# Multi-batch Reinforcement Learning

**Romain Laroche**
Microsoft Research Montréal
romain.laroche@microsoft.com

**Rémi Tachet des Combes**
Microsoft Research Montréal
remi.tachet@microsoft.com

## Abstract

We consider the problem of Reinforcement Learning (RL) in a multi-batch setting, also sometimes called growing-batch setting. It consists in successive rounds: at each round, a batch of data is collected with a fixed policy, then the policy may be updated for the next round. In comparison with the more classical online setting, one cannot afford to train and use a bad policy and therefore exploration must be carefully controlled. This is even more dramatic when the batch size is indexed on the past policies performance. In comparison with the mono-batch setting, also called offline setting, one should not be too conservative and keep some form of exploration because it may compromise the asymptotic convergence to an optimal policy.

In this article, we investigate the desired properties of RL algorithms in the multi-batch setting. Under some minimal assumptions, we show that the population of subjects either depletes or grows geometrically over time. This allows us to characterize conditions under which a safe policy update is preferred, and those conditions may be assessed in-between batches. We conclude the paper by advocating the benefits of using a portfolio of policies, to better control the desired amount of risk.

**Keywords:** Multi-Batch Reinforcement Learning, Algorithm Selection

# 1 Introduction

The most common setting for Reinforcement Learning (RL) is *online*: the algorithm directly interacts with the true environment and is allowed to be updated anytime. This setting is the less restrictive one from the algorithmic point of view, but real world problems (RWP) have all sorts of additional constraints that make it – most of the time – inapplicable. First example, RWP generally have a high complexity and a complete policy update would be too expensive to compute at every time step, hence the use of *online RL algorithms* that only perform small updates on the policy or the value-function estimators through temporal difference or gradient descent. The online RL algorithms comply with the complexity constraint but are less sample efficient. Second example, RWP are also generally meant to be widely deployed, on different devices with limited bandwidth, memory and computational power, which prevents frequent policy updates. As a consequence, while the online setting does not suffer from bad intermediate policies, since those can be fixed promptly, we argue that bad intermediate policies would jeopardize most RWP services.

At the opposite, the *single batch* setting, in the words of [3], refers to a reinforcement learning setting, where the complete amount of learning experience, usually a set of transitions sampled from the system, is fixed, without any access to the true environment. The literature on single batch RL focuses on safe policy improvement of the baseline policy that was used to generate the batch [6]. RWP never amount to a single policy update. Instead, we argue that most of them consist in a *multi-batch* setting, also sometimes referred to as growing batch in the literature, where the policy is successively trained on the past batches of data. This setting is commonly encountered in the following domains: dialogue systems, crop management or pharmaceutical treatment. The single batch setting might therefore be regarded as a greedily myopic study of the multi-batch setting where the former objective is a mix of safety and expected performance, neglecting the longer-term impact of the chosen policy on the quality of the next batches. As a consequence, algorithm safety might be counterproductive as it punishes exploratory strategies, which is detrimental to the asymptotic performance.

Our contributions are the following:

- We make the first attempt to model the multi-batch setting process.
- Under a set of minimal assumptions, we prove that, asymptotically, either the pool of subjects depletes, or grows geometrically.
- We conclude the paper with a set of recommendation for situational desired properties of the algorithms and argue that the situation may be assessed during the process with mild assumptions.

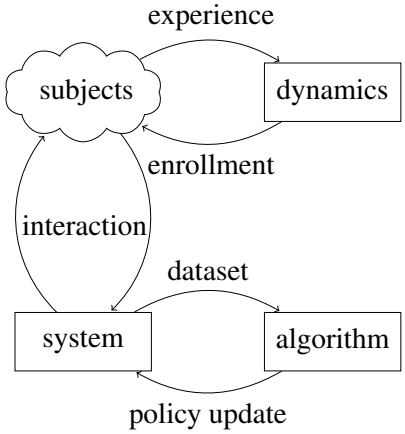


Figure 1: Multi-batch setting.

# 2 Multi-Batch Reinforcement Learning Process

Process 1 formalizes the generic process involved in the multi-batch setting: at every batch, the RL algorithm trains/updates a policy (Step 1). This policy is used to collect a dataset (Step 3) through interactions with a set of subjects, whose enrollment depends on the past subjects experience (Step 2). Figure 1 is an illustration of the multi-batch setting.

Step 2 is generally overlooked in the literature. However, we will show that it is crucial. Indeed, the size of the dataset $\mathcal{D}_\beta$, called crowd and denoted by $\kappa_\beta = |\mathcal{D}_\beta|$ in the following, is dependent on the past subject experience. For instance, if the algorithm generates a bad policy, it is likely to lose its subjects and later, it may only get a handful of additional experience in the next batch. Then, it may be slow to regain subjects' trust.

The goal is to optimize the cumulative return after $B \in \mathbb{R}^+$ batches. More formally, we have:

$$\mathcal{J}\left(\alpha, \kappa_0, \{\pi_0, \mathcal{D}_0\}, B\right) = \sum_{\beta=1}^{B} \sum_{k=1}^{\kappa_\beta} \dot{\rho}_{\pi_\beta, \tau_k} = \mathcal{J}\left(\alpha, \kappa_1, \{\pi_0, \mathcal{D}_0\} \cup \{\pi_1, \mathcal{D}_1\}, B-1\right) + \sum_{\tau=1}^{\kappa_1} \dot{\rho}_{\pi_1, \tau}, \tag{1}$$

**Process 1:** Multi-batch setting process

**Input:** Initial policy $\pi_0$ **Input:** Initial crowd $\kappa_0$ **Input:** Unknown environment MDP: $M = \langle \mathcal{X}, \mathcal{A}, P, R, \gamma \rangle$

**Input:** Initial dataset $\mathcal{D}_0$ **Input:** Multi-batch algorithm $\alpha$ **Input:** Horizon of the process (number of batches): $B$

**for** *each batch* $\beta \in [\![1, B]\!]$ **do**

**Step 1:** with $\alpha$, train the new policy $\pi_\beta$ on past datasets and their behavioural policies: $\pi_\beta \sim \alpha\left(\{\pi_{\beta'}, \mathcal{D}_{\beta'}\}_{\beta' \in [\![0, \beta-1]\!]}\right)$.

**Step 2:** enroll a crowd of $\kappa_\beta$ subjects, in function of the past subjects experience: $\kappa_\beta \sim g\left(\{\mathcal{D}_{\beta'}\}_{\beta' \in [\![0, \beta-1]\!]}\right)$.

**Step 3:** collect dataset $\mathcal{D}_\beta$ of size $\kappa_\beta$, by following policy $\pi_\beta$: $\mathcal{D}_\beta = \left\{\tau_k \sim \langle \mathcal{X}, \mathcal{A}, P, \pi_\beta, R, \gamma \rangle\right\}_{k \in [\![1, \kappa_\beta]\!]}$.

**end for**

where $k$ is the index of trajectory $\tau_k$, $\beta$ is a batch index, $\alpha$ is a multi-batch RL algorithm, $\pi_\beta$ is the policy trained by algorithm $\alpha$ at batch $\beta$ on dataset $\bigcup_{\beta'=0}^{\beta-1}\{\pi_{\beta'}, \mathcal{D}_{\beta'}\}$, $\dot{\rho}_{\pi_\beta,\tau_k}$ is the random variable denoting the performance of trajectory $\tau_k$, when following policy $\pi_\beta$, $\kappa_\beta$ is the crowd at batch $\beta$, $\pi_0$ is the initial policy, and $\mathcal{D}_0$ denotes the possibly empty initial batch of data.

In the following, the performance $\dot{\rho}_{\pi_\beta,\tau}$ of a given trajectory $\tau$ will be defined as the classical infinite horizon RL discounted return: $\dot{\rho}_{\pi_\beta,\tau} = \sum_{t=0}^{\infty}\gamma^t r_{\pi_\beta,\tau,t}$, (we assume that all trajectories of batch $B-1$ have terminated before starting batch $B$) but any other trajectory-sighted objective function may be considered and, at the exception of Proposition 2 (for which similar bounds may still be found under some other mild assumptions), all results hold. For instance, $\dot{\rho}_{\pi_\beta,\tau}$ may be defined as the binary task completion.

**Remark 1.** *We make the following remarks about Process 1:*

*(i) Step 1: some algorithms are randomized, hence the sampling sign '$\sim$'.*

*(ii) Step 1: algorithms cannot be considered monotonous with respect to the samples they are trained on. Indeed, some data may be misleading and lead to bad policies [4]. Some algorithms (such as vanilla model-based RL) may train policies that are performing worse with larger datasets, even in expectation [5].*

*(iii) Step 2: the function $g$ for the crowd update is stochastic, hence the sampling sign '$\sim$'.*

*(iv) Step 2: the function $g$ depends on individual factors: "did the subject have a good past experience with the system?" ; and global factors: "does the system have a good image?", "what is the pool size for the crowd?".*

*(v) Step 2: the function $g$ is dependent on the task. In some domains, it may be unacceptable for the system to fail: it is essentially evaluated on its efficiency (e.g. autonomous cars). In others, it is acceptable for it to fail regularly (e.g. dialogue systems).*

*(vi) Step 2: the function $g$ may not be monotonous: it has happened in the past that some systems got hyped because they were failing in an entertaining way. We may cite three famous examples: Tay, Baidu and Youtube Rewind 2018.*

*(vii) Step 3: the dataset collection involves several sources of stochasticity: $\pi_\beta$, $P$, and $R$.*

*(viii) Steps 2 & 3: subjects behave differently from one another and overtime.*

## 3 Analysis

In order to give some insights on the dynamics, and similarly to [2], we make a series of assumptions that should account for a large variety of multi-batch RL settings:

**Assumption 1.** *The multi-batch RL process is simplified as follows:*

*(i) The performance $\dot{\rho}_{\pi_\beta,\tau}$ of trajectory $\tau$ generated at batch $\beta$ is a random variable that belongs to $[-1,1]$.*

*(ii) The crowd $\kappa_\beta$ at any given batch $\beta$ is assumed to be subject-centered, i.i.d., linearly bounded, and stationary over time.*

Assumption 1*(ii)* states that each subject having followed a trajectory $\tau$ during batch $\beta$ enrolls $\dot{g}_{\pi_\beta,\tau} \in \mathbb{N}$ subjects for the next batch. $\dot{g}_{\pi_\beta,\tau}$ is assumed to be a random variable that only depends on its last individual experience and that is bounded by some maximal value $\dot{g}_{max}$. While $\dot{\rho}_{\pi_\beta,\tau}$ and $\dot{g}_{\pi_\beta,\tau}$ are only depending on the generated trajectory, it is more convenient to consider them as being directly sampled from a distribution only dependent on the policy $\pi_\beta$: $\dot{\rho}_{\pi_\beta,\tau} \sim \dot{\rho}(\pi_\beta)$ and $\dot{g}_{\pi_\beta,\tau} \sim \dot{g}(\pi_\beta)$, but one has to keep in mind that $\dot{\rho}_{\pi_\beta,\tau}$ and $\dot{g}_{\pi_\beta,\tau}$ are thus correlated through $\tau$.

**Assumption 2.** *Additionally, we assume that the random function $\dot{g}(\pi)$ is $\Lambda$-Lipschitz with respect to $\pi$ and $\ell_{1,\infty}$ :*

$$\mathbb{E}_\tau\left[|\dot{g}_{\pi_1,\tau} - \dot{g}_{\pi_2,\tau}|\right] \leq \Lambda\|\pi_1 - \pi_2\|_{1,\infty}, \tag{2}$$

*where $\ell_{1,\infty}$ is defined as follows: $\|\pi_1 - \pi_2\|_{1,\infty} = \sup_{x\in\mathcal{X}}\int_{a\in\mathcal{A}}|\pi_1(a|x) - \pi_2(a|x)|da$.*

Assumption 2 is satisfied in most cases, and in particular when the trajectory length is bounded by $t_{max}$. We denote the empirical mean performance during batch $\beta$ with the random variable $\widehat{\rho}_\beta$ and the empirical batch growth with $\widehat{g}_\beta$:

$$\widehat{\rho}_\beta = \frac{1}{\kappa_\beta}\sum_{\tau=1}^{\kappa_\beta}\dot{\rho}_{\pi_\beta,\tau} \qquad \text{and} \qquad \widehat{g}_\beta = \frac{1}{\kappa_\beta}\sum_{\tau=1}^{\kappa_\beta}\dot{g}_{\pi_\beta,\tau}. \tag{3}$$

Then, we may write:

$$\kappa_\beta = \kappa_{\beta-1}\widehat{g}_{\beta-1} = \kappa_0\prod_{\beta'=0}^{\beta-1}\widehat{g}_{\beta'}, \tag{4}$$

As a consequence of the assumption on $\kappa_\beta$'s dynamics, if $\dot{g}_{\pi,\tau}$ can take values greater than 1 for some policy $\pi$, then the crowd is automatically assumed unbounded.

This section makes the analysis of the multi-batch process under Assumptions 1 and 2 in a tacit manner. Due to space constraints, all the proofs are omitted in the extended abstract.

**Proposition 1.** *The objective function $\mathcal{J}$ unfolds as follows:*

$$\mathcal{J}\left(\alpha, \kappa_0, \{\pi_0, \mathcal{D}_0\}, B\right) = \kappa_0 \sum_{\beta=1}^{B} \widehat{\rho}_\beta \prod_{\beta'=0}^{\beta-1} \widehat{g}_{\beta'}. \tag{5}$$

From now on, we focus the analysis on the asymptotic behaviour of $\mathcal{J}$ with respect to $B$.

**Assumption 3.** *We assume that $\pi_\beta$ uniformly converges in probability with respect to the $\ell_{1,\infty}$-norm to some policy $\pi_\infty$ as $\beta$ tends to infinity:*

$$\forall \epsilon > 0, \exists \beta_0 \in \mathbb{N}, \text{ such that }, \forall \beta > \beta_0, \|\pi_\beta - \pi_\infty\|_{1,\infty} < \epsilon. \tag{6}$$

Assumption 3 only states the convergence of $\pi_\beta$, it does not say anything on the quality of the policy $\pi_\infty$ in the limit. This assumption generally holds for unbiased algorithms since the incoming data depletes either in quantity: the crowd equals 0 at some batch and thereafter, no more data may be collected, or in informative content as a consequence of the strong law of large numbers, and of the increasing nature of the data collection (as long as the trained policies do not oscillate between several equally (sub-)optimal solutions). More specifically, in finite spaces $\mathcal{X}$ and $\mathcal{A}$, this assumption is satisfied by most greedy-in-the-limit algorithms. In continuous space, it has to be noted that most algorithms include a bias that makes them sensitive to the data distribution, and this distribution is in turn dependent on the previously used policy. In case of crowd depletion, there is also the possibility for the algorithm to be randomized and therefore to generate a different policy at every batch. But, since these policies are never used because of the crowd depletion, this special case could be treated separately in a trivial way, and would not be a real issue for the generality of the theory.

**Proposition 2.** *We consider $t \in \mathbb{N}$ and two policies $\pi_1$ and $\pi_2$. Then, with probability larger than $1 - t\|\pi_1 - \pi_2\|_{1,\infty}$ :*

$$|\dot{\rho}_{\pi_1,\tau} - \dot{\rho}_{\pi_2,\tau}| \leq 2\gamma^t.^{1} \tag{7}$$

**Corollary 1.** *If $\|\pi_1 - \pi_2\|_{1,\infty} < \epsilon$, then, with probability larger than $1 + \epsilon\left\lfloor \log_{\gamma^{-1}} \epsilon \right\rfloor$: $|\dot{\rho}_{\pi_1,\tau} - \dot{\rho}_{\pi_2,\tau}| \leq 2\epsilon$.*

Proposition 2 and its corollary relate the distance between the trajectories generated by two policies with their distance.

**Proposition 3.** *If $\|\pi_1 - \pi_2\|_{1,\infty} < \epsilon$, then, $\dot{g}_{\pi_1,\tau}$ and $\dot{g}_{\pi_2,\tau}$ are close in mean and variance:*

$$|\mathbb{E}\dot{g}_{\pi_1,\tau} - \mathbb{E}\dot{g}_{\pi_2,\tau}| \leq \Lambda\epsilon \qquad \text{and} \qquad |\mathbb{V}\dot{g}_{\pi_1,\tau} - \mathbb{V}\dot{g}_{\pi_2,\tau}| \leq \Lambda\epsilon\dot{g}_{max}, \tag{8}$$

*and are equal with probability higher than $1 - \Lambda\epsilon$.*

Proposition 3 relates the distance between the enrollments implied by two policies with their distance.

**Corollary 2.** *Under Assumption 3, $\dot{\rho}_{\pi_\beta,\tau}$ and $\dot{g}_{\pi_\beta,\tau}$ respectively converge in probability to $\dot{\rho}_{\pi_\infty,\tau} \sim \dot{\rho}(\pi_\infty)$ and $\dot{g}_{\pi_\infty,\tau} \sim \dot{g}(\pi_\infty)$ :*

$$\forall \epsilon > 0, \lim_{\beta\to\infty} \mathbb{P}\left(|\dot{\rho}_{\pi_\beta,\tau} - \dot{\rho}_{\pi_\infty,\tau}| > \epsilon\right) = 0, \tag{9}$$

$$\forall \epsilon > 0, \lim_{\beta\to\infty} \mathbb{P}\left(|\dot{g}_{\pi_\beta,\tau} - \dot{g}_{\pi_\infty,\tau}| > \epsilon\right) = 0. \tag{10}$$

Corollary 2 proves, under Assumption 3, the convergence in probability of $\dot{\rho}_{\pi_\beta,\tau}$ and $\dot{g}_{\pi_\beta,\tau}$, when $\beta$ tends to $\infty$. Now, we define two modes of asymptotic behaviour of the multi-batch process. Then, we show as our main contribution that, except for identified degenerate cases, the multi-batch process follows one of the two following modes:

**Definition 1.** *The crowd depletion mode ($\mathcal{CDM}$) has the following properties:*

*(i) The crowd converges to 0 almost surely:* $\lim_{\beta\to\infty} \kappa_\beta = 0$.

*(ii) The dataset remains finite:* $|\bigcup_{\beta\in\mathbb{N}} \mathcal{D}_\beta| < \infty$.

*(iii) The objective function $\mathcal{J}$ remains finite:* $\lim_{B\to\infty} \mathcal{J} < \infty$.

[1] The use of the same subscript $\tau$ for both random variables $\dot{\rho}_{\pi_1,\tau}$ and $\dot{\rho}_{\pi_2,\tau}$ indicates that as long as both policies behave the same, the other random events follow the same realization. In other words, they behave as generated with the same random seed.

**Definition 2.** *The geometric crowd mode ($\mathcal{GCM}$) has the following properties:*

*(i) The crowd asymptotically grows geometrically with ratio $\mathbb{E}\dot{g}_{\pi_\infty,\tau}$ as a function of $\beta$.*

*(ii) The dataset size grows to infinity: $|\bigcup_{\beta\in\mathbb{N}} \mathcal{D}_\beta| = \infty$.*

*(iii) The objective function $\mathcal{J}$ asymptotically grows geometrically with ratio $\mathbb{E}\dot{g}_{\pi_\infty,\tau}$ as a function of $B$. As a consequence, $\mathcal{J}$ diverges either to $+\infty$ or $-\infty$, according to the sign of $\mathbb{E}\dot{\rho}_{\pi_\infty,\tau}$. If $\mathbb{E}\dot{\rho}_{\pi_\infty,\tau} = 0$, then nothing can be said about $\lim_{B\to\infty} \mathcal{J}$.*

**Theorem 1.** *In the degenerate case $\dot{g}_{\pi_\infty,\tau} = 1$, nothing may be said, except that it does not follow $\mathcal{GCM}$.*

*If $\mathbb{E}\dot{g}_{\pi_\infty,\tau} \leq 1$, then the process almost surely enters $\mathcal{CDM}$.*

*If $\mathbb{E}\dot{g}_{\pi_\infty,\tau} > 1$, then with some probability $p_\infty > 0$, the process asymptotically enters $\mathcal{GCM}$. With the complementary probability $1-p_\infty$, it enters $\mathcal{CDM}$. $p_\infty$ may be lower bounded from batch $\beta_0$ on, if $\beta_0$ is such that for all $\beta \geq \beta_0$, expectation $\mathbb{E}\dot{g}_{\pi_\beta,\tau} \geq \mu_0 > 1$, and variance $\mathbb{V}\dot{g}_{\pi_\beta,\tau} \leq \sigma_0^2$ :*

$$p_\infty \geq 1 - \min_{\mu\in(1,\mu_0)} \left\{ \frac{\sigma_0^2 \mu}{\kappa_{\beta_0}(\mu_0-\mu)^2(\mu-1)}; \frac{e^{-\frac{2\kappa_{\beta_0}(\mu_0-\mu)^2}{\dot{g}_{max}^2}}}{1-e^{-\frac{2\kappa_{\beta_0}(\mu-1)(\mu_0-\mu)^2}{\dot{g}_{max}^2}}} \right\}. \tag{11}$$

In particular $\mu = \frac{1}{4}\left(1+\sqrt{8\mu_0+1}\right)$ minimizes the first term (obtained with Chebyshev's bound), the second term (obtained with Hoeffding's bounds) may be shown to have a unique local minimum that does not admit a closed form expression. A numerical simulation (not reported here) shows that Equation 11 is not a tight bound and, more surprisingly, suggests that $p_\infty$ is close to being constant when $\mu_0 - 1$ is small, and when $\kappa_{\beta_0}(\mu_0-1)$ and $\sigma_0^2$ are constant. The derivation of tighter bounds is left for future work.

## 4 Concluding recommendations

The goal is to maximize the objective function $\mathcal{J}$, but given the stochasticity of the process, one has to consider the *expected indirect utility*. The concept of *indirect utility* classically refers to a measurement of the satisfaction obtained by the decision maker as a function of its objective function. It is generally assumed to be monotonically increasing with the objective function, concave in $\mathbb{R}^+$, reflecting aversion to risk and diminishing marginal utility, and asymmetric with respect to the origin. The logarithm utility function, first proposed by Bernoulli, is still commonly used: $\Upsilon_{\log}(\mathcal{J}) = \text{sign}(\mathcal{J})\log(1+|\mathcal{J}|)$.

Under the log-utility, the following recommendations may be made. The goals for a multi-batch algorithm are the following, by decreasing order of importance: to make $\mathbb{E}\dot{\rho}_{\pi_\infty,\tau}$ positive, to maximize $p_\infty \mathbb{E}\left[\log \mathbb{E}_\tau \dot{g}_{\pi_\infty,\tau} | \mathcal{GCM}\right]$, and only marginally, maximize the asymptotic expected performance $\mathbb{E}\dot{\rho}_{\pi_\infty,\tau}$. In other words, the most important for a service is to make it profitable for its owner ($\dot{\rho}_{\pi_\infty,\tau} > 0$), then to make it viable ($p_\infty \mathbb{E}\left[\log \dot{g}_{\pi_\infty,\tau} | \kappa_\beta \to \infty\right]$), and only then, to make it effective ($\mathbb{E}\dot{\rho}_{\pi_\infty,\tau}$). In most cases, maximizing $p_\infty$ and $\mathbb{E}\left[\log \dot{g}_{\pi_\infty,\tau}\right]$ is achieved through maximizing $\mathbb{E}\dot{\rho}_{\pi_\infty,\tau}$. But in some cases, a "too good" service has for consequence to lose potential future customers. This is, for instance, why planned obsolescence exists. Consequently, the multi-batch setting is multi-objective and its most salient objective is not the optimization of the performance.

The random variable $\dot{g}_{\pi,\tau}$ is generally strongly dependent on the trajectory performance random variable. Assuming past samples of $\dot{g}_{\pi,\tau}$ are observable, after several batches, one might have a good estimate of it as a random function of $\dot{\rho}_{\pi,\tau}$. Based on such a model, one is now able to assess whether risk should be taken in order to increase $\mathbb{E}\dot{g}_{\pi_\beta,\tau}$ or to be safe in order to optimize $p_\infty$. A/B testing over two or more policies, including a safe past policy, even without online optimization, is beneficial in order to balance the desired amount of risk and explore new policies at the same time. With online optimization, an algorithm selection for RL has been shown to outperform the most efficient one in the portfolio with a minimal amount of embedded computation power [4] and bandwidth [1].

## References


[1] Raphaël Féraud, Réda Alami, and Romain Laroche. Decentralized exploration in multi-armed bandits. *arXiv preprint arXiv:1811.07763*, 2018.

[2] Tatsunori B Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. *arXiv preprint arXiv:1806.08010*, 2018.

[3] Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*. Springer, 2012.

[4] Romain Laroche and Raphaël Féraud. Reinforcement learning algorithm selection. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[5] Romain Laroche and Paul Trichelair. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 14th European Workshop on Reinforcement Learning*, 2018.

[6] Marek Petrik, Mohammad Ghavamzadeh, and Yinlam Chow. Safe policy improvement by minimizing robust baseline regret. In *Proceedings of the 29th Advances in Neural Information Processing Systems (NIPS)*, 2016.