J}(\pi) = \left[J^1(\pi), \ldots, J^M(\pi)\right]$ defined over the control horizon $[0, H]$ as:

$$J^m(\pi) = \mathbb{E}\left\{\sum_{t=0}^{H} (\gamma^m)^t g^m(t+1) | x_0 \sim \mu\right\} \tag{1}$$

where $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_M] \in [0, 1]$ is the vector of discount factors relative to each objective, and μ is the initial state distribution.

The solution of the RL problem defined above is the policy π^* that yields the optimal value of objective vector $\mathbf{J}$ (here considered as a cost to be minimized) in its M dimensions:

$$\begin{aligned}\pi^* &= \arg\min_{\pi} \mathbf{J}(\pi, \mu) \\ &= \arg\min_{\pi} \left[J^1(\pi, \mu), \ldots, J^M(\pi, \mu)\right]\end{aligned} \tag{2}$$

In general, conflicts occur among different operating objectives, and it is thus not possible to define a single optimal policy, representing the optimum with respect to the M dimensions of $\mathbf{J}$. The solution of a MO problem is, in general, constituted by a set of non-dominated (or Pareto optimal) solutions $\mathcal{P}^* = \{\pi^* | \nexists \pi \prec \pi^*\}$, which maps onto the Pareto front $\mathcal{F}^* = \{\mathbf{J}(\cdot) | \pi^* \in \mathcal{P}^*\}$.

Definition 1: Policy π dominates policy π', denoted by $\pi \prec \pi'$, if: $\forall m \in \{1, \ldots, M\}, J^m(\pi) \leq J^m(\pi') \land \exists m \in \{1, \ldots, M\}, J^m(\pi) < J^m(\pi')$.

It is possible to solve a MO problem with a SO optimization algorithm by decomposing the MO problem into several SO tasks, each representing a different prespecified objective tradeoff. In particular, the M objectives are combined with a scalarization function $\Gamma : \mathbb{R}^M \rightarrow \mathbb{R}$. Traditionally, a convex combination of the objectives is applied using weights $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_M] \in \Lambda^{M-1}$, where Λ^{M-1} is the unit $(M-1)$-dimensional simplex (so that $\sum_{i=1}^{M} \lambda_i = 1$ and $\lambda_i \geq 0 \quad \forall i$). For a SO control routine, problem (2) is hence reformulated as:

$$\begin{aligned}\pi^* =& \arg\min_{\pi} \mathbf{J}(\pi, \mu) = \\ & \Gamma\left([J^1(\pi, \mu), \ldots, J^M(\pi, \mu)]\right)\end{aligned} \tag{3}$$

Given a desired precision of Pareto front approximation (i.e., number of solutions along a single objective axis), the computational cost required by the solution of Problem (3) grows factorially with the number of objectives M [22], and is defined by the following permutation:

$$S = \sum_{i=1}^{M} \frac{M!}{i!(M-i)!} + M \tag{4}$$

where S is the number of sub-tasks to be solved, equal to the number of Pareto approximate points produced.

Traditionally, the solution to Problem (2) is obtained by searching for the optimal action-value function $\mathbf{Q}^*(x_t, u_t)$, defined as the optimal cumulated future cost associated with each pair $(u_t,\, x_t)$ at time t. Such future cost is obtained by integrating in the state space (X) the immediate cost G and a discounted optimal future cost for time $t+1$:

$$\begin{aligned}\mathbf{Q}^*(x_t, u_t) = \int_{\mathcal{X}} & [\mathbf{G}(x_t, u_t, x_{t+1}) + \\ & \boldsymbol{\gamma} \min_{u_{t+1} \in \mathcal{U}} \mathbf{Q}^*(x_{t+1}, u_{t+1})] \mathcal{T}(dx_{t+1} | x_t, u_t)\end{aligned} \tag{5}$$

The exact complete estimation of the value function in its M dimensions is however possible only for a limited class of problems, while it quickly becomes computationally intractable for problems characterized by high dimensional action or state spaces (i.e., curse of dimensionality [37]) and objective space (i.e., curse of multiple objective [38]). Moreover, any variable considered into the problem formulation must be explicitly modeled in order to compute the value function (i.e., curse of modeling [39]).

In general, an approximated method is used when one or more curses prevent reaching an exact solution. The approximation can regard the action-value space (see e.g., [40], [41]), or the policy space, where the search for the optimal control policy is restricted to a prespecified parametric class of functions [42]. In this second approach, the control policy is applied to the system for a given horizon $[0, H]$. The sequence of states and controls produced is employed to compute the policy performance according to the problem's objectives.

This sequence defines a trajectory τ employed in the calculation of the objective $\mathbf{J}(\pi) = \mathbb{E}[\mathbf{G}(\tau) | \pi]$.

Direct Policy Search belongs to this class, and according to the taxonomy of Policy Search methods proposed in [2] configures as a stochastic, model-based and episode-based method. In particular, DPS approaches policy design as a problem of optimal functional parameterization, defining the control policy π_θ within a given function class, and then searching the parameters' space Θ to find the optimal parameterization $\theta^* \in \Theta$ with respect to the M-dimensional set of objectives $\mathbf{J}$. Hence, Problem (2) is reformulated as:

$$\pi^*_\theta = \arg\min_{\pi_\theta} \mathbf{J}(\pi_\theta, \tau) \tag{6}$$

Selecting an appropriate functional class for π_θ is critical, as DPS routines can find, at most, the best parameterization within the predefined class. In the absence of pre-existing knowledge of a (near-)optimal policy shape, highly flexible function classes (e.g., nonlinear approximating networks) are preferred [7], [30], [31], in order not to restrict the search to a policy subspace that, likely, does not contain the optimal one. Yet, optimizing the parameters of approximating networks requires searching high dimensional spaces that map to a noisy and multidimensional objective space. Comprehensive reviews (e.g., [43]) and extensive diagnostic assessments [44] have established the suitability and high algorithmic reliability of MO Evolutionary Algorithms (MOEAs) for tackling multi- and many- objective water control problems given their demonstrated ability to efficiently handle performance uncertainties [45], [46]. In state-of-the-art applications of DPS, an appropriate network dimension is selected by trials-and-errors, adjusting the number of neurons in a single-layer, fully connected, homogeneous network [3], [30]–[32], [47]. This architecture choice is motivated by theoretical results, which demonstrated that single- or multi-layer feedforward neural networks with continuous, non-constant, activation functions, could approximate any continuous bounded function to a desired accuracy, given enough nodes [48]. The nominal capacity of a neural network to absorb information is thus just limited by the number of its processing units, where numerous units imply large flexibility and approximation capacity. The network topology does not influence the theoretical expressiveness of a network; however, several studies show that, in practical applications, it significantly affects its training requirements, and approximation capacity. Firstly, fully connected networks offer high flexibility, but tend to force spurious connections that have no physical meaning, facilitating the overfitting to noise in training data [49], [50]. Secondly, the depth (i.e., number of layers) of a neural network affects its behavior in solving high complexity learning tasks. While a shallow (single-layer) network provides a direct input-output mapping described by the single hidden layer, the global mapping provided by a deep (multilayer) network is the result of the composition of several layers, a valuable asset in problems presenting regularities in the input-output mapping [16], [51]. Thirdly, comparative studies have demonstrated that the choice of nodes' activation functions plays a key role in determining convergence time and network accuracy [52]. An appropriate mix of activation functions generally reduces the number of processing units required for a task, and, accordingly, its training requirements and overfitting tendency [35]. Overall, these results indicate that in real-world applications, the network's topology plays a significant role in determining its suitability for a given task, and it should not be dismissed in DPS applications. Moreover, in MO problems, the multidimensional landscape defined by solutions mapped into corresponding value of objectives (i.e., fitness landscape) changes depending on the tradeoff. Every possible tradeoff combination originates a different sub-problem, and an efficient network topology should be set accordingly and tradeoff-dynamically.

B. Extending the scope of DPS

In this work, we extend the DPS problem formulation to search optimal policies in terms of architectures and relative parameterization Pareto dynamically. Accordingly, Problem (6) is reformulated as:

$$\pi^*_{\zeta(\theta)} = \arg\min_{\pi_{\zeta(\theta)}} \mathbf{J}(\pi_{\zeta(\theta)}, \tau) \tag{7}$$

where $\pi_{\zeta(\theta)}$ explicits the search for policy hyperparameters ζ defining a policy architecture as well as regular policy parameters θ, whose number and nature depend on the hyperparameters as in $\zeta(\theta)$. The policy search problem is thus expanded to conjunctively search architectural and parametrical spaces, enhancing DPS potential for single- and especially multi-objective problems.

C. NEAT

Problem (7) can be solved with Neuroevolution, a machine learning branch which employs evolutionary algorithms to automatically generate efficient artificial neural networks. NEAT (NeuroEvolution for Augmenting Topology, [9]) is the first prominent neuroevolution algorithm, and the benchmark for this field. It begins with a population of simple networks and progressively builds more complex topologies through a complexification process. In every generation of the evolutionary progress, the performance of each individual is evaluated with respect to a fitness function, and the fittest individuals survive onto the next generation. New derivative networks are created based upon the surviving networks by applying evolutionary operators (i.e., topological and parametrical mutation and crossover), to drive the search for efficient topologies and connection weights.

As the evolution proceeds and individuals complexify, increasingly sophisticated behaviors emerge. However, the addition of new structural elements with unoptimized coefficients is, at first, detrimental for an individual, and the usefulness of a topological innovation may become apparent only when given enough iterations to optimize. NEAT implements a niching scheme with the dual aim of protecting topological innovations from premature disappearance, and sustaining solution diversity. Topological innovation is protected by allowing individual competition only within niches of similar topologies. The population is partitioned into niches (or species), by evaluating a metric of topological distance δ between couples of individuals X_i and X_j:

$$\delta(X_i, X_j) = \frac{c_1 E_{i,j}}{NTE} + \frac{c_2 D_{i,j}}{NTE} + c_3 W_{i,j} \tag{8}$$

where $E_{i,j}$ is the difference in number of connections between X_i and X_j, $D_{i,j}$ is the difference in number of nodes, $W_{i,j}$ is the difference in average connection weights, c1, c2, c3 $\in [0,1]$ express the relative importance of each factor, and NTE is the maximum Number of Topological Elements in the networks. Individual X_i is assigned to species s if:

$$\delta(X_i, X_{j,s}) < \delta^* \tag{9}$$

where δ^* is a predefined speciation threshold, and $X_{j,s}$ is the reference individual for the species, extracted randomly from species s at each generation. A new species is created if (9) is not verified for any existing one.

Species compete among each other for their ability to reproduce, so that a larger offspring is assigned to well performing niches. However, a fitness sharing mechanism is introduced to penalize populous species and prevent them from taking over the entire population, thereby sustaining topological diversity.

In particular, a species' fitness is computed as the average shared fitness of its components. The Shared Fitness of individual X_i belonging to species s ($\mathrm{SF}_{X_{i,s}}$) is determined by normalizing its fitness f_{X_i} to the species' numerosity n_s with the following:

$$SF_{X_{i,s}} = \frac{f_{X_i}}{n_s} \tag{10}$$

The allotted number of individuals n'_s to species s in the next generation is determined by its average shared fitness normalized by the population average $\overline{SF}$:

$$n'_s = \frac{\frac{1}{n_s}\sum_{i=1}^{n_s} SF_{X_{i,s}}}{\overline{SF}} \tag{11}$$

D. NEMODPS

The implementation of NEMODPS inherits NEAT structure, and the literature of NEAT improvements targeting complex control design problems, vast decision spaces, and noisy environments. Additionally, we propose an original strategy to extend the search to MO problems. The meta-algorithm for NEMODPS is reported in the Supplementary Information attached to this manuscript in Algorithm S1. Below, we discuss the additional elements that differ from the original NEAT implementation.

First, NEMODPS assimilates the search dual timescale proposed in EANT [33]. Parametrical mutations occur in every generation to exploit existing structures. Topological innovations are injected every few generations, performing the exploration of the architectural hyperspace on a larger time scale. Second, when new neurons are injected into a network, the activation function is randomly selected among sigmoids and gaussians, allowing the generation of heterogeneous networks. Third, the speciation strategy is modified to reduce the criticality of the speciation threshold choice. An excessively low speciation threshold produces too many species and an overly fragmented population with restricted interaction between individuals, and weakened selection pressure. On the contrary, an excessively high speciation threshold produces overly homogeneous populations, an unfavorable environment for new emerging topologies competing against well optimized structures. Moreover, the appropriate speciation threshold can vary significantly throughout the evolution as the population complexifies. Alternatively to trying to guess a fair compromise for the selection threshold, some authors suggested to, instead, select an appropriate number of species to be maintained, and adjust the threshold accordingly during the evolution [53]. NEMODPS implements the latter technique, initializing a threshold for speciation $specThresh = \delta$, and an appropriate number of species to be maintained relatively constant during the search, $speciesNumerosity^* = \sigma$ In every generation, $specThresh$ is incremented if the number of species is above $speciesNumerosity^*$, and lowered if inferior. Lastly, NEAT supports SO optimization, and its application to a MO problems requires the iterated solution of several SO tasks with a scalarized monodimensional objective as in Problem (3). In a previous attempt to define a multi-objective neuroevolution routine, named MO-neuroevolution, the Non-dominated Sorting Genetic Algorithm II (NSGAII, [54]) was embedded in NEAT to perform the selection of the fittest individuals within niches in a multidimensional objective space [24]. The niching scheme supported by the Shared Fitness defined in eq. (10), however, does not seamlessly generalize to MO problems, given the difficulty to compare fitnesses with respect to multiple objectives, and therefore the MO-neuroevolution implementation sacrificed the speciation and fitness sharing operators. In NEMODPS, we employ the non-dominated sorting approach proposed in NSGAII (see [54] for details) for intra-species competition suitable to select a predetermined fraction of efficient individuals within a species. Additionally, we contribute an original definition of the fitness sharing operator for MO problems, thus restoring the speciation operator accordingly.

The Generalized Shared Fitness of individual X_i in species s, $GSF(X_{i,s})$, assigns a score to X_i equal to the number of individuals $X_j, j \neq i$ that are not dominating X_i.

$$GSF(X_{i,s}) = \sum_{j \in [1,\dots,N]: j \neq i} d_j; \quad d_j = \begin{cases} 0 & \text{if } X_j \prec X_i \\ 1 & \text{else} \end{cases} \tag{12}$$

where N is the total number of individuals in the population. The top score achievable is $GSF(X_i, s) = N - 1$, attained by individuals populating the best current approximation of the Pareto front. Solutions close to the approximate Pareto front are assigned good scores if they are located in sparsely populated regions, and lower scores if they are located in crowded areas, as they are more likely to be (semi-)dominated. An example of GSF computation for 2 individuals in a 2-objectives problem is presented in Fig. S1 of the Supplementary Information). Species grow or shrink depending on whether the average generalized shared fitness of their individuals is above or below the population average (lines 19-20), in accordance with the NEAT implementation in eq. (11). Species competition is thus based on a relative individual ranking, a strategy that is often featured in MOEAs, and has been demonstrated to handle performance uncertainties more effectively than relying on the estimation of absolute performance [46]. Additionally, in this formulation, the fitness sharing operator penalizes individuals' proximity in the objectives space, rather than in the topological space as originally conceived in NEAT. This transition is encouraged by several authors, who have observed that topological diversity does not necessarily induce a behavioral diversity of solutions for every task [55], [56]. This observation is key in MO problems: if a certain sector of the Pareto front can be approximated with a trivial solution, a broad set of topologies will succeed in reaching a high performance. By rewarding topological diversity, solutions will quickly concentrate in the trivial region, resulting in a topologically diverse population, but a poor approximation of the Pareto front, which instead should be the ultimate goal of MO policy search. With the proposed generalized fitness sharing, species are encouraged to achieve solution diversity intended as a good exploration of the tradeoffs in the Pareto front, rewarding ensembles that are well performing, and that occupy relatively empty and non-dominated regions of the objective space.

E. Metrics of Structural Analysis

As argued in Section II-B, the learning behavior of a network largely depends on its topology, therefore, topological analysis of Pareto-approximate networks could provide useful insights into the learning task. Neuro-optimized topologies are generally irregular, presenting sparse connections, hidden layers of different sizes, and heterogeneity in the activation functions. In order to characterize their topology, we use three metrics of structural analysis that capture critical network features, allowing us to compare and contrast different topologies.

The first metric, namely the Preference for Deep Learning (PDL), is measured as the ratio between number of hidden layers (L) and hidden nodes (H) in a structure.

$$PDL = \frac{L}{H} \tag{13}$$

PDL $\in (0, 1]$, tends to zero when hidden nodes are organized in one or few very populated layers, and assumes value one when there are as many layers as nodes.

The second metric is a measure of Network Complexity (NC), defined as the total number of parameters, namely connection weights and node biases, needed for its description.

The sum of the number of connections (C), hidden nodes (H), and output nodes (O), determines the Network Complexity as follows:

$$NC = C + H + O \tag{14}$$

High NC values are representative of complex networks, likely to reproduce sophisticated behaviors.

Lastly, the third considered metric of structural assessment is a measure of Network Heterogeneity, computed as the ratio of Gaussian nodes (GN) to the total number of hidden nodes.

$$NH = \frac{GN}{H} \tag{15}$$

By definition, NH$\in [0, 1]$ where NH = 0 indicates a homogeneous network comprising only sigmoidal nodes, and NH = 1 indicates a homogeneous Gaussian network. An even mixing of Gaussian and sigmoidal activation functions is verified for NH = 0.5. For examples and visual representations of the proposed metrics, refer to Fig. S2 in the Supplementary Information.

III. Case Study

NEMODPS is tested for a problem of designing the optimal control of a multi-purpose water resources system. Typical features of these problems are large decision spaces, presence of noise, and multiple conflicting objectives.

In this application, we design the control policy of Lake Como, a multipurpose regulated lake situated in the southern Alpine belt (Italy). The main tributary, and only emissary of the lake is the Adda river, whose waters are withdrawn downstream to the lake to irrigate four agricultural districts. The southwestern branch of Lake Como constitutes a dead end, and exposes the city of Como to flooding events.

The system is modeled as a discrete-time, periodic, nonlinear, stochastic process defined by a scalar state variable x_t (i.e., storage), a control variable u_t representing the release decision from the dam gates, stochastic disturbances ε_{t+1} (net reservoir inflow), and a state-transition function $f(\cdot)$: $x_{t+1} = x_t - r_{t+1} + \varepsilon_{t+1}$ where the effective release r_{t+1} coincides with the release decision u_t corrected, where appropriate, with a non-linear release function $R_t(x_t, \varepsilon_{t+1})$ determining the minimum and maximum releases feasible for the time interval $[t, t+1)$ to respect physical and legal constraints (for more on this, see the control scheme in the Supplementary Information in Fig. S3). The Adda River is described by a plug-flow model, which simulates the routing of the lake releases to the intake of the irrigation canals. The adopted time step is 1 day, and and the system is periodic with period T = 365 days.

The lake regulation generally considers two conflicting aims of minimizing flood risk on the lake shores, and supplying water to downstream users by storing spring snowmelt-driven inflow peak and releasing throughout summer when the irrigation demand is highest. On the basis of previous works [57], these two objectives are defined as:

Flooding: the average number of annual flood days, defined as days in which the lake level h_t is above the flood threshold $\bar{h} = 1.24$ m, i.e.:

$$J^{flood} = \frac{1}{N_y}\sum_{t=0}^{H-1} g^{flood}_{t+1}; \quad g^{flood}_{t+1} = \begin{cases} 1 & \text{if } h_{t+1} \geq \bar{h} \\ 0 & \text{if } h_{t+1} < \bar{h} \end{cases} \tag{16}$$

where N_y is the number of years in the simulation horizon.

Irrigation: the daily average squared water deficit with respect to the daily downstream demand w_t, subject to the minimum flow constraint $q^{MEF} = 5$ m^3/s to guarantee environmental stakes. The quadratic formulation is selected with the aim of penalizing severe deficits in a single time step, while allowing for more frequent, small shortages. i.e.,

$$J^{irr} = \frac{1}{H}\sum_{t=0}^{H-1}(max(w_t - (r_{t+1} - q^{MEF}), 0))^2 \tag{17}$$

In a second, more challenging experiment, we extend the Lake Como problem formulation to include two additional objectives, namely:

Navigation: the average number of annual dry days, defined as days in which the lake level h_t is below the navigation threshold, under which lake navigation is prohibited $\underline{h} = 0.205$ m, i.e.:

$$J^{nav} = \frac{1}{N_y}\sum_{t=0}^{H-1} g^{nav}_{t+1}; \quad g^{nav}_{t+1} = \begin{cases} 1 & \text{if } h_{t+1} \geq \underline{h} \\ 0 & \text{if } h_{t+1} > \underline{h} \end{cases} \tag{18}$$

Environment: the daily average squared deviation of the water released in Adda river with respect to the downstream undisturbed hydrological regime q_t, computed as a 30 years cyclostationary mean river regime downstream the lake on past data of undisturbed river flow data.

$$J^{env} = \frac{1}{H}\sum_{t=0}^{H-1}(q_t - r_{t+1})^2 \tag{19}$$

We hereby assume the considered simulation horizon H is sufficiently long to not require the addition of a penalty function to the final state.

IV. Computational Experiment

The problem of finding a set of Pareto approximate control policies for the Lake Como system was solved via three policy search methods, NEMODPS, NEAT, and traditional DPS, respectively. In these experiments, the designed optimal control policies provide the control u_t as a function of a three-dimensional input set I_t comprising the state of the system (i.e., the current reservoir storage) and two transformations of the time index t with sine and cosine, to embed time-variability and cyclostationarity in the control policy $I_t = |x_t, sin(t), cos(t)|$.

NEMODPS solves Problem (7), with $\mathbf{J} = [J^{flood}, J^{irr}]$ in the first, 2-objectives experiment, and $\mathbf{J} = [J^{flood}, J^{irr}, J^{nav}, J^{env}]$ in the second, 4-objectives, experiment. NEMODPS was run for 10 independently initialized and randomized seeds. Each seed comprises a Number of Function Evaluations (NFE) equal to 600,000, corresponding to a population size of 600 evolved for 1000 generations. The population is divided in a number of species that oscillates around the selected value of $speciesNumerosity^* = 15$. Individuals of the initial population consist of one hidden, one output node, and 4 connections, for a total of 6 parameters. Connections link inputs to the hidden node, and the hidden node to the output. Evolved individuals feature different complexities, spanning from 10 to 31 parameters across the 10 runs.

NEAT solves a SO version of Problem (7) where the objectives are aggregated using a weighted mean with 15 uniformly sampled combinations of weights. NEAT is only employed to solve the 2-objectives problem, as it would be computationally prohibitive to apply it to adequately represent the Pareto front in the 4-objectives problem. Each run of NEAT thus demanded the same computational effort of NEMODPS multiplied by the 15 tradeoff combinations considered.

Finally, the application of traditional DPS solves Problem (6) searching only the policy parameters $\theta \in \Theta$ for a predefined functional class. DPS requires the specification of a search algorithm, and of a policy structure. As search algorithm we selected the ϵ-NSGAII MOEA [58], which demonstrated consistently high levels of performance on an extensive diagnostic benchmarking for challenging MO problems [44]. ϵ-NSGAII extends the original NSGAII by including epsilon dominance archiving, adaptive population sizing, and time continuation that were demonstrated fundamental in discovering high quality solutions for similar problems characterized by multi and many heterogeneous objectives. Such problems are cursed by the issue of dominance resistance, i.e., the number of non-dominated solutions increases very quickly, and it becomes difficult to discriminate between solutions. ϵ-dominance is shown to alleviate the dominance resistance problem by allowing to discern among solutions with the desired precision for each objective, instead of burdening the search with a number of operationally equivalent solution with minor numerical differences in the objective values. For a detailed description of the algorithm refer to [58]. Concerning the policy structure, a single-layer, fully connected, homogeneous network was selected, as in state-of-the-art applications [3], [32]. The experiment was repeated for differently sized networks, from 1 node (corresponding to 6 parameters), to 6 nodes (31 parameters), covering an interval of parameters which contains the range delimited by optimized NEMODPS networks. These networks were populated homogeneously with sigmoidal activation functions, generating common Artificial Neural Networks (ANN), and with gaussian functions, generating Gaussian Perceptrons (GP).

Because the DPS problem formulation only searches the parameters' space, in contrast to the Neuroevolutionary formulation which searches the hyperspace comprising networks parameters and topologies, the number of function evaluations had to be adjusted to ensure a fair comparison across methodologies. By inspecting the search progression in NEMODPS, it was determined that, on average, the structures populating NEMODPS Pareto fronts remained fairly constant for the last 300 thousands evaluations. As a result, each DPS experiment was run for 10 seeds, and for NFE = 300,000.

The above policy search experiments were performed on a 10 years calibration horizon 1997-2006 comprising a mix of wet and dry years. Optimal policies were then tested on three validation chunks: an extended 20-years validation from 1977-1996, a combination of extreme dry years (1949, 1962, 1990, 1994, 2007), and wet years (1951, 1960, 1977, 2008, 2014) selected by searching the driest and wettest years from the available historical record of inflows to Lake Como (1947-2014), discarding the calibration years.

V. Numerical results

A. Benchmark analysis

The first experiment we present is a multi-objective benchmark analysis, contrasting the performance of Pareto-approximate control policies produced via NEMODPS, NEAT, and state-of-the-art DPS. The solutions displayed in this figure are the non-dominated solutions resulting from merging the Pareto front approximations of independent repetitions of the three policy search routines selecting non-dominated solutions among those generated by the multiple algorithmic runs. For all the considered policy search methods, the control policies are designed on a calibration dataset, and their performance is reported in Fig. 1a. The best performing solutions locate in the bottom left region of the objective space, corresponding to low values of J^{irr} and J^{flood}. Marker size is proportional to network dimension (i.e., number of parameters, or topological elements). State-of-the-art DPS networks with sigmoidal activation functions (namely, ANN, pink diamonds) and Gaussian functions (namely, GP, green diamonds) obtain the best calibration results, producing a Pareto front that completely dominates the one obtained via NEMODPS (blue circles) and NEAT (black triangles). However, when tested on unseen validation datasets, their performance significantly deteriorates. Benchmark DPS architectures thus demonstrate a tendency to overfit noise patterns in training data, which enables to attain impressive calibration results, but without effectively producing superior policies when compared to other policy search routines. On the contrary, Neuroevolutionary (i.e., NEMODPS and NEAT) control policies offer a much more stable validation/calibration ratio and consistently outperform benchmark DPS on all three validation datasets. NEMODPS, additionally, consistently offers an exhaustive exploration of the Pareto front, with very limited gaps even when tested on validation datasets. Conversely, solutions produced by fixed structure DPS tend to concentrate in restricted portions of the frontier, (e.g., panel (c)). NEAT policies almost overlap with NEMODPS solutions in the extremes of the Pareto front; however, the central region of the front is poorly characterized, presenting large gaps, and dominated solutions. This holds true both for the calibration and the validation experiments. Remarkably, selecting evenly spaced set of weights to aggregate the two objectives does not guarantee a uniform distribution of NEAT solutions in the Pareto front. The dishomogeneity in the solutions distribution could depend on concavities in the real unknown Pareto front, which are impossible to capture with a convex combination of objectives.

Intuitively, the higher reliability of neuro-evolved policies in contrast to traditional pre-defined structures against a suite of diverse validation experiments can be explained by the fact that

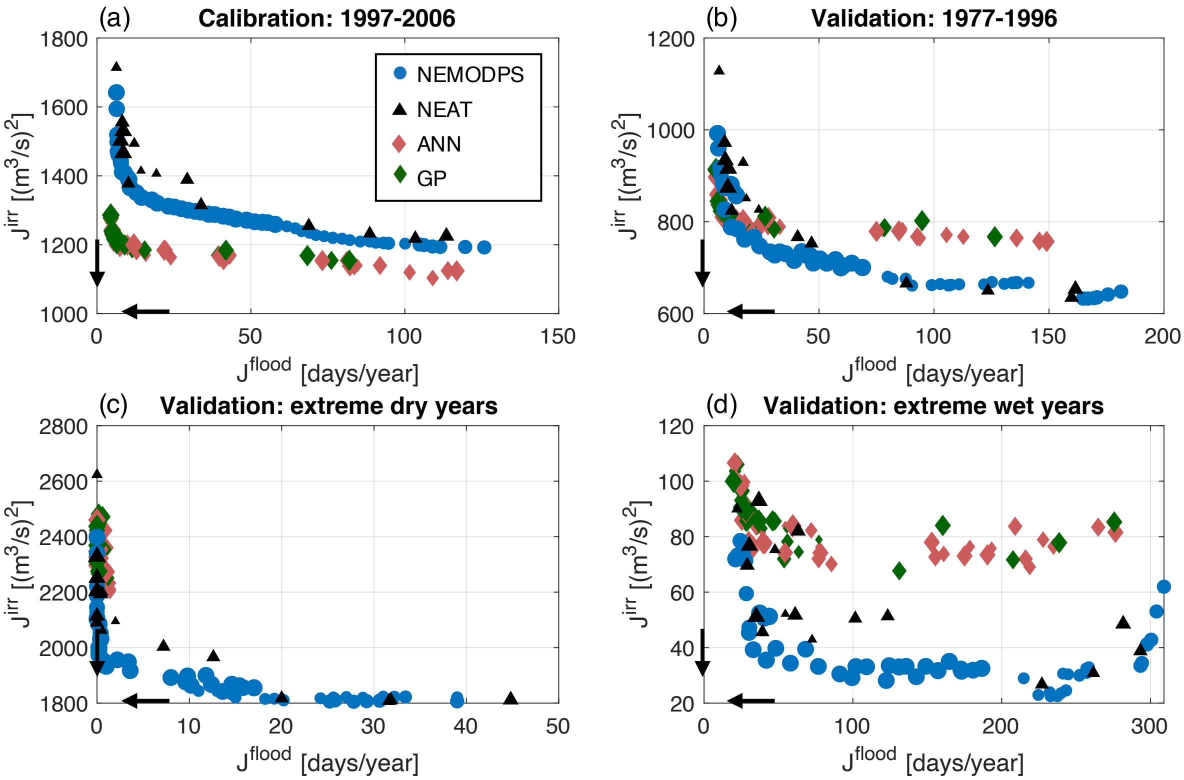

Fig. 1. Comparison of the control policies performances designed via NEMODPS (blue circles), NEAT (black triangles), and traditional DPS with fixed structures ANN and GP networks (pink and green diamonds). Policies are evaluated over a 10 years calibration period (panel (a)), a 20 years validation horizon of recorded inflows trajectory (panel (b)), and two 5 years extreme validation horizons (extreme dry in panel (c), and extreme wet in panel (d)).

each topological element of neuro-optimized networks was established as the result of a genetic selection [9]. Consequently, the added value of every element is tangible, otherwise simpler networks, with lower calibration requirements, would have prevailed. On the contrary, by pre-specifying a network structure, any superfluous element populating the network (e.g., connections with no physical sense) will contribute to overfit the noise patterns, ultimately undermining the network generalization capability.

B. Trends in policies architectural features

The following analysis is aimed at exploring more in detail NEMODPS topology selection, by uncovering possible trends and regularities in the architectural features of the Pareto-approximate solutions produced by the 10 independent runs of NEMODPS. This analysis is supported by the three structural metrics defined in Sec. II-E, computed for every solution, and plotted against their performance with respect to J^{flood} in Fig. 2. The flood objectives is used as a proxy to represent the solution tradeoff, as, for a given seed, lower J^{flood} values correspond to higher J^{irr} values.

The first panel of Fig. 2 displays the Preference for Deep Learning (PDL), defined in eq. (13). Each line represents one of the 10 independent runs of NEMODPS. By inspecting the lines ensemble a clear trend is visible: as J^{flood} increases, (corresponding to moving the tradeoff in favor of good J^{irr} performance) the values of PDL tend to increase as well, eventually reaching 1 in all the iterations. We notice more architectural variability for low values of J^{flood}, attributable to the fact that the J^{flood} minimization problem can be solved more easily than the J^{irr} problem (more on this in the next section and Fig. 3), and can therefore be tackled by a variety of alternative architectures.

The second panel of Fig. 2 shows the values of Network Complexity (NC, equation 14) that counts the number of topological elements present in the network, including hidden and output nodes, and connections, with respect to increasing values of J^{flood}. A visible trend persists in all the 10 runs, indicating that efficient architectures tend to simplify, on average, for high values of J^{flood}. Also the range of complexities covered by the solutions is sensible to a change in tradeoff. Low flood solutions display high variability in NC across different seeds, spanning from 12 to 41 parameters below 20 flood days. On the other end of the tradeoff curve, instead, solutions are confined within the 10 to 15 parameters range except for one seed stabilizing on 20 parameters. The last indicator of Network Heterogeneity (eq. 15) does not present any visible trend in response to the change of the J^{flood} objective. However, except for very few cases, Pareto-approximate networks select heterogeneous configurations comprising a mix of sigmoidal and gaussian functions. In this mix, generally, sigmoidal functions constitute

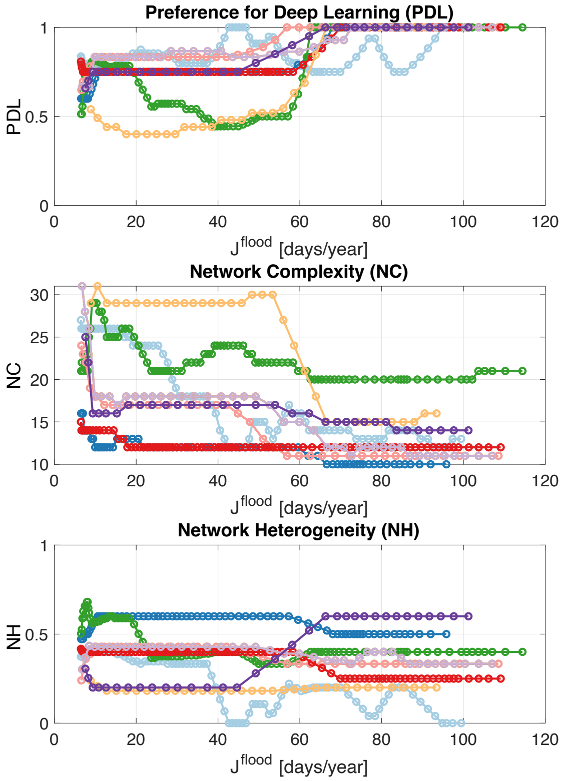

Fig. 2. Pareto dependent structural analysis of optimal solutions resulted from 10 independent runs of NEMODPS, represented by different line colors. The three metrics employed for structural analysis are Preference for Deep Learning (top panel), Network Complexity (middle panel), and Network Heterogeneity (bottom panel).

the greater portion (verified for NH < 0.5).

In summary, different runs of NEMODPS evolve independently to reach a coherency in the architecture of Pareto-approximate networks, indicating rationality in the network generation. The optimization routine, moreover, responds to changes in tradeoff by consistently adapting the solution topology, confirming that multi-objectives problems should be approached with a Pareto-dynamic selection of optimal architectures.

C. Convergence analysis from a multi-objective perspective

The last experiment is aimed at verifying the convergence of the solutions produced via NEMODPS across its independent runs from a multi-objectives perspective. First and second panel of Fig. 3 represent the minimum value of the two objectives, J^{flood} and J^{irr}, respectively, throughout the search until the maximum generation is reached. These two objectives present a remarkably different behavior: the best value of $J^{flood} = 6.3$ is consistently found at an early stage of the search by every algorithmic iteration, represented by differently colored lines, indicating that policies that minimize flood days (irrespective of their J^{irr} value) are relatively trivial to obtain. Conversely, the quest for an optimal irrigation deficit performance appears much more complex, given the slower progression towards low values of J^{irr}. Notably, a marked difference in computational effort required by different objectives poses an additional challenge to the use of SO policy search routines, as it complicates the selection of the set of weights employed in the objective aggregations. An example of this is visible in Fig. 1, where an evenly spaced set of aggregation weights for NEAT produces clusters of solutions scoring low values of J^{flood} and gaps in the Pareto front. As opposed to J^{flood}, the best J^{irr} solution obtained at the end of the search differs for every iteration, however, the final solutions place within a range of 5.19% with respect to the lowest, indicating a contained inter-seed variation even in regions of the Pareto front that appear more difficult to approximate. Despite the slower progression to the final optimal value of J^{irr}, we can consider the search converged, especially when examining metrics for assessing the quality of the approximation of the multi-dimensional Pareto front in Panels (c) and (d), namely the hypervolume indicator HV [59], and the Inverted Generational Distance (IGD) [60]. HV accounts for both convergence and diversity of an approximate set of solutions $\mathcal{F}$ capturing the behavior in intermediate regions of the front, with respect to the best known approximation Pareto optimal set $\mathcal{F}^*$, constituted by the front resulted from the combination of the 10 seeds approximation. The hypervolume measures the volume of objective space Y dominated ($\preceq$) by the considered approximate set. IGD is defined as the average Euclidean distance between each point in $\mathcal{F}^*$ and the closest point in $\mathcal{F}$. While still evaluating the convergence of the Pareto front, IGD is especially sensible to the presence of gaps in the Pareto front. We seek to maximize the value of the HV indicator, and minimize the value of IGD. For more details on their formulation refer to [36], [44] or the Supporting Information of this article.

Concerning the HV, its generational growth somewhat mirrors the search for the best irrigation solution, and by the end of the search, the worst solution covers over 90.5% of $\mathcal{F}^*$. IGD decreases even faster, with values that stabilize around the 500th generation. Evidently, the NFE assigned to the evolution appear more than sufficient to reach convergence, given that the HV or IGD indicators do not significantly improve for any seed in the second half of the search.

Additionally, we notice a contained intra-seed variability remarking a satisfying convergence and a limited dependency of NEMODPS solution on initial conditions. The maximum intra-seed variability (i.e., difference in performance between the best and the worst seed) is below 10% for all objectives and metrics considered, resulting in 5.19% for J^{irr}, 0% for J^{flood}, 9.37% in terms of HV and 7.14% for the IGD metric.

D. Many-objective application

In this section, we explore the potential of NEMODPS on a more challenging formulation of the Lake Como problem where we increase the number of objectives to four, by including the two additional objectives of navigation J^{nav} and environmental preservation J^{env}, yielding to a many-objective problem.

NEMODPS performance is again benchmarked against DPS with the 12 different prespecified policy architectures, namely ANN and GP architectures, each comprising 1 to 6 nodes

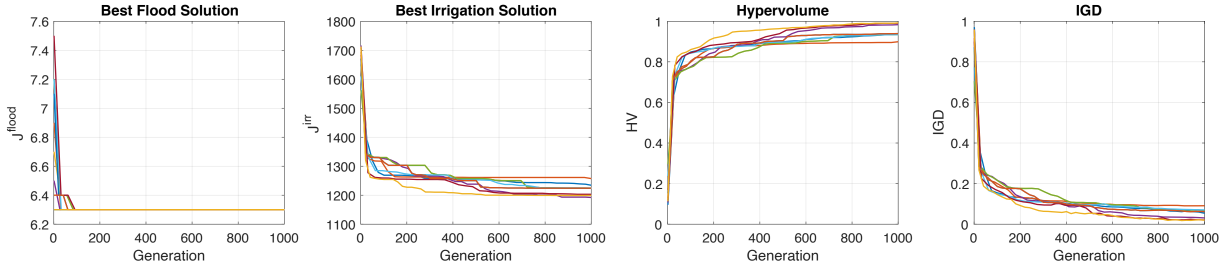

Fig. 3. Analysis of solution convergence with respect to multiple objectives. Each line represents the behavior of one of the 10 runs of NEMODPS. First and second panels report, respectively, the best value of the Flood and Irrigation objectives in the population, across the 1000 generations of the evolution. The third panel represents the value of the Hypervolume indicator during the evolution.

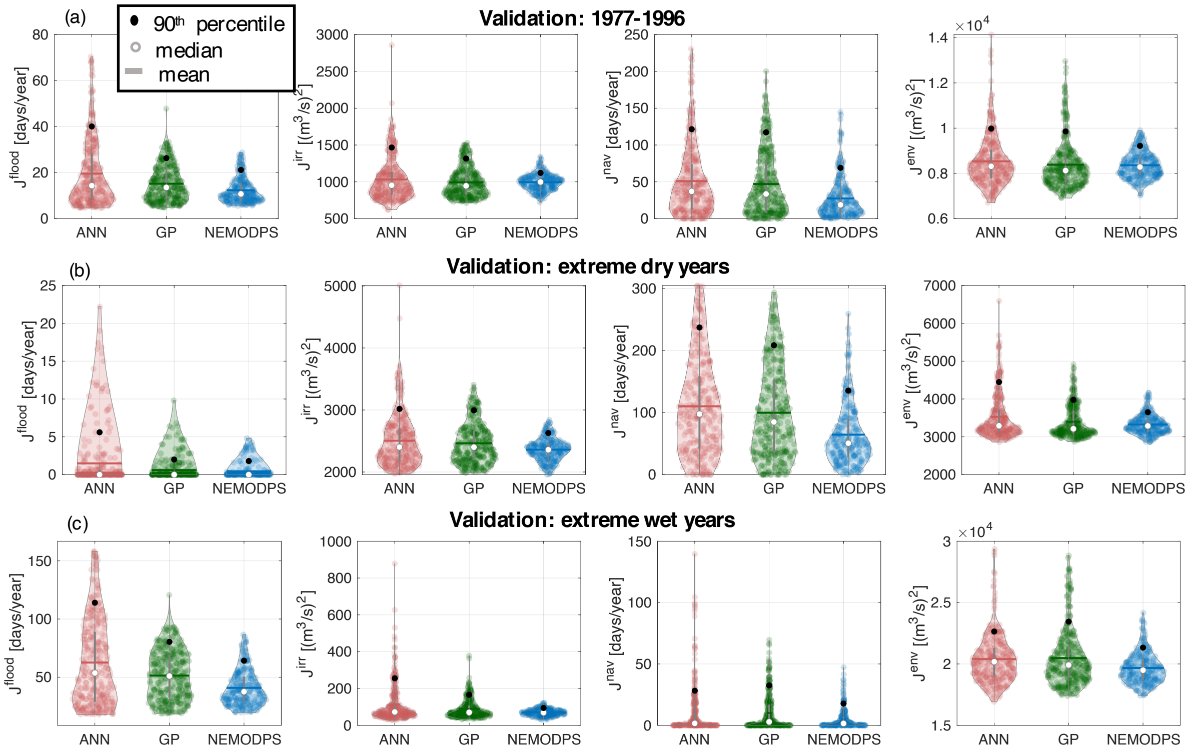

Fig. 4. Validation results for the many-objective formulation of the Lake Como problem. Each of the three panels corresponds to a different validation dataset, and is composed of 4 subpanels relative to the 4 optimization objectives, namely, from left to right, J^{flood}, J^{irr}, J^{nav}, J^{env}. The violin plot representation allows to compare the distribution of the validation performance of the generated policies and few relevant statistics, such as 90^{th} percentile, median, and mean.

in the single hidden layer. NEAT is not included in this benchmark experiment as generating a sufficiently characterized Pareto front for a many-objective problem with a SO algorithm would be computationally prohibitive. Calibration results are presented in Table I for the two DPS alternatives (ANN and GP), and NEMODPS, in terms of the two metrics of Hypervolume and IGD. Differently from the two-objectives optimization, NEMODPS policies dominate ANN and GP policies in the calibration experiment with respect to both indicators, thanks to their architectural flexibility.

TABLE I
MANY-OBJECTIVE OPTIMIZATION PERFORMANCE METRICS.

	ANN	GP	NEMODPS
Hypervolume	0.952	0.903	0.989
IGD	0.05	0.073	0.03

Efficient policies are then re-evaluated in a validation experiment that employs the same three validation datasets used for the 2-objectives optimization (Fig. 4), where efficient policies are re-evaluated on the three validation datasets used in the 2-objectives application of section V-A. Each validation panel is composed 4 subpanels relative to the 4 optimization objectives, namely, from left to right, J^{flood}, J^{irr}, J^{nav}, J^{env}. A violin plot is chosen to represent the distribution of performance of the re-evaluated policies. Compact distributions that lay in the lower portion of the axis are preferable, as they indicate that the policy search approach delivers a good and consistent performance for that objective across different solution tradeoffs.

On the contrary, expanded distributions with long tails indicate high variability in the objective's performance, and overall low reliability of the policy search method whose solutions may degrade significantly when tested on unseen conditions. Across validation experiments and objectives, NEMODPS consistently produces compact distributions concentrated in the lower portion of the axis, where objectives values are more desirable, indicating robustness against unseen conditions. Traditional DPS policies instead show long tailed distributions that extend in the upper region of the axis, with a more marked behavior displayed by ANN policy architectures. This confirms, on a many-objective setting, that DPS solutions have the potential to degrade significantly in validation with respect to one or more objective thereby producing high conflict between sectors. On top of the visual inspection of the distribution, this behaviour is quantifiable by comparing the statistics highlighted in the violin plot, namely the 90^{th} percentile (black dot), median (white dot), and mean (solid horizontal line). In risk neutral conditions (median and mean), and especially in risk averse conditions (90^{th} percentile), NEMODPS consistently ensures lower objective costs.

VI. Conclusions

In state-of-the-art applications of Direct Policy Search, the control policy is a priori defined as a fully-connected, single-layer, homogeneous neural network, independently from the problem characteristics or the objectives tradeoffs. This choice is motivated by theoretical results that assert the universal approximation capabilities of a wide range of network architectures. Many real-world applications, however, demonstrate a key role of topology in determining a network approximation skills and training requirements. Our results show that traditional DPS with such predefined policy topology is prone to overfitting in noisy environments, and does not offer enough flexibility in multi- and many-objectives problems, where different tradeoffs should be associated with different network architectures. By embedding NeuroEvolutionary (NE) techniques into the DPS framework, we extend the DPS problem to search a hyperspace containing control policy architectures and parameters. Yet, existing NE techniques, most notably NEAT and NEAT-inspired alternatives, are tailored to SO problems, and demonstrate a limited capacity to produce a high-quality approximation of the Pareto front in terms of solutions distribution and performance, while also requiring a substantially higher computational effort when compared to MO routines. This work contributes NEMODPS, a novel policy search algorithm which features the structure of the neuroevolutionary benchmark NEAT, several NEAT improvements proposed in the literature, and an original strategy to extend the routine to MO problems, exploring a multidimensional objective space in a single run of the algorithm. NEMODPS is a flexible framework that can support the exploration of alternative future research directions. One is to test its scalability to more complex control problem, for instance comprising multiple control decisions. New architectural and parametrical operators can be included in NEMODPS to enhance its effectiveness in exploring the architectural-parameteric hyperspace. Additional architectural operators can include new activation functions like ReLu, linear, or step functions, or the removal of existing nodes and connections. New parametrical operators can be included to target the investigation of a solution's proximity, or, alternatively, of unexplored regions of the parameter space (see, e.g., Reed et al., 2013). Numerical results show significant consistency in topological features of networks optimized across independent runs of NEMODPS, suggesting that the generated control policy architecture is rational and depends on the characteristics in the fitness landscape. Moreover, a change in objective tradeoff corresponds to a change in fitness landscape, and the Pareto-approximate topologies adjust accordingly. Finally, neuro-generated control policies demonstrate the ability to handle noisy environments featuring remarkable reliability, and generalization potential with respect to benchmark fixed-structure DPS solutions when tested on a suite of diverse validation experiments.

References

[1] C. Liu, X. Xu, and D. Hu, "Multiobjective reinforcement learning: A comprehensive overview," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 3, pp. 385–398, 2014.

[2] M. P. Deisenroth, G. Neumann, J. Peters *et al.*, "A survey on policy search for robotics," *Foundations and Trends® in Robotics*, vol. 2, no. 1–2, pp. 1–142, 2013.

[3] M. Giuliani, A. Castelletti, F. Pianosi, E. Mason, and P. M. Reed, "Curses, tradeoffs, and scalable management: Advancing evolutionary multiobjective direct policy search to improve water reservoir operations," *Journal of Water Resources Planning and Management*, vol. 142, no. 2, p. 04015050, 2015.

[4] V. Heidrich-Meisner and C. Igel, "Evolution strategies for direct policy search," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2008, pp. 428–437.

[5] M. Studley and L. Bull, "Using the xcs classifier system for multi-objective reinforcement learning problems," *Artificial Life*, vol. 13, no. 1, pp. 69–86, 2007.

[6] A. B. Celeste and M. Billib, "Evaluation of stochastic reservoir operation optimization models," *Advances in Water Resources*, vol. 32, no. 9, pp. 1429–1443, 2009.

[7] A. Rajeswaran, S. Ghotra, B. Ravindran, and S. Levine, "Epopt: Learning robust neural network policies using model ensembles," *arXiv preprint arXiv:1610.01283*, 2016.

[8] K. O. Stanley and R. Miikkulainen, "A taxonomy for artificial embryogeny," *Artif. Life*, vol. 9, no. 2, pp. 93–130, 2003.

[9] ——, "Efficient reinforcement learning through evolving neural network topologies," in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 2002, pp. 569–577.

[10] S. Risi and J. Togelius, "Neuroevolution in games: State of the art and open challenges," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 1, pp. 25–41, 2015.

[11] N. Kohl and R. Miikkulainen, "Evolving neural networks for fractured domains," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*. ACM, 2008, pp. 1405–1412.

[12] ——, "An integrated neuroevolutionary approach to reactive control and high-level strategy," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 4, pp. 472–488, 2012.

[13] S. Nichele, M. B. Ose, S. Risi, and G. Tufte, "Ca-neat: evolved compositional pattern producing networks for cellular automata morphogenesis and replication," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 687–700, 2018.

[14] M. Hausknecht, J. Lehman, R. Miikkulainen, and P. Stone, "A neuroevolution approach to general atari game playing," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 6, no. 4, pp. 355–366, 2014.

[15] S. Risi, C. E. Hughes, and K. O. Stanley, "Evolving plastic neural networks with novelty search," *Adaptive Behavior*, vol. 18, no. 6, pp. 470–491, 2010.

[16] J. Gauci and K. Stanley, "Generating large-scale neural networks through discovering geometric regularities," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, 2007, pp. 997–1004.

[17] P. Krčah, "Effects of speciation on evolution of neural networks in highly dynamic environments," in *International Conference on Learning and Intelligent Optimization*. Springer, 2012, pp. 425–430.

[18] M.-K. Jiau and S.-C. Huang, "Self-organizing neuroevolution for solving carpool service problem with dynamic capacity to alternate matches," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 4, pp. 1048–1060, 2018.

[19] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, "Real-time neuroevolution in the nero video game," *IEEE transactions on evolutionary computation*, vol. 9, no. 6, pp. 653–668, 2005.

[20] C. Rempis and F. Pasemann, "An interactively constrained neuro-evolution approach for behavior control of complex robots," in *Variants of Evolutionary Algorithms for Real-World Applications*. Springer, 2012, pp. 305–341.

[21] D. V. Vargas and J. Murata, "Spectrum-diverse neuroevolution with unified neural models," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 8, pp. 1759–1773, 2016.

[22] M. Giuliani, S. Galelli, and R. Soncini-Sessa, "A dimensionality reduction approach for many-objective markov decision processes: Application to a water reservoir operation problem," *Environmental Modelling & Software*, vol. 57, pp. 101–114, 2014.

[23] P. Vamplew, J. Yearwood, R. Dazeley, and A. Berry, "On the limitations of scalarisation for multi-objective reinforcement learning of pareto fronts," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2008, pp. 372–378.

[24] J. Schrum and R. Miikkulainen, "Constructing complex npc behavior via multi-objective neuroevolution." *AIIDE*, vol. 8, pp. 108–113, 2008.

[25] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *International conference on evolutionary multi-criterion optimization*. Springer, 2005, pp. 14–32.

[26] B. Lehner, C. R. Liermann, C. Revenga, C. Vörösmarty, B. Fekete, P. Crouzet, P. Döll, M. Endejan, K. Frenken, J. Magome *et al.*, "High-resolution mapping of the world's reservoirs and dams for sustainable river-flow management," *Frontiers in Ecology and the Environment*, vol. 9, no. 9, pp. 494–502, 2011.

[27] M. Franssen, "Arrow's theorem, multi-criteria decision problems and multi-attribute preferences in engineering design," *Research in engineering design*, vol. 16, no. 1-2, pp. 42–56, 2005.

[28] R. Soncini-Sessa, E. Weber, and A. Castelletti, *Integrated and participatory water resources management-theory*. Elsevier, 2007.

[29] A. Castelletti and R. Soncini-Sessa, "A procedural approach to strengthening integration and participation in water resource planning," *Environmental Modelling & Software*, vol. 21, no. 10, pp. 1455–1470, 2006.

[30] R. Zoppoli, M. Sanguineti, and T. Parisini, "Approximating networks and extended ritz method for the solution of functional optimization problems," *J. Optim. Theory Appl.*, vol. 112, no. 2, pp. 403–439, 2002.

[31] M. Baglietto, C. Cervellera, M. Sanguineti, and R. Zoppoli, "Management of water resource systems in the presence of uncertainties by nonlinear approximation techniques and deterministic sampling," *Computational Optimization and Applications*, vol. 47, no. 2, pp. 349–376, 2010.

[32] A. Castelletti, F. Pianosi, and M. Restelli, "A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run," *Water Resources Research*, vol. 49, no. 6, pp. 3476–3486, 2013.

[33] J. H. Metzen, M. Edgington, Y. Kassahun, and F. Kirchner, "Analysis of an evolutionary reinforcement learning method in a multiagent domain," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 291–298.

[34] M. Basirat and P. M. Roth, "The quest for the golden activation function," *arXiv preprint arXiv:1808.00783*, 2018.

[35] A. Hagg, M. Mensing, and A. Asteroth, "Evolving parsimonious networks by mixing activation functions," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2017, pp. 425–432.

[36] P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker, "Empirical evaluation methods for multiobjective reinforcement learning algorithms," *Machine learning*, vol. 84, no. 1-2, pp. 51–80, 2011.

[37] R. Bellman, "Dynamic programming (dp)," *Princeton University Press, Princeton, NJ*, 1957.

[38] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.

[39] J. N. Tsitsiklis and B. Van Roy, "Feature-based methods for large scale dynamic programming," *Machine Learning*, vol. 22, no. 1-3, pp. 59–94, 1996.

[40] A. Castelletti, F. Pianosi, and M. Restelli, "Multi-objective fitted q-iteration: Pareto frontier approximation in one single run," in *2011 International Conference on Networking, Sensing and Control*. IEEE, 2011, pp. 260–265.

[41] ——, "Tree-based fitted q-iteration for multi-objective markov decision problems," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2012, pp. 1–8.

[42] D. P. Bertsekas, "Reinforcement learning and optimal control," *Athena Scientific*, 2019.

[43] H. R. Maier, Z. Kapelan, J. Kasprzyk, J. Kollat, L. S. Matott, M. C. Cunha, G. C. Dandy, M. S. Gibbs, E. Keedwell, A. Marchi *et al.*, "Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions," *Environmental Modelling & Software*, vol. 62, pp. 271–299, 2014.

[44] J. S. Zatarain, P. M. Reed, J. D. Herman, M. Giuliani, and A. Castelletti, "A diagnostic assessment of evolutionary algorithms for multi-objective surface water reservoir control," *Advances in water resources*, vol. 92, pp. 172–185, 2016.

[45] P. M. Reed, D. Hadka, J. D. Herman, J. R. Kasprzyk, and J. B. Kollat, "Evolutionary multiobjective optimization in water resources: The past, present, and future," *Advances in water resources*, vol. 51, pp. 438–456, 2013.

[46] R. Busa-Fekete, B. Szörényi, P. Weng, W. Cheng, and E. Hüllermeier, "Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm," *Machine Learning*, vol. 97, no. 3, pp. 327–351, 2014.

[47] H. Xu and S. Jagannathan, "Neural network-based finite horizon stochastic optimal control design for nonlinear networked control systems," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 3, pp. 472–485, 2014.

[48] K. Hornik, "Multilayered feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[49] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[50] J. Liu, M. Gong, Q. Miao, X. Wang, and H. Li, "Structure learning for deep neural networks based on multiobjective optimization," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2450–2463, 2017.

[51] M. Bianchini and F. Scarselli, "On the complexity of neural network classifiers: A comparison between shallow and deep architectures," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 8, pp. 1553–1565, 2014.

[52] A. Laudani, G. M. Lozito, F. R. Fulginei, and A. Salvini, "On training efficiency and computational costs of a feed forward neural network: a review," *Computational intelligence and neuroscience*, vol. 2015, p. 83, 2015.

[53] S.-H. Jang, J.-W. Yoon, and S.-B. Cho, "Optimal strategy selection of non-player character on real time strategy game using a speciated evolutionary algorithm," in *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*. IEEE, 2009, pp. 75–79.

[54] M. Schoenauer, Ed., *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*, 2000.

[55] H. Moriguchi and S. Honiden, "Sustaining behavioral diversity in neat," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, 2010, pp. 611–618.

[56] J. Lehman and K. O. Stanley, "Revising the evolutionary computation abstraction: minimal criteria novelty search," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, 2010, pp. 103–110.

[57] A. Castelletti, S. Galelli, M. Restelli, and R. Soncini-Sessa, "Tree-based reinforcement learning for optimal water reservoir operation," *Water Resources Research*, vol. 46, no. 9, 2010.

[58] J. B. Kollat and P. M. Reed, "The value of online adaptive search: a performance comparison of nsgaii, ε-nsgaii and εmoea," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 386–398.

[59] E. Zitzler, M. Laumanns, L. Thiele, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, 2003.

[60] D. A. Van Veldhuizen and G. B. Lamont, “Evolutionary computation and convergence to a pareto front,” in *Late breaking papers at the genetic programming 1998 conference*. Citeseer, 1998, pp. 221–228.

Marta Zaniolo Marta Zaniolo received her PhD in Information Technology in the Environmental Intelligence Lab of Politecnico di Milano and is now a postdoc researcher in the Department of Civil and Environmental Engineering at Stanford University. In her research she fuses environmental, climate, and hydrologic disciplines, with machine learning, multi-objective optimal control, and evolutionary computation. She has experience in integrated water resources management in complex systems involving multiple actors and a changing climate and society, drought management and prediction, information theory for drought index design, and climate teleconnection analysis.

Matteo Giuliani Matteo Giuliani is assistant professor in the Environmental Intelligence Lab of Politecnico di Milano. The primary focus of his research is the integrated management of water resources in complex engineering systems involving multiple actors and exposed to evolving multisectoral demands and global change; his main research areas include multi-objective optimization and control algorithms, decision-making under uncertainty, machine learning and multi-agent systems. He is co-author of 51 publications in international journals, 24 conference proceedings, 2 book chapters, and more than 100 contributions to international conferences. He is Associate Editor of the ASCE Journal of Water Resources Planning and Management and of the ICE Water Management, and is currently member of the IFAC Technical Committee on Modelling and Control of Environmental Systems and of the ASCE/EWRI Environmental & Water Resources Systems Technical Committee.

Andrea Castelletti Andrea Castelletti is a full professor of Natural Resources Management and Environmental Systems Analysis at Politecnico di Milano, Italy, and a senior scientist at ETH Zurich. Dr. Castelletti research interest includes water systems planning and control under uncertainty and risk, decision-making for complex engineering systems, big environmental data analytics and smart sensing, information theory and selection for environmental decision making. Dr. Castelletti is co-author of two international books on integrated water resources management, and more than 150 publications in international journals, book chapters and conference proceedings. He is Associate Editor of Water Resources Research, the Journal of Hydrology, and Environmental Modelling and Software.