2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)
August 23-27, 2021. Lyon, France

Efficient Optimization-Based Falsification of Cyber-Physical Systems with Multiple Conjunctive Requirements

Logan Mathesen[1], Giulia Pedrielli[1], and Georgios Fainekos[1]

2021 IEEE 17th International Conference on Automation Science and Engineering (CASE) | 978-1-6654-1873-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/CASE49439.2021.9551474

***Abstract*— Optimization-based falsification, or search-based testing, is a method of automatic test generation for Cyber-Physical System (CPS) safety evaluation. CPS safety evaluation is guided by high level system requirements that are expressed in Signal Temporal Logic (STL). Trajectories from executed CPS simulations are evaluated against STL requirements using satisfaction robustness as a quantitative metric. In particular, robustness is the distance metric between the simulated system trajectory, associated to a specific input, and the known unsafe set, i.e., regions of the search space that violate the requirements. Identification of violations can be formulated as an optimization problem, where inputs that minimize the robustness function are of interest. In fact, an input falsifies a requirement if the associated robustness is negative.**

In this work, specifically, we consider the case where multiple requirements determine the unsafe set. Due to the computational burden of executing CPS simulations, practitioners often test all system requirements simultaneously by combining the requirement components and obtaining so-called "conjunctive requirements". Conjunctive requirements can challenge optimization-based falsification approaches due to the fact that the robustness function may "mask" the contributions of individual conjunctive requirement components. We propose a new algorithm, minimum Bayesian optimization (minBO), that deals with this problem by considering the contributions of each component of the conjunctive requirement. We show the advantages of the minBO optimization algorithm when applied to general non-linear non-convex optimization problems as well as when applied to realistic falsification applications.

I. INTRODUCTION

Falsification of Cyber-Physical Systems (CPS) refers to methods that attempt to demonstrate that a given system-level requirement is not satisfied over a CPS model. Falsification problems have gained prominence in the field of safety critical systems, where any behavior that violates system requirements must be detected and corrected prior to real world CPS implementation and deployment. Due to its importance in practice, CPS falsification via the discovery of counter-examples to system specifications (requirements) has drawn attention from academia and industry [1], [2].

Complex CPS requirements can be typically formalized using Signal Temporal Logic (STL). STL specifies requirements as sequences of unsafe states combined with timing and logical or reactive requirements [3]. Motivated by the need to falsify complex requirements that are composed via STL statements, the falsification problem has been translated into a minimization problem through the notion of STL robustness [4], [5]. In the following, we assume that the CPS under test can be represented as a model $\mathcal{M}$ which takes as input a vector of initial conditions x_0, a vector of parameters p and a vector signal u, and returns as output a vector signal $\mathcal{M}(x_0, p, u)$. In practice, the input signals u need to be finitely parameterized over time as well (see [4] for details and [6] on how this constraint can be removed). Therefore, the output behaviors of a CPS model $\mathcal{M}$ can be explored over a finite dimensional continuous search space X, i.e., for any point $\mathbf{x} \in X$, we can observe a model behavior $\mathcal{M}(\mathbf{x})$.

Formally, the falsification problem can be stated as: given a system specification φ in STL and a system model $\mathcal{M}$, find a point $\mathbf{x}$ (initial conditions x_0, parameters p and input signals u) such that $\mathcal{M}(\mathbf{x})$ violates φ. STL robustness is a quantitative measure that captures how robustly a system behavior, or trajectory, $\mathcal{M}(\mathbf{x})$ satisfies a requirement φ. In other words, the robustness quantifies how close $\mathcal{M}(\mathbf{x})$ came to falsifying requirement φ. We will use $\rho_\varphi(\mathbf{x})$ to refer to the robustness function that assigns an extended real number to the system behavior $\mathcal{M}(\mathbf{x})$, i.e., $\rho_\varphi(\mathbf{x}) \in \mathbb{R} \cup \{-\infty, \infty\}$. Large positive robustness values imply that the behavior is robustly satisfied (very far from violating the STL), while large negative values imply a behavior is heavily violating the STL and is robustly unsafe. The notion of STL robustness and robust semantics has allowed for the development of optimization-based falsification, where the resulting minimization problem searches for system inputs and parameters such that the resulting STL robustness $\rho_\varphi(\mathbf{x})$ is less than or equal to zero, thus falsifying the STL specification.

The transformation of the falsification problem into an optimization problem has initiated new research directions into re-purposing or developing optimization methods which take into account the structure of the falsification problem [7], [4], [8]. Common optimization methods such as Cross-Entropy [9], Simulated Annealing [4], and Tabu Search [10] have been adopted for falsification; techniques such as Monte Carlo Tree Search and Deep Reinforcement Learning have also been applied in recent years [11], [12]. The Stochastic Optimization with Adaptive Restart (SOAR) framework, which is a combined global local stochastic search described in [13], was applied to and proven quite successful over a range of falsification problems [14], [1], [2]. A distinguishing feature of SOAR ([14]) is that it can quantify the robustness uncertainty when no falsifications are found within the sampling budget – currently, SOAR is the only falsification

* Logan Mathesen is supported as a NSF Graduate Research Fellow, grant number 026257-001. This research has also been partially supported by NSF CNS 1932068.

[1] Computing Informatics and Decision Systems Engineering (CIDSE), Arizona State University, 699 S Mill Ave, Tempe, AZ 85281, USA Logan.Mathesen@asu.edu, Giulia.Pedrielli,fainekos@asu.edu

method capable of doing so.

In many practical applications of falsification, the system evaluator may desire to test several system requirements, φ_i $i = 1, \ldots, n$, to certify the safety of the system. The system model $\mathcal{M}$ often has a very long simulation execution time, such that a single observation $\mathcal{M}(\mathbf{x})$ can take minutes to hours to be generated [15]. On the other hand, the robustness calculations associated with each conjunctive component $\rho_{\varphi_i}(u)$ are fast. This imbalance in computation cost makes it undesirable to test each requirement φ_i individually. An alternative approach is to test all n requirements simultaneously by creating a *conjunctive requirement* such that $\varphi := \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$. Thus, if any conjunctive component φ_i is violated, then the collective conjunctive requirement φ is also violated. For conjunctive requirements, the robustness function is computed as the minimum of the robustness function of each sub-requirement, i.e.,

$$\rho_{\varphi}(\mathbf{x}) = \min_{i=1,\ldots,n} \rho_{\varphi_i}(\mathbf{x}).$$

Although using conjunctive requirements is computationally attractive, it presents the challenge of hiding the contributions of each requirement to the robustness function. This is a problem that is related to the "scale problem" in falsification as discussed in [16], but it also manifests itself even when all the variables have been normalized to the same range. The scale effect can be easily explained and conceptualized when the scale of the robustness of each conjunctive component is very different. For example, consider the conjunctive requirement where a system always maintains a speed less than 20 mph ($\varphi_1 := \Box speed < 20$) and the rpm are always less than 4000 ($\varphi_2 := rpm < 4000$), yielding $\varphi := \varphi_1 \wedge \varphi_2$. In this case the first component φ_1 has robustness values on the order of tens while φ_2 robustness is on the order of thousands. As such φ_2 is masked by φ_1 and it is not possible for optimization-based falsification to use the information of both components. The same principle can occur and reduce falsification efficiency even when all the variables have been normalized to the same scale.

Contribution: We present the problem of conjunctive requirement falsification as a minimization problem of a function which is the minimum of a number of components. In light of such structure knowledge, we propose a new Bayesian optimization algorithm that takes advantage of the structure of the conjunctive requirements and explicitly models information from each resulting robustness component. The result is the algorithm minimum Bayesian optimization (minBO). For testing purposes we imported the algorithm within the S-TaLiRo CPS falsification tool [17], thus verifying the increased falsification performance over two standard benchmark problems for the CPS community [2].

II. RELATED WORK

A. Scale Problem in CPS Falsification

The scale problem in its various instantiations has been reported in several works. In [18] and [19] the problem is encountered tangentially while investigating vacuity aware falsification, and interface-aware STL, respectively. These approaches rely on Boolean connectives to falsify antecedent-consequent pairs and, thus, tangentially mitigate the general scale problem by sequentially addressing the antecedent or consequent sub-formulas. The work in [16] is most related to our research since it directly addresses the scale problem. The authors aim to tackle the scale problem by making use of a multi-armed bandit approach, where sub-formulas are considered as arms in the multi-armed bandit problem. However, the foundation of [16] is built to falsify problem requirements of the form $\Box_I(\varphi_1 \vee \varphi_2)$ or $\Box_I(\varphi_1 \wedge \varphi_2)$. Thus, the method is not applicable to our more general requirement form $\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$. The most recent and related work, presented in [20], focuses on the conjunctive requirement *synthesis* problem, i.e., finding an input $\mathbf{x}$ such that $\rho_{\varphi_i}(\mathcal{M}(\mathbf{x})) > 0$ for all $i = 1, \ldots, n$. On the other hand, the conjunctive falsification problem aims to find $\mathbf{x}$ such that $\rho_{\varphi_i}(\mathcal{M}(\mathbf{x})) \leq 0$ for at least one i. Considering the structure of the conjunctive synthesis problem, the authors in [20] employ a constrained co-variance matrix adaption evolution strategy (CMA-ES).

B. Stochastic Search Optimization Methods

As mentioned in Section I, optimization-based falsification has seen a proliferation with respect to CPS [7], [4], [9], [11], [12]. Stochastic optimization techniques such as simulated annealing, genetic algorithms, ant colony optimization, and the cross-entropy method have been applied in this domain. However, these methods notably lack sample efficiency, partly due to the difficulty in setting the numerous hyper-parameters for methods with memory, and the inability of exploiting information from previous iterations for memory free methods. As an example of memory free sampling, hit and run, which in a common implementation of uniform random sampling, epitomizes myopic search: locations iteratively evaluated have no impact upon subsequent sampling decisions. The benefit of these stochastic search techniques is their easy to derive guarantees in terms of coverage.

On the other hand, local search and hill climbing techniques (which are deterministic with noiseless function evaluations) such as CMA-ES, simplex search, trust region search, response surface methodology, and gradient ascent/descent have increased sample efficiency. However, due to the notorious non-linearity of robustness landscapes in CPS falsification, these local techniques often get trapped in sub-optimal local regions as these searches lack explorative properties.

Bayesian optimization (BO) is a popular black-box stochastic optimization method which has proven quite successful in simulation-based optimization problems [21]. BO balances exploration and exploitation via surrogate modeling to produce high quality solutions in a relatively small number of iterations. However, due to the overhead costs associated to BO, such as surrogate model estimation and acquisition function optimization, the technique is recommended to be employed when observations of the objective function are expensive to collect- as in the case of observing the robust-

ness $\rho_\varphi(\mathcal{M}(\mathbf{x}))$ for a given input $\mathbf{x}$. BO has proven to be quite successful over CPS falsification problems [22], [23], [24]. Recently BO was combined with a local trust region search in an intelligent global-local optimization framework and proved highly effective for CPS falsification [13], [14].

III. PROPOSED METHOD: MINIMUM BAYESIAN OPTIMIZATION

We want to efficiently find, if they exist, falsifications of conjunctive requirements for CPS. We approach the falsification as a global optimization problem:

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x}) \quad (1)$$
$$\text{with } f(\mathbf{x}) = \min\left(h_1(\mathbf{x}), \ldots, h_n(\mathbf{x})\right)$$

where $\mathbb{X} \subset \mathbb{R}^d$ represents the input domain, $\mathbf{x}$ is the input to the model $\mathcal{M}$, and $h_i(\mathbf{x})$ is the robustness associated to the i^{th} conjunctive component, i.e, $h_i(\mathbf{x}) = \rho_{\varphi_i}(\mathcal{M}(\mathbf{x}))$. We assume that $h_i(\mathbf{x})$ for $i = 1, \ldots, n$ are simultaneously returned with a single evaluation of $f(\mathbf{x})$. This assumption aligns with the structure of the conjunctive falsification problem: once the simulated trajectory $\mathcal{M}(\mathbf{x})$ is observed, then the robustness calculation for φ_i is an iterative computation over the system output.

We propose a variant of Bayesian optimization (BO) for solving (1) that we call Minimum Bayesian Optimization (minBO). The goal of minBO is to remove the masking effect that occurs when observing $f(\mathbf{x})$ by leveraging the information learned about the individual components $h_i(\mathbf{x})$. Algorithm 1 outlines the inputs needed and algorithmic steps of minBO.

We initialize minBO by generating and sampling a random Latin hypercube design with b_0 locations. After initialization, each iteration of minBO includes n inner BO iterations where Gaussian processes (GPs) are estimated for each of the $i = 1, \ldots, n$ components. Each GP is built on the same sample of locations $\mathbf{x}_{\text{train}}$, collecting the specific sub-requirement data $\mathbf{y}^i_{\text{train}} = h_i(\mathbf{x}_{\text{train}})$; where $\mathbf{x}_{\text{train}} \in \mathbb{R}^{t\times d}$, $\mathbf{y}^i_{\text{train}} \in \mathbb{R}^{t\times 1}$, and t is the number of evaluations taken so far (the number of executions of the system simulator $\mathcal{M}$). For details on GP estimation we refer readers to [25]. After GPs have been estimated for all n components, we will have n models, i.e., $\left(\hat{h}_i(\mathbf{x}), \hat{s}_i^2(\mathbf{x})\right)$. We then maximize the Expected Improvement $\text{EI}_i(\mathbf{x})$ for each component, with respect to the best observed sample across all components $y^* = \min_{i=1,\ldots,n} \min_{j=1,\ldots,t} \mathbf{y}^i_{\text{train},j}$ where $\mathbf{y}^i_{\text{train},j}$ is the j^{th} observation of component i, we formalize as:

$$\text{EI}_i(\mathbf{x}) = E\Big[\max\left([y^* - \hat{h}_i(\mathbf{x})]\Phi\left(\frac{y^* - \hat{h}_i(\mathbf{x})}{\hat{s}_i(\mathbf{x})}\right)\right. \quad (2)$$
$$\left. + \hat{s}_i(\mathbf{x})\phi\left(\frac{y^* - \hat{h}_i(\mathbf{x})}{\hat{s}_i(\mathbf{x})}\right), 0\right)\Big].$$

We set $\mathbf{x}^i_{\text{EI}} \leftarrow \arg\max_{\mathbf{x}\in\mathbb{X}} \text{EI}_i(\mathbf{x}), i = 1, \ldots, n$ and set the next location to be sampled as: $\mathbf{x}^*_{\text{EI}} \leftarrow \arg\max_{i=1,\ldots,n} \mathbf{x}^i_{\text{EI}}$. After sampling $\mathbf{x}^*_{\text{EI}}$, we update $\mathbf{x}_{\text{train}}$ and all of the $\mathbf{y}^i_{\text{train}}$. This completes a single iteration and we begin a new iteration by re-estimating the GPs.

Algorithm 1 Minimum Bayesian Optimization: minBO

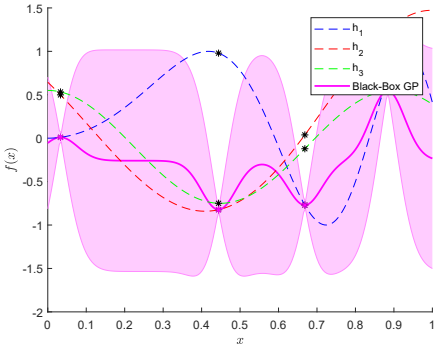
Input: domain $\mathbb{X} \subset \mathbb{R}^d$, n components $\{h_1(\mathbf{x}), \ldots, h_n(\mathbf{x})\}$, objective function $f(\mathbf{x}) = \min\left(h_1(\mathbf{x}), \ldots, h_n(\mathbf{x})\right)$, initialization budget b_0, and total budget T

Output: best location and value $\mathbf{x}^*_{\text{minBO}} \in \mathbb{X}$, $f(\mathbf{x}^*_{\text{minBO}})$

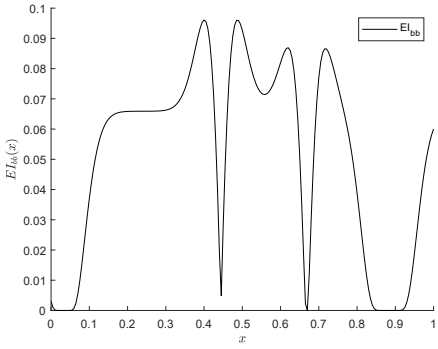
Step 1: Create initializing Latin Hypercube design $\mathbf{x}_{\text{train}}$ with b_0 locations from $\mathbb{X}$, $\mathbf{x}_{\text{train}} \in \mathbb{R}^{b_0\times d}$
Step 2: Sample $\mathbf{x}_{\text{train}}$ over the n composite functions, set $\mathbf{y}^i_{\text{train}} = h_i(\mathbf{x}_{\text{train}})$ for $i = 1, \ldots, n$.
Step 3: set $t \leftarrow b_0$
while $t < T$ **do**
  for $i = 1, \ldots, n$ **do**
    Step 4: Estimate a GP using the training data $\{\mathbf{x}_{\text{train}}, \mathbf{y}^i_{\text{train}}\}$, resulting in $\left(\hat{h}_i(\mathbf{x}), \hat{s}_i^2(\mathbf{x})\right)$ for all $\mathbf{x} \in \mathbb{X}$
    Step 5: $\mathbf{x}^i_{\text{EI}} \leftarrow \arg\max_{\mathbf{x}\in\mathbb{X}} \text{EI}_i(\mathbf{x})$
  end for
  Step 6: $\mathbf{x}^*_{\text{EI}} \leftarrow \arg\max_{i=1,\ldots,n} \mathbf{x}^i_{\text{EI}}$, append $\mathbf{x}^*_{\text{EI}}$ to $\mathbf{x}_{\text{train}}$
  Step 7: Sample and append $h_i(\mathbf{x}^*_{\text{EI}})$ to $\mathbf{y}^i_{\text{train}}$, for $i = 1, \ldots, n$.
  Step 8: $t \leftarrow t + 1$
end while
Step 9: Report best observed location and value $\mathbf{x}^*_{\text{minBO}}, f(\mathbf{x}^*_{\text{minBO}})$.

Figures 1 and 2 illustrate the difference between the standard BO approach and our proposed minBO approach over a 1-dimensional example. Figure 1(a) shows the GP associated to the black-box BO approach when optimizing $f(x)$ with four sample locations. The four sample locations yield the associated $h_1(x)$, $h_2(x)$, and $h_3(x)$ values, however the GP is fit only to the minimum value at each location. Figure 2(a) shows the three GPs obtained by fitting each of the individual components with the same sample locations and values as Figure 1(a). Figures 1(b)-2(b) show the respective EI functions. Note that the global minimum is achieved by $h_3(x)$ at $x = 0.72$. Inspecting Figures 1(b)-2(b), we see that the standard BO approach would sample $x = 0.4$ as the EI maximizer, while our minBO approach is immediately drawn towards the true global optimum and would sample $x = 0.67$ as the maximum EI maximizer across the three GPs. This example illustrates how modeling each of the components in minBO yields richer insight into the problem and effectively removes the *masking* problem.

In regards to the computational complexity of minBO, we know BO algorithms scale at $\mathcal{O}(m^2)$ (with a $\mathcal{O}(m^3)$ preparatory GP covariance matrix inversion steps), where m is the number of function evaluations [21]. Due to the nature of the minBO algorithm, which executes n independent BO iterations during each minBO iteration, minBO scales linearly with respect to BO with a computational complexity of $\mathcal{O}(nm^2)$ where n is the number of requirements. We note that it is important to distinguish between function evaluation costs, i.e., the cost of executing a simulation to observe the

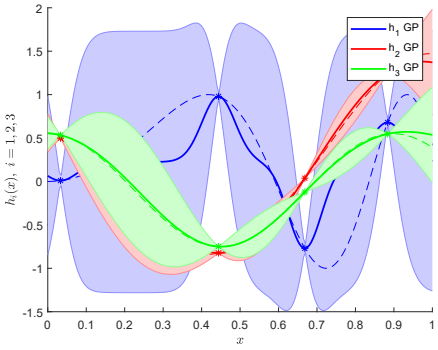


(a) GP estimation $\left(\hat{f}(\mathbf{x}), \hat{s}(\mathbf{x})\right)$.

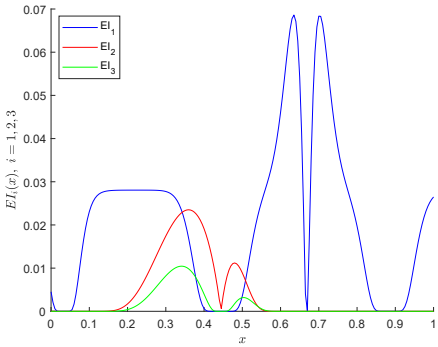


(b) EI function over $\left(\hat{f}(\mathbf{x}), \hat{s}(\mathbf{x})\right)$.

Fig. 1. Standard black-box Bayesian optimization approach for 1 dimensional problem $f(\mathbf{x}) = \min\left(h_1(\mathbf{x}), h_2(\mathbf{x}), h_3(\mathbf{x})\right)$, with $t = 4$ samples.



(a) Component GP estimations $\left(\hat{h}_i(\mathbf{x}), \hat{s}_i(\mathbf{x})\right)$.



(b) EI_i functions over $\left(\hat{h}_i(\mathbf{x}), \hat{s}_i(\mathbf{x})\right)$.

Fig. 2. Minimun Bayesian optimization approach for 1 dimensional problem $f(\mathbf{x}) = \min\left(h_1(\mathbf{x}), h_2(\mathbf{x}), h_3(\mathbf{x})\right)$, with $t = 4$ samples.

trajectory $\mathcal{M}(\mathbf{x})$, and the estimation and optimization costs of GP modeling and EI maximization. In CPS falsification, it is generally accepted that the observation cost of $\mathcal{M}(\mathbf{x})$ largely outweighs any other computational costs. Given that minBO takes a single observation at each iteration, the additional linearly scaling computational cost for estimation and optimization can be considered negligible to the overall falsification run time.

IV. EXPERIMENTATION

We execute two sets of experiments to evaluate minBO's performance relative to standard BO. The first set of experiments focuses on performance over a synthetic optimization problem, where the test functions and associated global optimum values and locations are known. The second set of experiments implements minBO within the S-TaLiRo falsification toolbox and tests falsification performance on benchmark CPS Simulink models with conjunctive requirements.

A. Multiple Component Function Optimization

Our theoretic function optimization tests over objective functions that are the minimum over multiple component functions, $f(\mathbf{x}) = \min(h_1(\mathbf{x}), \ldots, h_n(\mathbf{x}))$, matching the problem structure presented in (1). We perform experimentation over two and three dimensional problems, both problems have three component functions, i.e., $n = 3$.

The two dimensional problem has component functions:

$$h_1(\mathbf{x}) = 205 - 100\left(\sin\left(\frac{x_1}{3}\right) + \sin\left(\frac{x_2}{3}\right)\right)$$
$$h_2(\mathbf{x}) = 155 - 75\left(\cos\left(\frac{x_1}{2.5} + 15\right) + \cos\left(\frac{x_2}{2.5} + 15\right)\right)$$
$$h_3(\mathbf{x}) = (x_1 - 7)^2 + (x_2 - 7)^2 - \cos\left(\frac{x_1 - 7}{2.75}\right) - \cos\left(\frac{x_2 - 7}{2.75}\right)$$

the global minimum is at $\mathbf{x}^* = [7, 7]$, where the third component realizes the global minimum value $f(\mathbf{x}^*) = h_3(\mathbf{x}^*) = 0$; the first two components both have minimum values of 5. The three dimensional problem has components:

$$h_1(\mathbf{x}) = 305 - 100\left(\sum_{j=1}^{3} \sin\left(\frac{x_i}{3}\right)\right)$$
$$h_2(\mathbf{x}) = 230 - 75\left(\sum_{j=1}^{3} \cos\left(\frac{x_i}{2.5} + 15\right)\right)$$
$$h_3(\mathbf{x}) = \sum_{j=1}^{3}(x_i - 7)^2 - \sum_{j=1}^{3} \cos\left(\frac{x_i - 7}{2.75}\right)$$

the global minimum is at $\mathbf{x}^* = [7, 7, 7]$ with $f(\mathbf{x}^*) = h_3(\mathbf{x}^*) = 0$, and the first two components both have minimum values of 5. For both problems the search domain is $[-15, 15]$ in all dimensions.

For both the two and three dimensional tests we execute 50 algorithm macro-replications; for the two dimensional test we allow 50 samples per replication, with 10 initializing samples, and 80 samples per replication with 15 initializing samples for the three dimensional test. Note, for each replication, initializing designs are identical for

minBO and BO to remove initialization influence from the algorithm behavior. Table I shows the final results for the two and three dimensional problems. Figures 3 and 4 show the average performance per sample with respect to best observed function value. We see that minBO outperforms BO over both problems by a statistically significant margin, and minBO shows progress towards the global minimum location, while the progression of BO is unclear.

TABLE I

THEORETIC OPTIMIZATION PROBLEM RESULTS AFTER 50 SAMPLES, WITH 95% CONFIDENCE INTERVALS PRODUCED OVER 50 MACRO-REPLICATIONS. BOLD ENTRIES INDICATE SIGNIFICANTLY SUPERIOR PERFORMANCE AT $\alpha = 0.05$.

Dimension	Algorithm	$\hat{f} - f^*$	$\|\|\hat{\mathbf{x}} - \mathbf{x}^*\|\|$
2	minBO	$\mathbf{0.4085 \pm 0.3567}$	$\mathbf{1.491 \pm 1.519}$
	BO	6.338 ± 0.7705	8.302 ± 2.672
3	minBO	$\mathbf{3.680 \pm 0.5339}$	$\mathbf{13.318 \pm 3.434}$
	BO	6.672 ± 0.7884	23.828 ± 2.973

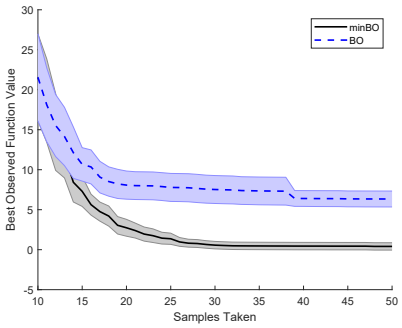


Fig. 3. Average best observed function value per sample over 2 dimensional problem, with 95% confidence interval. Average and CI taken over 50 replications.

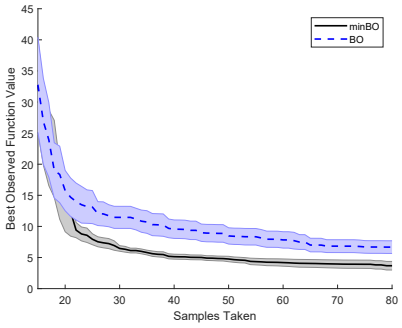


Fig. 4. Average best observed function value per sample over 3 dimensional problem, with 95% confidence interval. Average and CI taken over 50 replications.

B. Application to Benchmark Problems

In this section, we present the experimental results when minBO is applied for falsification of conjunctive requirements. We implement minBO as the optimization engine within the S-TaLiRo falsification tool [17], and make use of two industry benchmark models: the automatic transmission (AT) model from [2], and the ground collision avoidance system (GCAS) autopilot model for the F-16 fighter jet [26]. For each experiment, while attempting to falsify the requirements outlined below, we allow 300 calls to the AT or GCAS model and complete 50 macro-replications. Again, we compare minBO to BO and ensure identical starting designs, for each replication, for the two algorithms.

For both models, we have conjunctive requirements with 3 components, $\varphi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3$. The AT model uses requirement components:

$$\varphi_1 = \square_{[0,30]}(\text{rpm} < 3000) \rightarrow \square_{[0,4]}(\text{speed} < 35)$$
$$\varphi_2 = \square_{[0,30]}(\text{rpm} < 3000) \rightarrow \square_{[0,8]}(\text{speed} < 50)$$
$$\varphi_3 = \square_{[0,30]}(\text{rpm} < 3000) \rightarrow \square_{[0,20]}(\text{speed} < 65).$$

These specifications force the system to stay below a specified speed, for a specified time interval, whenever the rpm's are below the 3000 rpm threshold. The GCAS model uses the requirement components:

$$\varphi_1 = \square_{[0,15]}\ \text{alt} > 0$$
$$\varphi_2 = \square_{[0,15]}\left((\text{ap}_1) \wedge \text{X}\ (\text{ap}_0)\right) \rightarrow \text{X}\ \ (p_1 \wedge p_2)$$
$$\varphi_3 = \Diamond_{[0,15]}((\text{ap}_1) \wedge \text{X}\ (\text{ap}_0))$$

where φ_1 requires that the jet does not crash, and φ_3 specifies that the autopilot system eventually turns off, and φ_2 requires that the jet is in a stable roll and pitch position when the autopilot is turned off, specifically $p_1 \equiv (\text{roll} \in [0.02, 0.04])$ and $p_2 \equiv (\text{pitch} \in [0.28, 0.28])$.

Final results over the two models are reported in Table II. We report [2]: (1) falsification rate (FR), i.e., the number of replications that find a falsifying input out of 50, (2) average number of samples to produce a falsification ($\bar{S}$), given a falsification was found, and (3) median number of samples to produce a falsification ($\tilde{S}$), given a falsification was found.

TABLE II

FALSIFICATION RESULTS WITH 300 MODEL EVALUATIONS ALLOWED AND 50 REPLICATIONS. 95% CONFIDENCE INTERVALS REPORTED FOR MEAN EVALUATIONS TO FIND A FALSIFICATION ($\bar{S}$). BOLD ENTRIES INDICATE SIGNIFICANTLY SUPERIOR PERFORMANCE AT $\alpha = 0.05$.

Model	Algorithm	FR	$\bar{S}$	$\tilde{S}$
AT	minBO	48/50	68.6875 ± 7.2235	61
	BO	49/50	65.8163 ± 6.3645	57
GCAS	minBO	$\mathbf{50/50}$	$\mathbf{16.3200 \pm 0.1227}$	**16**
	BO	38/50	24.4737 ± 1.6611	21.5

Table II shows that minBO significantly outperforms BO over the GCAS model; finding a falsifying input every replication quite efficiently, using only a handful of observations. On the other hand, minBO and BO perform statistically equivalently for the AT model; both algorithms failing to find a falsification on at least one replication. Since the requirement components for the AT model are so similar, we see that the associated robustness values are all on the same scale, thus the scale problem discussed in Section I is not present. However, the component robustness values for the

GCAS model fall on vastly different scales, with φ_2 values being much smaller and masking the information from φ_1. We see that minBO substantially outperforms BO in cases where the scale problem exists, and performs no worse than BO in cases where the scale problem is not present.

V. CONCLUSIONS

We propose the minimum Bayesian optimization (minBO) algorithm for efficient falsification of CPS with conjunctive requirements. In such cases, the scale problem often presents a barrier to optimization-based falsification approaches. The goal of minBO is to exploit the structure of the underlying optimization problem by exposing and leveraging the information provided by the components of the objective function, i.e., the conjunctive requirement components. Empirical results over both theoretic optimization and practical falsification application problems, show that minBO performs significantly better than standard BO when the scale problem is present. Moreover, in the application cases where the scale problem is not present we see that minBO performs just as well as standard BO. In fact, when minBO is used, the computational gains from simulation reduction far out weight any additional computation needed. Future work includes: a comprehensive benchmarking study to test minBO against other falsification approaches; fully characterizing the theoretic properties of minBO; and, studying how the minBO approach can be applied to other CPS problems such as the conjunctive synthesis problem.

ACKNOWLEDGMENT

The research in this paper has been partially supported by the grant DARPA FA8750-20-C-0507, NSF #2046588, and NSF #1829238.

REFERENCES

[1] G. Ernst, P. Arcaini, A. Donze, G. Fainekos, L. Mathesen, G. Pedrielli, S. Yaghoubi, Y. Yamagata, and Z. Zhang, "Arch-comp 2019 category report: Falsification," in *ARCH19. 6th International Workshop on Applied Verification of Continuous and Hybrid Systems*, ser. EPiC Series in Computing, G. Frehse and M. Althoff, Eds., vol. 61. EasyChair, 2019, pp. 129–140. [Online]. Available: https://easychair.org/publications/paper/5VWq

[2] G. Ernst, P. Arcaini, I. Bennani, A. Donze, G. Fainekos, G. Frehse, L. Mathesen, C. Menghi, G. Pedrinelli, M. Pouzet, *et al.*, "Arch-comp 2020 category report: Falsification," *EPiC Series in Computing*, 2020.

[3] E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Nickovic, and S. Sankaranarayanan, "Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications," in *Lectures on Runtime Verification - Introductory and Advanced Topics*, ser. LNCS. Springer, 2018, vol. 10457, pp. 128–168.

[4] H. Abbas, G. Fainekos, S. Sankaranarayanan, F. Ivančić, and A. Gupta, "Probabilistic temporal logic falsification of cyber-physical systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 12, no. 2s, p. 95, 2013.

[5] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications," in *Formal Approaches to Testing and Runtime Verification*, ser. LNCS, vol. 4262. Springer, 2006, pp. 178–192.

[6] S. Yaghoubi and G. Fainekos, "Gray-box adversarial testing for control systems with machine learning components," in *International Conference on Hybrid Systems: Computation and Control (HSSC)*, 2019.

[7] J. Kapinski, J. V. Deshmukh, X. Jin, H. Ito, and K. Butts, "Simulation-based approaches for verification of embedded control systems: An overview of traditional and advanced modeling, testing, and verification techniques," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 45–64, 2016.

[8] A. Dokhanchi, A. Zutshi, R. T. Sriniva, S. Sankaranarayanan, and G. Fainekos, "Requirements driven falsification with coverage metrics," in *12th International Conference on Embedded Software (EMSOFT)*, 2015.

[9] S. Sankaranarayanan and G. Fainekos, "Falsification of temporal properties of hybrid systems using the cross-entropy method," in *ACM International Conference on Hybrid Systems: Computation and Control*, 2012.

[10] J. Deshmukh, X. Jin, J. Kapinski, and O. Maler, "Stochastic local search for falsification of hybrid systems," in *International Symposium on Automated Technology for Verification and Analysis*. Springer, 2015, pp. 500–517.

[11] Z. Zhang, G. Ernst, S. Sedwards, P. Arcaini, and I. Hasuo, "Two-layered falsification of hybrid systems guided by monte carlo tree search," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2894–2905, 2018.

[12] T. Akazaki, S. Liu, Y. Yamagata, Y. Duan, and J. Hao, "Falsification of cyber-physical systems using deep reinforcement learning," in *Formal Methods (FM)*, ser. LNCS, vol. 10951, 2018, pp. 456–465.

[13] L. Mathesen, G. Pedrielli, S. H. Ng, and Z. B. Zabinsky, "Stochastic optimization with adaptive restart: A framework for integrated local and global learning," *Journal of Global Optimization*, vol. 79, no. 1, pp. 87–110, 2021.

[14] L. Mathesen, S. Yaghoubi, G. Pedrielli, and G. Fainekos, "Falsification of cyber-physical systems with robustness uncertainty quantification through stochastic optimization with adaptive restart," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 991–997.

[15] S. Sankaranarayanan, S. A. Kumar, F. Cameron, B. W. Bequette, G. Fainekos, and D. M. Maahs, "Model-based falsification of an artificial pancreas control system," *ACM SIGBED Review*, vol. 14, no. 2, pp. 24–33, 2017.

[16] Z. Zhang, I. Hasuo, and P. Arcaini, "Multi-armed bandits for boolean connectives in hybrid system falsification," in *International Conference on Computer Aided Verification*. Springer, 2019, pp. 401–420.

[17] Y. Annpureddy, C. Liu, G. E. Fainekos, and S. Sankaranarayanan, "S-taliro: A tool for temporal logic falsification for hybrid systems," in *TACAS*, 2011.

[18] A. Dokhanchi, S. Yaghoubi, B. Hoxha, and G. Fainekos, "Vacuity aware falsification for mtl request-response specifications," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*. IEEE, 2017, pp. 1332–1337.

[19] T. Ferrère, D. Nickovic, A. Donzé, H. Ito, and J. Kapinski, "Interface-aware signal temporal logic," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019, pp. 57–66.

[20] S. Sato, M. Waga, and I. Hasuo, "Constrained optimization for falsification and conjunctive synthesis," *arXiv preprint arXiv:2012.00319*, 2020.

[21] P. I. Frazier, "Bayesian optimization," in *Recent Advances in Optimization and Modeling of Contemporary Problems*. INFORMS, 2018, pp. 255–278.

[22] J. Deshmukh, M. Horvat, X. Jin, R. Majumdar, and V. S. Prabhu, "Testing cyber-physical systems through bayesian optimization," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, pp. 1–18, 2017.

[23] S. Ghosh, F. Berkenkamp, G. Ranade, S. Qadeer, and A. Kapoor, "Verifying controllers against adversarial examples with bayesian optimization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7306–7313.

[24] M. Waga, "Falsification of cyber-physical systems with robustness-guided black-box checking," in *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, 2020, pp. 1–13.

[25] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments*. Springer Science & Business Media, 2013.

[26] P. Heidlauf, A. Collins, M. Bolender, and S. Bak, "Verification challenges in f-16 ground collision avoidance and other automated maneuvers." in *ARCH@ ADHS*, 2018, pp. 208–217.