Analyzing bandit-based adaptive operator selection mechanisms

Álvaro Fialho, Luis da Costa, Marc Schoenauer, Michèle Sebag

AMAI – Special Issue on LION manuscript No.
(will be inserted by the editor)

Analyzing Bandit-based Adaptive Operator Selection Mechanisms*

Álvaro Fialho · Luis Da Costa · Marc Schoenauer · Michèle Sebag

Received: date / Accepted: date

Abstract Several techniques have been proposed to tackle the *Adaptive Operator Selection* (AOS) issue in Evolutionary Algorithms. Some recent proposals are based on the *Multi-Armed Bandit* (MAB) paradigm: each operator is viewed as one arm of a MAB problem, and the rewards are mainly based on the fitness improvement brought by the corresponding operator to the individual it is applied to. However, the AOS problem is dynamic, whereas standard MAB algorithms are known to optimally solve the exploitation versus exploration trade-off in static settings. An original dynamic variant of the standard MAB *Upper Confidence Bound* algorithm is proposed here, using a sliding time window to compute both its exploitation and exploration terms. In order to perform sound comparisons between AOS algorithms, artificial scenarios have been proposed in the literature. They are extended here toward smoother transitions between different reward settings. The resulting original testbed also includes a real evolutionary algorithm that is applied to the well-known Royal Road problem. It is used here to perform a thorough analysis of the behavior of AOS algorithms, to assess their sensitivity with respect to their own hyper-parameters, and to propose a sound comparison of their performances.

* Author's draft version, the original publication is available at www.springerlink.com

Álvaro Fialho
Microsoft Research–INRIA Joint Centre, 28 rue Jean Rostand, 91893 Orsay Cedex, France
E-mail: Alvaro.Fialho@inria.fr, Tel.: +33 1 6935 6998, Fax: +33 1 6935 6969

Luis Da Costa
Project-Team TAO, INRIA Saclay – Île-de-France & LRI (UMR CNRS 8623), Bât. 490, Université Paris-Sud, 91405 Orsay Cedex, France. E-mail: Luis.DaCosta@inria.fr

Marc Schoenauer
Project-Team TAO, INRIA Saclay – Île-de-France & LRI (UMR CNRS 8623), Bât. 490, Université Paris-Sud, 91405 Orsay Cedex, France
and Microsoft Research–INRIA Joint Centre, 28 rue Jean Rostand, 91893 Orsay Cedex, France.
E-mail: Marc.Schoenauer@inria.fr

Michèle Sebag
Project-Team TAO, LRI (UMR CNRS 8623) & INRIA Saclay – Île-de-France, Bât. 490, Université Paris-Sud, 91405 Orsay Cedex, France
and Microsoft Research–INRIA Joint Centre, 28 rue Jean Rostand, 91893 Orsay Cedex, France.
E-mail: Michele.Sebag@inria.fr

Keywords Parameter Control · Adaptive Operator Selection · Multi-Armed Bandits

1 Introduction

Evolutionary Algorithms (EAs) are stochastic optimization algorithms remotely inspired by the Darwinian "survival of the fittest" paradigm. Let the goal be to optimize some objective function, referred to as *fitness* function, defined on search space X; elements of X are called *individuals*, and a set of individuals is termed a *population*. EAs evolve a population of individuals by iteratively (i) selecting some individuals (the *parents*), favoring those with better fitness; (ii) perturbing stochastically the parents using some *variation operators*, thus generating *offspring*; (iii) evaluating the offspring (i.e. computing their fitness); (iv) selecting some individuals among the parents and the offspring to become the next parents, again favoring fitter individuals. The reader is referred to [34,13] for a comprehensive presentation.

Evolutionary Computation is an exciting research field with the power to assist scientists, researchers, and engineers in the task of solving hard optimization problems, *e.g.*, problems in which the solution space is too big to be completely checked, or in which prior knowledge about the solution space is very hard and/or expensive to be obtained. EAs consistently perform well in approximating solutions to many different types of problems (see e.g. all applications described in [44]), largely because they do not require any strong assumption about the underlying search space.

However, EAs are rarely used outside the circle of *knowledgeable practitioners*; they still have to reach the status of off-the-shelf tools. There are several reasons for that, all boiling down to **a lack of practical support** when it comes to actually designing an EA for a given application. On a conceptual level, despite Michalewicz' seminal book [34] and the two more recent books by Eiben and Smith [13] and DeJong [9], the terminology used by authors still reflect the evolutionary trend they historically belong to. On a practical level, while some software packages provide a unifying framework for the various evolutionary approaches (see e.g. [5]), the success of EAs is still very sensitive to the setting of a number of parameters, e.g. population size, types of variation operators and their respective application rates, types of selection mechanisms and their internal parameters. In this respect, EAs are not different from other stochastic optimization methods, heuristics and meta-heuristics [39].

In early days, the Evolutionary Computation techniques actually benefited from those numerous parameters, which ensure their outstanding flexibility, making them applicable to a wide spectrum of applications. The contemporary view of EAs, however, acknowledges that specific problems require specific setups for satisfactory performance [12]: when it comes to solving a given problem, parameter setting is viewed as the Achilles' heel of EAs, on par with their high computational cost.

Accordingly, the topic of parameter setting has been extensively investigated in the Evolutionary Computation literature, and still is a very active field of research [30] (more in Section 2.1). This paper focuses on on-line parameter setting, a.k.a. Parameter Control, and more specifically on *Adaptive Operator Selection*.

In essence, the goal of *Adaptive Operator Selection* is to select on the fly the operator[1] that maximizes some measure of quality, usually, though not exclusively [33,

[1] Or the mixture of operators [38].

31], reflecting the fitness improvement brought by its application. *Adaptive Operator Selection* thus raises two main issues.

The first issue might be seen as an *Exploration vs Exploitation* (EvE) dilemma. While an operator that has performed well in the recent past should certainly be used again (exploitation), other operators that did not perform so well should also be tried (exploration). The rationale for exploration is rooted in the stochastic nature of the evolutionary process, on the one hand (some seemingly poorly-performing operators might just have been unlucky); and on its dynamics on the other hand. Specifically, the quality of an operator depends on the region of the fitness landscape being explored by the current population: good operators might become poor as evolution goes on, and vice-versa; the operators quality assessment being the second issue.

Notably, the *Exploration vs Exploitation* trade-off has been intensively studied in the context of Game Theory, in the so-called Multi-Armed Bandit (MAB) framework [28, 1] (Section 2.3.2). The use of MAB algorithms to solve the EvE dilemma has been investigated in the context of selecting between different algorithm portfolios to solve decision problems [18], and in the framework of *Adaptive Operator Selection* by the authors. For the latter case, the *Upper Confidence Bound* (UCB) technique [1] was used, being referred to as the original (or basic) MAB algorithm in the following. However, although providing asymptotic optimality guarantees with relation to the total cumulated reward in a stationary context, the UCB algorithm has to be extended to cope with the dynamics of evolution. The *Dynamic Multi-Armed Bandit* (DMAB) algorithm [7] proceeds by coupling the original MAB technique with a statistical change-point test, the Page-Hinkley test [23]; upon detecting a change in the operator quality distribution, the *Multi-Armed Bandit* process is restarted from scratch.

The first contribution of the current paper is to propose a smoother way to account for dynamic environments in the *Multi-Armed Bandit* framework, referred to as *Sliding Multi-Armed Bandit* (SlMAB): it uses a sliding time window to gracefully update the operator quality estimates, discarding ancient events while preserving the information from the recent ones. Contrasting with DMAB, the *Sliding Multi-Armed Bandit* does not call upon an external monitoring of the evolution process and involves only 2 hyper-parameters (while DMAB has 3).

The empirical validation of the *Sliding Multi-Armed Bandit* considers artificial settings, as proposed by Thierens [41] and later extended by the authors [7]. However, the *Adaptive Operator Selection* efficiency should also be thoroughly investigated in relation with the operator quality distribution, specifically the mean and variance of the fitness improvement brought by the operator. Hence, a new family of artificial *Adaptive Operator Selection* problems is proposed and considered for comparisons. Besides the artificial problems, a simple though real optimization problem extensively investigated in the Evolutionary Computation literature, the Royal Road [25], is also considered, with the *Adaptive Operator Selection* schemes being used to choose between different crossover and mutation operators.

The last contribution of the present paper is a principled and systematic comparison of different *Adaptive Operator Selection* schemes, ranging from *Sliding Multi-Armed Bandit* to *Multi-Armed Bandit*, *Dynamic Multi-Armed Bandit* and *Adaptive Pursuit* [40], depending on the order parameters of the operator quality distribution.

The paper is organized as follows. Section 2 briefly introduces the problem of Parameter Setting in Evolutionary Algorithms, focusing on *Adaptive Operator Selection*, with the mechanisms introduced in early work being reminded for the sake of self-containedness. In Section 3, the *Sliding Multi-Armed Bandit* mechanism is described

in detail and discussed. Section 4 is devoted to the presentation of the artificial and real problems considered to assess the strengths and weaknesses and, more generally, to identify the niche of the diverse *Adaptive Operator Selection* mechanisms. In Section 5, the experimental setting that has been used for all results presented here is detailed, and Section 6 reports on the empirical results obtained by the considered *Adaptive Operator Selection* mechanisms on all the surveyed and newly proposed artificial settings, as well as on the Royal Road problem. Finally, Section 7 summarizes the findings of this paper, and discusses some perspectives for further research.

2 Background and State of the Art

After a brief introduction to Parameter Setting in Evolutionary Algorithms, this section focuses on *Credit Assignment* and *Operator Selection* and recalls the most popular *Adaptive Operator Selection* algorithms to-date: *Probability Matching* and *Adaptive Pursuit* for probability-based mechanisms; and *Multi-Armed Bandit* and *Dynamic Multi-Armed Bandit* for bandit-based algorithms.

2.1 Parameter Setting in Evolutionary Algorithms

After [11,12], parameter setting in EAs proceeds along two main modes. Off-line or external tuning, referred to as **parameter tuning**, determines *a priori* the appropriate parameter values. Parameter tuning thus takes place before the run, *e.g.*, exploiting the lessons learned from previous runs. Standard approaches from experimental studies such as *ANOVA* or *Design Of Experiments* have been used for parameter tuning, *e.g.*, modeling the impact of parameter values on the overall performance and accordingly determining the optimal values [4,45,3,36]. These methods, however, are very computationally expensive, as each observation corresponds to the average of a few evolutionary runs; furthermore, static settings are usually considered (the parameter value is fixed along the run), whereas the optimal setting likely depends on the local landscape explored by the genetic population.

On-line or internal tuning, referred to as **parameter control**, determines the appropriate parameter values at each time step during the evolutionary run. One further level distinguishes between *deterministic* (parameter values are predefined functions of time), *self-adaptive* (parameters are part of the genotypic information and optimized by evolution itself), and *adaptive* (parameter values are predefined functions of the history of the run) parameter control strategies.

Deterministic parameter control essentially raises the same difficulties as parameter tuning: while the parameter values depend on the time step, these control functions must still be defined *a priori*. Self-adaptive parameter control is acknowledged as one of the most effective approaches to evolutionary parameter setting, specifically in the framework of continuous parameter optimization (see the discussion in [10] and references therein). In the general case, however, self-adaptive approaches often significantly increase the size of the search space, and/or the complexity of the optimization problem: not only should a successful individual have good genes, it should also bear parameter values enforcing some effective transmission of its genes.

Adaptive parameter control, also referred to as feedback-based control, uses information from the history of evolution to modify the parameter values while solving the

problem. *Adaptive Operator Selection* (AOS), a particular case of adaptive parameter control, aims at defining an on-line strategy for selecting the most appropriate variation operators. As shown on Fig. 1, AOS involves two sub-problems, detailed in the next subsections: (i) giving some *reward* to an operator (*Credit Assignment*), according to the impact brought by its recent application on the current search/optimization process; and (ii) using these assessments to actually select the next operator to be applied (*Operator Selection*), *e.g.*, the operator currently with best impact expectation.

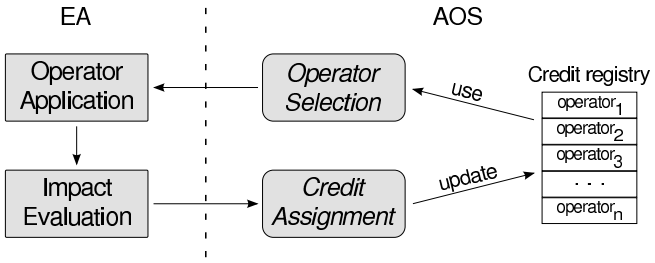


Fig. 1 The Adaptive Operator Selection scheme.

2.2 Credit Assignment

Several *Credit Assignment* mechanisms have been proposed in the literature, following Davis' seminal paper [8]. These mechanisms mostly differ by the measure used to compute the credit, and by the individuals taken into consideration to compute this measure.

Most approaches consider the new offspring and use its fitness improvement as a credit measure. The fitness improvement is assessed by comparison with (i) the current best individual [8]; (ii) the median fitness [26]; or (iii) the parent fitness [29, 42, 2]. When there is no improvement, a null credit is awarded to the operator.

In the case of multi-modal optimization, relevant measures should include the population diversity, which must be enforced to avoid premature convergence. Along this line, [33] proposed another credit measure called *Compass*, defined as a weighted sum of fitness improvement (intensification) and offspring diversity (diversification). In [32], a different aggregation between both impact measures was proposed, based on the *Pareto Dominance* paradigm.

From thereon, the actual reward given to an operator after it has been applied can be either the instantaneous value of the proposed measure, or its average over a sliding time window (i.e. over the last W applications, where W is the size of the time window). Though the instantaneous version can be viewed as an average over a window of size 1, both will be distinguished in the following, termed respectively *Instantaneous* and *Average* rewards.

A very different approach is the one proposed in [43], which rewards the operators based on their ability to generate outlier solutions, following some statistics over the received credit measures. The underlying idea is that the generation of rare but highly beneficial improvements matter as much or more than frequent small improvements;

experimental results show that the method significantly outperforms its competitors on a set of continuous benchmark problems. This proposal was adapted and used in the framework of *Adaptive Operator Selection* by the authors [7,14,16], and will be considered here too: instead of using all the complex statistics to detect outlier production, the reward is simply set to the maximum fitness improvement over a sliding time-window of size W. This *Credit Assignment* is termed *Extreme* in the following.

Another independent issue is that of the choice of the individuals that should be considered when computing the reward of an individual. Some authors consider that the operator impact should be measured after the genealogy of the outstanding offspring, *e.g.*, rewarding the operators producing the ancestors of a good offspring according to a bucket brigade algorithm [8,26]. To the best of our knowledge, however, no clear indication about the benefits of this approach can be found in the literature. Hence, only the current offspring (and its possible fitness improvement over its parent) will be considered here when computing the rewards.

2.3 Operator Selection Rules

Most *Operator Selection* rules attach an empirical quality to each operator. Those qualities can in turn be used to update operator probabilities[2], and these probabilities are then used for selection by a roulette wheel-like process, like in the Probability Matching (PM) and Adaptive Pursuit (AP) methods (section 2.3.1) [41]. Another possibility, however, is based on the so-called Multi-Armed Bandit framework [1], and uses directly the empirical reward gathered by each operator together with an exploration term to deterministically choose amongst the different available operators (section 2.3.2).

2.3.1 Probability Matching and Adaptive Pursuit

Let K denote the number of available variation operators. *Probability Matching* and *Adaptive Pursuit* both maintain a probability vector $(s_{i,t})_{i=1,K}$ and an estimate of the current operator reward noted $\hat{q}_{i,t}$. At each time t:
(i) the i-th operator is selected with probability $s_{i,t}$, and gets a reward r computed after the credit assignment at hand;
(ii) the reward estimate $\hat{q}_{i,t}$ of the i-th operator is updated using an additive relaxation mechanism with adaptation rate α ($0 < \alpha \leq 1$, the memory span decreases as α increases):

$$\hat{q}_{i,t+1} = (1-\alpha)\,\hat{q}_{i,t} + \alpha\, r \tag{1}$$

Probability Matching: PM mostly selects the i-th operator proportionally to $\hat{q}_{i,t}$, except for the fact that a minimum amount of exploration can be enforced. If the selection probability of an operator would become too low at some point, it would never be used again, thus precluding the AOS from discovering it in case it becomes optimal in further stages of evolution.

Formally, let p_{min} denote the minimal selection probability. The selection probability of the i-th operator is defined as:

$$s_{i,t+1} = p_{min} + (1 - K * p_{min}) \frac{\hat{q}_{i,t+1}}{\sum_{j=1}^{K} \hat{q}_{j,t+1}} \tag{2}$$

[2] Methods that recompute those probabilities from scratch from the most recent rewards [26,42] will not be considered here.

After Eq. (2), any ineffective operator (not getting any reward) would be selected with probability p_{min}, while the best operator (getting maximal rewards after some time) would be selected with probability $p_{max} = (1 - (K-1) * p_{min})$. In practice, all mildly relevant operators keep being selected, hindering the performance of *Probability Matching* (all the more so as the number of operators increases) [40].

Adaptive Pursuit: Originally proposed for learning automata, AP has been used in the AOS context [40] to address the above shortcoming of *Probability Matching*; a winner-take-all strategy is used to push forward the best current operator, noted i_t^*, as follows:

$$\begin{cases} i_t^* & = \arg\max_{i=1\ldots K} \{ \hat{q}_{i,t} \} \\ s_{i,t+1} & = \begin{cases} s_{i,t} + \beta \left(p_{max} - s_{i,t} \right) & \text{if } i = i_t^* \\ s_{i,t} + \beta \left(p_{min} - s_{i,t} \right) & \text{otherwise} \end{cases} \end{cases} \tag{3}$$

with the learning rate β controlling the greediness of the winner-take-all strategy.

Finally, both *Probability Matching* and *Adaptive Pursuit* are controlled by the p_{min} parameter (enforcing the operators exploration) and the adaptation rate α (ruling the memory span of the AOS). AP is also guided by the learning rate β.

2.3.2 Bandit-based Operator Selection

Operator selection can be seen as another *Exploration vs Exploitation* (EvE) dilemma, in which exploitation aims at selecting the best rewarded operators in the last stages of evolution, whereas exploration is concerned with checking whether other operators might in fact become the best ones at some later stages. The EvE dilemma has been intensively studied in *Game Theory*, more specifically within the so-called Multi-Armed Bandit (MAB) framework [28, 1].

Multi-Armed Bandit: The MAB framework considers a set of K independent arms (each operator will be considered as one arm), each one of which having some unknown probability of getting a (originally boolean) reward. An optimal selection strategy is the one that maximizes the cumulative reward along time. The *Upper Confidence Bound* (UCB) selection strategy, proposed by Auer et al. [1], provides asymptotic optimality guarantees. Although MAB is the name of the problem itself, for the sake of convenience, in the following of this text we refer to the UCB selection strategy as *the MAB algorithm*.

Formally, the i-th arm is associated to (i) its empirical reward $\hat{q}_i$ (the average reward obtained); and (ii) a confidence interval, depending on the number of times n_i the i-th arm has been tried. UCB selects, at each time step, the arm with best upper bound of the confidence interval:

$$\text{Select } \arg\max_{i=1\ldots K} \left(\hat{q}_{i,t} + C \sqrt{\frac{2 \log \sum_k n_{k,t}}{n_{i,t}}} \right) \tag{4}$$

The C parameter, referred to as *scaling* factor, controls the trade-off between exploitation (left term in Eq. (4), favoring the arms with best empirical reward) and exploration (right term, favoring the infrequently tried arms). It is frequently phrased as *"Be optimistic in front of the Unknown"*, choosing the arm that can possibly give the highest reward. The efficiency of the UCB rule follows from the fact that, although every arm

is selected an infinite number of times, the lapse of time between two selections of an under-optimal arm increases exponentially.

The standard MAB framework and the UCB algorithm, however, consider a static environment (the unknown reward probability of any arm is fixed along time), whereas the AOS framework is intrinsically dynamic (the quality of any operator is bound to vary along evolution). Even though every operator keeps being selected from time to time, in practice UCB would need to wait way too long before realizing that some new operator has become the best one.

Dynamic Multi-Armed Bandit: Originally proposed in another dynamic context [21], a restart strategy has been used in [7] to address this limitation, coupling UCB with a change detection test, the statistical Page-Hinkley (PH) test [23]. Basically, the PH test is in charge of checking whether the operator reward distribution has changed; upon its triggering, the UCB process is restarted (i.e. the empirical rewards and confidence intervals are re-initialized) in order to quickly identify the new best operator without being slowed down by now irrelevant information. Formally, the PH test works as follows:

$$\begin{cases} \bar{r}_t = \frac{1}{t}\sum_{i=1}^{t} r_i \qquad e_t = (r_t - \bar{r}_t + \delta) \qquad m_t = \sum_{i=1}^{t} e_i \\ \\ \text{Return } (\max_{i=1\ldots t}\{|m_i|\} - |m_t| > \gamma) \end{cases} \tag{5}$$

where $\bar{r}_t$ denotes the average reward over the last t steps, e_t refers to the difference between the instant and the average reward, plus some small tolerance δ, and the random variable m_t is the accumulation of those differences e_t. The PH test is triggered when the difference between the maximum value of $|m_t|$ in the past and its current value is greater than some user-specified threshold γ.

The PH test is thus parametrized by γ (controlling the test sensitivity and the rate of false alarms) and δ (enforcing the test robustness w.r.t slowly varying environments). Following early experiments, δ has been kept fixed to 0.15 throughout this work.

This combination of an optimal MAB algorithm (UCB) with the Page-Hinkley test has been initially validated on an artificial *Credit Assignment* scenario [7], in order to study the *Operator Selection* rules independently. Fed by the *Extreme* value of fitness improvements, it was later assessed on some EA binary benchmark problems [14–16], and also on some SAT instances [31].

3 Sliding Multi-Armed Bandit

This section first discusses the rationale for the *Multi-Armed Bandit* process, and the strengths and weaknesses of the *Dynamic Multi-Armed Bandit* process presented above. The new *Sliding Multi-Armed Bandit* process is thereafter described and discussed.

3.1 Discussion

The *Multi-Armed Bandit* UCB algorithm has been designed in order to minimize the regret, that is, the loss compared to the cumulative reward obtained by the oracle strategy (always selecting the best arm/operator) [1]. This makes it compulsory to determine *the* best arm (say with reward r): in case the algorithm settles on the second

best arm (say with reward r'), it incurs some loss $r - r'$ at each time step, and its regret increases linearly with the number of time steps. In the meanwhile, unpromising arms are tried exponentially less and less; since the reward distribution is assumed to be stationary, the chances of mistaking the best arm for an unpromising one decreases exponentially with the number of trials.

The main rationale behind the *Multi-Armed Bandit* exploration (trying other arms than the one with best empirical $\hat{q}$) thus is to determine the best arm among the most promising ones.

Adaptive Operator Selection faces quite different priorities. On the one hand, the main need for exploration comes from the dynamics of the environment; one cannot assume the reward distribution to be stationary. Henceforth, mistaking the best and second best operator has moderate consequences as the loss is small (provided r and r' are sufficiently close) compared to the cost of exploration. The point thus becomes to identify *as fast as possible* a sufficiently good operator.

Note that if the reward distribution is not stationary, the *Multi-Armed Bandit* regret cannot but linearly increase with the number of time steps in the worst case. The worst case is when the reward distribution of the supposedly best arm does not change, whereas a previously bad arm covertly becomes the best one. The only way to detect such a worst-case change would be to try all arms sufficiently often, that is, to define a minimal selection probability, along the same lines as PM (Section 2.3.1).

In the evolutionary framework, however, such a worst-case change scenario is unlikely to occur. On the one hand, the average reward of every operator tends to decrease as evolution goes on (diminishing returns). In the One-Max problem, for instance, the best mutation operator is the 5-bit mutation when the population is far away from the optimum; but the reward of the 5-bit mutation gracefully decreases as the population goes to more fit regions, and at some point the 3-bit mutation operator catches up (more details on this can be found in [15]). This suggests that when a good operator has been identified, there is no need for exploration as long as this operator remains sufficiently good.

The above remark motivated the *Dynamic Multi-Armed Bandit* approach, *i.e.*, upon a change detection only, the MAB process is restarted. This restart is made necessary due to the inertia of the reward update: as the weight of the instant reward is inversely proportional to the number of times the operator has been tried, adjusting the estimate of an often used operator takes a long time.

However, the *Dynamic Multi-Armed Bandit* process presents two weaknesses. On the one hand, the change-point detection test is triggered just in the case of an *abrupt* change, whereas the reward of an operator usually decreases gradually. This makes it difficult to calibrate the test. On the other hand, upon triggering the test, the whole memory of the *Multi-Armed Bandit* process is lost, and the exploration must start anew. These remarks motivated the introduction of the *Sliding Multi-Armed Bandit*.

3.2 Using a sliding window

Several heuristics have been proposed to update statistical estimates in a non-stationary context. The most natural one is the relaxation update rule, as involved in the *Probability Matching* and *Adaptive Pursuit* rules (Section 2.3.1), in which the weight of the instant information r_t is set to some constant learning rate α ($0 < \alpha < 1$) :

$$\hat{q}_{i,t+1} = (1 - \alpha)\hat{q}_{i,t} + \alpha r_{i,t}$$

The difficulty with the above rule is that the instant reward $r_{i,t}$ has a constant weight α, regardless of how frequently the i-th operator has been applied in the last time steps. Still, in the *Adaptive Operator Selection* framework different operators are applied with different frequencies; if an operator has not been applied for a long time, the weight of the instant reward should be higher everything else being equal, in order to enable a more rapid adjustment of $\hat{q}_{i,t}$.

The update rule must thus take into account the number of time steps elapsed since the previous time step t_i in which the i-th operator has been applied. Finally, in order to preserve the *Multi-Armed Bandit* trade-off between exploration and exploitation, one must also maintain the $n_{i,t}$ counters reflecting the frequency of application of operators up to time step t.

Considering a window of size W, the sliding exploitation and exploration terms can then be defined as follows:

$$\hat{q}_{i,t+1} = \hat{q}_{i,t} \frac{W}{W+(t-t_i)} + r_{i,t} \frac{1}{n_{i,t}+1}$$
$$n_{i,t+1} = n_{i,t} \left(\frac{W}{W+(t-t_i)} + \frac{1}{n_{i,t}+1} \right)$$

The above update rule is designed in such a way that, if an operator is applied with frequency W/n_t, then n_t is constant[3].

If an operator is performing well and is almost always applied, the counter $n_{i,t}$ rapidly increases up to W and sticks to this value, while its reward estimate $\hat{q}_{i,t}$ accurately reflects the reward expectation for the current stage of the search. The main difference compared to the *Multi-Armed Bandit* and *Dynamic Multi-Armed Bandit* settings is that $n_{i,t}$ is upper bounded by W. Equivalently, the inertia of the reward estimate is bounded: the weight of the instant reward cannot be less than $1/W$.

If an operator is rarely applied, $\hat{q}_{i,t}$ is an outdated, hence optimistic, estimation of the actual reward expectation (assuming that the operator reward decreases on average as evolution goes on). On the other hand, if the operator is rarely applied, it can only be because its reward estimate is lower than that of the best operator. The over-estimation of $\hat{q}_{i,t}$ thus modestly favors its exploration. If, however, the operator has not been tried in the previous W time steps, $n_{i,t}$ is low and the reward estimate rapidly goes toward the instant reward, potentially correcting the over-estimation.

4 Benchmark Problems

While some artificial benchmarks or scenarios have been used for the comparison of *Adaptive Operator Selection* mechanisms [40, 7], their representativeness with respect to actual situations faced by *Adaptive Operator Selection* schemes along evolution remains limited. After describing these early scenarios for the sake of self-containedness, this section discusses some complexity factors hindering *Adaptive Operator Selection* and proposes a comprehensive benchmark generator. Finally, the *Royal Road* [25] is described, an optimization problem intensively investigated in the literature of Evolutionary Computation, which is here used as a real benchmark to be solved by a Genetic Algorithm.

[3] The reward estimate attached to some operator is updated only when the operator has been applied.

4.1 Earlier Artificial Scenarios

Thierens' original benchmark [40] involves a set of 5 operators, in which the reward distribution associated to each operator is constant during an epoch (ΔT time steps). During every epoch, the operator reward is uniformly drawn in some interval: $\{4, 6\}$ for the current best operator, $\{3, 5\}$ for the second best, and so forth, until $\{0, 2\}$ for the worst operator. Note that, since these intervals overlap, the second best operator occasionally gets better rewards than the best one. The reward distributions associated to all operators are permuted at the end of every epoch, using pre-defined permutations[4] to decrease the experimental noise. The performance associated to an *Adaptive Operator Selection* scheme is the cumulative reward obtained during this sequence of 10 epochs.

Within this benchmark, thereafter referred to as *Uniform*, an operator always gets some positive reward. Still, in an actual evolutionary context, an operator would most often bring no improvement at all (after the first generations), thus providing the *Adaptive Operator Selection* with no information whatsoever. For this reason, two variants of the *Uniform* benchmark, respectively referred to as *Boolean* and *Outlier*, were considered in [7].

In the *Boolean* scenario, the best operator gets a reward of 10 with probability 50% (and 0 otherwise). The second best operator gets a reward of 10 with probability 40% and 0 otherwise, and so forth, until the worst operator, getting a reward of 10 with probability 10% and 0 otherwise. In this scenario, operators only differ by their probability of getting a non-null reward; the reward takes the same value in all cases. In particular, the best operator has the same reward expectation than in the Uniform scenario, though with a much higher variance.

Quite the contrary, in the *Outlier* scenario all operators get a non-null reward with the same probability (10%); the difference lies in the reward value, set to 50 for the best operator, 40 for the second best and so forth. While the reward expectation is still the same as in the Uniform benchmark, the *Adaptive Operator Selection* is provided with much less information (only 10% of the trials produce some information) and the reward variance is much higher than in the previous Boolean scenario.

The *Adaptive Operator Selection* ability to match the dynamics of evolution is assessed by varying the length of the epoch, set to $\Delta T = 50$ for fast dynamics and $\Delta T = 200$ for slow ones. As the reward expectation of the best operator is 5 in all scenarios, the maximal cumulative reward is 2,500 in the fast case ($5 \times 10 \times 50$) and 10,000 in the slower one ($5 \times 10 \times 200$).

4.2 Discussion

The operator reward distribution can challenge an *Adaptive Operator Selection* scheme in two different ways. Firstly, by varying the probability of getting some useful information about the actual quality of the operator. This probability is high (for the best operator) in the Boolean benchmark; it is low (for all operators) in the Outlier benchmark. Secondly, by varying the usefulness of the information it provides. Typically, the

[4] These permutations are: $41203 \mapsto 01234 \mapsto 24301 \mapsto 12043 \mapsto 41230 \mapsto 31420 \mapsto 04213 \mapsto 23104 \mapsto 14302 \mapsto 40213$. More precisely, the best operator in the first epoch is the op_4, which becomes the worst one in the second epoch. The best operator in the second epoch is op_0, which was the fourth one in the first epoch.

Boolean scenario does not provide useful information (all rewards have the same values, only the probabilities differ), while the Outlier scenario involves very informative rewards.

In brief, any *Adaptive Operator Selection* is provided with some (more or less) informative results (the reward amount, everything else being equal); and it is more or less likely to be provided with any information at all. Typically, the *Multi-Armed Bandit* process is well equipped to deal with Boolean-like settings, where operators (arms) get the same reward in case of success and only the probability of success differs.

Along these lines, a general frame for *Adaptive Operator Selection* benchmarks, referred to as Two-Values benchmarks, is proposed. Every operator is assumed to get one out of two reward values, a small one noted r and a large one noted R. The informativeness of the reward distribution is defined by the ratio R/r. The reward distribution is specified from r, R and the probability p to get reward R.

Formally, the reward distribution specified from the triple (p, r, R) is defined as:

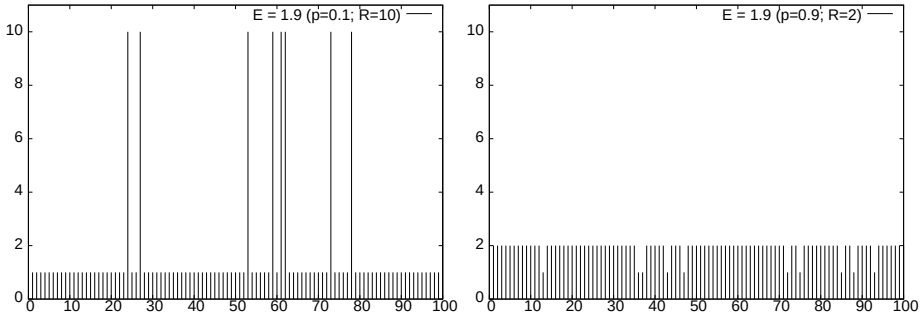
$$\begin{cases} \mathcal{TV}(p, r, R) = R \text{ with probability } p \\ \qquad\qquad\qquad\quad\ r \text{ with probability } 1 - p \end{cases}$$

with expectation and variance respectively noted $\mathbb{E}(p, r, R)$ and $V(p, r, R)$:

$$\begin{aligned} \mathbb{E}(p, r, R) &= pR + (1 - p)r \\ V(p, r, R) &= p(1 - p)(R - r)^2 \end{aligned}$$

In the rest of the paper, r is set to 1 and will be omitted in the notations for the sake of simplicity – a reward distribution will be noted $\mathcal{TV}(p, R)$ instead of $\mathcal{TV}(p, 1, R)$.

The respective roles of p and R are exemplified in Fig. 2, displaying two samples of size 100 of distributions with the same expectation and high versus low variance.



(a) $\mathcal{TV}(0.1, 10)$, $V(.1, 10) = 7.29$ (b) $\mathcal{TV}(0.9, 2)$, $V(.9, 2) = .09$

Fig. 2 Two samples drawn from two Two-Values distributions with same expectation $\mathbb{E}(.1, 10) = \mathbb{E}(.9, .2) = 1.9$; the distribution on the left picture presenting a much higher variance ($V(.1, 10) = 7.29$) then the one on the right ($V(.9, 2) = .09$).

4.3 Two-Values Artificial Scenarios

The experiments presented in Section 6 will aim at answering three questions.

The first one examines what happens when the *Adaptive Operator Selection* faces two operators with same reward expectation. Is there a bias, e.g. does the *Credit Assignment* used (e.g. Instantaneous vs Extreme vs Average) tend to favor the operator with high or low variance? More generally, the point is whether a given *Adaptive Operator Selection* can be considered as a *Risk-taker* (favoring high variance operators) or a *Risk-adverse* (favoring low variance operators) one.

The second question regards the soundness and agility of the *Adaptive Operator Selection* scheme. Specifically, does the candidate *Adaptive Operator Selection* pick up the operator with best reward expectation? Does it catch up after each change of the reward distributions?

The third one investigates the sensitivity of an *Adaptive Operator Selection* with respect to its hyper-parameters. How much attention/effort should be paid for the tuning of each parameter? Identifying the parameters that most affect its performance becomes even more important in case just a limited budget is available for the off-line tuning phase.

In all cases, the experimental setting involves two operators, the reward distributions, which are modified every ΔT time steps (considering short and long epochs, $\Delta T = 50$ and 200, as in Section 4.1). The exchanges done after each epoch will obey a fixed sequence[5] to decrease the experimental noise. Finally, to check how fast each *Adaptive Operator Selection* mechanism can adapt to a new situation after a long period of stability, experiments with 2 longer epochs will also be performed ($\Delta T = 500$ and 2000 respectively), with just one permutation of rewards happening between the two epochs ($01 \rightarrow 10$). The *Uniform*, *Boolean* and *Outlier* scenarios are also used by the agility and the sensitivity analyses, as well as the embedded *Royal Road* scenario, described in the following.

In the following, a scenario involving $\mathcal{TV}(p_1, R_1)$ and $\mathcal{TV}(p_2, R_2)$ will be denoted by $\mathcal{ARTV}(p_1, R_1, p_2, R_2)$. While many different and more general settings could have been used (considering more Two-Values operators, or more complex reward distributions, e.g., taking uniformly drawn values in two intervals centered in the r and R values, or smoother transitions between epochs), this experimental study primarily aims at analyzing the effect of the reward margin R/r and probability p on the *Adaptive Operator Selection* performance. More complex and realistic reward landscapes will be considered in further studies.

4.4 The Royal Road Problem

The *Royal Road* (RR) is an optimization problem that was intentionally created to be easy for GAs [35], with the crossover operators exploring the "building blocks" of the function, while being difficult for hill-climbing algorithms. Due to unexpected difficulties (the so-called hitch-hiking phenomenon), a revised version was later proposed in [24] and analyzed in [25]. The revised version is the one considered in this work.

The solutions are represented as bit-strings. Each bit-string is composed of 2^k regions, referred to as first-level *schemata*. Higher level schemata are formed by combining

[5] as follows: $01 \mapsto 01 \mapsto 10 \mapsto 01 \mapsto 10 \mapsto 10 \mapsto 10 \mapsto 01 \mapsto 01 \mapsto 10$.

lower-level ones; formally, a higher level L has 2^{k-L} schemata composed by 2^L first-level ones (the building-blocks, supposedly defining a crossover-friendly landscape). Each first-level schema is further divided into a *block* and a *gap* string, of respective lengths b and g. A bit-string is thus represented by $2^k \times (b+g)$ bits.

For the calculation of the fitness of a candidate solution, each first-level schema is independently evaluated, with the fitness resulting in the sum of the evaluations of all the schemata. Just the block region of each low level schema is considered, the gap region being completely ignored. The fitness is measured by the PART function or by the BONUS one, as follows. The PART function computes the number z of correct bits in the b-length block, resulting in a function value of $z \times v$ if $z < m$ and $(b-z) \times v$ for $m < z < b$, where m is a threshold that tunes the level of local deception in the function. The completed blocks in the bit-string, *i.e.*, the ones that have $z = b$, are evaluated by the BONUS function instead, which accounts a score of u^* for the first block to be completed, and u for the additional ones.

The *Royal Road* was found to be an interesting scenario to empirically analyze the *Adaptive Operator Selection* combinations within a real evolutionary algorithm as, by considering common crossover operators, the following can be stated (intuitively, and confirmed in [37]): the uniform crossover is the best operator during the initial evolution stages (exploration), 1-point crossover is the best in the final stages (exploitation), while the 4-point crossover is the best in-between. This operator set was used for the experiments presented in the following, adding yet another crossover operator, the 2-point crossover, as well as a disruptive bit-flip mutation operator that flips on average 8 bits (and hence possibly one block); after every crossover application, a mutation operator was also systematically applied, flipping each bit with a probability of 1%. These operators were applied within a (100,100)-GA with weak elitism, *i.e.*, at every generation, the entire population of 100 individuals is completely replaced by the newly generated 100 offspring, with the possible exception of the best parent, which is maintained if better than the best offspring (and the worst offspring is removed).

The problem function was defined using the default parameter values proposed by Holland [24]: $k = 4$, $b = 8$, $g = 7$, $m = 4$, $v = 0.02$, $u^* = 1.0$ and $u = 0.3$. The parameter $m = 4$ defines a medium level of deception; the fully deceptive case ($m = 1$) and the not deceptive one ($m = 7$) were also investigated, but the former was found too difficult to be solved within the given budget of 25,000 generations, while the latter was too easy, thus not allowing any distinction to be made between the *Adaptive Operator Selection* schemes. With 2^k regions involving $(b+g)$ bits, the dimension of the considered search space accounts to 240 bits.

5 Experimental Setting

This section summarizes the different *Adaptive Operator Selection* combinations, which will be experimentally assessed and compared with the *Sliding Multi-Armed Bandit*. Further, the details of the experiments are described together with the performance indicators and plots that will be used in Section 6.

5.1 *Adaptive Operator Selection* schemes and their parameters

An *Adaptive Operator Selection* mechanism is made of an *Operator Selection* and a *Credit Assignment*, as described in Section 2.

The *Sliding Multi-Armed Bandit Operator Selection*, presented in Section 3, has been compared with other *Operator Selection* schemes: *Probability Matching*, *Adaptive Pursuit*, *Multi-Armed Bandit*, and *Dynamic Multi-Armed Bandit* (Section 2.3). The corresponding parameters are listed in Table 1.

Table 1 Components of *Adaptive Operator Selection* schemes studied here, together with their hyper-parameters.

Probability-based *Operator Selection*		
Probability Matching (PM)	α	Adaptation rate
	P_{min}	Minimal value for operator probability
Adaptive Pursuit (AP)	α	Adaptation rate
	β	Learning or "greediness" rate
	P_{min}	Minimal value for operator probability
Bandit-based *Operator Selection*		
Multi-Armed Bandit (MAB)	C	Scaling factor
Dynamic Multi-Armed Bandit (DMAB)	C	Scaling factor
	γ	Threshold for Page-Hinkley test
Sliding Multi-Armed Bandit (SlMAB)	C	Scaling factor
Credit Assignment		
Instantaneous	-	-
Average	W	Window length
Extreme	W	Window length

All *Operator Selection* schemes can be combined with any of the *Credit Assignment* mechanisms (Section 2.2), respectively using the *Instantaneous*, the *Average* or the *Extreme* reward associated to an operator. In the artificial scenarios, the operator reward is drawn from the operator reward distribution (Section 4). In the case of the *Royal Road* problem within a real GA, the reward reflects the fitness improvement of the offspring over its parent (its best parent in the case of crossover operator), although the diversity of the offspring could also be taken into account, as discussed in Section 2.2.

In the *Average* and *Extreme* cases, there is an additional parameter involved, the size of the window W from which the average and the extreme reward are computed. Specifically, the average reward (respectively the extreme reward) associated to an operator is averaged over (resp. is the maximum reward gathered out of) the last W times this operator has been applied.

5.2 Hyper-Parameter Setting

It is generally desirable to compare the *Adaptive Operator Selection* schemes at their best, in the sense that an equal amount of resources should be devoted to tune the hyper-parameters of the new proposed scheme, and those of the baseline schemes. Accordingly, an off-line tuning was performed preliminarily to every experiment in order to determine the best hyper-parameter values in the ranges listed in Table 2.

Table 2 Value Range for *Adaptive Operator Selection* parameters

Parameter	Used by	|values|	Values
P_{min}	PM and AP	4	$\{0, .05, .1, .2\}$
α	PM and AP	4	$\{.1, .3, .6, .9\}$
β	AP	4	$\{.1, .3, .6, .9\}$
C	all MABs	14	$\left\{[1, 5] \times [10^{-4}, ..., 10^{1}], 25, 100\right\}$
γ	DMAB	17	$\left\{[1, 5] \times [10^{-4}, ..., 10^{2}], 25, 250, 1000\right\}$
W	Avg. and Ext.	4	$\{10, 50, 100, 500\}$

In brief, the *Operator Selection* settings include: 16 possible configurations for PM, 64 for AP, 14 for MAB and SlMAB, and 238 for DMAB. The *Credit Assignment* setting involves 4 configurations for the *Extreme* and the *Average*, the possible values for window-size W. For the SlMAB, the same W value was used by its update rules, except on its combination with the *Instantaneous*, when the *Credit Assignment* window size was set to 1, but all the 4 values were also tried for its update rule.

Instead of a complete factorial design of experiments, we used the F-Race off-line parameter tuning method [4] in order to find the optimal values of all hyper-parameters for each *Adaptive Operator Selection* combination on each of the analyzed scenarios. The general idea of Racing techniques is to run all configurations, discarding some of them as soon as there is enough statistical evidence showing that they will not likely be the best one. Racing thus allows one to allocate the computational resources in an optimal way, by focusing on the most promising configurations, and consequently achieving lower variance estimates for them.

After running each configuration for a minimal number of 11 runs, the *Friedman two-way analysis of variance by ranks* statistical test [6] used in F-Race is applied with a confidence level of 95%. The stopping criterion is when a single configuration remains; or all "survivors" have been run on the maximal number of runs, set to 50 in the experiments. In the latter case, the retained configuration is the one with the best mean amongst the survivors, as done in [4] (although different alternatives could be used, *e.g.*, in a critical situation the configuration with best worst case could be considered).

In all cases, 50 runs are launched for the retained configuration and the results presented in this paper are based on statistics over these 50 runs unless otherwise stated.

5.3 Performance indicators and displays

Several complementary views and indicators for analyzing the *Adaptive Operator Selection* performance will be used, distinguishing off-line performance, on-line performance, and empirical cumulative distribution functions.

For the artificial scenarios, the most natural performance indicator is the total gain brought by an *Adaptive Operator Selection* scheme, the *Total Cumulated Reward* (TCR), computed as the sum of the rewards gathered by the algorithm over the com-

plete run. A related but not equivalent indicator, the percentage of times the best operator was selected, is also presented for an illustrative purpose[6].

The off-line performance results associated to each *Adaptive Operator Selection* scheme[7] are represented within a matrix (*e.g.*, Table 3) in which each cell corresponding to a *Credit Assignment* (line) and an *Operator Selection* (row) indicates:

- the average and standard deviation of the TCR;
- the values of the best hyper-parameter configuration determined after the racing procedure (Section 5.2).
- the average percentage of choice of the best[8] operator.

The best results are indicated in bold; the results which are statistically equivalent[9] are displayed with a grey background; small result variations among the different *Adaptive Operator Selection* schemes thus translate into many grey cells. The caption of the table finally indicates the performances of the Oracle strategy (always choosing the best operator) and the Naive strategy (uniformly selecting among all operators). The former corresponds to the best possible performance, while the latter corresponds to the baseline.

The off-line performance, however, does not tell whether an *Adaptive Operator Selection* fails to detect the best operator due to an excess of exploration, or exploitation. In the former case, the *Adaptive Operator Selection* fails to stick to the best operator after the change; in the latter case, it fails to swiftly adapt whenever a change occurs. The on-line performance plots (*e.g.*, Fig. 3) depict for each operator its instant selection rate (averaged over 50 runs) along time. The best parameter configuration is recalled in the captions of the figures. Additionally, on all such plots for DMAB, the small peaks below the x-axis indicate the restarts (still averaged over 50 runs).

For the case embedding the *Adaptive Operator Selection* schemes within a real evolutionary algorithm, applied to the *Royal Road* problem, the results are assessed as the number of generations needed to achieve the optimal solution. Given the high dispersion of the results, it is not meaningful to present the averages and standard deviations: no significantly difference could be found between the performance of all the *Adaptive Operator Selection* schemes, according to the same statistical tests used for the artificial scenarios (see Footnote 9).

Because of this, in order to be able to figure out the difference of performance between the techniques, the empirical distribution of the results over 50 runs is presented, depicted using *Empirical Cumulative Distribution Functions* (ECDFs): for each level of performance (number of generations to achieve the optimum, on x-axis) the percentage of runs reaching this score is indicated (on y-axis; see Fig. 8). Better than standard boxplot diagrams, ECDFs display the full distribution, enabling a fine-grained comparison of the different schemes, accounting for the fact that one scheme might outperform

[6] Two techniques might present the same rate of "best operator selection", although presenting very different cumulated reward due to the difference between the sub-optimal choices done by each of them, the so-called error costs, which are not considered by this measure.

[7] In fact, the *Probability Matching Operator Selection* will be omitted in the rest of the paper due to its poor performances in all scenarios.

[8] In the experiments involving two operators with same expectation and different variances, the operator with highest variance is considered by convention.

[9] According to at least one of both unsigned Wilcoxon rank sum, and Kolmogorov-Smirnov non-parametric tests, applied at confidence level 90%.

another scheme with regard to some quantile performance, despite the fact that it is outperformed with respect to the average or median value.

In Section 6.3, ECDFs are also used to assess the sensitivity of the scheme with respect to its parameters. More precisely, ECDFs aggregating series of runs corresponding to several parameter configurations will be presented for each *Adaptive Operator Selection*; these plots graphically show how fast the performance decreases when leaving the optimal parameter configuration.

6 Experimental results

This section reports on the experimental comparative analysis of all *Adaptive Operator Selection* schemes presented in the paper, combining *Adaptive Pursuit*, *Multi-Armed Bandit*, *Dynamic Multi-Armed Bandit* and *Sliding Multi-Armed Bandit* with *Instantaneous*, *Average* and *Extreme* rewards. The goal of the experiments is to answer the three questions presented in Section 4.3.

The long-term goal indeed is to determine the niche of the diverse *Adaptive Operator Selection* schemes, the operator reward distribution and the dynamics for which they best fit. A first step towards this goal, investigated in Section 6.1, regards the *Adaptive Operator Selection* bias with respect to the reward variance. Specifically, the question is whether and when a given *Adaptive Operator Selection* tends to be *Risk-taker* or *Risk-adverse*, preferring the operators with higher or lower variance everything else being equal. This is answered using a set of Two-Values scenarios in Section 4.3.

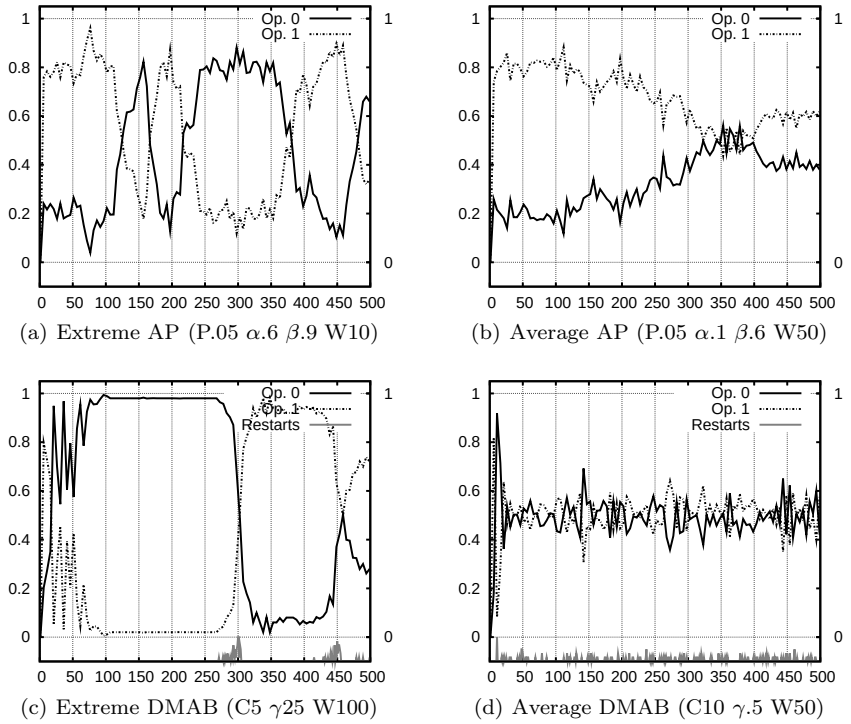
The other two questions, respectively aimed at the general soundness and agility of the schemes (Section 6.2) and their sensitivity to the hyper-parameters (Section 6.3) are answered using some Two-Values scenarios, and also the so-called *Uniform*, *Boolean* and *Outlier* scenarios (presented in Section 4.1), that involve 5 operators. The *Royal Road* problem is also used for both analyses, assessing the performance of the *Adaptive Operator Selection* techniques within a real evolutionary setting, as described in Section 4.4.

6.1 *Adaptive Operator Selection* Bias and Risk-Adversity

As discussed in Section 4.2, the variance of the reward distribution is the most important factor of complexity for every *Adaptive Operator Selection* technique. Accordingly, the behavior of the diverse *Adaptive Operator Selection* schemes when facing two operators with same reward expectation and different variance, is investigated, examining the instant operator selection rate (since the *Total Cumulated Reward* or ECDFs do not bring any information when considering operators with same reward expectation).

Let us consider the artificial benchmark $\mathcal{ART}(0.1, 10, 0.9, 2)$, involving the two reward distributions illustrated in Fig. 2. While both distributions have the same expectation ($\mathbb{E}_1 = \mathbb{E}_2 = 1.9$), the former one has a high variance, $V_1 = 7.29$, while the latter has a low one, $V_2 = 0.09$.

The behavior of the diverse *Adaptive Operator Selection* schemes on this benchmark is depicted in Figs. 3-4; operator op_0 is associated to the high variance reward distribution for the first two epochs (solid black line), and the reward distributions



(a) Extreme AP (P.05 α.6 β.9 W10) (b) Average AP (P.05 α.1 β.6 W50)

(c) Extreme DMAB (C5 γ25 W100) (d) Average DMAB (C10 γ.5 W50)

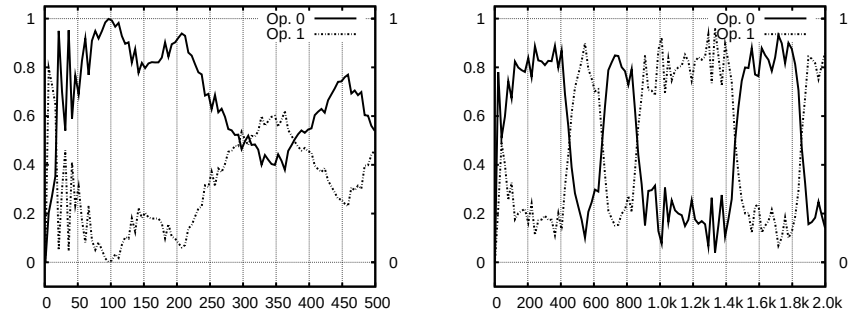
Fig. 3 Behavior of AP and DMAB with Extreme and Average reward on $\mathcal{ART}(0.1, 10, 0.9, 2)$ scenario for $\Delta T = 50$ (averaged over 50 runs). The solid black line depicts the instant selection probability of operator op_0 (high-variance reward during the first two epochs). The small peaks below the x-axis indicate *Dynamic Multi-Armed Bandit* restarts, still averaged over 50 runs.

are swapped between both operators after the second, third, fourth, seventh and ninth epochs.

The behavior of AP with Extreme and Average *Credit Assignment* (Figs 3(a) and 3(b)) can be considered as risk-adverse. The operator associated with the low-variance reward distribution is selected in about 80% of the time. AP quickly adapts to the changes when Extreme *Credit Assignment* is used, always following the low-variance operator, contrasting with the Average *Credit Assignment*, which takes longer to switch and cannot catch up with the changes.

Quite the contrary, *Dynamic Multi-Armed Bandit* with Extreme *Credit Assignment* shows highly risk-taker at the beginning (mostly using the high-variance operator during the first two epochs) and stubborn thereafter (sticking to the same operator although it has become low-variance, Fig. 3(c)). Meanwhile, *Dynamic Multi-Armed Bandit* with Average *Credit Assignment* shows an unbiased behavior, selecting both operators with equal probability. Interestingly, except when combined with AP (Fig. 3(b)) the Average *Credit Assignment* leads to an unbiased behavior. The same holds for the Instantaneous *Credit Assignment* mechanism.

The behavior of *Multi-Armed Bandit* with Extreme *Credit Assignment* is depicted on Figs. 4(a) and 4(b) for an epoch length ΔT respectively set to 50 and 200. *Multi-Armed Bandit* also shows a clear risk-taker bias; when dealing with a fast dynamics



(a) Extreme MAB (C5 W10) $\Delta T = 50$ (b) Extreme MAB (C50 W50) $\Delta T = 200$

Fig. 4 Behavior of MAB with Extreme reward on $\mathcal{ART}(0.1, 10, 0.9, 2)$ scenario for $\Delta T = 50$ (left) and $\Delta T = 200$ (right), averaged over 50 runs. The solid black line depicts the instant selection probability of operator op_0 (high-variance reward during the first two epochs).

($\Delta T = 50$) however, it is hardly able to follow the changes. For a slow dynamics ($\Delta T = 200$), *Multi-Armed Bandit* follows the changes almost perfectly and sticks to the high-variance operator. The same trends hold for the *Sliding Multi-Armed Bandit*.

Complementary experiments done ($\mathcal{ART}(0.1, 10, 0.6, 2.5)$, $\mathcal{ART}(0.1, 10, 0.225, 5)$ and $\mathcal{ART}(0.1, 10, 0.1385, 7.5)$) show that the differences between behaviors tend to vanish when the difference between the variances goes to 0, as could have been expected.

In summary, the lessons drawn from these experiments are the following. For all bandit-based *Operator Selection* rules, the Extreme *Credit Assignment* mechanism is risk-taker (biased toward high variance reward operators); the Average *Credit Assignment* is unbiased. For AP, the Extreme *Credit Assignment* mechanism is risk-adverse (biased toward low variance reward operators); the Average *Credit Assignment* equally is risk-adverse, though less so.

Additionally, no scheme proved able to swiftly follow the changes, which might be explained as both operators have the same reward expectation. Further experiments are considered in the following Section to confirm/infirm this tentative explanation.

This analysis explains previously published results in the context of the *Long 3-Path* problem [16], in which the Extreme - *Dynamic Multi-Armed Bandit* (embedded within a (1+50)-GA with 4 different mutation operators) greatly outperformed the probability-based approaches. In such a scenario, being a risk-taker in the initial stages can be highly beneficial, as follows: the 1-bit mutation (flips one randomly chosen bit) will very probably increase the fitness by 1 each time it is applied, while the 3-bit mutation has a much higher chance of taking the solution out off the path; however, with a minimal probability, "shortcuts" can be taken by the application of the latter, what provides very large fitness gains, possibly taking a shortcut from the early stages directly to the optimal value.

6.2 *Adaptive Operator Selection*, Soundness and Agility

Several Two-Values scenarios (Section 4.3) involving two operators with different expectations and variances, the *Uniform*, *Boolean* and *Outlier* scenarios (which involve 5 operators), and the *Royal Road* problem within a real GA, are considered here.

6.2.1 Soundness and Agility on the Two-Values Scenarios

The $\mathcal{ART}(0.01, 101, 0.5, 10)$ problem involves a low average/high variance distribution ($\mathbb{E}_1 = 2$, $V_1 = 99$) and a high average/low variance distribution operator ($\mathbb{E}_2 = 6$ and $V_2 = 20.25$. The fact that the high-variance operator also is the one with lower expectation should make it easy to be discarded.

The overall results (Table 3) show that *Dynamic Multi-Armed Bandit* and *Sliding Multi-Armed Bandit* significantly outperform *Multi-Armed Bandit* and *Adaptive Pursuit* in terms of TCR. *Sliding Multi-Armed Bandit* has a slight edge due to the many restarts of *Dynamic Multi-Armed Bandit* (plots not shown). It seems that, as far as the operators have distinct variances, the diverse *Operator Selection* rules do not matter, considering long or short time dynamics.

The AP scheme hardly deals with the *Exploration vs Exploitation* issue, being hindered by the minimum selection rate in the fast dynamics case ($p_{min} = .05$ for $\Delta T = 50$). For slower dynamics, no minimum selection rate is retained by the Racing procedure ($p_{min} = 0$ for $\Delta T = 500$), but AP still remains dominated by *Dynamic Multi-Armed Bandit* and *Sliding Multi-Armed Bandit*, taking more iterations to adapt to the changes. For very slow dynamics ($\Delta T = 2,000$), there is no longer any significant differences between bandit-based and AP schemes.

Table 3 $\mathcal{ART}(0.01, 101, 0.5, 10)$ scenario, $\Delta T = 200$ (Naive = 8000, Optimal = 12000).

Reward	SlMAB	DMAB	MAB	AP
Inst.	10656 ± 226 $C5$ $W10$ 5.1 ± 1.40	10503 ± 223 $C5$ $\gamma 50$ $W1$ 7.1 ± 1.33	10182 ± 243 $C10$ $W1$ 11.6 ± 1.94	10200 ± 308 $P.05$ $\alpha.9$ $\beta.9$ $W1$ 11.6 ± 3.38
Ext.	**10680 ± 216** **$C10$ $W10$** **4.6 ± 1.73**	10636 ± 440 $C1$ $\gamma.5$ $W10$ 5.5 ± 6.67	9322 ± 1010 $C10$ $W10$ 23.9 ± 15.52	10149 ± 300 $P.05$ $\alpha.9$ $\beta.9$ $W10$ 12.4 ± 3.26
Avg.	10637 ± 229 $C5$ $W10$ 5.2 ± 1.38	10483 ± 262 $C1$ $\gamma 10$ $W10$ 7.5 ± 2.66	10063 ± 260 $C10$ $W10$ 13.5 ± 2.34	10159 ± 303 $P.05$ $\alpha.9$ $\beta.9$ $W10$ 12.3 ± 3.27

Let us consider the opposite situation, illustrated by $\mathcal{ART}(0.1, 39, 0.5, 3)$. This problem involves a high average/high variance distribution ($\mathbb{E}_1 = 4.8$, $V_1 = 130$) and a low average/low variance distribution operator ($\mathbb{E}_2 = 2$ and $V_2 = 1$). Other problems with similar characteristics have been investigated with same empirical results and will be omitted for the sake of brevity.

Tables 4(a)-4(d) display the overall behavior of all *Adaptive Operator Selection* schemes for $\Delta T \in \{50, 200, 500, 2000\}$. In all cases, the best *Adaptive Operator Selection* is *Dynamic Multi-Armed Bandit* with Extreme reward. Broadly, *Dynamic Multi-Armed Bandit* and *Sliding Multi-Armed Bandit* dominate *Multi-Armed Bandit* (even more in terms of best operator choice than in terms of *Total Cumulated Reward*), and all bandit-based approaches very largely dominate AP. The Extreme-based *Credit Assignment* performs slightly better than the Average and Instantaneous, especially for large ΔT.

Interestingly, the improvement over the Naive strategy increases with ΔT, which is explained as the relative importance of the transient regime decreases. Notably, both *Dynamic Multi-Armed Bandit* and *Sliding Multi-Armed Bandit* reach a quasi-perfect score when $\Delta T = 2000$ (Table 4(d)), being reminded that only two epochs are

considered for $\Delta T \in \{500, 2000\}$, whereas ten epochs are considered for $\Delta T \in \{50, 200\}$ (Section 4.3).

Table 4 Results on the $\mathcal{ART}(0.1, 39, 0.5, 3)$ scenario for all 4 analyzed dynamics

(a) $\mathcal{ART}(0.1, 39, 0.5, 3)$ scenario ΔT=50 (Naive = 1700, Optimal = 2400)

Reward	SlMAB	DMAB	MAB	AP
Inst.	1980 ± 246 C25 W100 67.2 ± 6.07	1869 ± 207 C25 γ500 W1 61.5 ± 4.23	1869 ± 207 C25 W1 61.5 ± 4.23	1724 ± 224 P.2 α.1 β.3 W1 51.5 ± 7.01
Ext.	2012 ± 301 C100 W10 71.1 ± 5.87	**2040 ± 238** **C100 γ1000 W10** **69.4 ± 4.52**	2036 ± 239 C100 W10 69.5 ± 4.51	1793 ± 252 P.1 α.1 β.6 W10 54.2 ± 9.08
Avg.	1890 ± 245 C25 W10 62.8 ± 4.77	1946 ± 247 C10 γ100 W10 63.3 ± 6.20	1894 ± 228 C10 W10 61.8 ± 6.54	1761 ± 229 P.1 α.6 β.6 W50 53.0 ± 7.99

(b) $\mathcal{ART}(0.1, 39, 0.5, 3)$ scenario, ΔT=200 (Naive = 6800, Optimal = 9600)

Reward	SlMAB	DMAB	MAB	AP
Inst.	8337 ± 537 C10 W50 77.5 ± 3.78	8232 ± 475 C10 γ100 W1 74.2 ± 2.95	8009 ± 549 C25 W1 71.0 ± 3.45	7277 ± 639 P.2 α.1 β.1 W1 59.1 ± 5.00
Ext.	8746 ± 511 C100 W50 84.3 ± 2.94	**8834 ± 451** **C25 γ5 W50** **85.8 ± 1.64**	8555 ± 530 C100 W10 80.5 ± 3.28	7931 ± 606 P.1 α.3 β.6 W50 70.6 ± 5.27
Avg.	8522 ± 497 C10 W50 79.4 ± 3.35	8372 ± 545 C5 γ25 W50 77.3 ± 3.71	8160 ± 466 C10 W10 73.0 ± 4.35	7795 ± 741 P.1 α.6 β.3 W50 67.9 ± 7.43

(c) $\mathcal{ART}(0.1, 39, 0.5, 3)$ scenario, ΔT=500 (Naive = 3400, Optimal = 4800)

Reward	SlMAB	DMAB	MAB	AP
Inst.	4357 ± 312 C25 W500 84.6 ± 4.98	4265 ± 506 C5 γ250 W1 79.2 ± 12.74	4202 ± 431 C10 W1 77.2 ± 8.59	3737 ± 402 P.2 α.1 β.1 W1 61.6 ± 5.56
Ext.	4590 ± 324 C50 W50 91.8 ± 1.92	**4597 ± 339** **C25 γ250 W50** **92.0 ± 4.26**	4421 ± 297 C100 W50 85.8 ± 3.46	4148 ± 397 P.1 α.1 β.3 W50 76.1 ± 7.76
Avg.	4309 ± 371 C5 W50 80.7 ± 6.38	4377 ± 366 C5 γ100 W10 83.6 ± 7.28	4175 ± 400 C10 W10 76.5 ± 8.14	4116 ± 481 P.05 α.6 β.1 W100 74.7 ± 11.43

(d) $\mathcal{ART}(0.1, 39, 0.5, 3)$ scenario, ΔT=2000 (Naive = 13600, Optimal = 19200)

Reward	SlMAB	DMAB	MAB	AP
Inst.	17725 ± 835 C5 W50 86.6 ± 3.65	18261 ± 928 C5 γ500 W1 91.2 ± 5.43	17994 ± 848 C10 W1 89.0 ± 4.06	15021 ± 860 P.2 α.1 β.1 W1 62.5 ± 3.40
Ext.	18970 ± 716 C50 W50 97.7 ± 0.64	**18992 ± 714** **C25 γ100 W50** **97.9 ± 0.75**	18606 ± 743 C100 W50 94.4 ± 1.86	17901 ± 1083 P.05 α.1 β.9 W50 87.8 ± 6.21
Avg.	18169 ± 755 C5 W50 90.4 ± 2.45	18310 ± 865 C5 γ250 W50 92.0 ± 5.44	17962 ± 837 C10 W10 88.7 ± 3.86	17873 ± 1272 P.05 α.9 β.1 W100 87.7 ± 7.14

The differences among the performances can be explained from the behavioral plots (Figs. 5-6). For all *Operator Selection* rules, the Average *Credit Assignment* adapts

much more slowly to the changes than the Extreme one. Similarly, when combining the Extreme *Credit Assignment* with different *Operator Selection*, it appears that AP reacts slowly, even compared to MAB.

Quite the contrary, *Dynamic Multi-Armed Bandit* and *Sliding Multi-Armed Bandit* react quite rapidly to the changes, as shown by the steep slopes in Figs. 5(e) and 5(g), respectively. This good behavior of *Dynamic Multi-Armed Bandit* is explained by the well-positioned restarts (see the grey traces below the x-axis on Fig. 5(e)). For *Sliding Multi-Armed Bandit*, which does not use any change detection test and involves one parameter less than the *Dynamic Multi-Armed Bandit*, this robust pursuit behavior is good news. Both *Dynamic Multi-Armed Bandit* and *Sliding Multi-Armed Bandit* found the window size of 50 to be optimal, and both reach almost 100% of use of the optimal 'operator', except during transition phases, or, more suprisingly, during the very first epoch for *Sliding Multi-Armed Bandit*. Note also that the Average strategy for DMAB results in too many restarts, that hinder the global search for optimal reward.

This analysis is confirmed by the results obtained for a very slow dynamics ($\Delta T = 2000$), where the reward distribution is changed once after 2000 time steps, as shown by Figs. 6(a)-6(d). AP (Fig. 6(a)) needs quite some time to focus on the best operator; additionally it does not use it 100% due to the value of $p_{min} = 0.05$. Even MAB (Fig. 6(b)) switches faster than AP. In the meanwhile, *Dynamic Multi-Armed Bandit* (due to the well-tuned PH test) and *Sliding Multi-Armed Bandit* (thanks to its sliding window and the associated forgetting mechanism) select the best operator about 98% of the times (Table 4(d), Figs. 6(c) and 6(d)).

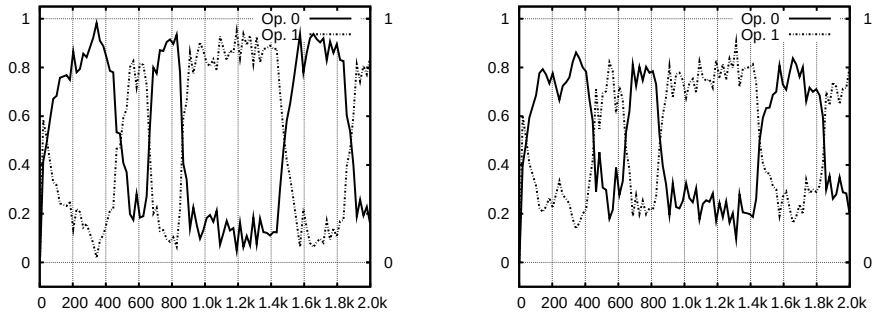
6.2.2 Soundness and Agility on the Uniform, Boolean, and Outlier Scenarios

Let us examine the soundness and agility of the *Adaptive Operator Selection* schemes when considering five instead of two operators. All three Uniform, Boolean and Outlier scenarios have been detailed in Section 4.1.

For the **Uniform scenario**, DMAB with *Instantaneous* reward is a clear winner for all epoch lengths ($\Delta T = 50, 200, 500$ or 2000). This outstanding performance ($TCR = 99.7\%$ of the Oracle *Total Cumulated Reward*; the best operator is selected about 99.5% of the times) is explained from the well calibrated PH test (Fig. 7). For faster dynamics (ΔT goes to 50), the performance is gracefully degraded (*Total Cumulated Reward* becomes 98%, 95% and 77% respectively for ΔT 500, 200 and 50) due to the extra exploration needed after each restart.

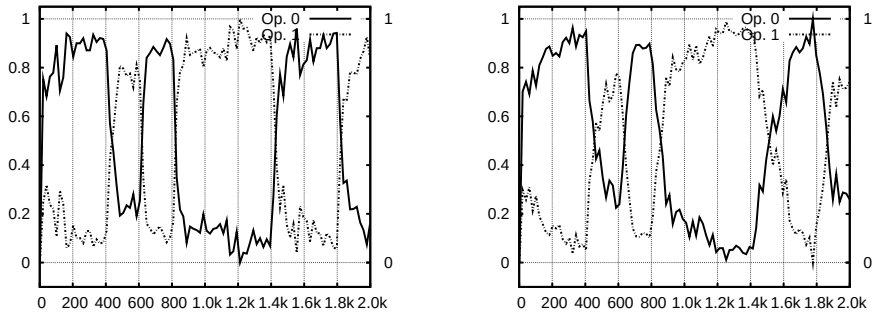
In all cases, *Adaptive Pursuit* is significantly outperformed by bandit-based *Adaptive Operator Selection* mechanisms; this limitation is blamed on an excess of exploration (lower bounded by the p_{min} parameter; the behavior plots are omitted for the sake of brevity). Meanwhile, *Multi-Armed Bandit* and *Sliding Multi-Armed Bandit*, both require a large scaling factor in order to enforce a sufficient exploration.

Finally, for all epoch lengths and all *Operator Selection*s, the *Instantaneous* reward gives the best results, though the difference with the other *Credit Assignment* variants is rather small. A tentative interpretation is that the number of operators (5 in the Uniform, Boolean and Outlier scenarios), the subtle overlapping difference between their performances, and the number of changes, make it necessary to build operator reward estimates as accurately as possible. In the Extreme or Average cases, for instance, up to W steps are needed before adjusting the operator estimates, thus entailing a significant delay for the *Adaptive Operator Selection* adaptation. This interpretation is supported by the fact that the best configuration retained by the Racing process



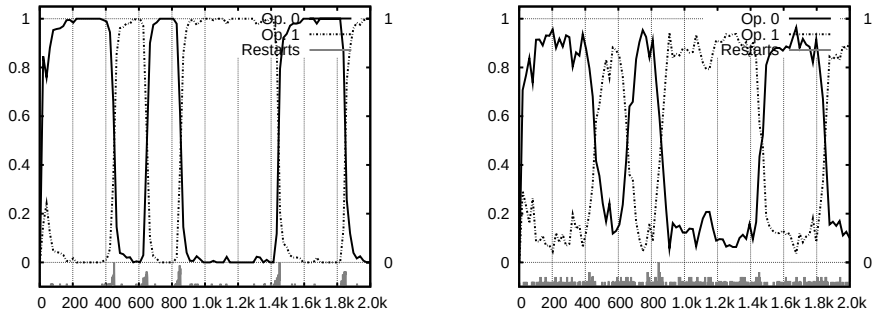
(a) Extreme AP (P.1 α.3 β.6 W50)

(b) Average AP (P.1 α.6 β.3 W50)



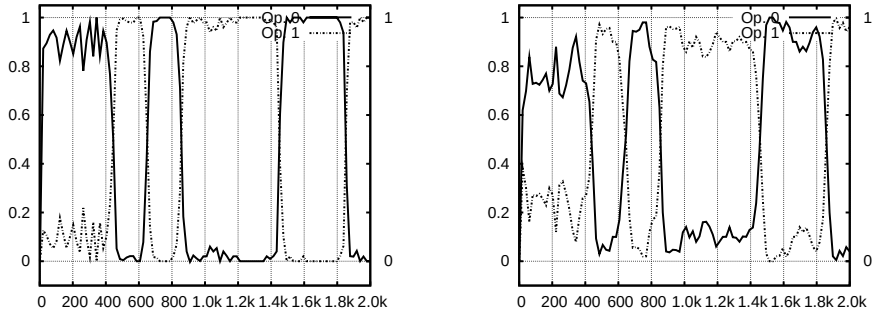
(c) Extreme MAB (C100 W10)

(d) Average MAB (C10 W10)



(e) Extreme DMAB (C25 γ5 W50)

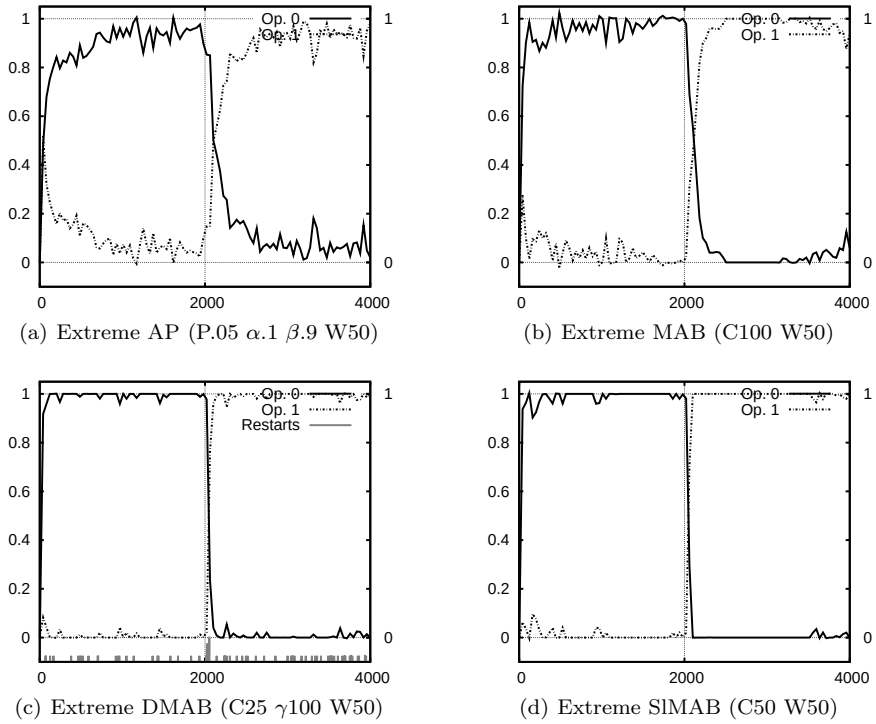
(f) Average DMAB (C5 γ25 W50)



(g) Extreme SlMAB (C100 W50)

(h) Average SlMAB (C10 W50)

Fig. 5 Behavior of AP, MAB, DMAB and SlMAB combined with *Extreme* and *Average Credit Assignment*, on $\mathcal{ART}(0.1, 39, 0.5, 3)$ for $\Delta T = 200$. The solid black line depicts the instant selection probability of operator op_0 (high-variance reward during the first two epochs). The small peaks below the x-axis of *Dynamic Multi-Armed Bandit* plots indicate restarts, still averaged over 50 runs.



(a) Extreme AP (P.05 α.1 β.9 W50)

(b) Extreme MAB (C100 W50)

(c) Extreme DMAB (C25 γ100 W50)

(d) Extreme SlMAB (C50 W50)

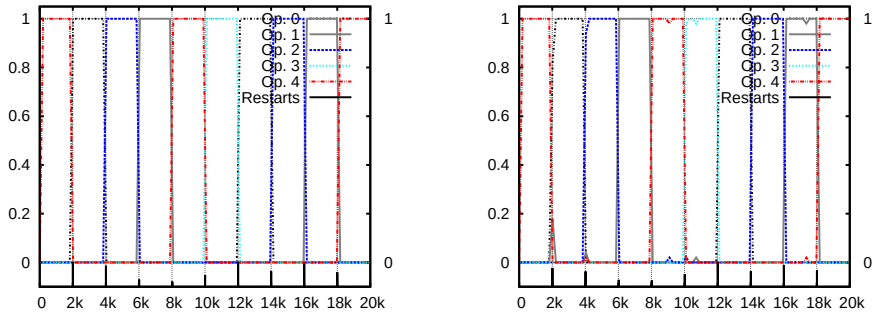
Fig. 6 Behavior of Extreme AP, MAB, DMAB and SlMAB on $\mathcal{ART}(0.1, 39, 0.5, 3)$ for $\Delta T = 2000$. The solid black line depicts the instant selection probability of operator op_0 (high-variance reward during the first two epochs). The small peaks below the x-axis of *Dynamic Multi-Armed Bandit* plots indicate restarts, still averaged over 50 runs.

for all combinations involving parameter W sets it to 10, *i.e.*, the lowest value in the parameter range tried.

Table 5 Uniform Scenario, ΔT=2000 (Naive = 60000, Optimal = 100000).

Reward	SlMAB	DMAB	MAB	AP
Inst.	95872 ± 2745 $C5$ $W10$ 83.3 ± 13.10	**99781 ± 72** **$C.1$ γ10 $W1$** **99.5 ± 0.03**	96954 ± 128 $C5$ $W1$ 92.2 ± 0.29	89617 ± 156 $P.05$ $\alpha.3$ $\beta.9$ $W1$ 78.5 ± 0.46
Ext.	94716 ± 613 $C10$ $W10$ 85.6 ± 1.90	99492 ± 79 $C.5$ $\gamma10$ $W10$ 98.9 ± 0.09	96658 ± 99 $C5$ $W10$ 91.7 ± 0.23	89560 ± 154 $P.05$ $\alpha.9$ $\beta.9$ $W10$ 78.9 ± 0.40
Avg.	95135 ± 663 $C10$ $W10$ 86.2 ± 2.38	97915 ± 985 $C1$ $\gamma25$ $W10$ 91.6 ± 4.88	96332 ± 330 $C5$ $W10$ 91.1 ± 0.74	89242 ± 174 $P.05$ $\alpha.9$ $\beta.9$ $W10$ 76.9 ± 0.64

On the **Boolean scenario**, the situation is slightly different for small values of ΔT, where all *Adaptive Operator Selection* perform equally poorly: For $\Delta T = 50$, only 47% of good choices for SlMAB, less than 40% for DMAB and MAB, and less than



(a) Instantaneous DMAB (C.1 γ10 W1) (b) Extreme DMAB (C.5 γ10 W10)

Fig. 7 Behavior of DMAB on the Uniform scenario combined with *Instantaneous* and *Extreme Credit Assignment*: restarts are perfectly triggered by the PH test in the transitions, as indicated by the small peaks below the x-axis.

30% for AP; with SlMAB being slightly (but not statistically significantly) better than the others, and AP a little worse.

When ΔT increases, DMAB gradually becomes significantly better than the others, and its overall performance increases. Interestingly, for $\Delta T = 200$ (Table 6), SlMAB still has the better proportion of best-operator-choice (almost 60% vs 52% for DMAB) though its TCR is significantly less than that of DMAB. And for larger values of ΔT, DMAB is significantly better than all other techniques.

As for the Uniform scenario, the best *Credit Assignment* is the *Instantaneous* reward; the *Average* performs almost as good; while the *Extreme* reward is outperformed by far: the only values that are taken here are 0 or 10, which makes it difficult for the Extreme reward to distinguish among operators. On the other hand, the changes in averages (or in $\hat{q}$ in Eq. (4)) are slow: DMAB performs very few restarts (the PH test is almost never triggered), and MAB performs almost as good here as DMAB.

Again similarly to the Uniform scenario, all bandit-based techniques outperform AP, although the p_{min} value is set to 0 most of times (all the time for $\Delta T = 500$ or 2000). The behavior plots (omitted here) suggest that AP fails to identify the best 'operator' in all cases in which the best becomes the second best in the following epoch, due to its high inertia.

Table 6 Boolean scenario, ΔT=200 (Naive = 6000, Optimal = 10000)

Reward	SlMAB	DMAB	MAB	AP
Inst.	**8167 ± 271** C**25** W**500** **59.6 ± 6.27**	8164 ± 338 $C5$ $\gamma250$ $W1$ 51.1 ± 8.37	8154 ± 348 $C5$ $W1$ 50.8 ± 8.58	7966 ± 249 $P.05$ $\alpha.1$ $\beta.6$ $W1$ 47.5 ± 6.07
Ext.	7414 ± 420 $C10$ $W10$ 30.5 ± 9.45	7451 ± 502 $C.5$ $\gamma10$ $W10$ 31.1 ± 9.44	7402 ± 429 $C5$ $W10$ 31.3 ± 5.46	7477 ± 526 $P0$ $\alpha.9$ $\beta.9$ $W10$ 31.4 ± 9.66
Avg.	7979 ± 380 $C5$ $W10$ 46.5 ± 9.63	8122 ± 265 $C1$ $\gamma5$ $W10$ 48.3 ± 6.44	7921 ± 313 $C5$ $W10$ 45.7 ± 6.15	7871 ± 271 $P.05$ $\alpha.9$ $\beta.6$ $W10$ 45.0 ± 4.98

On the **Outlier scenario**, all techniques perform very poorly for small values of ΔT. This is not a surprise, because of the small chance of seeing some outlier reward within 50 or even 200 time steps. Best TCR for instance is 1722 for $\Delta T = 50$ (naive would reach 1500) and 7560 (vs 6000 for the naive strategy) for $\Delta T = 200$; the situation changes as ΔT increases. For $\Delta T = 2000$ (Table 7), DMAB and MAB reach about 90% of the maximum TCR with a slight advantage to MAB, outperforming SlMAB and AP. For $\Delta T = 500$ however, AP is not significantly worse than DMAB and MAB.

Here, the Extreme *Credit Assignment* becomes clearly the best, as could be expected: when an outlier value is triggered, the Extreme *Credit Assignment* will maintain this operator in some top position longer than any other strategy. Interestingly, the Instantaneous reward is clearly a complete disaster for AP, while maintaining a fair level of performance (at least similar to that of the Average reward) for the bandit-based techniques: some average actually takes place in the computation of $\hat{q}$ in Eq. (4), keeping some memory of the outlier value, while the benefit of such value vanishes more rapidly within the two-tiered mechanism of AP.

Table 7 Outlier scenario, $\Delta T = 2000$ (Naive = 60000, Optimal = 100000)

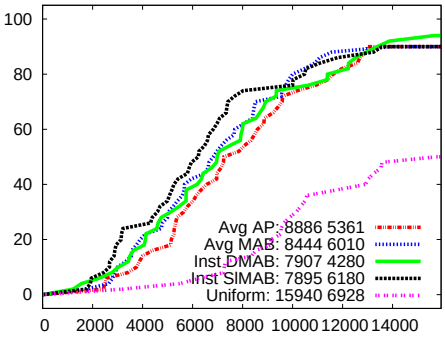
Reward	SlMAB	DMAB	MAB	AP
Inst.	79965 ± 3385 $C25\ W500$ 48.8 ± 7.08	84169 ± 3217 $C10\ \gamma500\ W1$ 55.8 ± 7.93	81842 ± 2705 $C10\ W1$ 50.4 ± 7.78	69735 ± 2054 $P.05\ \alpha.1\ \beta.6\ W1$ 29.5 ± 2.51
Ext.	86372 ± 2602 $C100\ W50$ 60.8 ± 9.15	91622 ± 2673 $C50\ \gamma500\ W50$ 76.1 ± 5.96	**92119 ± 1982** C**100** W**50** **81.2 ± 2.40**	86595 ± 2035 $P.05\ \alpha.9\ \beta.1\ W50$ 70.6 ± 3.07
Avg.	80723 ± 3381 $C5\ W50$ 41.7 ± 10.61	84601 ± 2580 $C10\ \gamma500\ W10$ 56.7 ± 6.57	81949 ± 2759 $C10\ W10$ 51.3 ± 7.71	79059 ± 2394 $P.05\ \alpha.6\ \beta.1\ W100$ 44.5 ± 5.72

6.2.3 Soundness and Agility on the Royal Road Problem

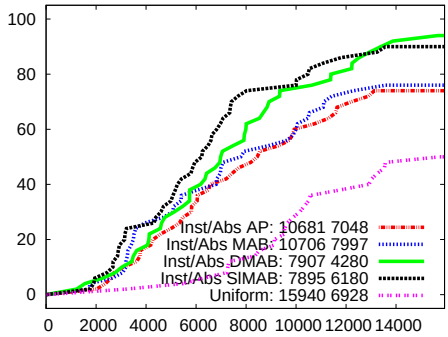
The *Royal Road* problem is used here to assess the *Adaptive Operator Selection* techniques within a real evolutionary algorithm, a *(100,100)-GA*, as described in Section 4.4. The objective is to autonomously and efficiently select among the following operators: 1-Point, 2-Points, 4-Points and Uniform crossover operators, and 1/30 mutation operator (that flips 8 bits on average); the impact of the application of each operator is measured by the fitness improvement achieved by the newborn offspring compared to that of its best parent.

The main difference with the artificial scenarii is that the behavior of each operator is very difficult to guess: besides being real evolutionary operators (crossover and mutation), their performance is related to the region of the landscape that is being locally explored by each of the 100 parents. Despite the 1/30 mutation operator, the other 4 crossover operators (especially the n-point ones) tend to present a similar behavior, all exploring the building blocks of the intentionally designed search space. Besides, there is no "fine-tuning operator" between them: even the 1-point crossover substantially modifies the solution to which it is applied to, what makes it easier to miss the target, thus explaining the high variance of the results.

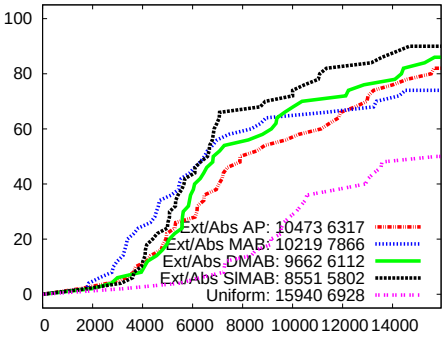
The comparison of the *Adaptive Operator Selection* schemes under scrutiny is thus only presented by means of *Empirical Cumulative Distribution Functions* (see Section



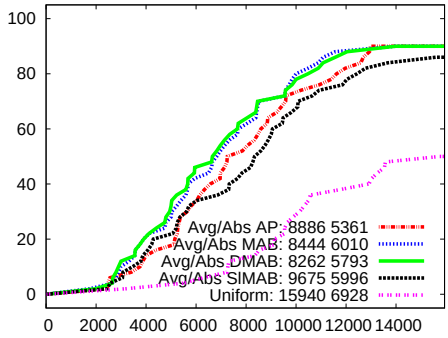
(a) Comparison of each *Operator Selection* rule with its best *Credit Assignment*.



(b) Comparison of each *Operator Selection* rule with the *Instantaneous Credit Assignment*.



(c) Comparison of each *Operator Selection* rule with the *Extreme Credit Assignment*.



(d) Comparison of each *Operator Selection* rule with the *Average Credit Assignment*.

Fig. 8 ECDF performance plots for the Royal Road scenario.

5.3), which shows in a very detailed way the performance of each technique (as well as that of the naive uniform approach). Fig. 8(a) somehow summarizes the results by comparing all *Operator Selection* rules at its best, i.e., associated with the *Credit Assignment* that gave the best results with it. In Figs. 8(b)-8(d), the *Operator Selection* rules are analyzed using the *Instantaneous*, *Extreme*, and *Average Credit Assignment* schemes. The performance averages and standard deviations of each technique are recalled in the legends of the ECDF figures.

The best *Adaptive Operator Selection* combination was found to be the *Instantaneous* - SlMAB, very closely followed by the *Instantaneous* - DMAB. By considering just the *Extreme* values, SlMAB was also the best, while DMAB achieved the best performance among all *Operator Selection* using *Average* values. MAB and AP showed their best performance using *Average* values, generating rather poor results with the other two *Credit Assignment*. Although no significant difference could be found between them (given the high variances of the results), an improvement in the order of 2 was achieved with relation to the average performance of the naive uniform approach.

These results extend those published in [16] by considering also the newly proposed *Sliding Multi-Armed Bandit*. Another difference is the use of ECDFs, that we believe gives another point of view on the results. We refer the reader to [14,15,31,16] for other

comparative results of *Adaptive Operator Selection* applied to different binary and SAT problems, for some of the *Adaptive Operator Selection* combinations analyzed in the present work.

6.3 Sensitivity Analysis

As discussed in Section 5.1, the *Adaptive Operator Selection* mechanisms analyzed in this work have their own (hyper-)parameters (Table 1), which values might critically impact their performance. This issue has been addressed using the F-Race described in Section 5.2, and the best hyper-parameter combination for each considered *Adaptive Operator Selection* has been determined and further used on the empirical comparisons. It is, however, important to study the sensitivity of these hyper-parameters, in order to identify which parameters should receive more attention when addressing a new problem.

When only 1 or 2 parameters are concerned, a 3-D plot of the response surface of these parameters would give a clear picture (as has been done in [7] for AP and DMAB on Thierens' scenarii, for instance). However, by including the window size W, the common parameter for *Average* and *Extreme Credit Assignment*, AP and DMAB involve, respectively, 4 and 3 parameters. This is why ECDFs plots will be also used for this analysis, as described in Section 5.3.

6.3.1 ECDF Sensitivity Plots

Though ECDF sensitivity plots have been generated for all *Adaptive Operator Selection* schemes and all scenarios presented before, only 3 series of plots offering typical behavior will be presented in detail here: Fig. 9 displays the ECDF sensitivity plots for all *Operator Selection* rules on the $\mathcal{ART}(0.1, 39, 0.5, 3)$ scenario for ΔT200, Fig. 10 considers the *Uniform* scenario for ΔT500, and Fig. 11 refers to the *Royal Road* scenario. All of them consider the *Operator Selection* rules with the *Extreme Credit Assignment*. The *Extreme Credit Assignment* was chosen because it performs best in most cases. For a few exceptions, the *Instantaneous* performed slightly better, but the corresponding ECDF plots looked rather similar, though involving one parameter less, the size W of the rewards window (expect for SlMAB, which needs W anyway).

All sub-figures display a number of ECDFs representing the results of the same *Adaptive Operator Selection* on the same scenario. For the artificial scenarios, the x-axis represents the TCR, ranging from the value of the optimal strategy (*i.e.*, the highest possible value on average for any *Adaptive Operator Selection*) **down to** the value that would, on average, be gathered by the naive uniform strategy (hopefully the lowest possible value on average for any *Adaptive Operator Selection*). For the *Royal Road* case, the x axis represents the number of generations to achieve the optimum, starting from zero (the optimal strategy is not known) **up to** the average number of generations taken by the naive uniform choice. The y-axis shows the proportion of runs that reached the corresponding x value, out of 50 runs: even the parameter combinations that did not make it through the end of the Racing procedure have been run 50 times for the sake of the sensitivity analysis.

Two lines are given as references on each plot: the top-left-most line (continuous, and blue on color printouts), labelled with a "**" and referred to as "Best/All",

represents the overall best results, in terms of average TCR (or number of generations to the optimum in the *Royal Road* case), obtained on this scenario by a single hyper-parameter configuration, between all the *Adaptive Operator Selection* combinations considered. The next line going down/right, labelled with a "*" and referred to as "Best/This", represents the results obtained by the best combination of hyper-parameters for the *Adaptive Operator Selection* scheme named on the caption of the sub-figure, under the plot. The discrepancy between both lines shows in detail how different are the performances of the considered *Adaptive Operator Selection* and of the one that performed best on this scenario.

At the other extreme of each sub-figure, the bottom-right-most line represents the ECDF of all runs for the particular *Adaptive Operator Selection*, *i.e.*, for all hyper-parameter combinations ever tried (the average performance of the factorial DOE from values given on Table 2). The lines in-between represent partial aggregations of these runs. More precisely, if the best results have been obtained for a given set of hyper-parameters (recalled in the legend), each line represents the ECDF obtained when one hyper-parameter is varied over all its values used in the racing procedure, while all others are kept to the optimal value found. If a line corresponding to a given parameter is close to the line of the best combination, it is a clear indication that the results are not very sensitive to this parameter. Oppositely, a high difference indicates a high sensitivity.

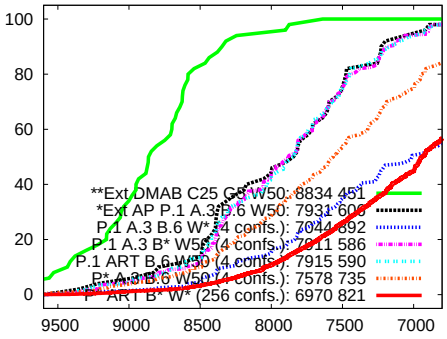
6.3.2 Sensitivity Analysis on the $\mathcal{ART}(0.1, 39, 0.5, 3)$ scenario with $\Delta T = 200$

Global comments: As can be seen from Fig. 9, the three bandit-like techniques (associated with the *Extreme Credit Assignment*) perform rather similarly on this scenario: the same (small) proportion of runs reach the best possible value. SlMAB even has slightly more runs reaching the best values than DMAB, the champion here (for which the two "Best" curves are on top of one another). At the other end, all runs are much better than the Naive strategy (plateau on the top right for all "Best/This" curves). Oppositely, AP perform significantly worse, both for the TCR high values (no run does reach the 'Optimal' value) and for the low values (no plateau at 100% on the right).

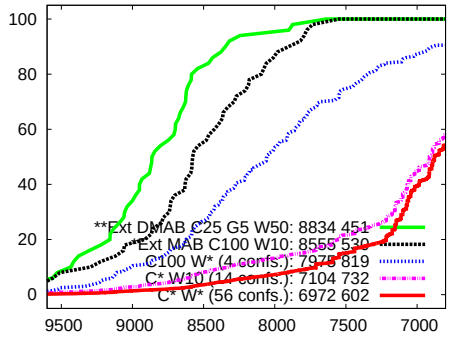
Sensitivity hints: For the AP, the adaptation rate α and the learning rate β are very robust parameters indeed: their aggregated ECDF plots are very close to the one of the best combination for this technique. The minimal probability p_{min} is a much more sensitive parameter, which was expected, as discussed in Section 2.3.1. The window size W is the most sensitive parameter from those plots. Altogether, AP seems to be robust with respect to its parameters, but its poor global performance make it a poor choice anyway.

The MAB and SlMAB only have 2 parameters. For both techniques, the scaling C factor is definitely a very sensitive parameter – much more sensitive than the window size W. Indeed, C controls the heart of the bandit algorithm, the balance between the exploration and exploitation terms, and thus determines the actual behavior of the strategy. The seemingly robustness with respect to W for SlMAB is slightly surprising, as W plays a double role in the SlMAB algorithm: it is used both to compute the extreme of the reward, and to tune the size of the memory of the armed bandit.

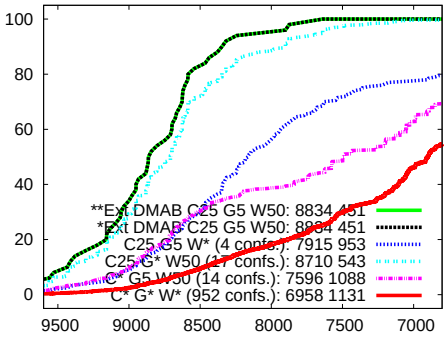
The DMAB has one additional parameter, the threshold γ of the Page-Hinkley test. Although 17 different values were tried for γ, it surprisingly demonstrates to be quite robust, with its aggregated distribution performing very close to the best one.



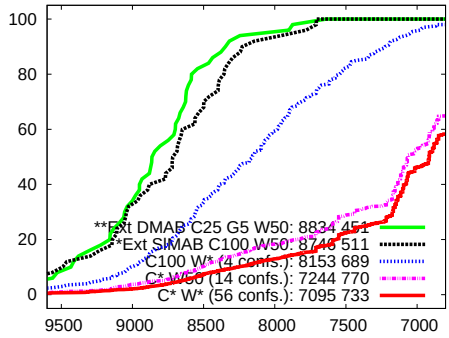
(a) Extreme **AP**. From top-left to bottom-right: *Best/All*; 3 almost identical lines, *Best/This*, adaptation α and learning rate β; minimal probability p_{min}; Window size W; *Aggreg(All)*.



(b) Extreme **MAB**. From top-left to bottom-right: *Best/All*; *Best/This*; Window size W; Scaling factor C; *Aggreg(All)*.



(c) Extreme **DMAB**. From top-left to bottom-right: Identical *Best/All* and *Best/This*; PH threshold γ; Window size W; Scaling factor C; *Aggreg(All)*.



(d) Extreme **SlMAB**. From top-left to bottom-right: *Best/All*; *Best/This*; Window size W; Scaling factor C; *Aggreg(All)*.

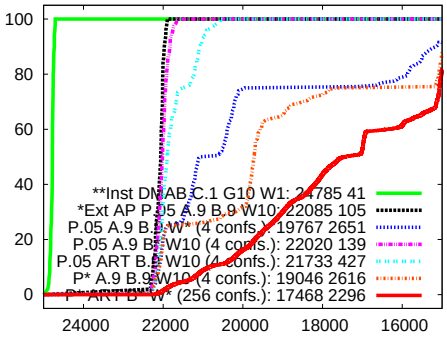
Fig. 9 ECDF sensitivity plots for $\mathcal{ART}(0.1, 39, 0.5, 3)$, $\Delta T200$. See text for line label details.

On the other hand, the window size W and, even more, the scaling factor C, are more sensitive, in a similar way than for the MAB *Adaptive Operator Selection* scheme.

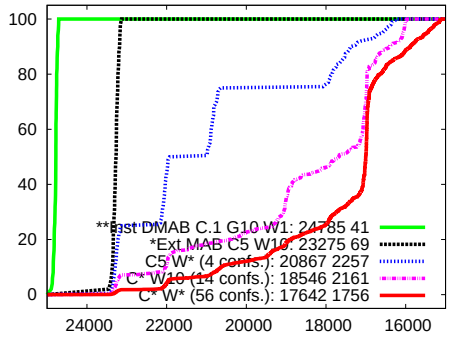
These conclusions hold quite well for the other epoch sizes experimented with (50, 500, 2000). However, the longer the epoch, the more sensitive the scaling factor C (for all the bandit-based techniques), and the less sensitive the γ for DMAB: the scaling C controls the UCB *Exploration vs. Exploitation* balance during the whole run, while γ is important only in the transition phases, which represent a smaller fraction of the total run as ΔT increases.

6.3.3 Sensitivity Analysis on the Uniform scenario with $\Delta T = 500$

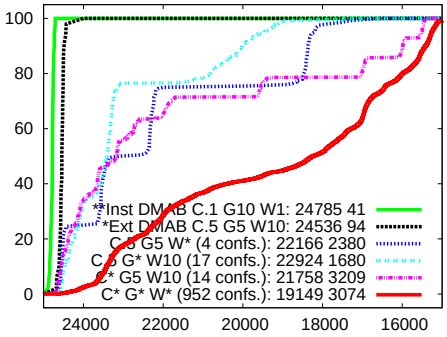
Global comments (see Fig. 10): DMAB is again here the best performing *Adaptive Operator Selection*, even though the overall best is DMAB together with the instantaneous reward (the two curves "Best/All" and "Best/This" are distinct here). However,



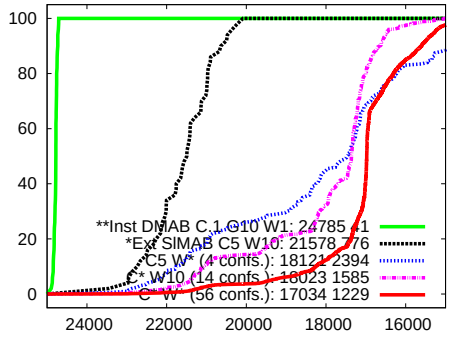
(a) Extreme **AP**. From top-left to bottom-right: *Best/All*; *Best/This*; learning rate β; adaptation α; Window size W; minimal probability p_{min}; *Aggreg(All)*.



(b) Extreme **MAB**. From top-left to bottom-right: *Best/All*; *Best/This*; Window size W; Scaling factor C; *Aggreg(All)*.



(c) Extreme **DMAB**. From top-left to bottom-right: *Best/All*; *Best/This*; PH threshold γ; Window size W (best on top); Scaling factor C; *Aggreg(All)*.



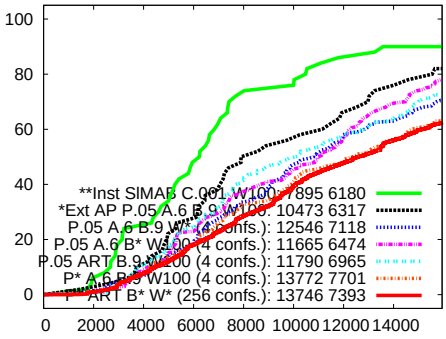
(d) Extreme **SlMAB**. From top-left to bottom-right: *Best/All*; *Best/This*; Window size W (best at bottom); Scaling factor C; *Aggreg(All)*.

Fig. 10 ECDF sensitivity plots for the *Uniform* scenario for $\Delta T500$.

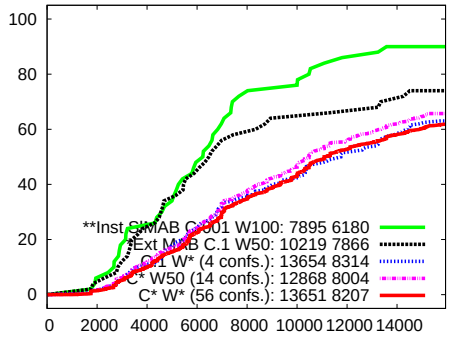
MAB comes next, followed by AP, and SlMAB performs rather poorly. Also note that the variance of the results is way higher for SlMAB than for all others, which display almost vertical curves (*i.e.*, no variance at all).

Sensitivity hints: For AP, again the adaptation and the learning rates α and β are very robust. However, the sensitivity of p_{min} is here higher than that of W. This is clearly a consequence of having more operators, as for a given value of p_{min}, the amount of exploration is proportional to the number of operators.

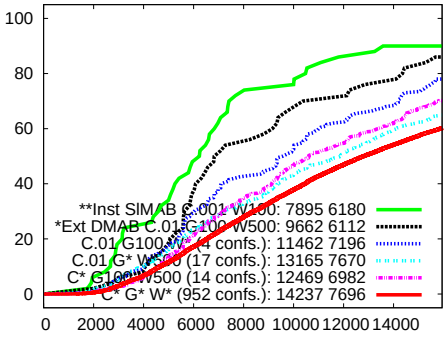
Regarding the bandit-based *Operator Selection* mechanisms, the conclusions that can be drawn from those plots are very similar to the findings in the case of the $\mathcal{ART}(0.1, 39, 0.5, 3)$ scenario: the scaling factor C is always the most sensitive parameter. However, for DMAB, parameter γ seems relatively more sensitive than in the previous scenario; and for SlMAB, W and C now seem equally sensitive.



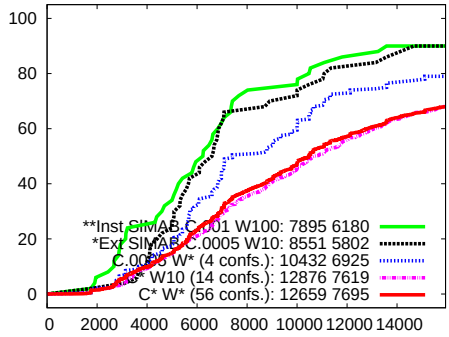
(a) Extreme **AP**. From top-left to bottom-right: *Best/All*; *Best/This*; learning rate β; adaptation α; Window size W; minimal probability p_{min}; *Aggreg(All)*.



(b) Extreme **MAB**. From top-left to bottom-right: *Best/All*; *Best/This*; Window size W; Scaling factor C; *Aggreg(All)*.



(c) Extreme **DMAB**. From top-left to bottom-right: *Best/All*; *Best/This*; PH threshold γ; Window size W (best on top); Scaling factor C; *Aggreg(All)*.



(d) Extreme **SlMAB**. From top-left to bottom-right: *Best/All*; *Best/This*; Window size W (best at bottom); Scaling factor C; *Aggreg(All)*.

Fig. 11 ECDF sensitivity plots for the *Royal Road* problem.

6.3.4 Sensitivity Analysis on the Royal Road *scenario*

Global comments: As shown in Fig. 11, in this case the SlMAB with instantaneous rewards is the overall best one, presenting a performance very similar to the extreme SlMAB. Given the high dispersion of the results, all of them are likely to be statistically equivalent. As shown by the absence of plateaus at 100% on the right, none of the techniques is able to outperform the expected performance of the naive uniform on all the runs; however, the instantaneous and the extreme variants of SlMAB outperform the naive strategy on around 90% of the runs – around 10% more than the extreme AP and DMAB, and 20% more than the extreme MAB.

Sensitivity hints: For AP, the same remarks done for the sensitivity analysis on the *Uniform* scenario are totally valid here: α and β very robust, with p_{min} being very sensitive (the more available operators, the more p_{min} will affect the AP performance).

For the three bandit-based techniques, the scaling factor C shows again to be very sensitive, but less than the change-detection threshold γ for the DMAB. Differently from the analyzed artificial scenarios, the size of the sliding window W is not the most sensitive parameter, except for the MAB technique.

6.3.5 Sensitivity Analysis: Discussion

Although further analyses should be done in more complex problems to confirm these sensitivity hints, the agreements found between the analyzed scenarios in that respect is somehow comforting. These findings can thus be used to guide the parameter tuning of the proposed *Adaptive Operator Selection* combinations, *e.g.*, the most important parameter to tune for all bandit-based procedures is definitely the scaling factor C, while the PH threshold γ for DMAB is also quite sensitive.

These two parameters become more difficult to tune in *Royal Road* case (and this is likely to be the case for every real scenario), probably because the range of the rewards tends to vary as the search advances. Because of this, there is no value for C and γ that performs in an optimal fashion during the whole run. A solution to try to alleviate this issue is currently being explored, considering the use of rank-based rewards, that seems to be the path towards more robust settings.

Regarding the high sensitivity of the window size W, independently of the *Adaptive Operator Selection* scheme, it seems to be more important in the artificial scenarios, mainly because in these cases there is a strong link between W and the overall performance, *e.g.*, values of W larger than the epoch size ΔT, the period of the changes, will result in using too old information. Furthermore, even in the case of longer epochs, each operator has its own window, of size W: in the worst case, a window might contain rewards as old as $K \times W$ time steps ago, K being the number of operators (*e.g.*, the "steps" on the "W" curves of Figure 10 clearly shows how large values for W are hindering the overall performance of the aggregated distribution in this case). This parameter becomes less sensitive in the more realistic *Royal Road* scenario, in which there is no clear relation between operator qualities and time. Anyway, very little (or nothing) is in general known about the dynamics of change of operator rewards within an evolutionary run, which is likely to be different from one run to another. Trying to adaptively define W is also a mandatory step for further research.

7 Discussion and Conclusion

The work presented here continues earlier studies devoted to *Adaptive Operator Selection*, investigating the use of bandit-based approaches [7] as an alternative to the popular *Probability Matching* [19] and *Adaptive Pursuit* [40].

While Multi-Armed Bandit algorithms provide guarantees for an optimal *Exploration vs Exploitation* trade-off in a static setting [1], they hardly deal with dynamic environments. *Dynamic Multi-Armed Bandit* (DMAB), coupling the so-called UCB algorithm [1] with a change-point detection test, has been first proposed in [22] to preserve the good UCB properties in an abruptly changing environment, through the restart of UCB when a change in the reward distribution is detected. The main limitation of DMAB thus relates with the adjustment of the detection test; not only does this test require a hyper-parameter to be tuned; it further hardly adapts to different

dynamics of the operator reward distributions, whereas *Adaptive Operator Selection* takes place in the highly dynamic environment of evolution.

A first contribution of the present paper has been to propose another dynamic bandit-based algorithm, referred to as *Sliding Multi-Armed Bandit* (SlMAB), addressing the limitations of DMAB. The SlMAB uses a window-based relaxation mechanism to update the reward estimates and the counters involved in the UCB algorithm, taking into account how many time steps have occurred since an operator has last been tried. While this relaxation enforces the forgetting of past events, it preserves an adaptive minimal level of exploration (as opposed to the fixed minimal level of exploration involved in *Probability Matching* and *Adaptive Pursuit*). Given that most *Credit Assignment* schemes proposed in the literature rely on windowing the operator reward distribution, the SlMAB can almost be parametrized "for free", except for the scaling factor C, which is needed by all bandit-based *Operator Selection* mechanisms.

The second contribution of the paper is an extensive and principled experimental comparison of the considered *Adaptive Operator Selection* combinations. The most critical factors of difficulty have been discussed and an original set of artificial scenarios has been proposed in order to understand how the reward frequency and variance impact the *Adaptive Operator Selection*. These scenarios generalize previous benchmarks proposed in the literature [41, 7]. All these benchmarks have been used to fairly compare (after a preliminary Racing phase to tune the hyper-parameters) the combination of *Operator Selection* rules (PM, AP, MAB, DMAB and SlMAB) and *Credit Assignment* schemes (*Instantaneous*, *Average* and *Extreme*). We believe, as also argued in [17, 27, 43], that operators that are capable of rare but highly beneficial moves might be more important to evolutionary optimization than operators that repeatedly make small progress[10], and the design of the artificial scenarios was somehow targeted to reflect such situations. However, the results obtained on the artificial scenarios were confirmed by those in the *Royal Road* context: although being also artificial, the *Royal Road* problem does not a priori seem biased toward risk-taking operators.

Several lessons can be drawn from the above experiments:

- With respect to *Credit Assignment*:
  - While the *Instantaneous Credit Assignment* performs well on smooth scenarios, its performance degrades as the rewards becomes more and more extreme (large values happening rarely and being too rapidly forgotten).
  - On the opposite, the *Average Credit Assignment* keeps track of too many events, which may delay the adaptation of the *Adaptive Operator Selection* mechanism at hand.
  - The *Extreme Credit Assignment* obtains the best performance for most of the Two-Values scenarios. While it efficiently captures the critical information even in the case of smooth reward distributions (as in the *Uniform* scenario), its limitation is when operator reward distributions only differ by the reward probability (as in the *Boolean* scenario).
- With respect to *Operator Selection* rule:
  - *Probability Matching* has not been presented in the detailed results due to its consistently poor behavior.
  - While *Adaptive Pursuit* obtains good results on average, it is dominated by the bandit-based approaches, mainly *Dynamic Multi-Armed Bandit* and *Slid-*

[10] In the latter case indeed, Hill Climbing might be more efficient than Evolutionary Algorithms.

ing Multi-Armed Bandit. This failure is blamed on the fixed exploration mechanism (the minimal selection probability p_{min} attached to all operators) and the two-tiered reward allocation, seemingly slowing down the change detection and biasing the *Adaptive Operator Selection* toward low-variance distributions. Additionally, *Adaptive Pursuit* requires the tuning of 4 hyper-parameters, against 2 or 3 for the bandit-based approaches.
- *Multi-Armed Bandit* was also found to react slowly, especially if a change happens after a long steady-state period, as expected.
- *Dynamic Multi-Armed Bandit* very often obtained the best performance on the testbed proposed here. If the PH test triggers at the right time, *Dynamic Multi-Armed Bandit* is very strong (as witnessed by the results on the *Uniform* scenario for instance). But this adds another hyper-parameter, the threshold to decide when to do a restart. Though this parameter is not as sensitive as it might have seemed a priori, it must nevertheless be tuned accurately, in particular in the actual evolutionary optimization context.
- *Sliding Multi-Armed Bandit* performs almost as good as *Dynamic Multi-Armed Bandit*, being able to reach almost 100% of use of the best operator once identified, while also reacting rapidly to abrupt changes, thanks to its limited memory. It is sensitive, however, to the window size; no automatic method or adaptive principle has been proposed insofar to tune this critical parameter.

After the ample empirical evidence gathered in this work, the Extreme *Credit Assignment* together with the *Sliding Multi-Armed Bandit Operator Selection* rule seems to be a good choice for a robust *Adaptive Operator Selection*. A critical issue regards the parameter tuning[11], essentially the scaling parameter, which has been found by far the most sensitive parameter for all bandit-based approaches presented here.

Another limitation of the presented results is that they rely mostly on artificial scenarios; the Two-Values scenarios involve abrupt changes of the reward distribution whereas the dynamics of the operator rewards is much more complex in an actual evolutionary setting, as in the *Royal Road* embedded case. On the other hand, it is also often the case in Evolutionary Algorithms, as in natural evolution, that relatively stable epochs are separated by (a series of) large jumps, after the so-called *punctuated equilibria* phenomenon [20].

A first perspective for further research is the extension of the Two-Values scenarios proposed here, to account for arbitrary frequencies and amplitude of change, and to analytically investigate the probability of missing the current best operator. A second perspective is to assess the efficiency of the presented *Adaptive Operator Selection* approaches on real optimization problems, with both unimodal and multimodal fitness functions.

Additionally, as mentioned in the discussion about the sensitivity analysis (Section 6.3.5), different *Credit Assignment* schemes should be explored in the near future, considering rank-based rewards instead of raw values. The use of rank-based rewards will hopefully lead to robust settings for the scaling factor C (and for the DMAB PH threshold γ), because the range of the received rewards would then be invariant, no matter the problem and the current stage of the search process. Furthermore, in

[11] The reader might wonder what is the point of designing *Adaptive Operator Selection* schemes to handle the parameter tuning, if these schemes themselves involve parameters to be tuned. The gain is that *Adaptive Operator Selection* schemes take care of many shallow parameters (e.g. the selection rate of all variation operators) while involving a few general hyper-parameters (Table 2).

order to efficiently tackle multi-modal problems, the diversity should also be considered somehow by the *Credit Assignment*, following the ideas of [31,32].

References

1. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multi-armed bandit problem. Machine Learning **47**(2-3), 235–256 (2002)
2. Barbosa, H.J.C., Sá, A.M.: On adaptive operator probabilities in real coded genetic algorithms. In: XX Intl. Conference of the Chilean Computer Science Society (2000)
3. Bartz-Beielstein, T., Lasarczyk, C., Preuss, M.: Sequential parameter optimization. In: B. McKay (ed.) Proc. Congress on Evolutionary Computation, pp. 773–780. IEEE (2005)
4. Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In: W. B. Langdon et al. (ed.) Proc. Genetic and Evolutionary Computation Conference, pp. 11–18. Morgan Kaufmann (2002)
5. Collet, P., Schoenauer, M.: GUIDE: Unifying evolutionary engines through a graphical user interface. In: P. Liardet et al. (ed.) Proc. Intl. Conference on Artificial Evolution, *LNCS*, vol. 2936, pp. 203–215. Springer (2003)
6. Conover, W.J.: Practical Nonparametric Statistics. John Wiley & Sons (1999)
7. Da Costa, L., Fialho, A., Schoenauer, M., Sebag, M.: Adaptive operator selection with dynamic multi-armed bandits. In: M. Keijzer et al. (ed.) Proc. Genetic and Evolutionary Computation Conference, pp. 913–920. ACM (2008)
8. Davis, L.: Adapting operator probabilities in genetic algorithms. In: J.D. Schaffer (ed.) Proc. Intl. Conference on Genetic Algorithms, pp. 61–69. Morgan Kaufmann (1989)
9. DeJong, K.: Evolutionary Computation. A unified Approach. MIT (2006)
10. DeJong, K.: Parameter setting in EAs: a 30 year perspective. In: Lobo et al. [30], pp. 1–18
11. Eiben, A.E., Hinterding, R., Michalewicz, Z.: Parameter control in Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation **3**(2), 124–141 (1999)
12. Eiben, A.E., Michalewicz, Z., Schoenauer, M., Smith, J.E.: Parameter control in evolutionary algorithms. In: Lobo et al. [30], pp. 19–46
13. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer (2003)
14. Fialho, A., Da Costa, L., Schoenauer, M., Sebag, M.: Extreme value based adaptive operator selection. In: G. Rudolph et al. (ed.) Proc. Intl. Conference on Parallel Solving from Nature, *LNCS*, vol. 5199, pp. 175–184. Springer (2008)
15. Fialho, A., Da Costa, L., Schoenauer, M., Sebag, M.: Dynamic multi-armed bandits and extreme value-based rewards for adaptive operator selection in evolutionary algorithms. In: Stützle [39], pp. 176–190
16. Fialho, A., Schoenauer, M., Sebag, M.: Analysis of adaptive operator selection techniques on the royal road and long k-path problems. In: G. Raidl et al. (ed.) Proc. Genetic and Evolutionary Computation Conference, pp. 779–786. ACM (2009)
17. Fogel, D.B.: Phenotypes, genotypes and operators in evolutionary computation. In: Proc. Intl. Conference on Evolutionary Computation. IEEE (1995)
18. Gagliolo, M., Schmidhuber, J.: Algorithm selection as a bandit problem with unbounded losses. Tech. Rep. IDSIA - 07 - 08, IDSIA (2008)
19. Goldberg, D.: Probability matching, the magnitude of reinforcement, and classifier system bidding. Machine Learning **5**(4), 407–426 (1990)
20. Gould, S., Eldredge, N.: Punctuated equilibria: the tempo and mode of evolution reconsidered. Paleobiology **3**(2), 115–151 (1977)
21. Hartland, C., Baskiotis, N., Gelly, S., Teytaud, O., Sebag, M.: Change point detection and meta-bandits for online learning in dynamic environments. In: Proc. Conférence Francophone sur l'Apprentissage Automatique (2007)
22. Hartland, C., Gelly, S., Baskiotis, N., Teytaud, O., Sebag, M.: Multi-armed bandit, dynamic environments and meta-bandits. In: Online Trading of Exploration and Exploitation Workshop, NIPS (2006)
23. Hinkley, D.: Inference about the change point from cumulative sum-tests. Biometrika **58**(3), 509–523 (1970)
24. Holland, J.H.: Royal road functions. In: Internet Genetic Algorithms Digest 7:22. Massachusetts Institute of Technology (1993)
25. Jones, T.: A description of Holland's Royal Road. Evolutionary Computation **2**(4), 409–415 (1994)

26. Julstrom, B.: What have you done for me lately? Adapting operator probabilities in a steady-state genetic algorithm. In: L. J. Eshelman et al. (ed.) Proc. Intl. Conference on Genetic Algorithms, pp. 81–87. Morgan Kaufmann (1995)
27. Kallel, L., Schoenauer, M.: Fitness distance correlation for variable length representations. Tech. Rep. 363, CMAP, Ecole Polytechnique (1996)
28. Lai, T., Robbins, H.: Asymptotically efficient adaptive allocation rules. Advances in applied mathematics **6**(1), 4–22 (1985)
29. Lobo, F., Goldberg, D.: Decision making in a hybrid genetic algorithm. In: B. Porto (ed.) Proc. Intl. Conference on Evolutionary Computation, pp. 121–125. IEEE (1997)
30. Lobo, F., Lima, C., Michalewicz, Z. (eds.): Parameter Setting in Evolutionary Algorithms, *Studies in Computational Intelligence*, vol. 54. Springer (2007)
31. Maturana, J., Fialho, A., Saubion, F., Schoenauer, M., Sebag, M.: Extreme compass and dynamic multi-armed bandits for adaptive operator selection. In: Proc. Congress on Evolutionary Computation, pp. 365–372. IEEE (2009)
32. Maturana, J., Lardeux, F., Saubion, F.: Autonomous operator management for evolutionary algorithms. Journal of Heuristics (2010)
33. Maturana, J., Saubion, F.: A compass to guide genetic algorithms. In: G. Rudolph et al. (ed.) Proc. Intl. Conference on Parallel Solving from Nature, *LNCS*, vol. 5199, pp. 256–265. Springer (2008)
34. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3 edn. Springer, New-York (1996)
35. Mitchell, M., Forrest, S., Holland, J.H.: The royal road for genetic algorithms: Fitness landscapes and GA performance. In: Proc. European Conference on Artificial Life, pp. 245–254 (1992)
36. Nannen, V., Eiben, A.E.: Relevance estimation and value calibration of evolutionary algorithm parameters. In: M. Veloso (ed.) Proc. Intl. Joint Conference on Artificial Intelligence, pp. 975–980 (2007)
37. Quick, R.J., Rayward-Smith, V.J., Smith, G.D.: The royal road functions: Description, intent and experimentation. In: Selected Papers from AISB Workshop on Evolutionary Computing, *LNCS*, vol. 1143, pp. 223–235. Springer (1996)
38. Spears, W.: Adapting crossover in evolutionary algorithms. In: J.R. McDonnell et al. (ed.) Proc. Conference on Evolutionary Programming, pp. 367–384. MIT (1995)
39. Stützle, T. (ed.): Proc. 3rd Intl. Conference on Learning and Intelligent Optimization, *LNCS*, vol. 5851. Springer (2009)
40. Thierens, D.: An adaptive pursuit strategy for allocating operator probabilities. In: H.G. Beyer (ed.) Proc. Genetic and Evolutionary Computation Conference, pp. 1539–1546. ACM (2005)
41. Thierens, D.: Adaptive strategies for operator allocation. In: Lobo et al. [30], pp. 77–90
42. Tuson, A., Ross, P.: Adapting operator settings in genetic algorithms. Evolutionary Computation **6**(2), 161–184 (1998)
43. Whitacre, J., Pham, T., Sarker, R.: Use of statistical outlier detection method in adaptive evolutionary algorithms. In: M. Keijzer (ed.) Proc. Genetic and Evolutionary Computation Conference, pp. 1345–1352. ACM (2006)
44. Yu, T., Davis, D., Baydar, C., Roy, R. (eds.): Evolutionary Computation in Practice, *Studies in Computational Intelligence*, vol. 88. Springer (2008)
45. Yuan, B., Gallagher, M.: Statistical racing techniques for improved empirical evaluation of evolutionary algorithms. In: X. Yao et al. (ed.) Proc. Intl. Conference on Parallel Solving from Nature, *LNCS*, vol. 3242, pp. 172–181. Springer (2004)