Let us assume that a path is feasible iff the first and the third occurrences of symbol v are followed by the same symbol (s feasible iff $a_{v,1}(s) = a_{v,3}(s)$). Let us further assume that $\mathcal{E}$ includes $s_1 = vwvxvw$, $s_2 = vxvwvx$ and $s_3 = vxvwvw$; s_1 and s_2 are feasible while s_3 is infeasible. Consider the current path $s = vwvxv$; the next step is to select the successor of the 3rd occurrence of v. It can be seen that $p(s, w) = .5$ while $p(s, x) = 1.$, as the first event (the 3rd occurrence of v is followed by w and there are at least 2 occurrences of w) is satisfied by s_1 and s_3 while the second event (the 3rd occurrence of v is followed by x and there are at least 2 occurrences of x) only covers s_2.

A *Seeded Init* option is devised to remedy the above limitation. The idea is to estimate $p_s(w)$ from a subset of $\mathcal{E}$, called Seed set, including feasible paths belonging to one single conjunctive subconcept. A necessary condition for a set of positive examples (feasible paths) to represent a conjunctive sub-concept is that its least general generalisation[3] be correct, i.e. it does not cover any negative example. In our toy example problem, the lgg of s_1 and s_2 is not correct as it covers s_3.

Seed sets are stochastically constructed as follows. Let $\mathcal{E}^+$ be the randomly ordered set of feasible paths in $\mathcal{E}$. Let the seed set E be initialized to s_1 and let h denote the lgg of elements in E. At step $i \geq 2$, the i-th path s_i in $\mathcal{E}^+$ is considered, the lgg of h and s_i is constructed and its correctness is tested against the infeasible paths in $\mathcal{E}$; if the lgg is correct s_i is added to E. By construction, if the infeasible paths are sufficiently representative, E will only include feasible paths belonging to a conjunctive concept (a single branch of the XORs); therefore the probabilities estimated from E will reflect the long range dependencies among the node transitions.

The exploration strength of *EXIST* is enforced by using a restart mechanism to construct another seed set after a while, and by discounting the events related to feasible paths that have been found several times; see [3] for more details.

4 Experimental validation

EXIST is empirically validated on a real-world program and on artificial problems. The real-world Fct4 program includes 36 nodes and 46 edges; the ratio of feasible paths is about 10^{-5} for a maximum path length $T = 250$. The artificial problems are derived from a stochastic generator, varying the number of nodes in $[20, 40]$ and the path length in $[120, 250]$ (available on demand from the first author). Three series of results, related to representative "Easy", "Medium" and "Hard" SST problems are presented in Table 1 and in Fig. 2, 3, 4 (Appendix B). The ratio of feasible paths respectively ranges in $[5 \times 10^{-3}, 10^{-2}]$ for the Easy problems, in $[10^{-5}, 10^{-3}]$ for the Medium problems, and in $[10^{-15}, 10^{-14}]$ for the Hard problems. For each *EXIST* variant and each problem, the reported result is the number of distinct feasible paths found out of 10,000 generated paths, averaged over 10 independent runs; the initial $\mathcal{E}$ set includes 50 feasible/50 infeasible paths. The computational time ranges from 1 to 10 minutes on PC Pentium 3 GHz depending on the problem and the variant considered (labelling cost non included).

In the considered range of problems, the most robust variant is the *SeededGreedy* one (SG); although *BandiST* and *SeededRouletteWheel* (SRW) are efficient on Easy problems, their efficiency decreases with the ratio of feasible paths. The *Seeded* option is almost always beneficial, especially so when combined with the *Greedy* and *RouletteWheel* options, and when applied on hard problems. The *Seeded* option is comparatively less beneficial for *BandiST* than for the other options, as it increases the *BandiST* bias toward exploration; unsurprisingly, exploration is poorly rewarded on hard problems.

The sensitivity of the *EXIST* performances wrt the size of the initial training set is studied experimentally, varying the number of initial feasible and infeasible paths in $50, 200, 1000$. The results obtained on a representative medium problem are displayed in Fig. 5 (Appendix B).

A first remark is that increasing the number of infeasible paths does not improve the results, everything else being equal. Concretely, it makes almost no difference to provide the system with 50 or 1000 infeasible paths besides 50 feasible paths. Even more surprisingly, increasing the number of feasible paths rather degrades the results (the 1000/1000 curve is usually well below the 50/50 curve in Fig. 5).

Both remarks can be explained by modeling the *Seeded* procedure as a 3-state automaton. In each step t, the *Seeded* procedure considers a feasible path s_t, and the resulting lgg is tested for correctness against the infeasible paths. If s_t belongs to the same subconcept as the previous feasible paths (state A), the lgg will be found correct, and the

[3] The least general generalisation (lgg) of a set of propositional examples is the conjunction of constraints of the type *[attribute = value]* that are satisfied by all examples. For instance, the lgg of examples s_1 and s_2 in the extended Parikh representation is $[a_v = 3] \wedge [a_{w,1} = v] \wedge [a_{x,1} = v]$.