DOI 10.4430/bgo00411

Inversion of geophysical data supported by Reinforcement Learning

P. Dell'Aversana

Eni S.p.A., San Donato Milanese (MI), Italy

(Received: 6 September 2022; accepted: 8 November 2022; published online: 7 December 2022)

ABSTRACT Local optimisation algorithms allow exploring a given region of the search space and getting close to, or finding exactly, the extrema of the function in that region (local optima). However, we are more frequently interested in finding global optima of a given objective function. For this purpose, there are many global, or quasi-global, exploration strategies that allow performing an expanded exploration of the model space. Unfortunately, their effectiveness is counterbalanced by their slow convergence towards the solution and, consequently, by long or prohibitive computation times. In order to improve their effectiveness, it is possible to combine global optimisation techniques with local search methods. Based on such hybrid approach, in this paper, we introduce a novel geophysical inversion strategy aimed at optimising the exploration of the model space with the support of Reinforcement Learning (RL) techniques. These include a suite of Machine Learning methods concerned with how intelligent agents ought to take actions by interacting with their environment with the objective to maximise the notion of Cumulative Reward (CR). Following the Bellman equation, this is given by the contribution of a Short-Term and a Long-Term reward, balanced by a trade-off term called discount factor. Adopting these RL concepts, our goal is to teach an artificial agent to search for the global optimum of the cost function, limiting the risk of being trapped in local optima. This can be done by relating the CR to various indicators of the inversion performance, including the trend of the objective function over a limited number of iterations. Using multiple inversion tests on synthetic as well as real geo-electric data, we verified that our approach fits the purpose of expanding the exploration of the model space, finding optimal solutions (global optima) even in complex 3D inversion scenarios.

Key words: geophysical inversion, optimisation problems, Reinforcement Learning, Q-Learning, Model space exploration, Epsilon-Greedy.

1. Introduction

Inversion of geophysical data is generally an ill-conditioned and ill-posed problem (Tarantola, 2005), with solutions that depend closely on data uncertainties and model space parameterisation. This means that the data have extremely variable sensitivity to the model parameters and that small error bars on the data produce major variations in the inverted models. For instance, solution instability and non-uniqueness represent frequent issues when dealing with electric, electromagnetic, magnetic, and gravity data inversion. The same problems can arise in seismic data inversion too, such as in many cases of reflection/refraction tomography. When prior reliable information is not available, selecting one or more representative models among the

ensemble produced by the inversion is often difficult or even arbitrary. Furthermore, inverted models can be very different depending on the initial guess model and the inversion hyper-parameters setting. For instance, using different regularisation operators, or different inversion meshes, can produce significantly variable results.

A different, but related, open question is how to avoid the cost function converging towards one or more local optima. These represent extrema (minima or maxima) of the objective function in a restricted region of the model space. Instead, a global optimum is the extremum of the objective function over the entire model space. When possible, depending on the size of data and model space, global optimisation methods can be applied for finding the global minimum (or maximum) of the objective function. Well-known global optimisation techniques include the Annealing-Simplex method (Pan and Wu, 1998), Genetic algorithms (Ines and Droogers, 2002), Ant Colony Optimisation (Abbaspour *et al*., 2001), Shuffled Complex methods (Duan *et al*., 1993; Vrugt *et al*., 2003), Particle Swarm Optimisation (Brunetti *et al*., 2016, 2018), and many others. All these methods have benefits and limitations. By definition, global exploration strategies allow performing an expanded exploration of the model space, but their effectiveness is counterbalanced by their slow convergence towards the solution and, consequently, by long or prohibitive computation times. In order to improve their effectiveness, several authors have coupled global optimisation techniques with local search methods. For instance, Noel (2012) proposed a method that combines Particle Swarm Optimisation (Eberhart and Kennedy, 1995) with a steepest descent approach. This is applied periodically to sample the model space in a restricted area around the swarm's best solution.

Based on a similar idea of coupling global exploration techniques with traditional local search methods, in this paper, we propose a geophysical inversion strategy aimed at optimising the exploration of the model space and the entire inversion workflow. The novelty of our approach is that it takes advantage of Reinforcement Learning (RL) algorithms (Littman, 1994; Lample and Chaplot, 2017; Nagabandi *et al*., 2018; Ravichandiran, 2020). We have already introduced our approach in recent papers and conferences (Dell'Aversana, 2022a, 2022b), but in this new article we intend to further clarify the details of our approach and to test it on new data. Two strictly related objectives motivated and addressed our research: first, to optimise the exploration of the model space; second, to reduce the risk that the inverted solution is trapped in one or more local minima of the cost function. Our basic idea is that RL techniques can support the exploration of the model space as well as the setting of the inversion hyper-parameters. We recall that RL includes a suite of Machine Learning methods concerned with how intelligent agents ought to take actions by interacting with their environment with the objective to maximise the notion of Cumulative Reward (CR). As we shall see in detail in the methodological section, this reward can be defined as a balanced combination of an immediate reward plus a long-term reward. The environment of an RL problem can be a physical space as well as a virtual, mathematical space. For instance, it is a physical space when we want to teach a robot to optimise its actions in a real 3D space. Instead, we deal with a virtual environment when we desire to optimise the actions of an artificial agent in a *n*-dimensional virtual space, like a decision space or the model space of an inverse problem. The value of the reward received by the agent while interacting with its environment depends on the 'quality' of the agent's actions. High rewards correspond with the positive impact of the actions on the agent's target, and vice versa.

Our approach aims to take advantage of the above-mentioned RL concepts in order to define an effective strategy, or policy, for optimising the exploration of the model space in geophysical inverse problems. Our objective is to teach an 'artificial agent' to search for the global minimum of the cost function, avoiding being trapped in local minima. This can be done by relating the

RL concept of CR to various indicators of the inversion performance, including the trend of the objective function. When the inversion produces good results in terms of significant reduction of the objective function, we assign a high reward to our RL agent, and vice versa. In such a way, we assume that the agent can learn, after a sufficiently high number of iterations, how to move in the model space in an optimal way (that is towards some minima of the cost function). Furthermore, in order to avoid getting trapped in local minima, we go beyond the Greedy approach (Cormen *et al*., 2001), which is based solely on reduction of the objective function. In fact, it is well-known that every Greedy algorithm applies the problem-solving heuristic of making the locally optimal choice at each stage. Consequently, in many inversion and optimisation problems, a Greedy strategy is not able to produce an optimal solution. Instead, using an Epsilon-Greedy approach (Sutton and Barto, 1998), we do not neglect exploring the model space through alternative directions where it is possible to find lower minima of the objective function. This can happen especially in those inverse problems where the topology of the cost function is complex and includes many basins, hills, and local optima.

In the next methodological section, we will elaborate on all these introductive concepts. We start by recalling the basics of the RL problem and its related methods. Next, we continue by explaining how every geophysical inverse problem can be reformulated in terms of RL strategy. To this end, we will use a combination of the abovementioned Epsilon-Greedy method and the Q-Learning algorithm (Sutton and Barto, 1998), that is a specific RL approach. In particular, we will use the Bellman equation (Jones and Peet, 2021) for relating, in a simple and pragmatic way, the RL CR with the misfit between observed and predicted responses over a limited range of inversion runs. We will see that our approach fits the purpose of optimising the exploration of the parameter-space in inversion problems, not necessarily restricted to the geophysical field. Finally, we will test our workflow using synthetic as well as real 3D geoelectric data.

2. Methodology

In this section, we summarise the key concepts of RL, Q-Learning, and Epsilon-Greedy methods, as they represent fundamental steps of our workflow. Additional details on these techniques and methods are widely available in the scientific literature, consequently we prefer to focus the discussion on the novel concepts introduced in this paper [an extensive and detailed discussion about Deep RL can be found, among the others, in Ravichandiran (2020)].

2.1. Reinforcement Learning, Q-Learning and Epsilon-Greedy

RL includes a suite of algorithms and techniques through which an artificial agent learns an optimal 'policy' through continuous interaction with its environment. The objective of the agent is to maximise a reward metric for the task, without being explicitly programmed for that task. The artificial agent selects by trial and error the actions that improve the CR, $r \in$ R, achievable from a given state, $s \in$ S. The agent's goal is learning a policy, π, that maximises the total (or cumulative) reward. Such a policy depends on the current environment state, s, belonging to the set S of all possible states, and returns an action, a, belonging to the set A of all possible actions:

$$\pi(s) : S \rightarrow A. \tag{1}$$

In this paper, we focus on the Q-Learning method because we assume that it is particularly suited for tackling optimisation/inverse problems. Such a choice will appear clearer after explaining the entire workflow. However, other RL techniques can be used for the same purpose. The name of the Q-Learning method derives from the Q-function that provides a measure of the Quality, *Q*, of an action that the agent takes starting from a certain state:

$$Q(s,a) = S \times A \rightarrow R. \tag{2}$$

We have already anticipated that, in RL approaches, the agent's objective is to maximise a CR, rather than just the immediate reward associate with its actions. Such a CR is formally defined by Bellman's equation reported below. This is given by the reward *r* that the agent received for entering the current state *s* and action *a*, plus the maximum future reward for the next state *s'*, taking all the possible actions *a*' from that state:

$$Q(s,a) = r + \gamma\, max_{a'}\, Q(s',a'). \tag{3}$$

The 'discount factor', *γ*, allows balancing the trade-off between immediate and future rewards. The values of *Q*(*s, a*) are updated using a recursive procedure. We start by assuming an initial Q-Table consisting of hypothetical vales of *Q* for each state and for each action of the agent. These values are commonly set to zero or are initially set as random values. Next, the agent proceeds by interacting with its environment, and the initial guess *Q* values are progressively changed taking into account the positive and/or negative CRs that the agent receives from its environment.

The 'Temporal Difference' method (Eq. 4 below) provides a practical way for updating the *Q* values, as follows:

$$Q^{new}(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \cdot \left[r_t + \gamma \cdot \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right]. \tag{4}$$

We can see that the new value of *Q* for state s_t and action a_t, is obtained by adding to the previous Q value a new term (in the square brackets) called 'temporal difference'. This, in turn, is multiplied by a factor *α* that represents the 'learning rate' and is commonly determined empirically by the user. The temporal difference consists of the immediate reward, r_t, plus the difference between the maximum *Q* value for all the actions that the agent can take from the state s_{t+1}, minus the old value of *Q*. The $\max_a Q(s_{t+1}, a)$ term is multiplied by the above mentioned discount factor, *γ*.

As noted, another fundamental aspect of our inversion workflow is to explore the model space using an Epsilon-Greedy strategy. This is a well-known approach aimed at solving the dilemma between exploration and exploitation (Sutton and Barto, 1998; Ravichandiran, 2020). Such a dilemma is crucial in geophysical inverse problems (and, more generally, in optimisation problems) where we need to define an optimal policy for moving in the model-space and finding global minima, rather than local minima. If we decide to use a Greedy approach (for instance, applying an optimisation method based on steepest gradient descent), we risk excluding a priori other portions of the model space that could include more promising solutions. Using the Epsilon-Greedy approach, we allow the optimisation/inversion algorithm exploring the model space (with low probability: epsilon << 1) in alternative directions too, even if that choice could imply a temporary increase of the cost function.

2.2. Inversion supported by Reinforcement Learning (RL-Inv)

At this point, we need to clarify how we combine the Epsilon-Greedy method with the Q-Learning concepts and the Bellman formula, with the final goal to improve the inversion workflow. The scheme of Fig. 1 represents an extreme simplification of our approach, showing that it consists of two cooperating loops, linked with each other.

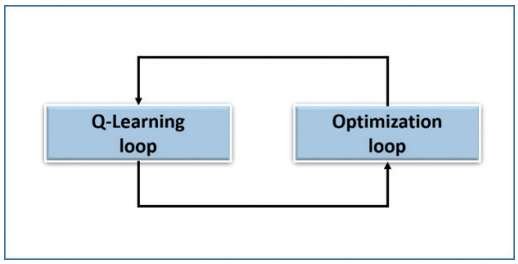

Fig. 1 - General scheme of inversion supported by RL (called briefly RL-Inv).

Let us expand on the left block of Fig. 1, as shown in Fig. 2.

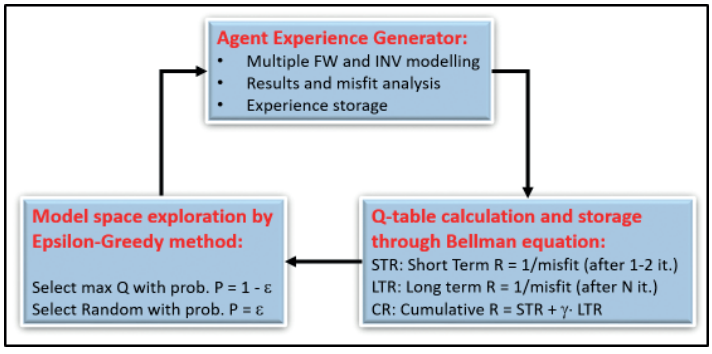

Fig. 2 - The main blocks of the Q-Learning loop.

The first block (on the top) of the Q-Learning loop consists of an 'Agent Experience Generator'. The agent is the optimisation algorithm that we are using (in the right block of Fig. 1) for minimising the misfit between observed and predicted responses. In the examples discussed in this paper (see next section dedicated to demonstrative applications), we apply a Least Square optimisation algorithm (as well as its variations, like Damped Least Square) to solve our inverse problems. In Least Square optimisation, the cost function $\Phi(\boldsymbol{m})$ *is*:

$$\Phi(\boldsymbol{m}) = [\boldsymbol{d}_{obs} - g(\boldsymbol{m})]^T \boldsymbol{W}_d [\boldsymbol{d}_{obs} - g(\boldsymbol{m})] + \eta \cdot \boldsymbol{m}^T \boldsymbol{R}\boldsymbol{m}, \quad (5)$$

where $\boldsymbol{m}$ represents the vector of model parameters, or model vector; $\boldsymbol{d}_{obs}$ represents the data vector (observations); $g(\boldsymbol{m})$ is the forward operator by which we calculate the predicted response in the model vector $\boldsymbol{m}$; the symbol T indicates 'transpose'; $\boldsymbol{W}_d$ is he data covariance matrix for

taking data uncertainties into account; ***R*** is a smoothing operator applied to the model vector ***m*** as a regularisation term; η is a factor regulating the weight of the smoothing term in the cost function.

The second block (bottom-right block in Fig. 2) receives in input the results of the Agent Experience Generator. This feeds and updates iteratively a Q-Table. This is the table of the Q values calculated through the Bellman equation using the inverse of the values of the cost function, as explained below. In our procedure, we calculate $\Phi(\boldsymbol{m})$ *and the RMS*(%) (Root Mean Square misfit) at each iteration and store their values. In such a way, we can calculate and store the building terms of the *Q* value as follows:

$$Q(s_t, a_t) \approx {}^{1}\!/_{\Phi(\boldsymbol{m})}, \text{ or} \tag{6}$$

$$Q(s_t, a_t) \approx {}^{1}\!/_{RMS\ (\%)}. \tag{7}$$

The Bellman formula (Eq. 3) includes both a Short-Term Reward (STR) as well as a Long-Term Reward (LTR). We can consider STR proportional to the inverse value of the cost function estimated after just one or two iterations, as in Eq. 6 or, equivalently, Eq. 7. Instead, LTR is proportional to the inverse of the cost function (or RMS) estimated after a significant number, *N*, of iterations (such number *N* depends on the inverse problem and is decided by the user, case by case). In such a way, our goal is to set a policy that minimises the cost function through a balanced combination of both short-term and-long term views. This approach is summarised in the bottom-right block of Fig. 2, where the total Q function is represented by the CR. This, in turn, consists of the sum of the STR and the LTR, balanced by a trade-off parameter, γ (the discount factor). Such a CR is calculated at each inversion run performed in the optimisation loop. The idea behind this approach is that we assign a CR at each optimisation trial that depends on the objective function trend or, more simply, on the RMS misfit at different iterations of each inversion run. In fact, the RMS (as well as the value of the objective function) after 1 or 2 iterations depends mainly on the starting guess that we are using for triggering the inversion itself. If the RMS is high (low), it means that the starting model is far from (close to) the true one. Consequently, the STR must be low (high). Instead, the RMS after a sufficiently large number of iterations, *N*, represents the effectiveness of the optimisation algorithm and of its hyper-parameters. High (low) RMS after *N* iterations will provide low (high) LTRs. In plain terms, the second block of Fig. 2 allows us to reformulate the RMS (and the objective function) in terms of Q-function and Q-Table, simply by using the Bellman formula. This step is useful for the next step, where we use the Q-Table for applying the Epsilon-Greedy method.

The third block (bottom-left block in Fig. 2) is aimed at exploring the model space through the Epsilon-Greedy approach. First, we choose a value for ε ($0<\varepsilon<1$), that we can define as 'exploration probability'. Next, we randomly select a value p ($0<p<1$). If $p>\varepsilon$, then we select the model (from the agent's memory) with the highest Q value (exploitation). Otherwise, if $p\leq\varepsilon$, then we pick a model at random (exploration). Following that procedure, the agent (the optimisation algorithm) updates the model trying to reduce the cost function at each iteration (exploitation) with probability $p>\varepsilon$. However, it explores the model space (with lower probability: $p\leq\varepsilon$) in different directions (exploration). This strategy allows limiting the risk of the cost function being trapped in relative minima. At the same time, it is based on the Q-Table defined through the Bellman formula as explained earlier.

After the model has been selected in block three, it is randomly perturbed many times in order to create a new ensemble of starting models. These are used for activating again the block 1, by triggering a new cycle of inversion runs through the Agent Experience Generator. These new inversions will update the agent's memory with new inverted models. Furthermore, the Q-Table will be updated, and the cycle continues until the Q-Table becomes stationary (no further improvements). As shown in Fig. 1, the Q-Learning loop is chained with the optimisation loop. For sake of simplicity, the two loops are represented as two separate blocks, but it is more appropriate to consider the optimisation loop interconnected with (or even embedded into) the Q-Learning loop. In fact, in the frame of the Reinforcement Q-Learning approach, the optimisation loop corresponds with the agent that continuously interacts with its environment, which is the model space. Such an agent can be any type of local optimiser aimed at reducing the misfit between observed and predicted responses. In the illustrative examples discussed in the next section, we used a (Damped) Least Square algorithm; however, our RL-Inv approach can combine Q-Learning with other types of optimisation algorithm.

3. Synthetic tests

The first demonstrative example concerns modelling and inversion of synthetic 2D resistivity data. This test is apparently simple and is based on a schematic model in order to clarify the workflow through its main steps. We simulated an acquisition test along a line of 550 m, with electrode spacing of 10 m, using a standard Wenner-Schlumberger geo-electrical array (Telford *et al.*, 1990). Fig. 3 shows the data pseudo-section in terms of apparent resistivity (upper panel) and the 'true model' in terms of true resistivity distribution (lower panel).

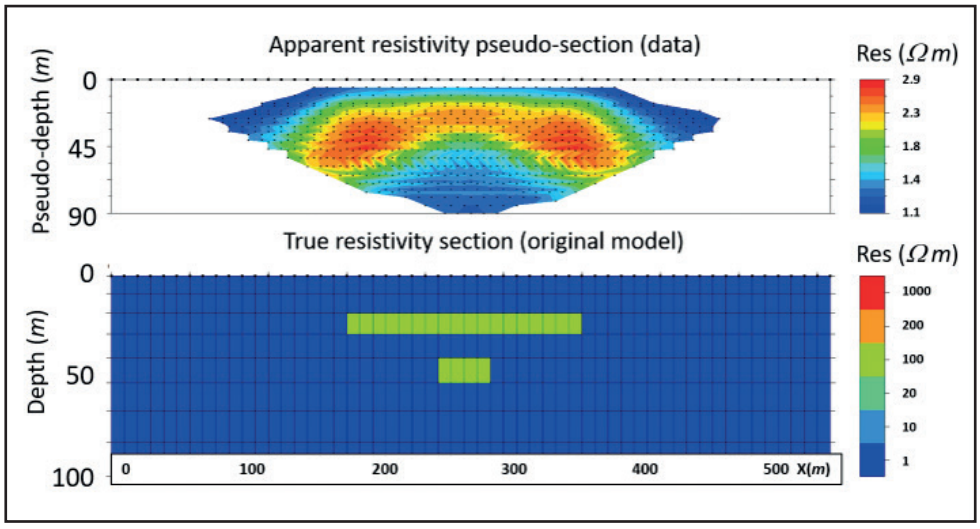

Fig. 3 - Input synthetic apparent resistivity pseudo-section (upper panel) and 'true' resistivity model (lower panel).

First, we performed a sequence of inversion runs using a standard Damped Least Square algorithm, a suite of variable starting models (uniform half spaces) and variable inversion parameters (damping factor, smoothing factor, mesh parameters size, etc.). As expected, despite

its apparent simplicity, retrieving the original model shown in Fig. 3 represents a challenging geophysical inverse problem. The upper resistivity layer tends to mask the shorter layer below, and the smoothing operator of the inversion algorithm tends to create a unique thick resistivity layer rather than two distinct resistors.

Fig. 4 shows the best inversion result (in terms of RMS) that we were able to obtain after many trials and without assuming any prior knowledge about the original resistivity model. Looking at the misfit cross-plot, we can observe two distinct clusters of points. These indicate a double over-estimation of the retrieved resistivity model, caused by the fact that the inverted solution includes a unique thick resistive layer instead of two thinner stacked resistors embedded in a conductive background.

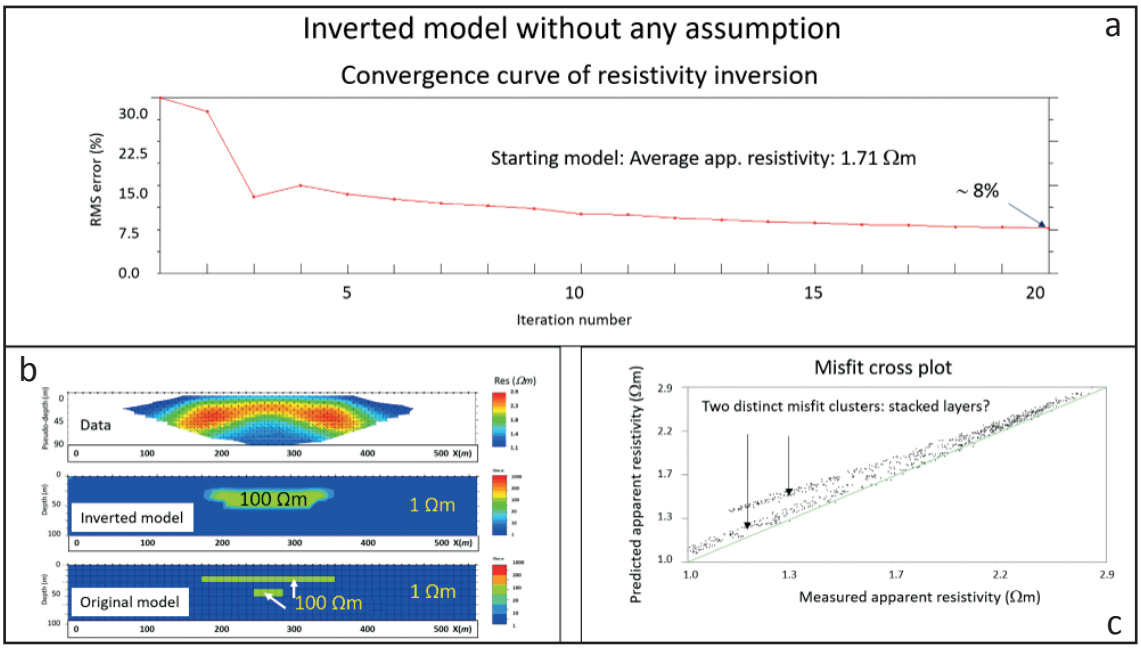

Fig. 4 - Best result of Damped Least Square inversion: a) RMS convergence curve; b) data, inverted model and original model; c) misfit cross-plot (predicted vs. measured apparent resistivity).

As an alternative inversion strategy, we applied our RL-Inv workflow described in the methodological part. In this particular inversion test, we followed a progressive RL-Inv workflow (Fig. 5) aimed at defining the different parts of the resistivity models with an increasing level of complexity. First, we defined the background, and then focused on the definition of number, geometry and resistivity of the layers/anomalies.

In order to expand the exploration of the model space, we applied the Agent Experience Generator. This represents the first step of the RL-Inv approach, as shown in Fig. 2. We recall that the Agent Experience Generator is a procedure (triggered by a batch file) aimed at producing and storing a large set of inverted results by launching automatically a sequence of many inversion runs, using different starting models and setting variable inversion hyper-parameters. Next, using the results of the Experience Generator, we calculated the CR for each inversion run by using the Bellman formula, as explained earlier. As an illustrative example, Fig. 6 shows a histogram of normalised Q-values estimated for an ensemble of 20 different models. In this specific case, the

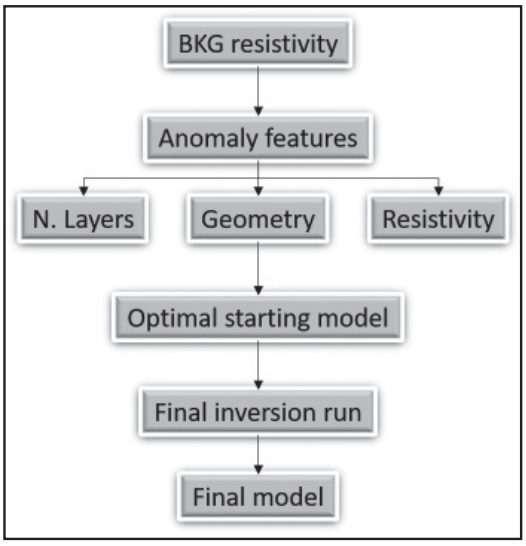

Fig. 5 - RL-Inv progressive inversion scheme.

model N.9 corresponds to the inversion result obtained using an optimal starting guess (half-space at 1.71 Wm).

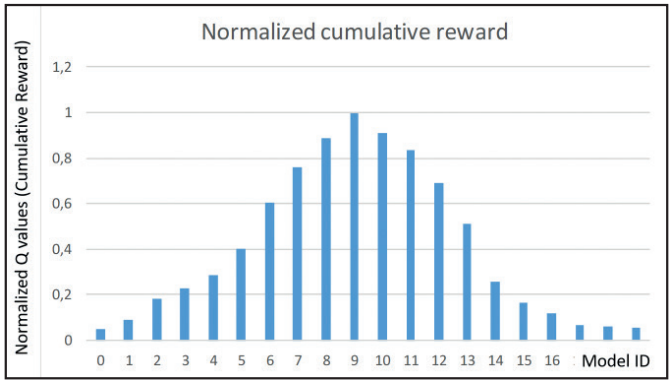

Fig. 6 - Normalised CRs (Q-values) for a sample of twenty inverted models.

The Q values for each inversion result were stored in order to apply the Epsilon-Greedy strategy. It is worth recalling that this approach does not necessarily select the model with the temporary highest Q-value, like the model N.9 shown in the example of Fig. 5 (exploitation). In fact, the Epsilon-Greedy method is designed for exploring, with lower probability than exploitation, alternative portions of the model space. Such a probability depends on the epsilon parameter. Thus, we can explore or exploit the model space with extreme variability, depending on how we set such ε parameter. A rule of thumb is to set ε = 0.1 or 0.2, in order to guarantee a probability of 10-20% of exploration of alternative directions in the model space (explorative policy) and 80-90% of exploitation (Greedy policy). Another approach is to progressively increase the value of this parameter until the objective functional becomes stationary.

Next, the model selected through the Epsilon-Greedy strategy is randomly perturbed many times in order to trigger a new sequence of inversion runs. Such a sequence is automatised using

again the Agent Experience Generator. In this way, the inversion workflow continues through a balanced trade-off between exploitation and exploration of the parameters space. Fig. 7 shows the final results obtained through this workflow, together with the RMS convergence curve of the final inversion run and its relative misfit cross-plot.

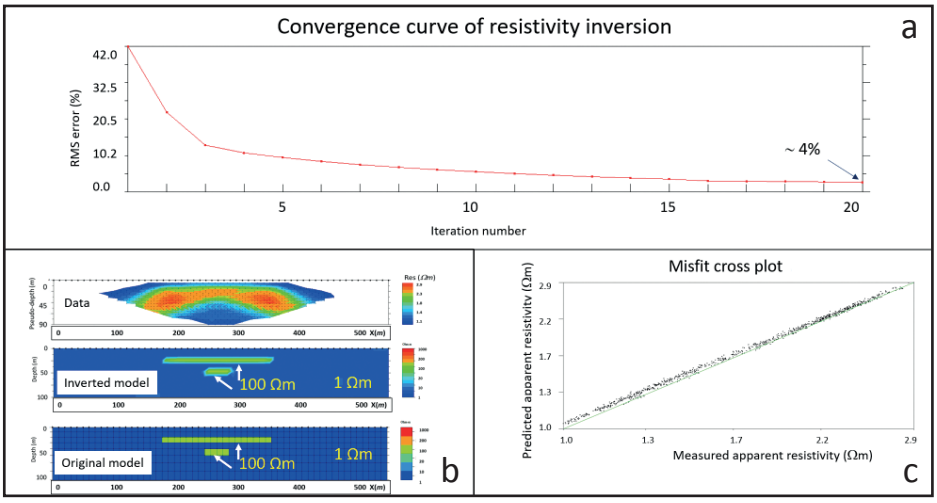

Fig. 7 - Result of RL-Inv inversion strategy: a) RMS convergence curve; b) data, inverted model and original model; c) misfit cross-plot.

Comparing the inverted model with the original model (bottom-left panel of Fig. 7), we can see that the inverted solution is very close to the true resistivity distribution. Furthermore, the misfit cross-plot shows a unique cluster of points properly distributed along the 45° line of the predicted vs. measured apparent resistivity values.

4. Inversion of real data

We applied the RL-Inv approach to a real data set of DC (Direct Current) laboratory measurements. This experiment was addressed to verify, through a small scale and fully controlled test, the sensitivity of cross-well geoelectric data to map reservoir fluids displacement over time (Dell'Aversana *et al*., 2017). For that purpose, we installed electrodes on several pipes (simulating vertical wells) in a plexiglas sand box in which we created a layered sedimentary sequence. We intended to reproduce a realistic scenario of an oil field during production monitored by 3D DC cross-well tomography. Fig. 8 shows two pictures our experimental system and, on the left side, the schematic cartoon of three phases of the production test. The pictures show the installation of the wells equipped with 120 electrodes in our experimental sandbox. The simulated oil-filled reservoir consists of a coarse sand layer embedded in a homogeneous fine sand background, saturated by salty water. The reservoir is sealed by a thin layer of clay (thickness = 3 cm). The top layer consists of a homogeneous sand formation just above the clay layer. We used a dipole-dipole geoelectric acquisition layout (Telford *et al.*, 1990), recording single-well as well as cross-well measurements during different stages of oil production. We performed the production phases using a small tube crossing the reservoir layer and pumping

oil, taking care to preserve the hydrostatic reservoir pressure. For each set of measurements, we recorded about 2200 values of electrical potentials.

We used part of this data set to verify the effectiveness of our RL-Inv approach, motivated by the fact that this experiment was controlled through accurate measurements of the fluids (water and oil) displacement over time. In such a way, we were able to compare directly the physical state of our experimental system with the results of the resistivity inversion for each experimental phase.

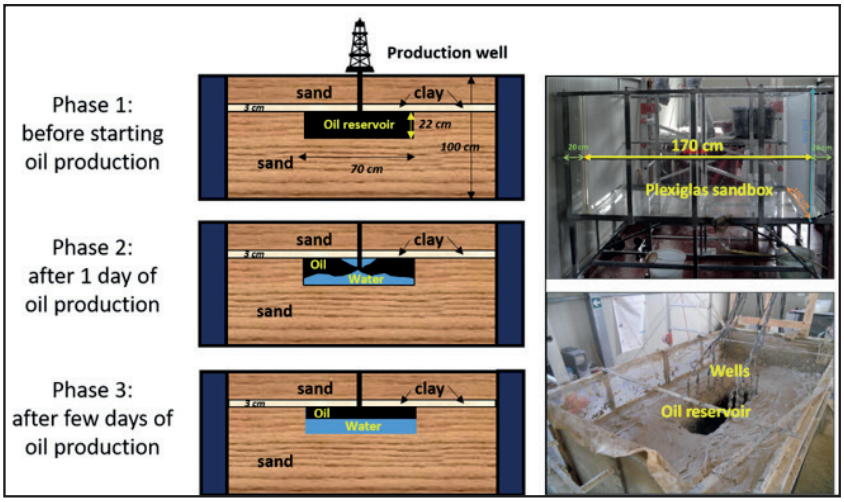

Fig. 8 - Scheme of three phases of the production experiment (left panels); two illustrative pictures of the laboratory experimental setup (right panels) (after Dell'Aversana *et al*., 2017, modified).

Following the RL-Inv workflow shown in Fig. 2, we applied the Agent Experience Generator for running automatically a long sequence of 3D inversions with variable starting models and hyper-parameters. As noted in the synthetic test discussed in the previous section, the inversion

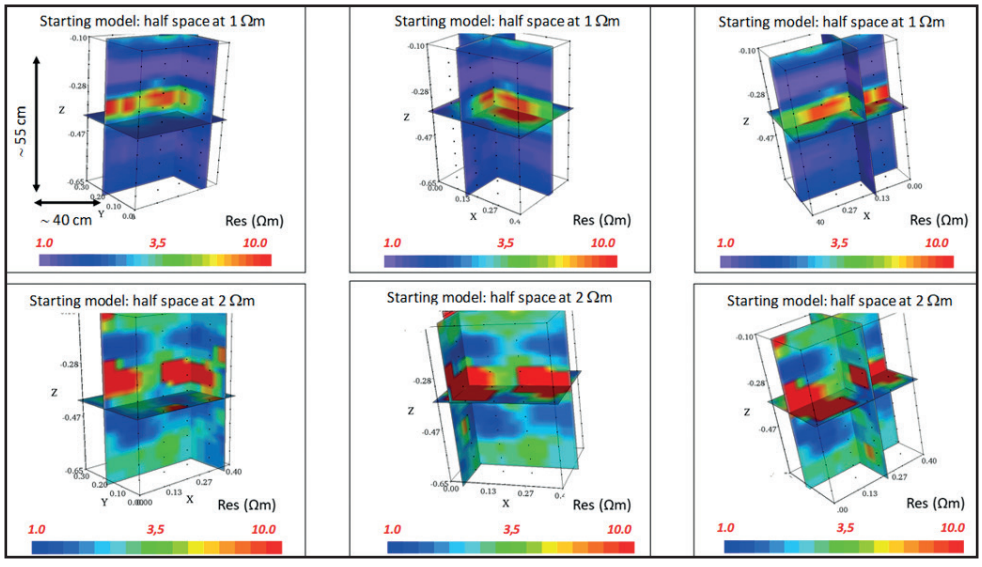

Fig. 9 - Three different views of the inversion results obtained by using two different starting models: half space of 1 Ωm (upper panels) and half space of 2 Ωm (lower panels).

results are sensitive to the starting models as well as to the inversion hyper-parameters. For instance, Fig. 9 shows the inverted 3D results obtained by using two different starting models. In this case, we show the inversion of data acquired when oil saturated the entire volume of the reservoir formation, creating a high-resistivity layer embedded in a more conductive background. We applied a 3D Damped Least Square inversion scheme in this case too.

Fig. 10 shows the misfit curves and the apparent resistivity cross-plots for four different inversion runs (with different starting models, but with the same inversion setting). In the same figure, the CR is reported for each inversion.

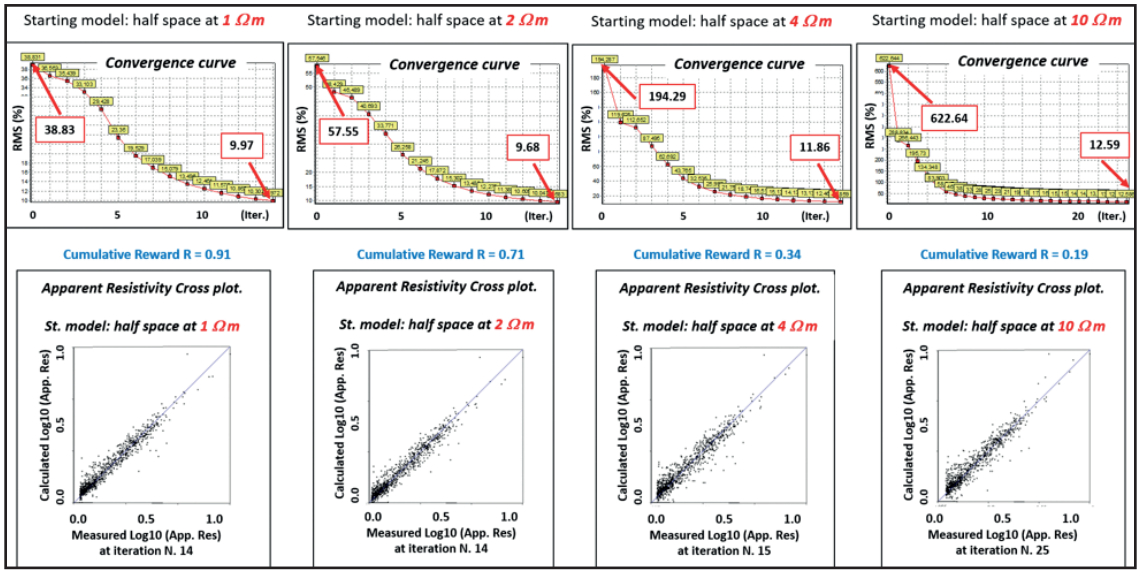

Fig. 10 - Misfit results and CR, for four different inversion runs.

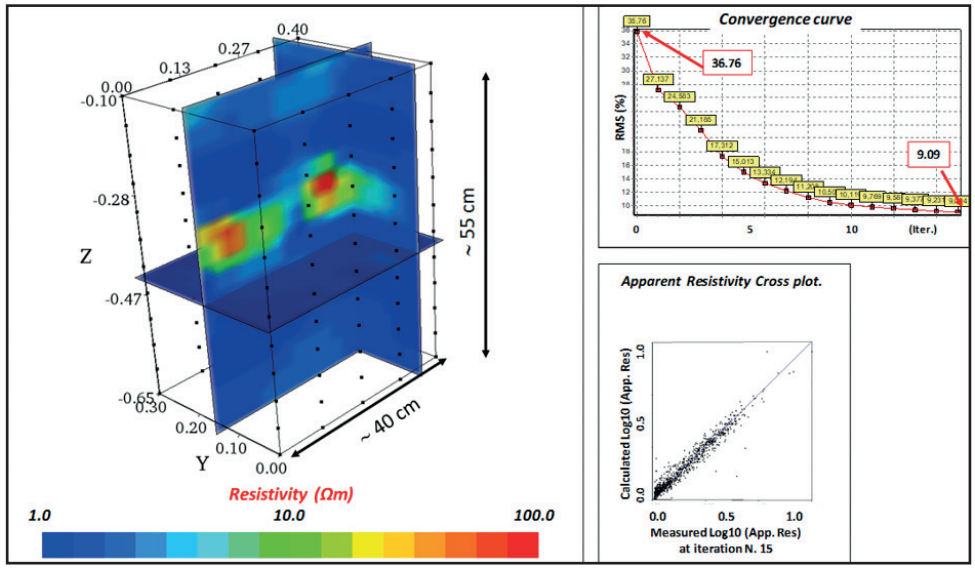

Fig. 11 - Final 3D resistivity model.

After collecting a large volume of inversion results with their associated CRs (Q-values), we used the Epsilon-Greedy method for selecting the model for the next step of the RL-Inv scheme. That model was used for generating new starting guesses aimed at triggering a new sequence of inversions. Then, we updated the Q-Table using the Bellman formula and iterated that procedure many times. Finally, we exited from the Q-Learning loop when the Q-Table remained stationary. Fig. 11 shows the final inverted model, together with its final RMS convergence curve, and its misfit cross-plot.

5. Discussion: benefits and limitations of RL-Inv

The RL-Inv approach is a methodology that combines complementary algorithms and methods for improving the exploration of the model space in optimisation problems, including geophysical inverse problems, with the support of RL methods. An interesting question is if and why RL-Inv adds something new with respect to any other inversion/optimisation scheme based on the progressive minimisation of the objective function, or of RMS. Effectively, we use a similar criterion for updating the models, because the Q-values are linearly dependent on the RMS misfit calculated during the optimisation process. The first difference with respect to other consolidated approaches is that such a linear combination of RMS is a simple although not trivial optimisation criterion. In fact, it allows expressing the objective function in terms of CR. It is important to remark again how such a CR is calculated. This is a balanced combination of STR and LTR. The first term is inversely proportional to the RMS value after 1 or 2 iterations. This allows linking the reward directly to the choice of the starting model. Instead, the second term is inversely proportional to the RMS value after a significant number N of iterations (for instance, in many of the tests discussed in this paper, $N = 8$). This allows linking the reward to the choice of the inversion hyper-parameters (regularisation operator, smoothing factor, etc.) as well as to the exploration of the model space. The Bellman formula (Eq. 3) provides a simple way to take into account all these factors in the same Q-function. Furthermore, we can use the discount factor γ as a trade-off parameter, for deciding which element is more determinant towards improving the inversion performances.

An additional motivation supporting the RL-Inv approach is the following: after having reformulated the inversion problem in terms of an RL problem, we can use a large suite of RL algorithms to find global solutions for many types of optimisation problems. For instance, we can couple Deep Neural Networks with the Q-Learning method to find global solutions to inverse problems even in the case of large and complex model spaces. This approach is commonly known as the 'Deep Q-Learning method' (Mnih *et al.*, 2015; Ravichandiran, 2020), and it has been widely and successfully applied for solving complex optimisation problems in other fields apart from geophysics.

Furthermore, the RL-Inv approach can involve many criteria for defining the reward metric in addition to the RMS values. All these criteria concur in defining a CR that can address the agent towards an optimal exploration/exploitation policy of interaction with its environment (the model space). The ambitious goal of this approach is to teach an artificial agent to explore autonomously the space of models in the best possible way, searching for global optima of the objective function even in complex geophysical inverse scenarios.

An additional important discussion point concerns the comparison of our RL-Inv method with other global inversion approaches. For instance, Aleardi *et al*. (2021) have successfully implemented a Markov Chain Monte Carlo (MCMC) inversion algorithm for Bayesian ERT (Electric

Resistivity Tomography) in which the subsurface resistivity model, the facies configuration, and the associated uncertainties are inferred from the measured apparent resistivity values. This approach works effectively without requiring any assumptions (such as Gaussian distribution) on the model property in each facies. Furthermore, in order to reduce the ill-conditioning of the inversion process, the authors include spatial constraints in the inversion workflow for both the continuous and discrete model parameters. In particular, they show that using a Gaussian variogram model for the resistivity value, and 1D Markov prior models for the facies, allows stabilising the inversion process. Unfortunately, from a general point of view, this approach as well as any other global or quasi-global inversion method, necessarily implies an increase of the computation costs. However, in the case of the RL-Inv method, this problem can be limited. In fact, in order to estimate the CR for each inversion trial, it is sufficient to estimate the RMS (or the cost function) after a few iterations. In all our tests, the number of iterations, N, is generally less than 20, but we verified that setting $N < 10$ is sufficient for estimating the CR through Bellman formula. This implies that we do not need to run every inversion completely for updating the Q-Table and for applying the RL-Inv workflow. This strategy allows an expanded exploration of the model space without increasing prohibitively the computation costs. These costs depend on the specific characteristics of the inverse problem, such as data dimensionality, data size, complexity of resistivity scenario, number of starting models tested in the inversion, number of iterations, number of hyper-parameters settings, and so forth. We compared the computation times required by the full application of RL-Inv methodology with respect to a standard deterministic inversion based on Damped Least Square inversion. When running our RL-Inv methodology on 2D ERT data, the computation time generally increases by about two orders of magnitude (using a standard computer with a dual core Intel processor, 2.5 GHz, RAM 12.0 GB, on Windows 10 system, 64 bit). For instance, Table 1 shows a few indicative examples of computation times for the 2D inversion tests discussed in this paper, comparing the Least Square inversion performance with different settings of the RL-Inv approach. We can see that the deterministic inversion algorithm produces a resistivity model in less than half a minute, in this specific case. Instead, the quasi-global RL-Inv approach requires a much longer computation time. However, the Least Square inversion result is highly inaccurate while the RL-Inv model is very close to the true resistivity distribution (as shown in Figs. 4 and 7).

A final discussion point concerns the comparison of RL-Inv with other Machine Learning inversion techniques (Wu and McMechan, 2019; Bai *et al*., 2020). For instance, Colombo *et al*. (2021) developed effective workflows and algorithms for embedding Machine Learning and physics-based inversions into the same unified approach. Their approach consists of training a neural network so that it learns the physical requirements of the inversion process, in order to steer its predictions toward the data misfit reduction. Using a different approach, Dell'Aversana (2020) applies a physics-based process based on a suite of supervised Machine Learning algorithms working in parallel on the same input data set. This methodology consists of a complex supervised learning approach. Different classifiers (including Deep Neural Networks, Random Forest, Adaptive Boosting, Naïve Bayes, Decision Trees, and so forth) are trained using the same labelled data set properly calibrated at well locations. Furthermore, the same classifiers are trained using the geophysical models obtained by inverting multidisciplinary data and properly calibrated at the same well locations. The final objective is to create a suite of probabilistic maps of fluids distribution at target depth, and to compare the different results obtained with the various Machine Learning methods. The main limitation of all the above-mentioned Machine Learning inversion techniques is that they require large labelled input data in order to retrieve reliable models/maps through a supervised Machine Learning workflow. Instead, the RL-Inv approach does not require any training data set because the optimal policy for exploring the model space is retrieved progressively through recursive application of Bellman equations, as explained in the methodological part of this paper.

Table 1 - Comparison of computation costs between Damped Least Square and RL-Inv approaches for the 2D synthetic inversion test discussed in section 3.

Inv. approach	*Dim.*	*N. of data*	*N. model par.*	*N. of models*	*N. inv. settings*	*N. Iterat.*	*Inv. time (s)*	*Inv. time (h)*
			(N. inv. runs)		(for each run)			
Least Square	**2D**	**620**	**440**	**1**	**1**	**40**	**22**	**0,01**
RL-Inv	2D	620	440	120	10	8	5280	1,47
RL-Inv	2D	620	440	200	5	6	3300	0,92
RL-Inv	2D	620	440	250	4	5	2750	0,76
RL-Inv	2D	620	440	280	2	5	1540	0,43

(System specs: Dual core Intel processor, 2.5 GHz, RAM 12.0 GB, Windows 10, 64 bit)

6. Conclusions

In this paper, we introduced a new approach, named 'RL supported Inversion' (RL-Inv), for inverting geophysical data with the support of RL techniques. After testing our workflow on synthetic and real geoelectric 3D data, we can draw the following main conclusions.

RL-Inv effectively combines the Q-Learning method with consolidated local optimisation techniques. There is no particular restriction in choosing the local optimisation algorithm: the workflow is designed in order to combine RL criteria with the Epsilon-Greedy approach and, at the same time, with standard optimisation techniques.

RL-Inv improves the exploration of the model space and helps select optimal inversion hyper-parameters, in order to limit the possibility of being trapped in local optima of the objective function. This allows improving the performance of the geophysical inversion with just a partial increase of the computation cost, depending on the specific inversion problem.

Finally, it is worth noting that RL-Inv is a general approach that can be applied to many other optimisation problems and not just geophysical inversion. In the case of inverse problems with large numbers of parameters, our method can be empowered by using Deep Neural Networks coupled with Q-Learning (Deep Q-Learning).

REFERENCES

Abbaspour K.C., Schulin R. and van Genuchten M.T.; 2001: *Estimating unsaturated soil hydraulic parameters using ant colony optimization*. Adv. Water Resour., 24, 827-841, doi: 10.1016/S0309-1708(01)00018-5.

Aleardi M., Vinciguerra A. and Hojat A.; 2021: *A geostatistical Markov chain Monte Carlo inversion algorithm for electrical resistivity tomography*. Near Surf. Geophys., 19, 7-26, doi: 10.1002/nsg.12133.

Bai P., Vignoli G., Viezzoli A., Nevalainen J. and Vacca G.; 2020: *(Quasi-) real-time inversion of airborne time-domain electromagnetic data via artificial neural network*. Remote Sens., 12, 3440, doi: 10.3390/rs12203440.

Brunetti G., Šimůnek J. and Piro P.; 2016: *A comprehensive numerical analysis of the hydraulic behavior of a permeable pavement*. J. Hydrol., 540, 1146-1161, doi: 10.1016/j.jhydrol.2016.07.030.

Brunetti G., Porti M. and Piro P.; 2018: *Multi-level numerical and statistical analysis of the hygrothermal behavior of a non-vegetated green roof in a Mediterranean climate*. Appl. Energy, 221, 204-219, doi: 10.1016/J.APENERGY.2018.03.190.

Colombo D., Turkoglu E., Li W., Sandoval-Curiel E. and Rovetta D.; 2021: *Physics-driven deep-learning inversion with application to transient electromagnetics*. Geophys., 86, E209-E224.

Cormen T.H., Leiserson C.E., Rivest R.L. and Stein C.; 2001: *Introduction to algorithms*. MIT Press, Cambridge, MA, USA, 370 pp., ISBN 978-0-262-03293-3.

Dell'Aversana P.; 2020: *An integrated multi-physics Machine Learning approach for exploration risk mitigation.* Boll. Geof. Teor. Appl., 61, 517-538, doi: 10.4430/bgta0325.

Dell'Aversana P.; 2022a: *Reinforcement Learning in optimization problems. Applications to geophysical data inversion*. AIMS Geosci., 8, 488-502, doi: 10.3934/geosci.2022027.

Dell'Aversana P.; 2022b: *Combining geophysical inversion with Reinforcement Learning*. In: Proc. 83rd EAGE Annual Conference & Exhibition, Madrid, Spain, Vol. 2022, pp. 1-5, doi: 10.3997/2214-4609.202210229.

Dell'Aversana P., Servodio R. and Rizzo E.; 2017: *4D borehole electric tomography for hydrocarbon reservoir monitoring*. In: Extended Abstract 79th EAGE Conference & Exhibition, Paris, France, Vol. 2017, pp. 1-5, doi: 10.3997/2214-4609.201701385.

Duan Q.Y., Gupta V.K. and Sorooshian S.; 1993: *Shuffled complex evolution approach for effective and efficient global minimization*. J. Optim. Theor. Appl., 76, 501-521, doi: 10.1007/BF00939380.

Eberhart R. and Kennedy J.; 1995: *A new optimizer using particle swarm theory.* In: Proc. 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39-43, doi: 10.1109/MHS.1995.494215

Ines A.V.M. and Droogers P.; 2002: *Inverse modelling in estimating soil hydraulic functions: a genetic algorithm approach*. Hydrol. Earth Syst. Sci., 6, 49-66, doi: 10.5194/hess-6-49-2002.

Jones M. and Peet M.M.; 2021: *A generalization of Bellman's equation with application to path planning, obstacle avoidance and invariant set estimation*. Automatica, 127, 109510, doi: 10.1016/j.automatica.2021.109510.

Lample G. and Chaplot D.S.; 2017: *Playing FPS games with Deep Reinforcement Learning*. Assoc. Adv. Artif. Intell., 2140-2146, doi: 10.48550/arXiv.1609.05521.

Littman M.L.; 1994: *Markov games as a framework for multi-agent Reinforcement Learning*. In: Proc. 11th Machine Learning International Conference, New Brunswick, NJ, USA, pp. 157-163, doi: 10.1016/b978-1-55860-335-6.50027-1.

Mnih V., Kavukcuoglu K., Silver, D., Rusu A.A., Veness J., Bellemare M.G., Graves A., Riedmiller M., Fidjeland A.K., Ostrovski G., Petersen S., Beattie C., Sadik A., Antonoglou I., King H., Kumaran D., Wierstra D., Legg S. and Hassabis D.; 2015: *Human-level control through Deep Reinforcement Learning*. Nature, 518, 529-533, doi: 10.1038/nature14236.

Nagabandi A., Kahn G., Fearing R.S. and Levine S.; 2018: *Neural network dynamics for model-based Deep Reinforcement Learning with model-free fine-tuning.* IEEE International Conference on Robotics and Automation (ICRA), 7559-7566, doi: 10.1109/ICRA.2018.8463189.

Noel M.M.; 2012: *A new gradient based particle swarm optimization algorithm for accurate computation of global minimum*. Appl. Soft Comput., 12, 353-359, doi.org/10.1016/j.asoc.2011.08.037.

Pan L. and Wu L.; 1998: *A hybrid global optimization method for inverse estimation of hydraulic parameters: annealing-simplex method*. Water Resour. Res., 34, 2261-2269, doi: 1029/98WR01672.

Ravichandiran S.; 2020: *Deep Reinforcement Learning with python, 2nd ed*. Packt, Birmingham, UK, 760 pp.

Sutton R.S. and Barto A.G.; 1998: *Reinforcement Learning: an introduction*. MIT Press, Cambridge, MA, USA, 340 pp., ISBN 0-262-19398-1.

Tarantola A.; 2005: *Inverse problem theory and methods for model parameter estimation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 358 pp.

Telford W.M., Geldart L.P. and Sheriff R.E.; 1990: *Applied Geophysics, 2nd ed.* Cambridge University Press, Cambridge, UK, 760 pp., ISBN 0-521-32693-1.

Vrugt J.A., Gupta H.V., Bouten W. and Sorooshian S.; 2003: *A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters.* Water Resour. Res., 39, 1201, doi: 10.1029/2002WR001642.

Wu Y. and McMechan G.A.; 2019: *Parametric convolutional neural network-domain full-waveform inversion*. Geophys., 84, R881-R896.

Corresponding author: Paolo Dell'Aversana
Eni S.p.A. Upstream and Technical Services
Via Emilia 1, 20097 San Donato Milanese (MI), Italy
Phone: +39 02 52063217; e-mail: Paolo.Dell'Aversana@eni.com