Dynamic Algorithm Configuration: Foundation of a New Meta-Algorithmic Framework — Supplementary —

André Biedenkapp[1] and **H. Furkan Bozkurt**[1] and **Theresa Eimer**[3] and **Frank Hutter**[1,2] and **Marius Lindauer**[3]

A Appendix

A.1 Importance of Temporal Information for Luby

For a sequence like *Luby*, an agent can benefit from additional information about the sequence, such as the length of the sequence. For example, imagine an agent has to learn the Luby sequence for length $T = 16$. Before time-step 8 the action value 3 would never have to be be played. For a real algorithm to be controlled, such a temporal feature could be encoded by the iteration number directly or some other measure of progress. The state an agent can observe therefore consists of such a time feature and a small history over the five last selected actions.

A.2 Luby Instance Sampling Strategies

Depending on the sampling strategy, the instances can be chosen to be more homogeneous or more heterogeneous. To generate homogeneous instances, we sample a small temporal error $i \sim \mathcal{N}(0, 0.15)_{-0.5 \leq i \leq .5}$ from a two-sided truncated normal distribution. This temporal error is added every step to t. After m steps $t + |\sum_{j=0}^{m} i| > t$ and therefore an element in the Luby sequence will be skipped ($i > 0$) or repeated ($i < 0$). Formally

$$\text{luby}(t, i) = \begin{cases} l_t & \text{if } mod(t, m) \neq 0 \\ l_{t-1} & \text{if } i < 0 \\ l_{t+1} & \text{otherwise} \end{cases} \tag{6}$$

where l_t is the t-th value of the Luby sequence (Equation 7 in the main paper). The resulting instances will overlap at most time-steps, giving a very homogeneous distribution of instances.

To generate heterogeneous instances, we sample different starting points $i \sim \mathcal{N}(L, 0.25)_{0 \leq i \leq T-L}$ of the Luby sequence from a two-sided truncated normal distribution. Every instance is therefore a subsequence of the original Luby sequence. Formally

$$\text{luby}(t, i) = l_{t+i} \tag{7}$$

where l_t is the t-th value of the Luby sequence (Equation 7 in the main paper) These different starting points cause most instances to have little overlap with other sequences.

[1] University of Freiburg, Germany, email: {biedenka, bozkurf, fh}@cs.uni-freiburg.de
[2] Bosch Center for Artificial Intelligence, Germany
[3] University of Hannover, Germany, email: lastname@tnt.uni-hannover.de

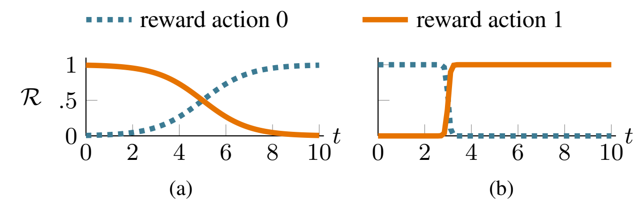


Figure 1: Example rewards for Benchmark *Sigmoid* with $T = 10$, $N = 1$ and $a_{0,t} \in \{0, 1\}$ on two instances, where $p_{(a)} = 5$, $s_{(a)} = 1$ on (a) and $p_{(b)} = 3$, $s_{(b)} = -20$ on (b). The solid/dashed line depicts the reward for action 1/0 at time-step t. On (a) it is preferable to select action 1 for the first halve of the sequence whereas on (b) it is better to start with action 0.

A.3 Sigmoid Example Policies

Figure 1 depicts how different scaling factors and inflection points can be used to construct different problem instances that require different optimal policies. In Figure 1a, the optimal policy is to play action 1 for the first 4 to 5 time-steps before switching over to action 0. In Figure 1b, the scaling factor is inverted, requiring to first play action 0 and then action 1. As the inflection point is shifted to the left the optimal policy selects this action for first 2 to 3 time-steps before playing action 1 the remaining steps.

A.4 Context-Oblivious Agents

Adriaensen and Nowé (2016) proposed two additional context-oblivious agents, (i) PURS, which selects a previously not selected action uniformly at random; otherwise, actions are selected in proportion to the expected number of remaining steps; (ii) GR, which selects an action greedily based on the expected future reward. In the following we discuss why we did not consider PURS and GR.

PURS PURS leverages information about the expected trajectory length, but it does not include the observed reward signal in the decision making process. For tasks where every execution path has the same length (e.g. *Luby*$_{L=T}$ and *Sigmoid*), PURS would fail to produce a policy other than a uniform random one. Further, when using PURS, we need to have some prior knowledge if shorter or longer trajectories should be preferred. For example on benchmarks like *Luby*$_{L<T}$, PURS is only able to find a meaningful policy if we know that longer sequences produce better rewards. Thus, we do not consider PURS further.

GR GR is comparable to an ϵ-greedy agent with constant $\epsilon = 0$. This makes GR very prone to getting stuck in local optima which can happen in our experiments quite often.

A.5 Details on Experimental Setup

DQN Details We used a double DQN in chainerRL (0.7.0), i.e., where the target network is updated every 10 episodes and the exploration fraction ϵ of the DQN is linearly decreased from 1.0 to 0.1. We used a fully connected Q-function with a training batch size of 32. In each training iteration only one episode is observed. We chose a network size of 1 layer with 50 hidden units as we only deal with very small state-vectors with at most 8 state features.

Hardware All experiments were run on a compute cluster with nodes equipped with two Intel Xeon E5-2630v4 and 128GB memory running CentOS 7.

A.6 Further Experimental Results

Effect of Short Effective Sequence Length Further experiments for the effect of short effective sequences are given in Tables 1 and 2b, with corresponding reward curves given in Figures 3 and 4 respectively. Table 1 corresponds to Table 1 in the main paper and additionally present the standard error for all presented agents. Table 2b shows similar results with a much higher noise level. In this setting however ϵ-greedy does not achieve the best AUC for short-effective sequences with $L = 32$. From Figure 4 we can observe that, although PS-SMAC results in a better AUC, it alreay converged to a suboptimal sequence, whereas the ϵ-greedy agent is still improving its reward and resulting in a slightly higher final reward.

Stochasticity of Reward Signal Further results for the effect of the stochasticity of the environment are given in Table 3a and 3b with the corresponding plots in Figures 5a and 5b respectively. Table 3 corresponds to Table 2 in the main paper and additionally gives the standard errors of all discussed methods. Figures 5a and 5b show that, no matter the noise level, the ϵ-greedy agent always outperforms both PS-SMAC and URS not only in final reward but also in anytime reward.

Effect of Self-Paced Learning In Figure 2 the trainings performance of SPL on only the selected instances is given. The curve corresponds to Equation (5) of the main paper. The agent starts training on an instance which gives it better initial performance. The agent continues its training on instances which resemble the first initial instance which gave it good performance. After around 5×10^3 training episodes, the agent also includes instances which require opposite policies into its training procedure. It quickly recovers from this shift in instances before it approaches the optimal possible reward.

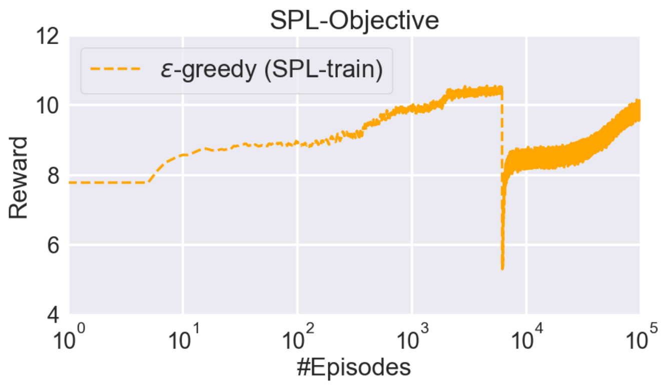


Figure 2: Training performance on instances selected by SPL for 1D-Sigmoid with binary actions and $T = 11$.

	8	16	32
ϵ-greedy	0.86 ± 0.04	0.72 ± 0.05	0.47 ± 0.04
PS-SMAC	0.62 ± 0.10	0.39 ± 0.01	0.39 ± 0.01
URS	0.17 ± 0.02	0.17 ± 0.02	0.17 ± 0.03

(a) Homogeneous

8	16	32	
ϵ-greedy	0.89 ± 0.03	0.75 ± 0.05	0.47 ± 0.03
PS-SMAC	0.56 ± 0.09	0.37 ± 0.01	0.37 ± 0.01
URS	0.17 ± 0.02	0.17 ± 0.02	0.17 ± 0.02

(b) Heterogeneous

Table 1: Results for the discussed agents on *Luby* with fuzzy rewards for $L \in \{8, 16, 32\}$ with $T = 64$ on two instance distributions and a noise factor such that roughly 15% of the actions returned a false positive reward. Results for homogeneous instances are shown in (a) and for heterogeneous instances in (b). The values represent the normalized area under the learning curve for 10^5 training episodes.

	8	16	32
ϵ-greedy	0.76 ± 0.06	0.56 ± 0.10	0.35 ± 0.05
PS-SMAC	0.66 ± 0.10	0.38 ± 0.01	0.38 ± 0.01
URS	0.17 ± 0.04	0.17 ± 0.04	0.17 ± 0.03

(a) Homogeneous

8	16	32	
ϵ-greedy	0.80 ± 0.05	0.59 ± 0.07	0.35 ± 0.05
PS-SMAC	0.55 ± 0.10	0.37 ± 0.01	0.38 ± 0.01
URS	0.16 ± 0.03	0.16 ± 0.05	0.16 ± 0.04

(b) Heterogeneous

Table 2: Results for the discussed agents on *Luby* with fuzzy rewards for $L \in \{8, 16, 32\}$ with $T = 64$ on two instance distributions and a noise factor such that roughly 25% of the actions returned a false positive reward. Results for homogeneous instances are shown in (a) and for heterogeneous instances in (b). The values represent the normalized area under the learning curve for 10^5 training episodes. The corresponding plots are contained in the supplementary material.

	$p(r_t > 0)$ 0.01	0.08	0.15	0.20	0.25
ϵ-greedy	0.96 ± 0.01	0.92 ± 0.03	0.86 ± 0.04	0.81 ± 0.04	0.76 ± 0.05
PS-SMAC	0.71 ± 0.09	0.63 ± 0.10	0.62 ± 0.10	0.62 ± 0.11	0.55 ± 0.10
URS	0.21 ± 0.01	0.18 ± 0.02	0.17 ± 0.02	0.17 ± 0.02	0.16 ± 0.02

(a) Homogeneous

ϵ-greedy	0.97 ± 0.01	0.94 ± 0.02	0.89 ± 0.03	0.84 ± 0.04	0.80 ± 0.05
PS-SMAC	0.60 ± 0.10	0.63 ± 0.11	0.56 ± 0.09	0.61 ± 0.10	0.52 ± 0.10
URS	0.21 ± 0.02	0.19 ± 0.02	0.17 ± 0.02	0.17 ± 0.03	0.16 ± 0.03

(b) Heterogeneous

Table 3: Sensitivity analysis of the presented agents for varying degrees of noise on Luby. The short effective sequence was set to 8 with a maximal length of 64. The values represent the normalized area under the learning curve for 10^5 training episodes. The corresponding plots are contained in the supplementary material.

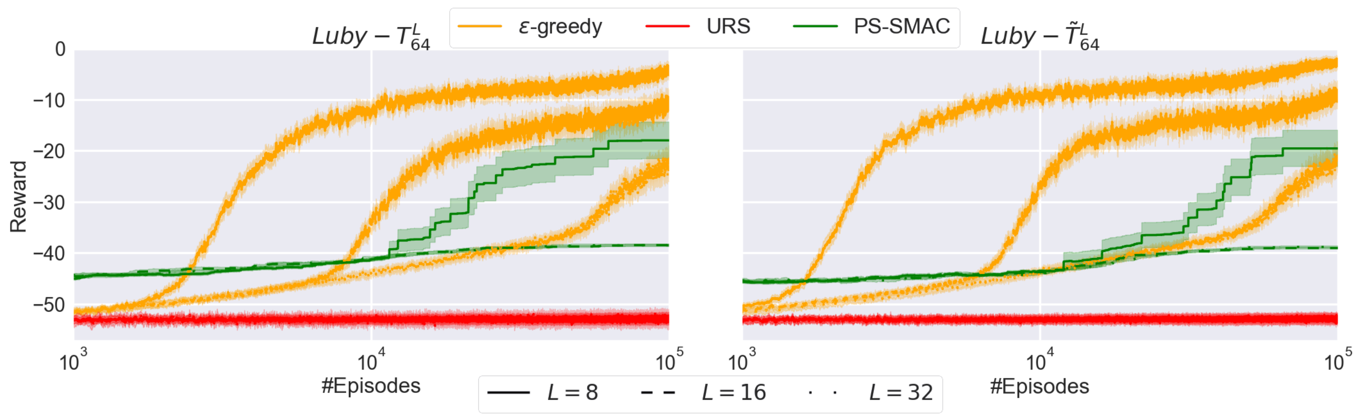


Figure 3: Comparison of ϵ-greedy, URS and PS-SMAC on Luby for varying short effective sequences. (Left) Results on the homogeneous version of Luby. (Right) Results on the heterogeneous version of Luby. Gaussian noise was added to the reward such that roughly 15% of the reward signals gave a false positive signal.

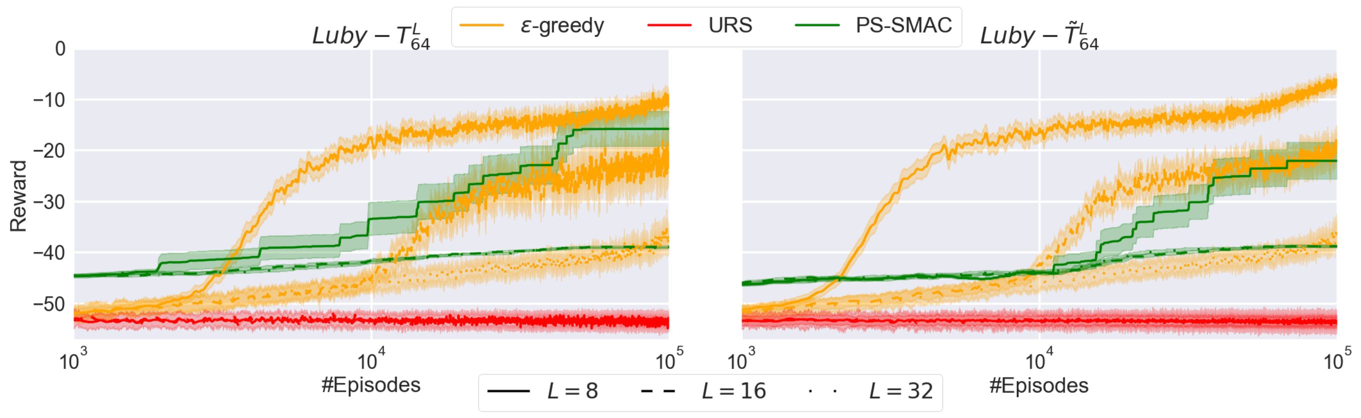
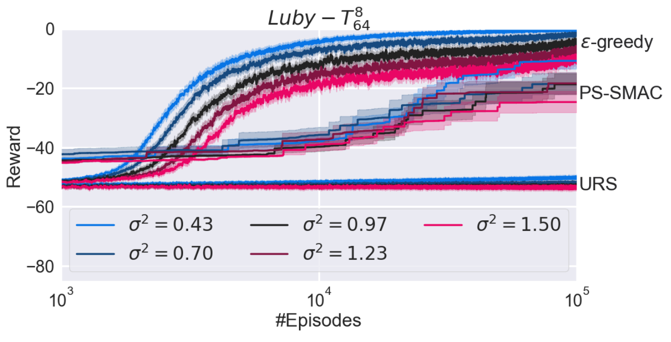
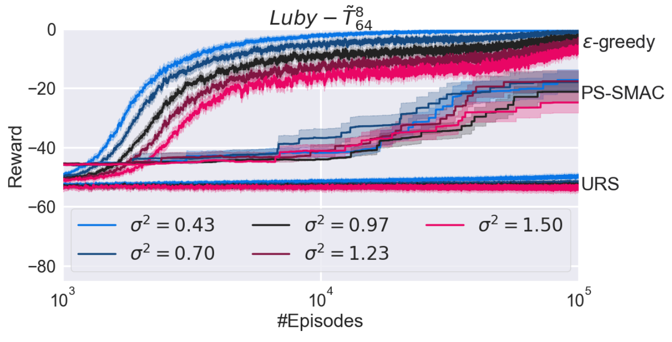


Figure 4: Comparison of ϵ-greedy, URS and PS-SMAC on Luby for varying short effective sequences. (Left) Results on the homogeneous version of Luby. (Right) Results on the heterogeneous version of Luby. Gaussian noise was added to the reward such that roughly 25% of the reward signals gave a false positive signal.



(a)



(b)

Figure 5: Sensitivity analysis on Luby (left homogeneous, right heterogeneous) for varying degrees of noise levels.