DATA-INFORMED PARAMETER SYNTHESIS FOR POPULATION MARKOV CHAINS

Matej Hajnal[2,4], Morgane Nouvian[1,3], Tatjana Petrov[2,3], David Šafránek[4]

[1]) Department of Biology [2]) Department of Computer and Information Sciences

[3]) Centre for the Advanced Study of Collective Behaviour - University of Konstanz, Konstanz, Germany

[4]) Systems Biology Laboratory, Faculty of Informatics, Masaryk University, Brno, Czech Republic

We present a workflow as Jupyter notebook for obtaining viable parameter space in a probabilistic scheme including information of experimental data [2]. First, parameters satisfying given a set of properties are **synthesised** in the form of rational functions. Further, data is used to constrain previously obtained functions. Finally, we solve these constrains by **refining** the parameter space.

Motivation

Honeybees protect their colonies against vertebrates by releasing an alarm pheromone to recruit a large number of defenders into a massive stinging response. In order to achieve a balanced trade-off towards efficient defence, yet no critical worker loss, each bee's response to the same amount of pheromone may vary greatly.

To unveil the mechanism of this collective behaviour we created models and estimated the parameters that comply with experimental data.

Input - model, temporal properties, and data

In this workflow, we support Parametric Discrete Markov Chains (pMC) to **model** probabilistic system under observation- e.g. a population of three bees:

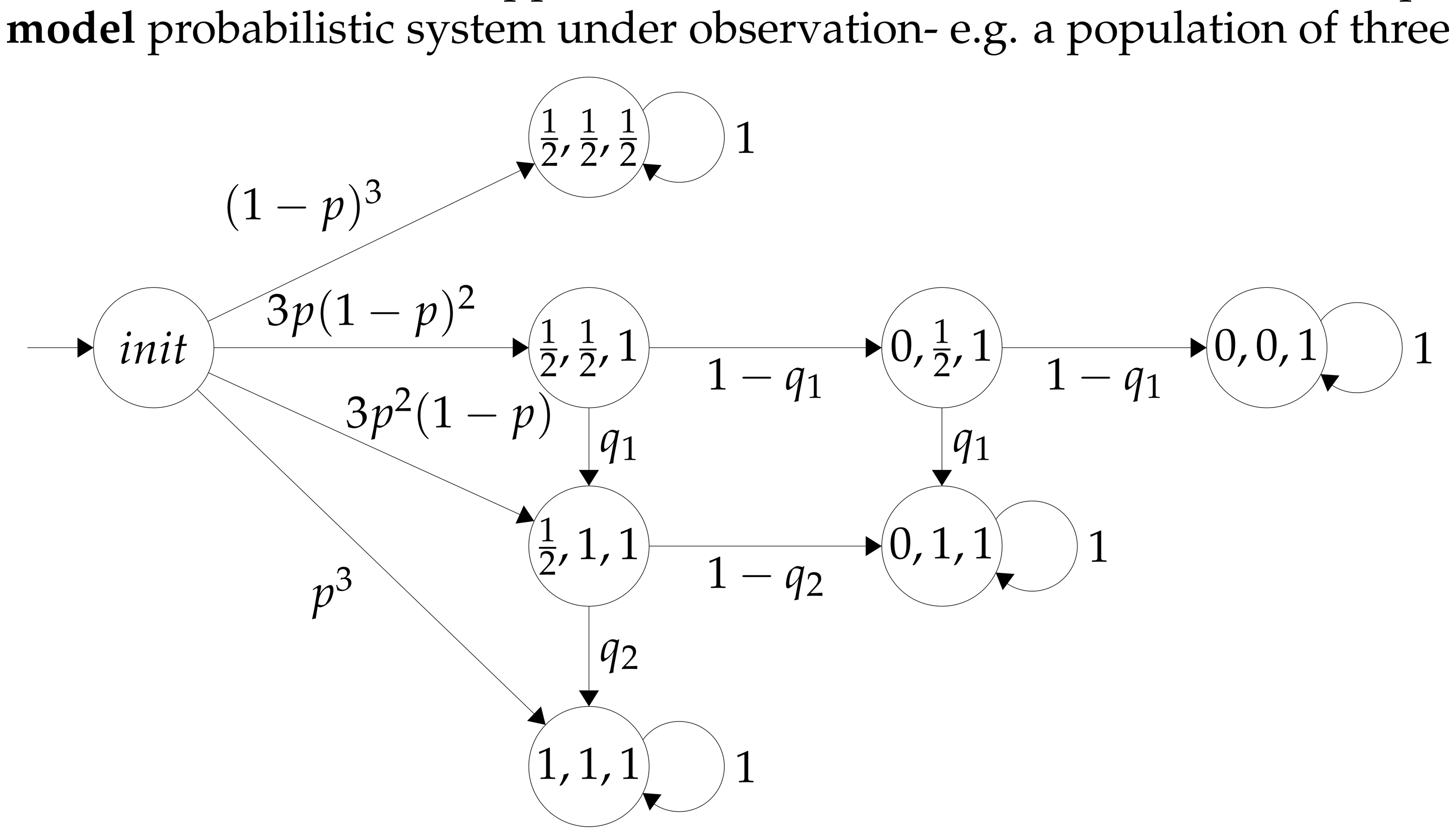

Figure 1: Parameter p represents the initial probability of a bee stinging, while q_i represents the probability of stinging after i bees released pheromone. A vector labelling states represents state of the individual agent (1 denotes decision to sting, $\frac{1}{2}$ denotes the second chance, 0 denotes decision of no stinging)

Observed behaviour of the system, **event**, is encoded as a **temporal property** - particularly Probabilistic Computational Tree Logic (PCTL) formula - e.g. exactly one bee stung

$$P_{=?}\ FG\ (\ s(0,0,1)\ )$$

Finally, the (experimental) **data** captures frequency of the event occurrence - e.g. twice in ten runs

0.2

Parameter Synthesis

First two inputs are used to obtain a symbolic characterisation of satisfaction of the temporal property in the form of (multivariate) rational functions leveraging the existing tools for parameter synthesis [1, 3]. In our running example:

$$P_{=?}\ FG\ (\ s(0,0,1)\ ) = 3p \cdot (1-p)^2 \cdot (1-q_1)^2$$

Confidence intervals

Considering the stochastic behaviour of the system, the frequency of the occurrence of the event may not be exactly the probability of the event. Therefore, margins, δ, are added to the data point resulting with these two constraints:

$$0.2\ - \delta \le 3p \cdot (1-p)^2 \cdot (1-q_1)^2 \le 0.2\ + \delta$$

Margins can be computed for example based on selected confidence level, α, and number of all observations, N - e.g.

$$\delta = z_{\alpha/2}\sqrt{\frac{data \cdot (1-data)}{N}} + \frac{0.5}{N}$$

Universität Konstanz

Centre for the Advanced Study of Collective Behaviour

sybila

Space Refinement

Second, we employ CEGAR-like reasoning (candidate region generation and checking) for determining the viable parameter space. In each step selected region is marked either as **safe** (all parameters in the region satisfy all constraints), **unsafe** (no parameter in the region satisfy all constraints), or **unknown** (otherwise). In the next step, a region marked *unknown* is selected unless the desired proportion of the space (called coverage) is marked *safe/unsafe*. In the case of 2-dimensional space, safe subspace can be visualised as a green area, unsafe as a red area, and unknown subspace as a white area.

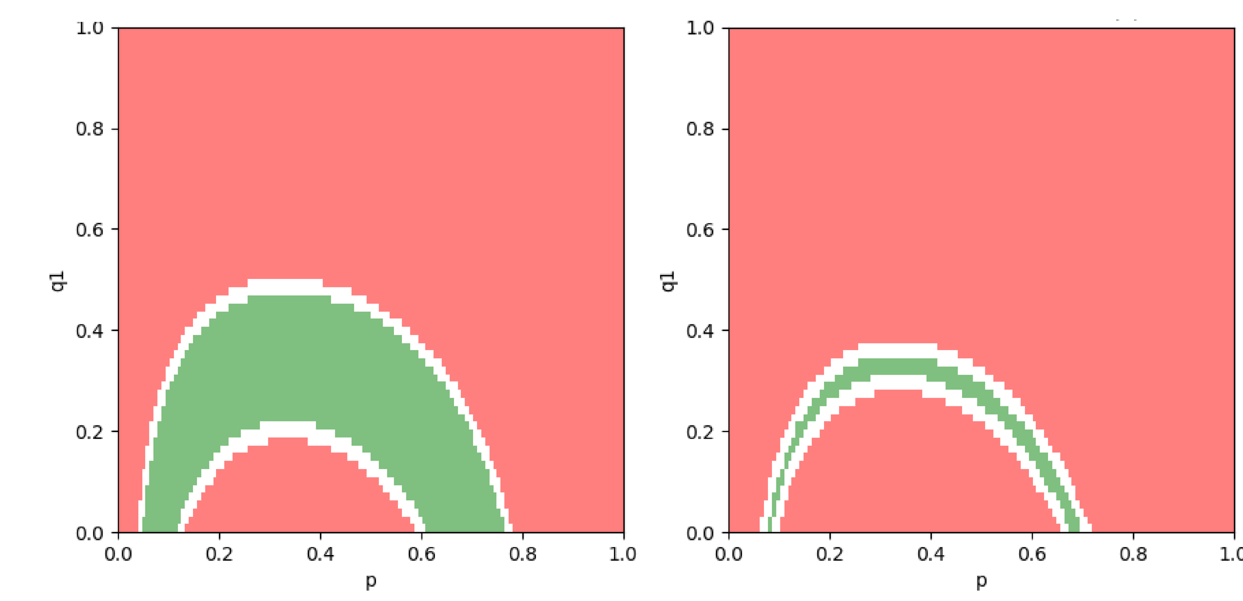

Figure 2: Parameter space refinement visualisation with two different intervals inferred from the data - $\alpha = 0.95$ (both), $N = 100$ (left) and $N = 1500$ (right).

Summary

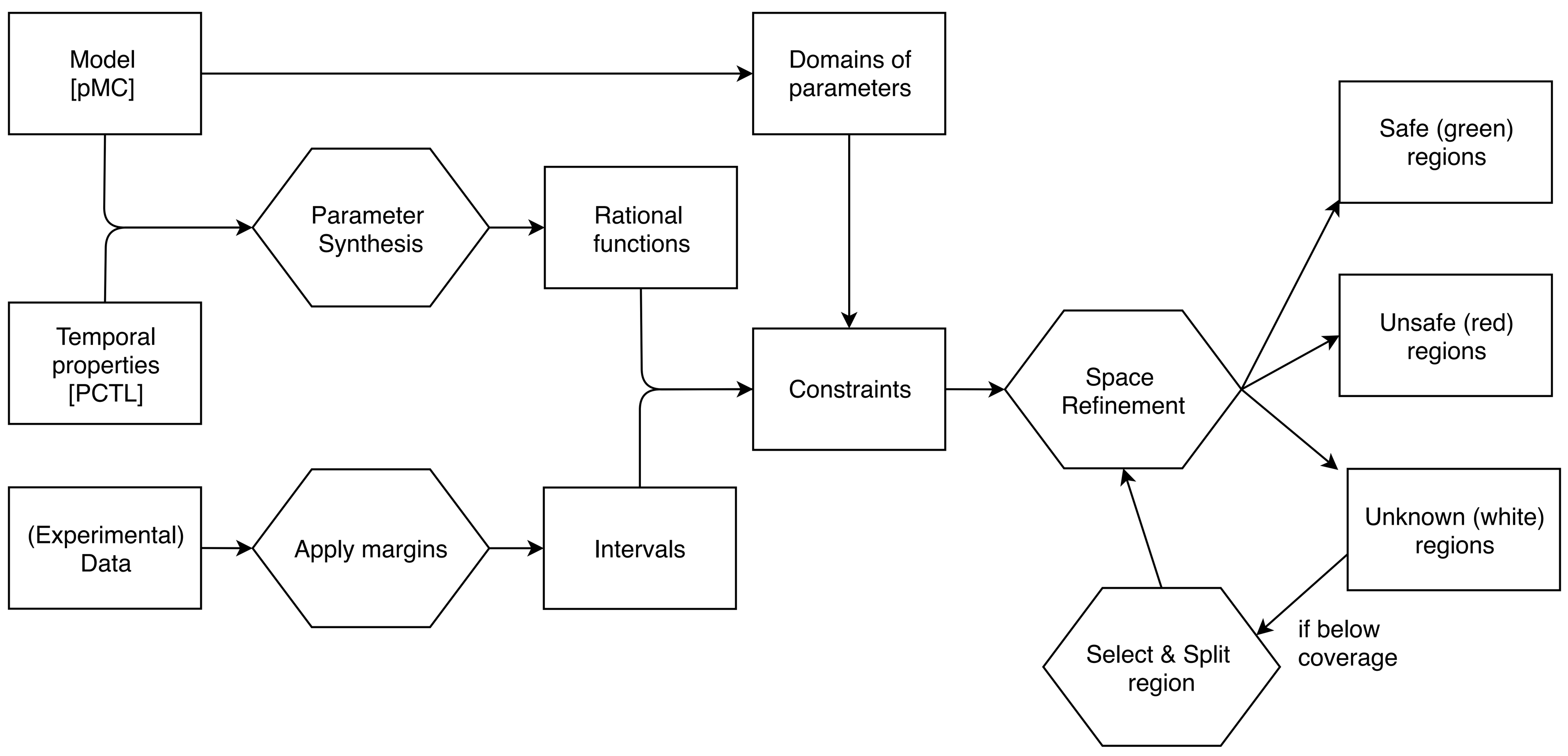

Figure 3: Scheme of the presented workflow.

Table 1: Brief scalability showcase of space refinement from the case study. Constaints on each number of sting $(0, 1, \dots, \#agents)$ were used. Time in seconds. Timeout (TO) is 2 hours.

	two param.								multi param.	
#agents	2		3		5		10		3	
data set	1	2	1	2	1	2	1	2	1	2
alpha, n_samples	coverage_thresh= 0.95, recursion_depth= 15									
0.95, **100**	6.8	1.7	11.1	2.1	41.2	TO	TO	TO	0.7	0.5
0.95, **1500**	0.2	0.2	0.3	0.3	0.1	0.5	307	0.3	0.5	0.6
0.95, **3500**	0.2	0.2	<0.1	0.3	0.1	0.5	3.7	0.3	0.5	0.3

Besides the presented workflow, we provide the following features:

- Sampling of rational functions and parameter space
- In the case of multiple properties, space refinement and sampling operates with conjunction of constrains obtained from the properties.
- Source code in Python available at github.com/xhajnal/mpm
- Currently we are working on a tool with a graphical user interface

References

[1] C. Dehnert, S. Junges, J.-P. Katoen, and M. Volk. A STORM is coming: A modern probabilistic model checker. In *Computer Aided Verification*, pages 592–600. Springer, 2017.

[2] M. Hajnal, M. Nouvian, D. Šafránek, and T. Petrov. Data-informed parameter synthesis for population markov chains. In *Hybrid Systems Biology - 6th International Workshop, HSB 2019*, pages 147–164, 2019.

[3] M. Kwiatkowska, G. Norman, and D. Parker. Prism 4.0: Verification of probabilistic real-time systems. In *International conference on computer aided verification*, pages 585–591. Springer, 2011.

– This research is supported by the Ministry of Science, Research and the Arts of the state of Baden-Württemberg, DFG Centre of Excellence 2117 'Centre for the Advanced Study of Collective Behaviour' (ID: 422037984), Young Scholar Fund (YSF), project no. P83943018FP430_/18, Mentorship grant from the Zukunftskolleg, and Czech Grant Agency grant no. GA18-00178S.