

Optimizing Control Strategy using Statistical Model Checking

Alexandre David¹, Dehui Du², Kim G. Larsen¹,
Axel Legay³, Marius Mikučionis¹

¹ Computer Science, Aalborg University, Denmark

² Laboratory of Trustworthy Computing, East China Normal University, Shanghai,
China

³ INRIA/IRISA, Rennes Cedex, France

This paper proposes a new efficient approach to optimize energy consumption for energy aware buildings. Our approach relies on stochastic hybrid automata for representing energy aware systems. The model is parameterized by several cost values that need to be optimized in order to minimize energy consumption. Our approach exploits a stochastic semantic together with simulation in order to estimate the best value for such parameters. Contrary to existing techniques that would estimate energy consumption for each value of the parameters, our approach relies on a new statistical engine that exploits ANOVA, a technique that can reduce the number of runs needed by the comparison algorithm to perform the estimates. Our approach has been implemented and our experiments show that we clearly outperform the naive approach.

1 Introduction

Cyber-Physical Systems. Cyber-physical systems are large-scale distributed systems, often viewed as networked embedded systems, where a large number of computational components are deployed in a physical environment. Each component collects information about and offers services to its environment (e.g., environmental monitoring and control, health-care monitoring and traffic control). This information is processed either at the component, in the network or at a remote location (e.g., the base station), or in any combination of these.

Characteristic for cyber-physical systems is that they have to meet a multitude of quantitative constraints, e.g., timing constraints, power consumption, memory usage, communication bandwidth, QoS, and often under uncertainty of the behavior of the environment. Existing model-driven methodologies for embedded systems are rather sophisticated in handling functional requirements, and some methods are good at handling special kinds of quantitative constraints. However, there is a lack of a mathematical foundation and supporting tools allowing to handle the combination of quantitative aspects concerning, for example, time, stochastic behavior, hybrid behavior including energy consumption.

In our previous work [10, 11] we have proposed to capture the behavior of cyber-physical systems with Priced Timed Automata (PTA). Those models are extensions of timed automata [2], where clocks may have different rates (even

potentially negative) in different locations. Several projects at the EU level promoted PTA as an adequate model for energy-aware systems. PTAs are as expressive as linear hybrid automata [1] providing high expressive power useful for modeling complex cyber-physical systems, but also rendering most problems either undecidable or too complex to be solved with classical model checking approaches. To overcome these limitations, we proposed in [10] to give a stochastic semantics to PTAs and then apply Statistical Model Checking (SMC) techniques [17, 24, 21, 14, 13], which is a highly scalable simulation-based approach. SMC consists in randomly generating and monitoring simulation runs of the system and verify whether they satisfy a given property written in some temporal logic. The results are then used by statistical algorithms in order to compute an estimate of the probability for the system to satisfy the property with some level of significance. Our work has been implemented in UPPAAL SMC, that is, an extension of UPPAAL that relies on verifying metric interval temporal logic (MITL) properties using our stochastic semantic and statistical model checking algorithms. Our tool comes together with a friendly user interface that allows a user to specify complex problems in an efficient manner as well as to get feedback in the form of probability distributions and compare probabilities to analyze performance aspects of systems. UPPAAL SMC has been applied to a wide range of examples from networking and Nash equilibrium [5] to system biology [9], real-time scheduling [8], and energy-aware systems [7]. A major difference with classical MATLAB Simulink[®] approaches is that ours relies on formal models for both the system and the requirements, hence allowing to express eventually complex properties and behaviors in a straight-forward way.

As a main contribution we improve over the framework presented in [7] for modeling, analyzing and in particular optimizing control strategies for energy aware buildings. The framework consists of several parameterized components (rooms, building, heaters, weather, user, . . .) as well as a collection of properties for evaluating comfort, and energy profiles of various control strategies, i.e., various values for the parameters. The challenge is to find the best value of a parameter to optimize a given property.

We then propose a systematic approach to encode the problem via stochastic hybrid automata. Our approach relies on a new framework to optimize parameters of the controller given in the Hybrid Systems Verification Benchmark of [12] to control temperatures of rooms in a given building.

We address the problem by working with a refined technique called analysis of variance (ANOVA) to compare many distributions in one method potentially more efficient than many pair-wise Student's t-test [18] applications and thus generalizes t-test for many distributions. There can be many different arrangements of those distributions and in particular we are interested in a so called two-factor factorial experiment design [18], where our two parameters become the two factors (two orthogonal dimensions), the parameter values become factor levels (discrete values on those two dimensions), and a cost (discomfort or energy) is the measured outcome value. We can then reuse the data gathered by ANOVA to estimate the energy and discomfort for interesting values of pa-

rameters. “Discomfort” is interpreted as a distance between a desired range of temperature and the current temperature.

We have implemented our technique and show that it works faster than the naive statistical model checking approach.

2 Stochastic Hybrid Automata

In [11], we proposed UPPAAL SMC that is a new release of UPPAAL that supports verification of probabilistic timed automata via simulation-based approaches [22]. Details can be found in [10]. In [7] we generalized the model to stochastic hybrid automata (SHA) that are timed automata whose clock rates can be changed to be constants or expressions depending on other clocks, effectively defining ordinary differential equations (ODEs). UPPAAL SMC⁴ supports fully hybrid automata with ODEs and a few built-in complex functions (such as `sin`, `cos`, `log`, `exp` and `sqrt`) since version 4.1.10. We only recall informally the modeling language of SHA through an example because it is not a contribution in this paper.

A Simple 2-Room Example To illustrate the SHA model as supported by UPPAAL SMC, we consider the case of two independent rooms that can be heated by a single heater shared by the two rooms, i.e., at most one room can be heated at a time. Figure 1a shows the automaton for the heater in UPPAAL notation where the circle inscribed means initial location and U inscribed means urgent location (no time delay is allowed). The heater starts in location `OFF` and turns itself on after picking a delay between $[0, 4]$ as there is no guard controlling lower bound and only an invariant $x \leq 4$ is setting the upper bound for time delay using clock `x`. The delay is chosen with a *uniform probability distribution* over that interval. Then one of the weighted transitions is taken: the plain edge transition from the location `OFF` is branched into either room 0 or room 1 (dashed edge). The dashed edges have probabilistic weights 1 and 3: the room 0 is chosen with probability $\frac{1}{1+3}$ and room 1 with probability $\frac{3}{1+3}$. The heater stays on (location `ON_0` or `ON_1`) for some time, potentially forever. The delay in this case is picked with an *exponential probability distribution*, for which we have to define the rate. We use rate 2 for room 0 and rate 1 for room 1 and the interpretation is that the stochastic controller is more eager to initiate the heating of room 1 than room 0, as well as less eager to stop heating room 1. Both rooms are similar and are modeled by the same template instantiated twice. Figure 1b shows an automaton for room 0. The room is initialized to its initial temperature `T=INIT[0]` by leaving its initial location `Init` which is also a committed location (inscribed with C). Then the temperature `T` evolves by the derivatives $T' = -T/10$ or $T' = K - T/10$ depending on whether the heater is turned on or off. The equations are defined as part of invariant expressions. Furthermore, when the heater is turned on, its heating is not exact and is picked with a uniform distribution of $K \in [9, 12]$, realized by the update `K=9+random(3)`. The variables `T` and `K` are clocks but

⁴ <http://www.uppaal.org/>.

are used more generally as floating point numbers in the hybrid model. For this purpose we *stop* K with the derivative expression $K'=0$. Furthermore, the usage of ODEs, such as $T' = -T/10$, prompts the checker to integrate the value of T .

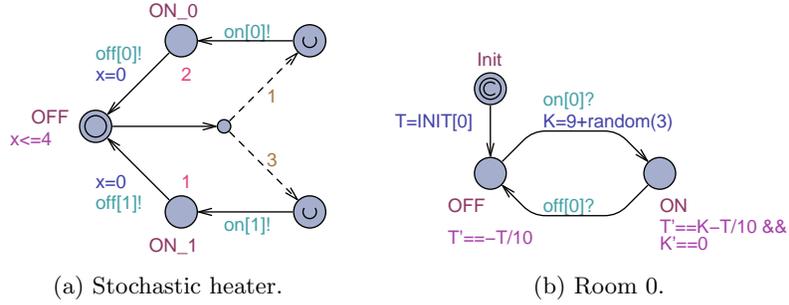


Fig. 1: A simple two room example.

Statistical Model Checking We use SMC [16, 21, 23, 3] to estimate and test on the probability that a random run of a network of SHAs will satisfy a given property. Given a model \mathcal{H} and a trace property φ (e.g. expressed in LTL [19] or MTL [15]), SMC refers to a series of simulation-based techniques that can be used to answer two questions: (1) *Qualitative*: is the probability that a random run of \mathcal{H} will satisfy φ greater or equal to a certain threshold θ (or greater or equal to the probability to satisfy another property φ')? and (2) *Quantitative*: what is the probability that a random run of \mathcal{H} will satisfy φ ? In both cases, the answer will be correct up to a user-specified level of significance or level of confidence that bounds the probability of making a wrong conclusion. Our UPPAAL SMC toolset implements a wide range of SMC algorithms for answering qualitative and quantitative questions on networks of SHAs. The tool supports not only classical reachability and safety properties, but also general weighted MTL properties [6, 4]. One can exploit the quantitative engine of SMC to do parameter sweep in order to optimize some quantity. In this paper, we will rather exploit the statistical method called ANOVA that will ease the estimation of parameters in a single step.

SMC on the 2-Room Example First we visualize the behaviour of our stochastic controller by checking the property $\text{simulate } 1 \ [\leq 120] \{ \text{Room}(0).T, \text{Room}(1).T \}$. Figure 2 shows the evolution of both temperatures (shown with the short names T_0 and T_1). The results are in accordance with the controller automaton. Based on this plot we can now do some quantitative analysis by checking the query

$$\text{Pr}[\leq 120] ([\Box \text{Room}(1).\text{Init} \ || \ \text{Room}(1).T \geq 10])$$

This query asks for an estimate of the probability that the temperature of room 1 stays above 10 degrees for runs bounded by 120 time units. After 4239 runs the checker returns the interval $[0.36, 0.42]$ with 99% confidence level. The tool can also test the hypothesis that this probability is greater or equal than 0.37. After 2523 runs, this is confirmed with a level of significance of 0.01 and an indifference region of size ± 0.01 .

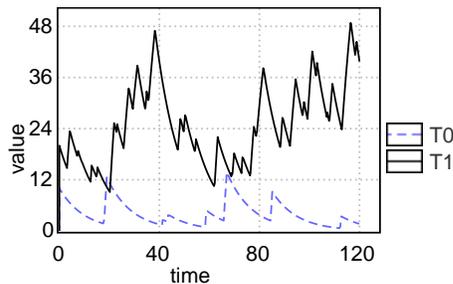


Fig. 2: Evolution of room temperatures.

3 The Energy-Aware Building Challenge

We consider the case of an energy-aware building where rooms are modeled according to the layout shown in Fig. 3. A number of heaters are available and can be moved between rooms. The rooms can transfer heat between each other and an adjacency matrix gives the heat transfer coefficients (not shown here for brevity). The goal is to design a controller that will maintain the room temperatures within acceptable comfort ranges despite adverse weather conditions.

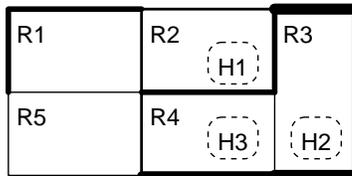


Fig. 3: Layout of the rooms R_i with the heaters H_k .

This case-study reproduces the model-checking challenge of hybrid systems proposed in HSCC [12]. In this paper, we focus on one type of controller for which we want to find good parameters to minimize energy consumption and maximize comfort. Furthermore, the weather model is fixed, though every run has an uncontrollable range of temperatures.

The room temperature dynamics is described by a differential equation:

$$T_i' = \sum_{j \neq i} a_{i,j}(T_j - T_i) + b_i(u - T_i) + c_i h_i$$

where T_i and T_j are the temperatures in room i and j respectively, u is the environment temperature, and h_i is equal to 1 when the heater is turned on in room i and 0 otherwise. The adjacency matrix a gives the heat exchange coefficients $a_{i,j}$ between rooms i and j . The heat exchange with the environment is encoded in a separate vector b , where b_i is a energy loss coefficient for room i . The power supply from heaters is encoded in a vector c , where c_i is a power coefficient for room i . The corresponding hybrid automaton is shown in Fig. 4a. The automaton is maintaining its own Boolean `need[id]` to inform the central controller that a heater is needed and `cold[id]` to keep track of uncomfortable rooms. This improves over the model of [7] since we can accumulate the discomfort over all the rooms with the (dynamic) rate in the (more compact) automaton of Fig. 5b. The

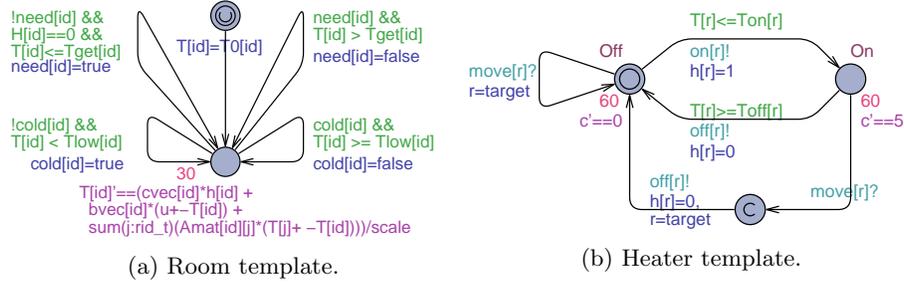


Fig. 4: Stochastic hybrid automata model.

change in modeling discomfort allows us to avoid local optima when searching for good parameters (in the next section) where one room could be heated with low energy while another room would be cold. There are two levels of heater control: primary (local) controller at the individual heaters and secondary (central) controller which determines how the heaters are switched over from one room to another. The local controllers use a *bang-bang* strategy, i.e., when the controlled value (here the temperature $T[r]$) goes below or above a threshold, the controller changes action. In our case, when the temperature $T[r]$ is below $Ton[r]$, the heating is turned on (with $h[r]=1$), and when the temperature $T[r]$ is above $Toff[r]$, the heating is turned off (with $h[r]=0$). The hybrid automaton of the heater is shown in Fig. 4b. The central controller can switch over the heating from one room to another. The room is said to be needing a heater if the temperature drops below its $Tget$ threshold and it is said to be outside the comfort zone if the temperature drops below $Tlow$. We used the thresholds according to [12], based on the heuristics that the temperature difference between rooms should not be too high. This controller is shown in Fig. 5a.

Whenever the heating is turned on, the heaters consume some energy whose rate is determined by the vector pow (power). The monitor automaton keeping track of discomfort (accumulated time spent when rooms are cold) and this energy consumption is shown in Fig. 5b.

Figure 6 shows the automaton used for the user profile: the room parameters are changed based on the time of the day and whether the room is occupied or not. For example, the room is preheated starting from 50'clock in the morning

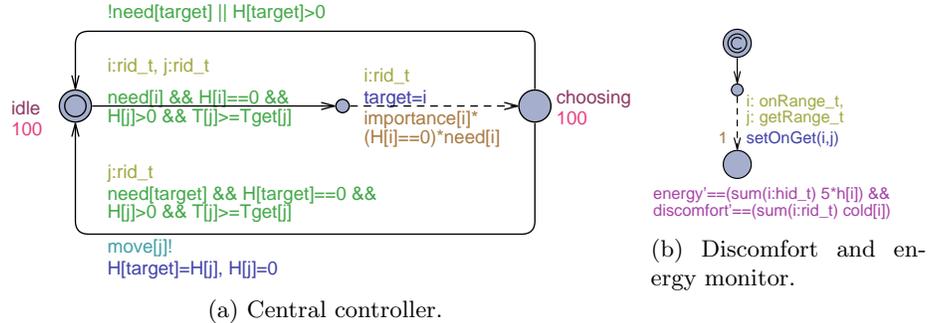


Fig. 5: Stochastic hybrid model for controller and outcome monitor.

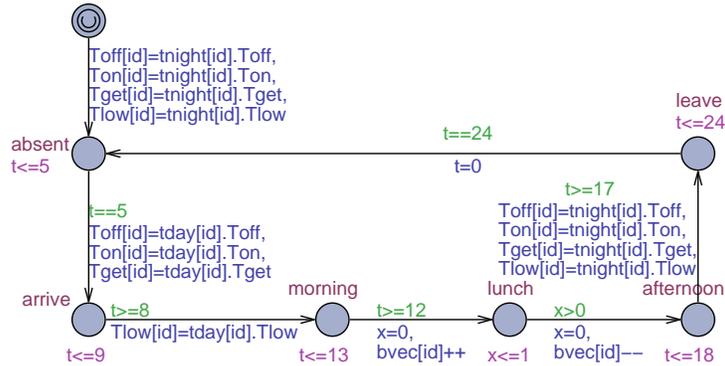


Fig. 6: Stochastic hybrid automaton of the user profile.

and the user arrives between 8 and 9 which increases the **Tlow** threshold requirement. The energy leakage into environment is increased by $bvec[id]++$ when the user opens a window before entering location **lunch** and is decreased by $bvec[id]--$ when the user comes back and closes the window.

The weather model in this study is fixed to be a daily cycle of temperatures varying between $\pm A + B$ where the amplitude A is picked uniformly in $[0, 2]$ and the offset B in $[0, 1]$.

In order to optimize the energy consumption we also model a user profile which assumes that at night it is acceptable for the temperatures to drop to $8^\circ C$. Furthermore, in the morning the rooms should be preheated to about $20^\circ C$ and should not be lower than $17^\circ C$ when the user arrives between 8 and 9 o'clock (with a uniform distribution).

Parameters The goal of this study is to find optimal values for temperature thresholds **Ton** (primary controller) and **Tget** (secondary controller) with respect to energy consumption and user discomfort time. The method is simply to parameterize the model by varying the **Ton** and **Tget** between the values $[16, 22]^\circ C$, and estimate energy consumption and comfort time for various configurations using UPPAAL SMC. The parameter value variation is modeled as a uniform choice during the first discrete stochastic transition in the automaton shown in Fig. 5b. Then the energy and discomfort time is computed using derivative expressions over all rooms by the same automaton.

4 Optimizing Control Strategies

First, we explain the methodology of our approach and then we present the empirical results following the methodology.

4.1 Methodology

We are interested in two aspects of the system: discomfort time and energy consumed – two notions of a cost that are in conflict if we try to optimize both of them, i.e. lowest energy might imply large discomfort and vice-versa, while

some configurations are better compromises. Thus our goal is to identify the best configurations when preferring a lower discomfort for some energy ranges or a lower energy consumption for some ranges of discomfort – the so called Pareto-optimal frontier. But before we find the Pareto frontier, we need to estimate both costs for every configuration.

A simple approach would be to estimate confidence intervals (average \pm standard error) for both costs with some given confidence level (say 95%) and for each configuration of the parameter values, and then compare them. The problem is that it can take a lot of measurements to achieve confidence for every configuration and even then the intervals may still overlap due to higher variance in some combinations. A better suited technique for comparisons is a pair-wise Student’s t-test [18] which reduces the comparison of two distributions to checking that the mean of their differences is below, above or equal to zero. The improvement here is that if two distributions are significantly different then we would observe earlier with less samples that the confidence interval for the differences does not include zero and thus we could conclude with fewer measurements compared to estimating all the individual means. However, the Student’s t-test is a pair-wise test but we have many more than just two configurations, thus we would need to apply this test at least $n \cdot \log(n)$ times with the best sorting algorithm.

Statisticians developed a more refined technique called analysis of variance (ANOVA) to compare many distributions in one method potentially more efficient than many t-test applications and thus generalizes t-test for many configurations. There can be different arrangements, but we are interested in a so called two-factor factorial experiment design [18] in particular: our two parameters become the two factors (two orthogonal dimensions), the parameter values become factor levels (discrete values on those two dimensions) and the cost (discomfort or energy) is the measured outcome value. In such design, we are interested in all pair-wise combinations of parameter values, and those combinations can be arranged on a two dimensional grid. In this experiment design, ANOVA is based on estimating the parameters for a linear model⁵ and computing how much influence each factor has on the outcome. The computed measure, called the F-statistic, is a ratio of a mean square for a particular factor and an error square. The F-statistic is then translated into a P-value by looking up the tables of F-distribution. P-value is called the factor significance: the probability of making an error by stating that the factor has influence on the outcome, thus the smaller the P-value the more confidence that the factor is significant.

An important assumption of this experiment design is that the measurements should be balanced (the amount of samples is the same across all configurations), therefore the minimum amount of data is one sample per each configuration, and at least two samples per configuration if we are also interested in the interaction between factors. Our overall method is described by the following steps:

1. Using UPPAAL SMC, generate enough runs to provide enough measurements for each parameter configuration so that the data is balanced. At least two

⁵ a linear equation predicting the outcomes given the concrete factor values.

runs per configuration are required, but our parameter values are chosen stochastically by UPPAAL SMC, thus there might be some negligible balancing overhead.

2. Apply ANOVA on the gathered and balanced data so far. For this pilot study we use the implementation from the statistical tool R [20], but it is well known in textbooks and simple to implement in any other tool.
3. If our factors are significant then stop data generation and proceed to the next step, otherwise loop back to Step 1 and append more samples.
4. Reuse the gathered samples so far and compute the confidence intervals (average with standard error) for the means of cost (discomfort and energy) for each pair-wise parameter value combination.
5. Compute the Pareto frontier of discomfort and energy over configurations.
6. Present the Pareto frontier as a set of optimal parameter values the user can choose from as a compromise between energy consumption and discomfort.

Figure 7 shows a pipeline overview of operations performed with some steps marked where data is visualized by rectangles and operations as rounded rectangles. The dashed arrows indicate the change of control flow in Step 3. In general, Step 2 is not guaranteed to show factor significance so that Step 3 could proceed to Step 4, even if lots of data is presented. For example significance will not be reported if some factor/parameter has no influence on the measured outcome. Therefore an alternative test is needed to detect the independence in order to terminate the data generation (there can be several options to explore, thus we leave this generalization as a future work).

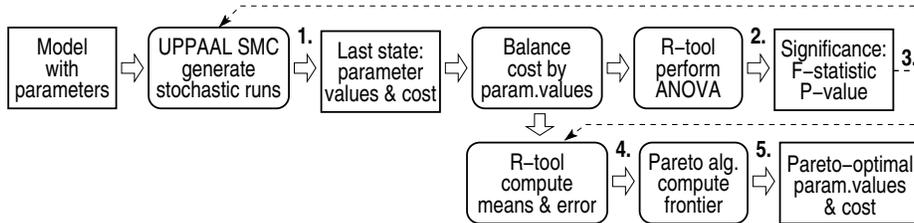


Fig. 7: Pipeline for finding the Pareto-optimal configurations of parameters.

In addition, we pick out a few configurations, estimate means using UPPAAL SMC probabilistic query and compare them to validate some of the results. We also compare the performance of ANOVA based method with a simple mean value estimation method.

4.2 Results

We have parameterized the model with two parameters (factors in our experiment design) represented by temperature thresholds T_{on} and T_{get} each varying between 7 values (factor levels resp.), thus yielding 49 distinct pair-wise combinations to be tested. We instrumented UPPAAL SMC simulator to record the

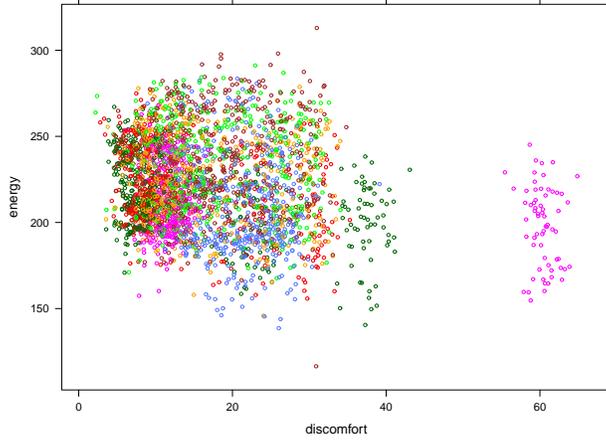


Fig. 8: Scatter plot of measurements colored by parameter value configuration.

last values from a run. Consequently the query `simulate 100 [≤=2*day] { Ton, Tget, discomfort, energy }` generates 100 runs and stores the values of `Ton`, `Tget`, `discomfort` and `energy` as tuples. The gathered results can be plotted in terms of discomfort and energy. However, the individual data is scattered so much that it is impossible to distinguish individual clusters. An example plot of 3136 points is displayed in Fig. 8, where a different color used for every combination, but most configurations result in overlapping clusters (except one on the right), configurations are visually indistinguishable and thus the results need to be processed further.

In our setup the newly produced simulation results are streamed into a small C++ program which balances the data across parameter combinations, i.e. it outputs the data when all combinations have enough data. The balanced data is then analyzed using R scripts applying ANOVA. Table 1 shows an output from ANOVA performed on 98 measurements (two measurements per each combination, and there are $7 \cdot 7$ combinations): the first column shows factors, the fifth contains F-statistic and the sixth – P-value. The coefficients of underlying linear model are appended at the bottom of the table, which means that the model

Table 1: ANOVA table with coefficients of the linear model provided by R [20].

Analysis of Variance Table						
Response: discomfort						
	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Ton	1	4159.3	4159.3	63.8874	3.303e-12	***
Tget	1	0.4	0.4	0.0063	0.9369	
Ton:Tget	1	4.1	4.1	0.0629	0.8026	
Residuals	94	6119.7	65.1			

(Intercept)			Ton	Tget	Ton:Tget	
	60.8283113	-2.2867466	1.0029695	-0.0510851		

Table 2: Summary of sequential applications of ANOVA.

Number of runs	Factor	Discomfort time		Energy consumption	
		<i>F</i> value	P-value	<i>F</i> value	P-value
2 · 49	<i>T_{on}</i>	63.8874	3.30e-12	0.7147	0.4000
	<i>T_{get}</i>	0.0063	0.9369	17.5777	6.24e-05
	<i>T_{on} : T_{get}</i>	0.0629	0.8026	0.7181	0.3989
4 · 49	<i>T_{on}</i>	136.1676	<2e-16	1.1647	0.2818
	<i>T_{get}</i>	0.1537	0.6955	17.9283	3.55e-05
	<i>T_{on} : T_{get}</i>	0.0003	0.9869	0.0582	0.8096
8 · 49	<i>T_{on}</i>	315.7978	<2e-16	2.4425	0.1189
	<i>T_{get}</i>	0.1202	0.7290	35.8938	4.76e-09
	<i>T_{on} : T_{get}</i>	0.0096	0.9218	0.8253	0.3642
16 · 49	<i>T_{on}</i>	629.1384	<2e-16	6.5909	0.01044
	<i>T_{get}</i>	0.5895	0.4429	90.9612	<2e-16
	<i>T_{on} : T_{get}</i>	0.2852	0.5935	5.3053	0.02152
32 · 49	<i>T_{on}</i>	1263.5390	<2e-16	27.9527	1.42e-07
	<i>T_{get}</i>	1.0840	0.2980	172.3296	<2.2e-16
	<i>T_{on} : T_{get}</i>	0.5401	0.4625	3.2632	0.07104
64 · 49	<i>T_{on}</i>	2575.3208	<2e-16	65.6245	7.74e-16
	<i>T_{get}</i>	4.6682	0.0308	405.4892	<2.2e-16
	<i>T_{on} : T_{get}</i>	0.5949	0.4406	0.1926	0.6608

predicts the mean discomfort with expression $60.82 - 2.29 \cdot \text{Ton} + 1.00 \cdot \text{Tget} - 0.05 \cdot \text{Ton} \cdot \text{Tget}$.

In our setup, we apply the ANOVA method each time new simulation data is appended. Table 2 displays a summary of ANOVA results for discomfort time and energy consumption when the amount of measurements is increased with each row. The first iteration with just one sample per configuration is not reliable to detect interactions, thus we start with two. The table shows that even in the first step **Ton** is significant for discomfort (since $P < 0.05$) and **get** is significant for energy. Later more significance emerges as more data is supplied. We stop at iteration of 64 samples per configuration ($64 \cdot 49 = 3136$ simulations) where both factors **Ton** and **Tget** are significant (since $P < 0.05$) in both outcomes. Their interaction (**Ton:Tget**) is still not significant in distinguishing outcomes (since $P > 0.05$). Figures 9a and 9b show the planes of linear models learned by ANOVA: higher **Ton** and **Tget** values are preferred to lower the discomfort, but smaller values are preferred for lower energy. Discomfort and energy are clearly in a conflict. Perhaps a compromise can be found in upper **Ton** and lower **Tget** levels as the plane seems more tilted in the opposite corner which should be avoided. Next, we look at the individual cost estimates.

Figure 9c shows estimated means with standard error bars (95% confidence intervals) aggregated by configurations. The plot shows Pareto-optimal frontier of mean values: points which dominate others by yielding smaller discomfort and energy values. We also plot the dominating minimum values to illustrate that original data is widely scattered.

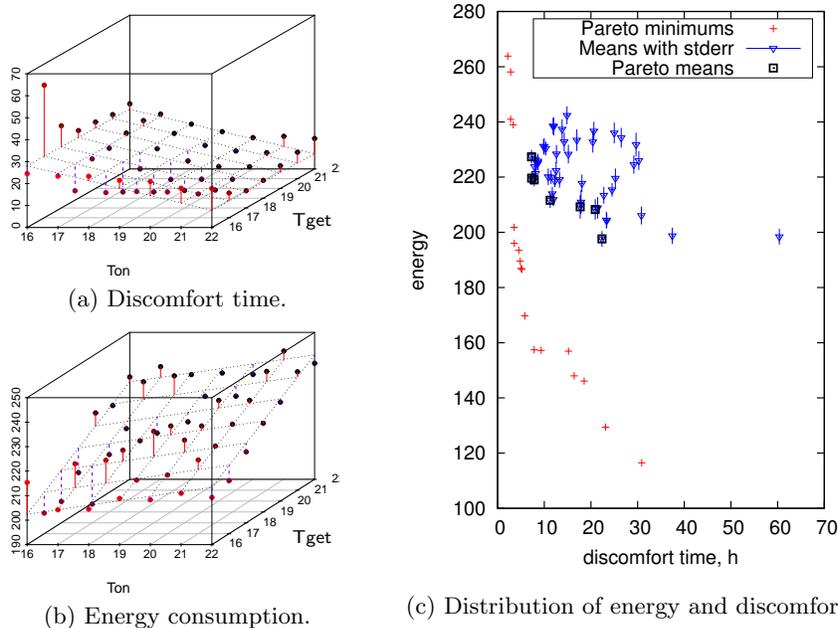


Fig. 9: Estimated energy and discomfort for various parameter combinations.

Figure 10 shows level maps of averages as shades of gray and contours signifying equal levels. Discomfort is greatest on the bottom-left and there is a valley of small preferred values around (17;17) and (18;18) (discomfort=7.29, energy=219.6). The energy consumption is greatest on the top-right with a valley at around (17;18). Pareto-optimal means are marked by larger circles and it seems that configuration (22;18) (discomfort=7.83, energy=219.0) offers nice compromise between discomfort and energy, it is also further from steep slopes which might be at risk of yielding long discomfort time.

The plots show that four (more than half) of Pareto-optimal configurations are in the bottom-right corner as predicted by the ANOVA linear model and the other three are found closer to the center – the surface curving which could not be predicted by a linear model. Thus ANOVA can be used to detect the significant parameters and provide a linear model of overall tendencies.

4.3 Comparison with Estimation

Next we evaluate the efficiency of ANOVA in discriminating the means of all the configurations in contrast to estimating simple costs (energy and discomfort). Two configurations are chosen for more detailed comparison: a Pareto-optimal configuration (22; 18; 7.83 ± 0.23 ; 218.99 ± 2.32) and a non-optimal configuration worse than this optimal (21; 18; 8.16 ± 0.26 ; 221.38 ± 2.33). To validate our results we would like to use our SMC technique in UPPAAL SMC to evaluate the means and compare them but we cannot directly do that. Instead we check the

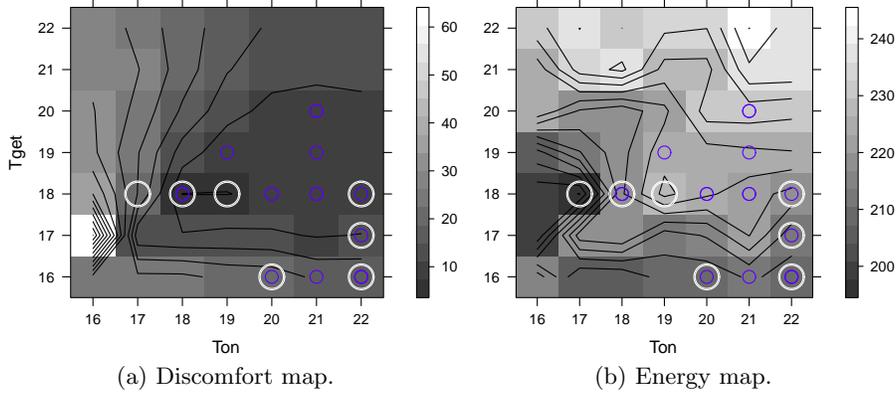


Fig. 10: Level maps of estimated mean values where Pareto-optimal configurations marked as: small circles for the minimum values and larger circles for mean.

following queries $\Pr[\text{discomfort} \leq 100] (\langle \text{time} \rangle = 2 \cdot \text{day})$ and $\Pr[\text{energy} \leq 1000] (\langle \text{time} \rangle = 2 \cdot \text{day})$. The actual probability is not interesting here (will be close to one), but rather the resulting distribution over discomfort and energy. We can then derive the mean that we want from this data as shown in Fig. 11.

The results for the Pareto-optimal configuration is in solid red lines. A summary of estimated means is described in a Table 3. The estimated means for combination (22;18) (discomfort 7.83 and energy 220.0) are smaller than in alternative configuration (21;18) (7.86 and 222.1 resp.), and thus ($\text{Ton}=22; \text{Tget}=18$) is a slightly better choice than ($\text{Ton}=21; \text{Tget}=18$). Another Pareto-optimal configuration ($\text{Ton}=22; \text{Tget}=17$) uses less energy, but the discomfort is noticeably larger, thus incomparable with (22;18).

Table 3: Estimated means.

Ton	Tget	discomfort	energy	Optimal?
22	18	7.83	220.0	Pareto
21	18	7.86	222.1	no
22	19	8.57	226.8	no
22	17	11.22	214.8	Pareto

UPPAAL SMC used 738 runs (80s) for each query to compute the cost with 95% confidence level (the confidence interval still needs to be computed). In principle it would take $7 \cdot 7$ such queries to estimate discomfort for each configuration (plus the same amount for energy), thus in total 36162 simulation

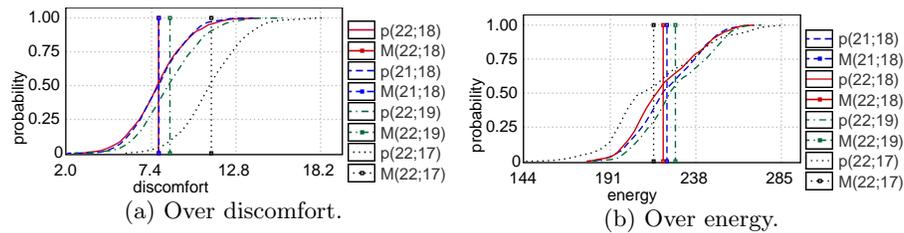


Fig. 11: Probability of reaching 2 days.

runs ($\approx 3920s \approx 1h\ 5min^6$). The ANOVA-based analysis required only 3136 simulation runs ($\approx 366s = 6min\ 6s$) – an improvement of 11.5 times. Thus we conclude that ANOVA method requires less measurements and consequently less simulations in order to differentiate and pick the optimal configurations.

5 Conclusion

The analysis of variance has been used in a sequential manner to decide if there is enough data for distinguishing the effects of two factors on two different costs in a two-factor factorial design. The ANOVA method can identify the significant factors by computing the F-statistic, however it can be problematic if the chosen factor has no influence on the outcomes (the P-value does not converge), thus an alternative test is needed to conclude independence to ensure termination.

We have demonstrated the technique on an energy aware buildings example and have identified Pareto-optimal configurations in terms of both discomfort and energy consumption. Thus SMC can be used to analyze complex models and determine cost-optimal parameter values using statistically efficient methods. The approach can also be distributed across a cluster of computers, but the load balancing algorithm need also to be fair with respect to parameter values as required by the analysis of variance.

There are many other experiment design variations including more factors and thus in the future it would be interesting to generalize the ANOVA method and implement the support for parametric SMC using general factorial designs inside the model checker.

References

1. R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.*, 138(1):3–34, 1995.
2. R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
3. S. Bensalem, B. Delahaye, and A. Legay. Statistical model checking: Present and future. In *RV*. Springer-Verlag, 2010.
4. P. Bulychev, A. David, K. G. Larsen, A. Legay, G. Li, and D. B. Poulsen. Rewrite-based statistical model checking of wmtl. In S. Qadeer and S. Tasiran, editors, *Runtime Verification*, volume 7687 of *Lecture Notes in Computer Science*, pages 260–275. Springer Berlin Heidelberg, 2013.
5. P. Bulychev, A. David, K. G. Larsen, A. Legay, and M. Mikučionis. Computing nash equilibrium in wireless ad hoc networks: A simulation-based approach. In J. Reich and B. Finkbeiner, editors, *Second International Workshop on Interactions, Games and Protocols*, volume 78 of *EPTCS*, pages 1–14, 2012.
6. P. E. Bulychev, A. David, K. G. Larsen, A. Legay, G. Li, D. B. Poulsen, and A. Stainer. Monitor-based statistical model checking for weighted metric temporal logic. In *LPAR*, pages 168–182, 2012.

⁶ as measured on an Intel Core i7-2600 processor.

7. A. David, D. Du, K. G. Larsen, A. Legay, M. Mikučionis, D. B. Poulsen, and S. Sedwards. Statistical model checking for stochastic hybrid systems. In E. Bartocci and L. Bortolussi, editors, *1st International Workshop on Hybrid Systems and Biology*, volume 92 of *Electronic Proceedings in Theoretical Computer Science*, pages 122–136. Open Publishing Association, 2012.
8. A. David, K. G. Larsen, A. Legay, and M. Mikučionis. Schedulability of herschel-planck revisited using statistical model checking. In T. Margaria and B. Steffen, editors, *ISoLA (2)*, volume 7610 of *Lecture Notes in Computer Science*, pages 293–307. Springer, 2012.
9. A. David, K. G. Larsen, A. Legay, M. Mikučionis, D. B. Poulsen, and S. Sedwards. Runtime verification of biological systems. In *ISoLA (1)*, pages 388–404, 2012.
10. A. David, K. G. Larsen, A. Legay, M. Mikučionis, D. B. Poulsen, J. V. Vliet, and Z. Wang. Statistical model checking for networks of priced timed automata. In *FORMATS*, LNCS, pages 80–96. Springer, 2011.
11. A. David, K. G. Larsen, A. Legay, M. Mikučionis, and Z. Wang. Time for statistical model checking of real-time systems. In *Proceedings of the 23rd international conference on Computer aided verification*, LNCS, pages 349–355, Berlin, Heidelberg, 2011. Springer-Verlag.
12. A. Fehnker and F. Ivancic. Benchmarks for hybrid systems verification. In R. Alur and G. J. Pappas, editors, *HSCC*, volume 2993 of *Lecture Notes in Computer Science*, pages 326–341. Springer, 2004.
13. A. Hartmanns and H. Hermanns. A modest approach to checking probabilistic timed automata. In *Quantitative Evaluation of Systems, 2009. QEST '09. Sixth International Conference on the*, pages 187–196, sept. 2009.
14. J.-P. Katoen, I. S. Zapreev, E. M. Hahn, H. Hermanns, and D. N. Jansen. The ins and outs of the probabilistic model checker MRMC. In *Proc. of 6th Int. Conference on the Quantitative Evaluation of Systems (QEST)*, pages 167–176. IEEE Computer Society, 2009.
15. R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.
16. K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Inf. Comput.*, 94(1):1–28, 1991.
17. A. Legay, B. Delahaye, and S. Bensalem. Statistical model checking: An overview. In *RV*, volume 6418 of *Lecture Notes in Computer Science*, pages 122–135. Springer, 2010.
18. D. C. Montgomery. *Design and Analysis of Experiments*. John Wiley & Sons, Inc, 1997,2001. ISBN 0-471-31649-0.
19. A. Pnueli. The temporal logic of programs. In *FOCS*, pages 46–57, 1977.
20. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.
21. K. Sen, M. Viswanathan, and G. Agha. Statistical model checking of black-box probabilistic systems. In *CAV*, LNCS 3114, pages 202–215. Springer, 2004.
22. A. Wald. Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(2):117–186, 1945.
23. H. L. S. Younes and R. G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *Proc. of 14th Int. Conference on Computer Aided Verification (CAV)*, LNCS 2404, pages 223–235. Springer, 2002.
24. H. L. S. Younes and R. G. Simmons. Statistical probabilistic model checking with a focus on time-bounded properties. *Inf. Comput.*, 204(9):1368–1409, 2006.