Proceedings of the ASME 2021 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2021
August 17-20, 2021, Online, Virtual

DRAFT: IDETC2021-70425

EVALUATING HEURISTICS IN ENGINEERING DESIGN: A REINFORCEMENT LEARNING APPROACH

Karim A. ElSayed
Graduate Research Assistant
School of Mechanical Engineering
Purdue University
West Lafayette, IN

Ilias Bilionis
Associate Professor
School of Mechanical Engineering
Purdue University
West Lafayette, IN

Jitesh H. Panchal
Associate Professor
School of Mechanical Engineering
Purdue University
West Lafayette, IN

ABSTRACT

Heuristics are essential for addressing the complexities of engineering design processes. The goodness of heuristics is context-dependent. Appropriately tailored heuristics can enable designers to find good solutions efficiently, and inappropriate heuristics can result in cognitive biases and inferior design outcomes. While there have been several efforts at understanding which heuristics are used by designers, there is a lack of normative understanding about when different heuristics are suitable. Towards addressing this gap, this paper presents a reinforcement learning-based approach to evaluate the goodness of heuristics for three sub-problems commonly faced by designers while carrying out design under resource constraints: (i) learning the mapping between the design space and the performance space, (ii) sequential information acquisition in design, and (iii) decision to stop information acquisition. Using a multi-armed bandit formulation and simulation studies, we learn the heuristics that are suitable for these sub-problems under different resource constraints and problem complexities. The results of our simulation study indicate that the proposed reinforcement learning-based approach can be effective for determining the quality of heuristics for different sub-problems, and how the effectiveness of the heuristics changes as a function of the designer's preference (e.g., performance versus cost), the complexity of the problem, and the resources available.

Keywords: Heuristics, Decision Making, Reinforcement Learning, Multi-armed Bandit

1 Introduction

Heuristics are context-dependent directives, based on intuition, tacit knowledge, or experiential understanding, which provide design process direction to increase the chance of reaching satisfactory (but not necessarily optimal) solutions [1]. Although they allow us to navigate the design process, sometimes, they are fallible depending on the context and circumstances [1]. Heuristics work well in environments for which they have evolved, but fail miserably in other situations [2]. They can result in cognitive biases. For example, the use of catalogs for framing the design space can result in the cognitive bias of "fixation" [3].

While there is now an improved understanding of which heuristics people use in engineering design and systems engineering, there is a lack of normative understanding of how good the heuristics are in the presence of finite (limited) resources, and how well heuristics transfer from one design problem to another. To address the research gap, the objective of the paper is to analyze the quality of design heuristics under different resource constraints and problem complexities.

To achieve this objective, we present an approach which consists of (i) abstracting the design problems into a set of idealized sub-problems, (ii) representing the design process as a set of process heuristics to solve the sub-problems, and (iii) using reinforcement learning (RL) to learn optimal heuristics through sequential solution of design problems sampled from the problem space. Specifically, we abstract the design problem as a sequential information acquisition and decision making process, which consists of the following three sub-problems (see Figure 1): (1) decision to choose function learning model for learning the mapping between the design space and the performance space, (2) information acquisition decision to choose next design point for evaluation, and (3) deciding whether to stop the information acquisition process. We choose a class of parametric design problems where the designer's goal is to select a set of design parameters that maximize performance. The designer is assumed to possess a set of simulation models/experimental apparatus, and a fixed budget to run the experiments.

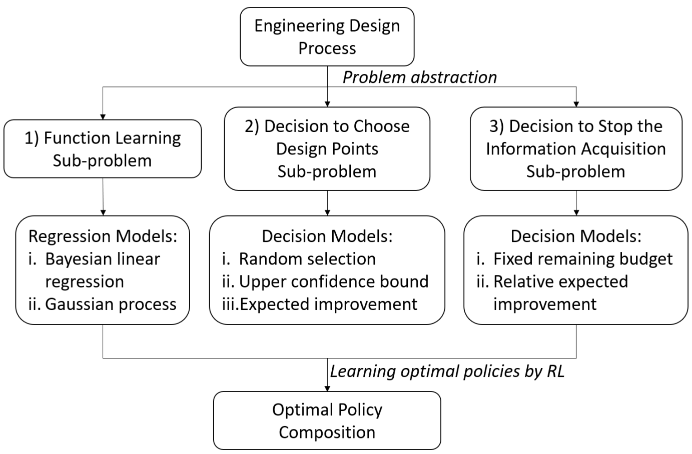


FIGURE 1: An overview of the research approach.

Our results indicate that reinforcement learning can be used as an effective approach to determine the quality of heuristics for different sub-problems, and how the effectiveness of the heuristics changes as a function of the designer's preference (e.g., performance versus cost), the complexity of the problem, and the resources available.

The paper is organized as follows. We present a review of literature on heuristics in engineering design, cognitive science, and artificial intelligence in Section 2. Section 3 provides an overview of the approach followed in this paper. Section 4 presents the formulation of the reinforcement learning problem, and Section 5 presents the results. Finally, closing comments are presented in Section 6.

2 Literature Review

Heuristics have been studied in several different domains. Within engineering design, design heuristics have been investigated from the perspective of creating novel design solutions while managing the complexity of the design process. Researchers in cognitive psychology have studied heuristics as deviations from rationality, and the resulting cognitive biases. Within computer science and artificial intelligence communities, heuristics have been studied with the aim of developing efficient algorithms to solve hard computational problems. In this section, we provide a brief review of the literature in these three domains.

2.1 Heuristics in Engineering Design

There has been substantial research on heuristics in the engineering design literature. Based on a thorough evaluation of the definitions of heuristics, Fu et al. [1] define heuristics as "context-dependent directives, based on intuition, tacit knowledge, or experiential understanding, which provides design process direction to increase the chance of reaching a satisfactory but not necessarily optimal solution".

Heuristics are used throughout the design process, including creation of new design concepts during the conceptual stage, decision making in the later stages of design. Through a protocol study, Yilmaz et al. [4] found that heuristics are frequently used by engineers in different domains during conceptual design. Yilmaz and Seifart [5] show how the use of heuristics by expert designers in the early stages of design leads to novel and creative solutions. In addition to early stage product design, heuristics play an important role in early stage systems design. By interviewing ten experts at the Jet Propulsion Laboratory (JPL), Fillingim et al. [6] identified 101 heuristics used in space mission design. In the later stages of design, such as in design optimization, heuristics are used to reduce the complexity of finding optimal designs. Deshmukh et al. [3] show that while these heuristics are helpful in managing design optimization tasks, they can also lead to unnecessary constraints and cognitive biases.

2.2 Heuristics in Cognitive Psychology

Heuristics have been studied extensively in psychology as an alternative to rational decision making. Tversky and Kahneman [7] classify decision making heuristics into representativeness, availability, adjustment and anchoring. Heuristics are considered efficient cognitive processes, conscious or unconscious, that ignore part of the information [8]. Heuristics help in making decisions more quickly, frugally, and/or accurately than more complex methods.

In contrast to the view that heuristics are procedural simplifications from rational models, recent work in cognitive science suggests that the use of heuristics is in fact rational if we account for the cognitive constraints [9, 10]. Lewis and co-authors [11] have coined the phrase "computational rationality" to emphasize that theories of rational behavior should account for the computational and cognitive constraints, and the cost of cognitive effort [12].

2.3 Heuristics in the Computer Science and Artificial Intelligence

Within the artificial intelligence (AI) and computer science (CS) literature, heuristics are used to solve complex computational problems such as the traveling salesman problem. Pearl defines heuristics are defined as "criteria, methods, or principles for deciding which among several alternate courses of action promises to be the most effective in order to achieve some goal" [13]. Several efforts have been devoted to developing good heuristics for specific computational problems.

Within AI, heuristics are used for designing computational agents that mimic human behavior. The use of heuristics is particularly prominent in the reinforcement learning literature, where agents use heuristic policies for learning, such as pure exploration, pure exploitation, excitation policies, epsilon-greedy exploration, Boltzmann exploration, Upper confidence bounding, and Thompson sampling [14].

As a summary, the usefulness of heuristics is well recognized in each of these fields. It is also recognized that different heuristics have different effectiveness under different conditions, such as different resource constraints. However, there is a lack of normative understanding of heuristics, which can guide designers to choose the right heuristics considering the nature of their own problem. Therefore, there is a need for approaches for determining the appropriateness of heuristics for the problems commonly encountered in engineering design. To address this need, we present a reinforcement learning based approach to determine the heuristics that a resourced constrained rational agent should use. The approach is discussed in the following section.

3 Approach

To focus our attention on an specific class of problems encountered in engineering design, we start with abstracting the engineering design process as a sequential decision making process [15]. The designer's goal is to find the design with the best performance by searching the design space while sequentially acquiring information about the performance of different design alternatives [16, 17]. We identify three sub-problems within the design process, and present different heuristics to solve the problems. For each sub-problem, we implement a reinforcement learning model to understand the effectiveness of heuristics under resource constraints, and how their goodness changes with changes in problem characteristics.

3.1 Engineering Design as a Sequential Decision Making Process

We consider a scenario where the designer chooses design parameters x that map to outcome y with constraints $g(x) \leq 0$. The designer does not explicitly know the mathematical relationship between x and y, but can evaluate the performance at specified design points using costly computational or physical experiments [16]. Many design tasks follow this scenario. For example, in additive manufacturing (AM), optimizing components' mechanical properties (outcome) requires an iterative selection of the AM process parameters [18]. Thus, in order to maximize the outcome, the designer sequentially evaluates the design performance (see Figure 2). At each step, the designer decides on a model, or heuristic, for (1) learning the mapping between the the design space and the performance, (2) choosing the next point in the design space to evaluate performance by running a costly experiment, and (3) determining whether to stop information acquisition (experimentation).

In this sequential decision making process, the designer has an initial state of knowledge $\mathscr{H}_0$ based on a set of observations $\mathscr{D} = [\mathbf{x}_n, \mathbf{y}_n]$, along with any prior beliefs. At each iteration, the designer first chooses a function learning model to predict how x maps to y. This is followed by selecting a heuristic to choose the next x_i, and evaluate its performance y_i. Lastly, the designer chooses whether to stop based on their updated state of knowledge $\mathscr{H}_i$ and constraints $g(x)$, which may be limited budget B. The process is illustrated in Figure 2.

With the aim of identifying the best heuristics for the design process, we partition the process into three sub-problems:

1. Decision to choose function learning model
2. Decision to choose next design point
3. Decision to choose when to stop the information acquisition process

We recognize that these three sub-problems are interdependent, and the optimal heuristics depend on each other. As a starting point, in this paper, we identify optimal heuristics for each of these decisions independently.

3.2 Learning Optimal Heuristics for Engineering Design

We use RL to identify the best heuristics for each sub-problem. RL approaches are advantageous and attractive when the objective is to maximize the total benefit over time, and there are no examples of desired actions given situations, but it is possible to reward the actions taken according to a performance criterion [19, 20]. Additionally, RL has been shown to hold promise in solving complex problems that require cost-sensitive decision making [21, 22, 23].

Specifically, we formulate the problem of learning optimal heuristics for each of the idealized sub-problems as a multi-armed bandit (MAB) problem. For the first sub-problem, we evaluate the performance of different function learning models based on various designer preferences, such as increasing prediction accuracy or reducing cost. Similarly, we analyze how different heuristics perform in the information acquisition sub-problem by measuring the improvement in performance from acquiring information about one additional point in the design space. Lastly, we assess the quality of different decisions to stop the information acquisition process under resource constraints, such as a fixed budget. For all three sub-problems, we also show how the optimal heuristics change with changes in problem complexity, designer's preferences, and the amount of resources & information available.

3.3 Design of the Simulation Experiment

We perform simulation experiments using a RL-based model. For the simulations, we make the following simplifying assumptions.

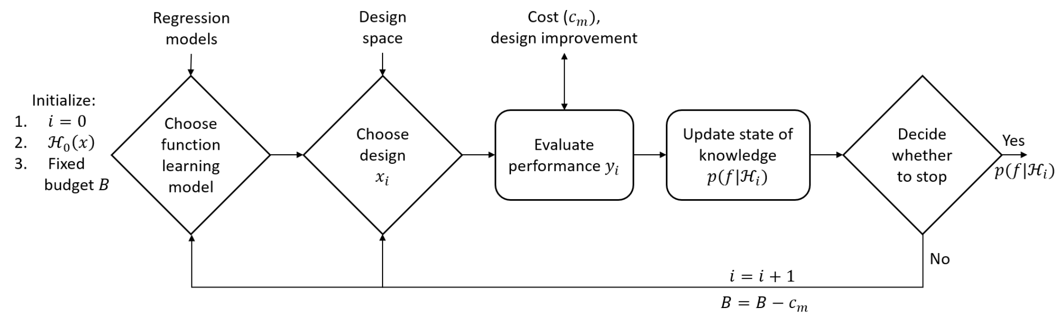


FIGURE 2: Engineering design as a sequential decision making process.

Assumption 1 (Continuous design space). The design performance $f(x)$ is a one-dimensional continuous function of a single design parameter $x \in \mathscr{X}$, where $\mathscr{X} = [-10, 10]$.

Assumption 2 (Objective functions as Gaussian processes). We represent the mapping between the design space and the performance space using a class of non-linear functions, called Gaussian processes (GPs):

$$f \sim GP(\mu = 0, k), \tag{1}$$

where $\mu = 0$ is the mean and k is the covariance kernel. We select the radial basis function (RBF) kernel, which is a squared exponential covariance function such that

$$k(x, x') = \sigma^2 \exp\left\{-\frac{1}{2}\frac{(x-x')^2}{\ell^2}\right\}, \tag{2}$$

with lengthscale $\ell \geq 0$ and variance $\sigma^2 \geq 0$. This allows us to easily adjust the complexity of the problem by changing ℓ and σ^2. Large ℓ values lead to smooth functions, whereas small values result in a more complex function. Generating new design problems is equivalent to sampling functions from the Gaussian process. The sampled function is not known to the reinforcement learning agent, but is used to evaluate the performance at the chosen design points.

Assumption 3 (Variations of objective function complexities). In this study, we examine the transferability of optimal heuristics for one objective function to another with different complexity. We increase the objective function's complexity by reducing ℓ and changing σ^2. The three different complexities used in the simulation study are: (i) low complexity [$\ell = 5$, $\sigma^2 = 4$], (ii) medium complexity [$\ell = 3$, $\sigma^2 = 3$], and (iii) high complexity [$\ell = 1.5$, $\sigma^2 = 5$]. A sample of these three function complexities is shown in Figure 3.

Assumption 4 (No noise in the output, y_i). We assume that there is no measurement error in design evaluations, such that $y_i = f(x_i) + \varepsilon_m$ and $\varepsilon_m = 0$.

Assumption 5 (Single information source). Typically in a sequential decision making process, design evaluations y_i are performed using multiple information sources with different uncertainties and costs. Here, we use a single information source with fixed cost c_m.

These assumptions limit the scope of applicability of the results from the experimental study to the specific scenario. However, the overall approach described in this paper and decision models in the next section are general and can be applied in the future to more complex design problems, with higher dimentionality, more information sources, and added noise.

4 Formulation of the Reinforcement Learning Models for the Sub-Problems

In this section, we describe the alternative heuristics that can be used for making decisions in the three sub-problems (Sections 4.1 through 4.3), and the formulation of the reinforcement learning problem (Section 4.4). The heuristics used in the RL framework are inspired from engineering design, cognitive psychology, and human behavior literature. For the first sub-problem, we use rule-based and similarity-based heuristics in order to learn the mapping between design parameters and performance [24, 25]. For choosing the next design point and deciding whether to stop, we utilize two main classes of heuristics: simple heuristics and heuristics close to the expected utility (EU) theory. Simple heuristic models include random selection (RS), upper confidence bound (UCB) [26], and fixed remaining budget (FRB) [17], in which the latter two use cues from the environment such as predictive mean, variance, and remaining budget. On the other hand, models closer to the EU theory portray rational decisions and are based on expected improvement (EI) [27]. A summary of the heuristics used for each sub problem is listed in Table 1.

4.1 Sub-Problem 1: Learning the mapping between the design space and the performance space

We utilize two main approaches for learning the space of problems in this study: rule-based and similarity-based models. Both approaches have been used frequently in modeling how people learn about functional relationships between continuous variables (i.e., mapping inputs onto outputs) [28, 29, 30, 24, 25,

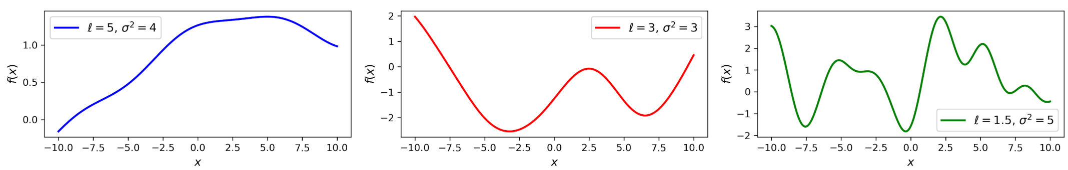


FIGURE 3: Samples of the three objective function complexities used in this study.

31].

Rule-based models are one of the earliest models used for human function learning [28]. They assume that function learning is based on learning explicitly represented functions, for example linear, polynomial, or exponential functions. In engineering design, using such representations allows for the extrapolation beyond the observed values, covering the continuous design space. Conversely, similarity-based approaches assume that function learning occurs by forming associations between observed input-output pairs and extend it based on the similarity of new inputs to old ones [25]. For example, if observed input x_i has output y_i, then inputs similar to x_i should have outputs similar to y_i. One common similarity-based approach is using GPs for functional learning [24, 25].

4.1.1 Bayesian Linear Regression models. We utilize Bayesian linear regression (BLR) to map x_i to y_i, where $x_i \in \mathscr{X}$ are the inputs and $y_i \in \mathbf{y}$ are the outputs [32, Chapter 3]. First, we consider a simple example of learning a function f from a set of observations $\mathscr{D} = [\mathbf{x_n}, \mathbf{y_n}]$. In a one-dimensional case, a linear regression model assumes that outputs, y_i are a linear function of the input x_i with additive Gaussian noise $\varepsilon \sim \mathscr{N}(0, \sigma^2)$ on each observation such that

$$y_i = f(x_i) + \varepsilon. \tag{3}$$

This can be written as a linear combination of weights, $\mathbf{w}$, and basis functions, $\phi(x_i)$:

$$y_i = \mathbf{w}^\mathbf{T} \phi(x_i) + \varepsilon. \tag{4}$$

These basis functions enable learning of non-linear problems by transforming the input space into feature space. However, this makes an assumption about the class of function being learned, thus leading to poor functional learning if there is a mismatch between the objective function and the selected basis function [33].

This can be extended to the Bayesian framework by assuming a Gaussian prior over the weights, $p(\mathbf{w}) = \mathscr{N}(\mathbf{w}|\mathbf{m_0}, \mathbf{S_0})$, and a Gaussian likelihood such that the Gaussian posterior is:

$$p(\mathbf{w}|\mathbf{x}_{1:n}, \mathbf{y}_{1:n}, \sigma, \alpha) = \mathscr{N}(\mathbf{w}|\mathbf{m}, \mathbf{S}), \tag{5}$$

where $\mathbf{S} = \left(\sigma^{-2}\Phi^\mathbf{T}\Phi + \alpha\mathbf{I}\right)^{-1}$, $\mathbf{m} = \sigma^{-2}\mathbf{S}\Phi^\mathbf{T}\mathbf{y}_{1:n}$, and α_i are hyperparameters governing the prior variance of each unknown coefficient [32, Chapter 3].

In our study, we utilize three BLR models with distinct basis functions as different heuristic rules for functional learning: Linear, Quadratic, and Cubic basis functions. The hyperparameters of the BLR models are estimated using Automatic Relevance Determination Regression (ARD) [34].

4.1.2 Gaussian Process model. A GP is a non-parametric regression method and it is completely defined by its mean, $\mu(x)$, and covariance function, or kernel, $k(x,x')$ [35, Chapter 2]. It defines a distribution over functions $f : \mathscr{X} \to \mathbb{R}$ that maps inputs, x_i, to outputs y_i as a random draw from a Gaussian distribution:

$$f \sim GP(\mu_n, k_n), \tag{6}$$

where μ_n and k_n are the posterior mean and covariance functions after n observations, respectively.

Since GPs provide a flexible approach to prediction, we utilize a GP with a RBF kernel (see Eq. **??**) to associate which inputs x_i are likely to have similar outputs y_i.

4.2 Sub-Problem 2: Deciding what Information to Acquire – Selecting the Next Design Point

In choosing the next design point, three main strategies are used: random selection, upper confidence bound [26], and expected improvement [27]. In the latter two, a $GP(0, k_{\text{RBF}})$ is used to learn the objective function and predict the mean and variance at every $x \in \mathscr{X}$ based on the observed input-output pairs, $\mathbf{x}_{1:n}$ and $\mathbf{y}_{1:n}$. The best design to pick is the one that maximizes output of the decision model.

4.2.1 Random selection. The random selection model selects the next design point as a uniform random sample from $\mathscr{X} \setminus x_n$. This gives each x_i an equal chance of selection. RS is one of the simplest heuristics to apply in engineering problems, and it is practical to use in some engineering tasks [36].

4.2.2 Upper confidence bound. The upper confidence bound model is defined as:

$$\text{UCB}_i(x) = \mu_i(x) + \psi\sigma_i(x), \tag{7}$$

where $\psi \geq 0$ is the exploration parameter, and $\mu_i(x)$ and $\sigma_i(x)$ are the predictive mean and standard deviation, respectively [26]. Higher values of ψ emphasize exploration, while $\psi = 0$ represents a pure exploitation strategy.

4.2.3 Expected improvement. The expected improvement model computes the expectation of improvement for each $x_i \in \mathscr{X}$ relative to the currently best observed point, u_i^* [27].

$$\mathrm{EI}_i(x)=\frac{\mu_i(x)-u_i^*}{\sigma_i(x)}\Phi\left(\frac{\mu_i(x)-u_i^*}{\sigma_i(x)}\right)+\sigma_i(x)\phi\left(\frac{\mu_i(x)-u_i^*}{\sigma_i(x)}\right) \tag{8}$$

4.3 Sub-Problem 3: Stopping the Information Acquisition

The decision about whether to stop is dependant on both the designer and the resources available to them. We assume that the designer has control over saving or consuming their entire resources, such as their Budget, B. Below are the two simple-heuristic and EU-based stopping strategies used in this study.

4.3.1 Fixed remaining budget. In this stopping strategy, design evaluations are stopped when the remaining budget, $B_r = B - ic_m$, reduces to a fixed value set by the designer.

4.3.2 Relative expected improvement. In this model, design evaluations are stopped when the maximum EI value of the next design point is below a fixed threshold relative to the initial EI value. These EI values are calculated in the same manner as shown in Eq. (**??**).

4.4 Reinforcement Learning Model

In the multi-armed bandit (MAB) problem, the agent is given a choice among k different actions (or arms). After taking action a at time step t, denoted by A_t, the agent receives a corresponding reward R_t. The goal of the agent is to maximize the reward signal by choosing the best action for each situation. Generally, each action has a value, or an expected reward, associated with it. For any arbitrary action a in action space $\mathscr{A}$ with n_a actions, the expected reward is given by:

$$q_*(a) \doteq \mathbb{E}[R_t|A_t=a]. \tag{9}$$

Since the value of actions is not known with certainty, one can compute an estimate of value of an action a at time step t as $Q_t(a)$. In the MAB problem, one often assumes that $Q_t(a)$ is close to $q_*(a)$ and these estimates are used to select actions. However, we implement the *Gradient Bandit Algorithm* (see Ref. [37, Chapter 2]) in this study. Instead of estimating rewards for each action, the agent learns a numerical preference, $H_t(a)$, which reflects its belief about which actions will maximize rewards. The higher the relative preference of one action over another, the more that action is selected. The probability of taking action a at time step t, denoted by $\pi_t(a)$, is determined by a soft-max distribution as shown:

$$\Pr\{A_t=a\} \doteq \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}} \doteq \pi_t(a), \tag{10}$$

and the action preferences, $H_t(a)$, are updated as follows:

$$H_{t+1}(a)=H_t(a)+\gamma(R_t-\bar{R}_t)(\mathbb{1}_{a=A_t}-\pi_t(a)), \tag{11}$$

where $\gamma \geq 0$ is the step-size parameter, and $\bar{R}_t$ is the mean of all rewards up to time t, which provides a baseline for all rewards R_t.

The Gradient MAB algorithm enables learning of optimal strategies for different variations of each sub-problem. They represent the task as learning which heuristics to choose in order to maximize the reward signals. The reward signal is the only feedback the agent receives, thus, it is instrumental to define one that correctly reflects the main objective of the task [37, Chapeter 1].

In our study, the arms correspond to alternate heuristic rules for the sub-problems (see Table 1). For example, the arms available to the agent in sub-problem 1 are: Linear, Quadratic, Cubic, or GP models. At each time step $t \leq T$, where T is a fixed number of steps, the agent takes an action (i.e., selects a heuristic) and updates its action preferences (probabilistic map) after receiving $R_{j,t}$ corresponding to the j^{th} sub-problem during time step t. When $t = T$, the simulation, or run, terminates and the agent resets their action preference. This reinforcement learning loop is shown in Figure 4. It is important to note that a new f and set of observations $\mathscr{D}$ are generated at each t.

In order to fully define our MAB problem, we formulate the reward signal, $R_{j,t}$, for the three sub-problems such that actions taken by the agent are evaluated according to a performance criterion [20]. The two main goals for all three sub-problems are (i) having high design performance, and (ii) low the computational cost, C_t. In our study, C_t is calculated based on the computational time taken to perform each action. Below are the formulations of the three reward signals, which increase as performance improves (or error reduces, in the case of sub-problem 1) and decrease as C_t increases.

For sub-problem 1, the reward is formulated with the goal of reducing error and cost as follows:

$$R_{1,t}=-(w_e\Delta+w_cC_t), \tag{12}$$

where Δ is the mean squared error (MSE) between test points $[\mathbf{x}_{test},\mathbf{y}_{test}]$ and predicted points $[\mathbf{x}_{test},\mathbf{y}_{pred}]$, w_e is the weight for MSE, and w_c serve as the weight for computational cost C_t of function learning. Simple models for function learning, such as linear regression require a lower computational cost than more sophisticated models such as the GP model.

The reward function for sub-problem 2, pick next design point, is dependent on the design performance and cost if design evaluation, such that:

$$R_{2,t}=w_vp-w_cC_t, \tag{13}$$

where w_v is a variable weight valuing the improvement in performance, p, which is calculated as follows:

$$p(y_i)=\begin{cases}\dfrac{y_i-\max(\mathbf{y}_{1:n})}{f_{\max}-f_{\min}}, & y_i \geq \max(\mathbf{y}_{1:n})\\ 0, & \text{otherwise}\end{cases} \tag{14}$$

where $f_{\max}$ and $f_{\min}$ are the true maximum and minimum of the objective function, respectively.

The reward formulation for the third sub-problem is similar to previous one with the addition of remaining budget B_r,

$$R_{3,t}=w_vp-w_cC_t+B_r. \tag{15}$$

This adds a preference to strategies that save budget, especially

TABLE 1: Heuristic Strategies for the three sub-problems in engineering design.

Decision model	**Underlying strategy**
1. Choice of function learning model	
Linear	Bayesian linear regression with linear basis functions.
Quadratic	Bayesian linear regression with second-degree polynomials as basis functions.
Cubic	Bayesian linear regression with third-degree polynomials as basis functions.
Gaussian Process (GP)	GP with zero mean and a radial basis function (RBF) kernel.
2. Decision to choose next design	
Random Selection (RS)	Randomly select a design point without replacement.
Upper confidence bound (UCB)	Explore design space during initial iterations while exploit during later iterations.
Expected improvement (EI)	Selection probability proportional to EI value.
3. Decision to stop	
Fixed remaining budget (FRB)	Stop after a fixed amount of budget is remaining.
Relative expected improvement (REI)	Stop after expected improvement (EI) is below a fixed value relative to the EI value from first iteration.

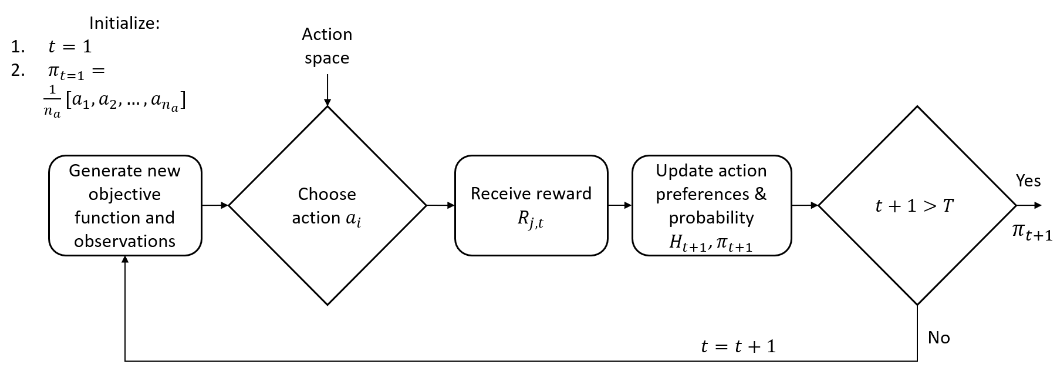


FIGURE 4: Reinforcement learning loop for the gradient multi-armed bandit algorithm.

when there is no significant improvement in performance.

5 Results from a Simulation Study

We present the modeling results of the three multi-arm bandit problems outlined in Section 3.3. For all RL simulations, we used $\gamma = 0.5$, $T = 150$, and ran the model for 1000 runs (50 samples of 20 runs). We varied the number of observation points, $n = 4:10$, the *performance-cost* ratio ($\frac{w_e}{w_c}$ for Experiment 1, and $\frac{w_v}{w_c}$ for Experiments 2 and 3), and the complexities of the objective functions (see Figure 3). The ranges for the performance-cost ratios were selected based on the reward signals, such that there is a balance between improving performance and lowering cost within the range. For testing the effect of these ratios, we used the high complexity objective function [$\ell = 1.5$, $\sigma^2 = 5$]. Additionally, we estimated the computational cost C_t for decision models with fixed number of operations by running each for 10^4 iterations to reduce the effect of variable power and memory allocation of the computer. The C_t values are shown in Table 2.

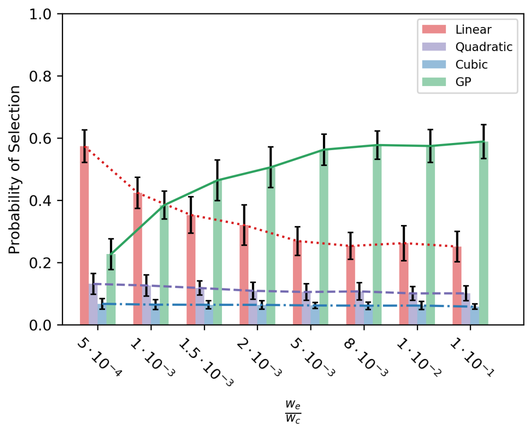
5.1 Experiment 1: Choosing Function Learning Model

At the beginning of each run, the RL agent chooses one of the four arms (function learning models) with an equal probability. Throughout the simulation, their action is evaluated using the reward function, $R_{1,t}$ (see Eq. **??**), where Δ is calculated using 10 test points, separate from the observed points. We hypothesized that more expensive models, such as GPs, outperform their less expensive counterparts (linear models), when there is an emphasis on accuracy. Comparing the models' performance results in

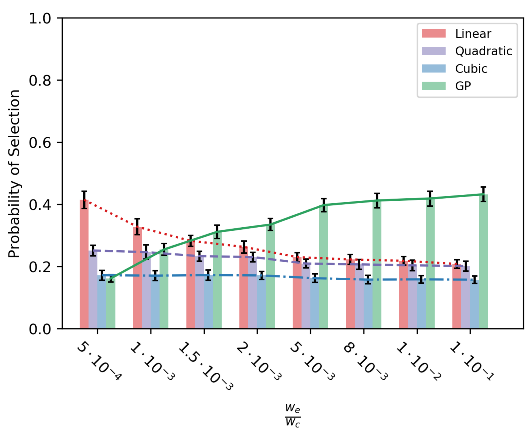
TABLE 2: C_t values for different heuristics (in milliseconds).

Observations Number (n)	Time required to run the heuristic models once						
	Linear	Quadratic	Cubic	GP	RS	UCB	EI
4	1.76	2.42	2.68	5.58	0.37	6.48	6.83
10	1.76	2.46	2.78	5.81	0.37	6.57	6.93

Figure 5, the bar plots show a similar trend in which the least expensive model had a higher probability of selection (POS) when there was a greater emphasis on cost. Conversely, the GP model, with the highest C_t, had higher POS when more weight was put on accuracy, consistent with our hypothesis.



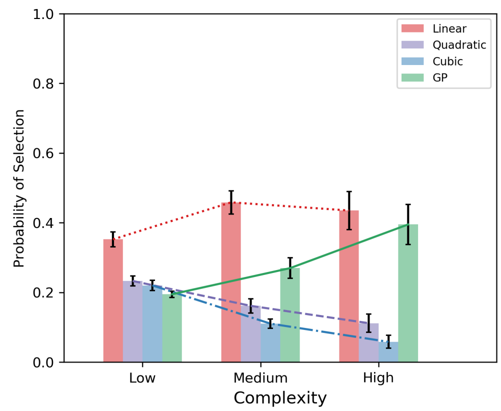
(a) $n = 4$



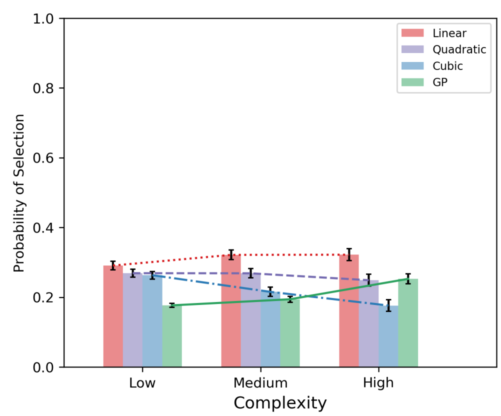
(b) $n = 10$

FIGURE 5: Performance of regression models at different $\frac{w_e}{w_c}$ values for different number of observations.

Looking at how different regression models perform at different objective function complexities for $\frac{w_e}{w_c} = 10^{-3}$, BLR models outperform the GP model when the complexity is low. Interestingly, the variance and difference in POS among all four models is significantly lower when n is increased to 10, as shown in Figures 5b and 6b. This is due to the increase of POS of both the Quadratic and Cubic models, suggesting that both polynomial models were underfitting when given 4 observation points.



(a) $n = 4$

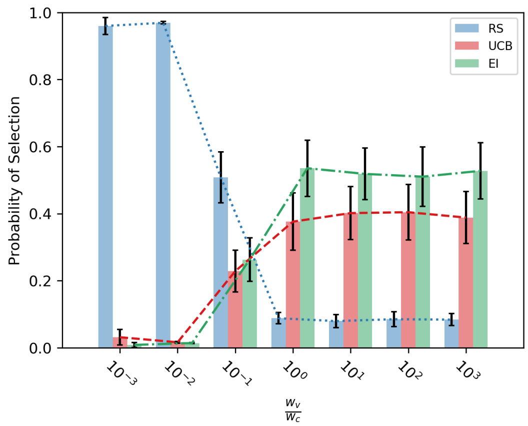


(b) $n = 10$

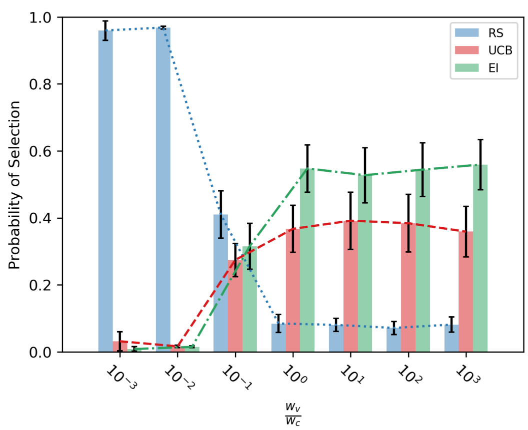
FIGURE 6: Performance of regression models for different objective function complexities and number of observations.

5.2 Experiment 2: Choosing Next Design Point

Experiment 2 used a GP to learn the objective functions for both the UCB and EI decision models, increasing their C_t by an order of magnitude from that of the RS model. Being correlated to reward functions, the POS for the RS model was consistently above 90% at low $\frac{w_v}{w_c}$ values, while dropping below 10% at higher values, as shown in Figure 7. Increasing the number of observed points did not affect this trend, however. Consistent with results from Experiment 1, the model with the highest C_t, which is the EI model here, performed best at high $\frac{w_v}{w_c}$ values. This corroborates our hypothesis that EU-based models perform well when the designer prioritizes quality over cost of design evaluations.



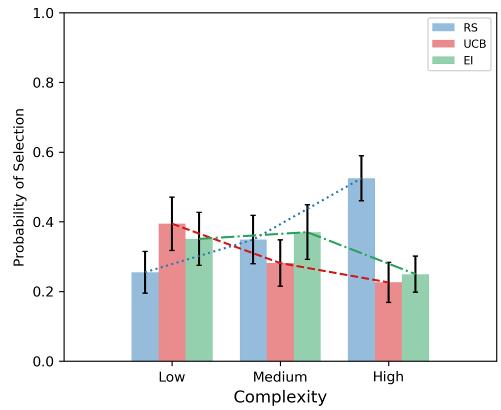
(a) $n = 4$



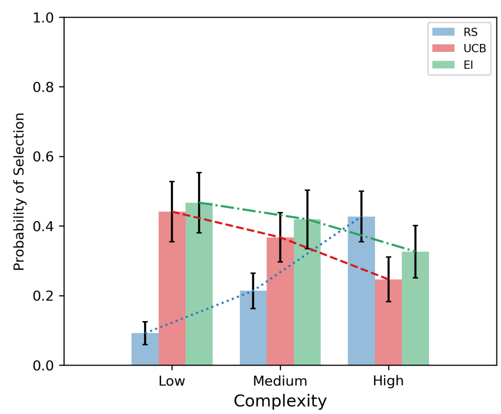
(b) $n = 10$

FIGURE 7: Performance of decision to choose next x models at different $\frac{w_v}{w_c}$ values for different number of observations.

To better understand the effect of complexity in design evaluations, we fixed the performance-cost ratio and varied the objective functions complexities. The $\frac{w_v}{w_c}$ ratio was fixed at 10^{-1} because the POS for all three models were closest for this ratio. Remarkably, the RS model underperformed similarly performing UCB and EI models at low complexities, specifically at $n = 10$ (see Figure 8). However, at higher complexities and low number of observations, the RS model outperforms, suggesting that a random search can be an effective strategy when little information is available and low cost is desired.



(a) $n = 4$



(b) $n = 10$

FIGURE 8: Performance of decision to choose next x models for different objective function complexities and number of observations.

5.3 Experiment 3: Choosing Whether to Stop

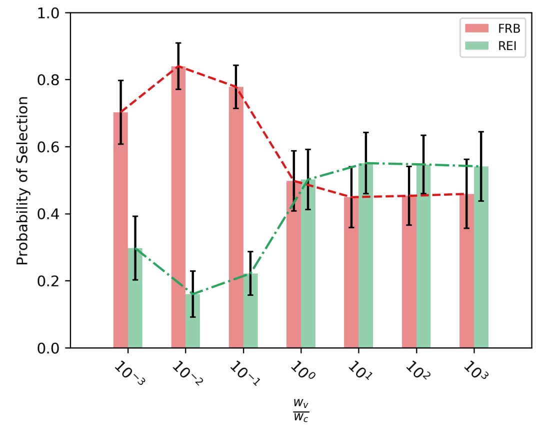
For each time step in a run, the agent decides whether to stop after picking a new design point and is rewarded according to $R_{3,t}$ (Eq. **??**). The decision to pick the next x_i was fixed to using EI, while the budget was set to $B = 1$, and the cost of design evaluation was $c = 0.1$. Some model parameters for both stopping strategies were also fixed in this experiment. Specifically, B_r was fixed to 0.5 (i.e., the FRB model evaluates five additional design points and saves half the budget), and the threshold for the REI model was set to 0.1 (i.e., the model stops design evaluations when the maximum EI value of next design point is below 10% of that of the first design evaluation).

Looking first at the results for $n = 4$ (Figure 9a), the FRB model had a higher POS than the REI one when reducing the computational cost was highly weighed (low $\frac{w_v}{w_c}$ values). On the other hand, both models had similar POS when design performance was emphasized. Since both models used EI for design evaluations, it can be assumed that the C_t values are comparable for same number of evaluations. This suggests that the average design evaluation of the REI model were greater than five, giving an edge to the FRB model at low $\frac{w_v}{w_c}$ values and a slight edge to the REI at high $\frac{w_v}{w_c}$ values.

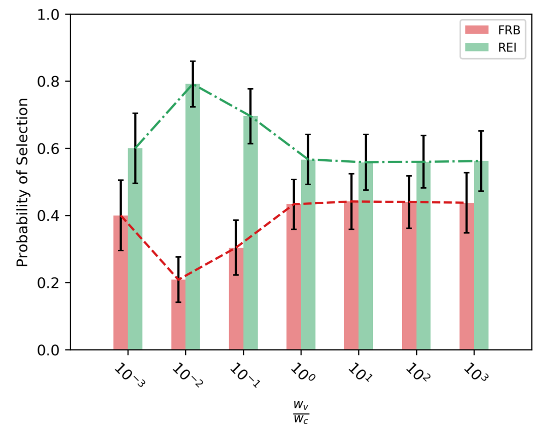
For higher number of observation points, the POS of the REI model was higher for all $\frac{w_v}{w_c}$ values, specifically for low $\frac{w_v}{w_c}$ values, as shown in Figure 9b. This shows that, on average, the REI model stopped the design process earlier than the FRB model, saving more of the budget. Also, the additional design evaluations of the FRB did not significantly improve its POS at higher $\frac{w_v}{w_c}$ values, indicating that the value of the best design point of each model was close to, or at, $f_{\max}$. Thus, the results from Experiment 3 closely align with the results from Experiment 2. They suggest that during the early stages of the design process (low n), a random search, or other simple-heuristic strategy, performs well, especially when total cost is to be minimized. During the later stages of the process (high n), EU-based decisions may lead to higher improvement in performance.

6 Conclusion

In this paper, we present a reinforcement learning-based approach for analyzing heuristics in engineering design processes. Our results indicate that the proposed framework can determine the effectiveness of different heuristics under different resource constraints and preferences set by the designer. For example, choosing a similarity-based approach for function learning, such as a Gaussian process with a RBF kernel, showed to be best when seeking high accuracy and little information is available. This choice also eliminates the need to know the class of objective functions, which is essential for Bayesian linear regression models [33]. During the new design evaluation sub-problem, we observed that a random selection model was most effective when (1) little information is provided, (2) the design problem was more complex, and (3) reducing cost was emphasized. However, for low complexity tasks or when seeking higher improvement in design, both simple-heuristic and EU-based models that utilize prior information are preferred. Again, results from Experiment 3 showed that the expensive EU-based decision was preferred over its simple-heuristic counterpart when high performance is needed. The results suggest that a random search is an effective design evaluation strategy early on in complex engineering design tasks. However, when more information is available and during the later stages of design process, utilizing expensive EU-based models may lead to higher performance improvement.



(a) $n = 4$



(b) $n = 10$

FIGURE 9: Performance of decision to stop models at different $\frac{w_v}{w_c}$ values for different number of observations.

Although the results are specific to the preferences and constraints we set in defining the sub-problems, designer researchers can utilize this framework to understand how their own preferences and resource constraints affect the goodness of different heuristics, and map the two together. This framework could also be extended in future studies using higher dimensionality design tasks. Additionally, some of the simplifying assumptions made in the RL models used in this paper can be relaxed in the future to account for the nuances of real engineering design processes.

Acknowledgements

The authors gratefully acknowledge financial support from the National Science Foundation through NSF CMMI Grant No. 1662230. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] Fu, K. K., Yang, M. C., and Wood, K. L., 2016. "Design principles: Literature review, analysis, and future directions". *Journal of Mechanical Design,* ***138***(10), Aug.

[2] Lieder, F., Chen, O. X., Krueger, P. M., and Griffiths, T. L., 2019. "Cognitive prostheses for goal achievement". *Nature Human Behaviour,* ***3***(10), Aug., pp. 1096–1106.

[3] Deshmukh, A. P., Thurston, D. L., and Allison, J. T., 2016. "Heuristics for formulating design optimization models: Their uses and pitfalls". In 5th Internations Engineering Systems Symposium, CESUN 2016.

[4] Yilmaz, S., Daly, S. R., Seifert, C. M., and Gonzalez, R., 2015. "How do designers generate new ideas? design heuristics across two disciplines". *Design Science,* ***1***, Nov.

[5] Yilmaz, S., and Seifert, C. M., 2011. "Creativity through design heuristics: A case study of expert product design". *Design Studies,* ***32***(4), July, pp. 384–415.

[6] Fillingim, K. B., Nwaeri, R. O., Borja, F., Fu, K., and Paredis, C. J. J., 2019. "Design heuristics: Extraction and classification methods with jet propulsion laboratory's architecture team". *Journal of Mechanical Design*, June, p. 1.

[7] Tversky, A., and Kahneman, D., 1974. "Judgment under uncertainty: Heuristics and biases". *Science,* ***185***(4157), Sept., pp. 1124–1131.

[8] Gigerenzer, G., and Gaissmaier, W., 2011. "Heuristic decision making". *Annual Review of Psychology,* ***62***(1), Jan., pp. 451–482.

[9] Griffiths, T. L., Lieder, F., and Goodman, N. D., 2015. "Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic". *Topics in Cognitive Science,* ***7***(2), Apr., pp. 217–229.

[10] Lieder, F., and Griffiths, T. L., 2019. "Resource-rational analysis: understanding human cognition as the optimal use of limited computational resources". *Behavioral and Brain Sciences*, p. 1–85.

[11] Lewis, R. L., Howes, A., and Singh, S., 2014. "Computational rationality: Linking mechanism and behavior through bounded utility maximization". *Topics in Cognitive Science,* ***6***(2), Mar., pp. 279–311.

[12] Westbrook, A., and Braver, T. S., 2015. "Cognitive effort: A neuroeconomic approach". *Cognitive, Affective, & Behavioral Neuroscience,* ***15***(2), Feb., pp. 395–415.

[13] Pearl, J., 1984. *Heuristics: Intelligent search strategies for computer problem solving*.

[14] Powell, W. B., and Ryzhov, I., 2018. *Optimal learning, 2nd Edition*. Wiley-Interscience, Hoboken, N.J.

[15] Miller, S. W., Yukish, M. A., and Simpson, T. W., 2017. "Design as a sequential decision process". *Structural and Multidisciplinary Optimization,* ***57***(1), July, pp. 305–324.

[16] Shergadwala, M., Bilionis, I., Kannan, K. N., and Panchal, J. H., 2018. "Quantifying the impact of domain knowledge and problem framing on sequential decisions in engineering design". *Journal of Mechanical Design,* ***140***(10).

[17] Chaudhari, A. M., Bilionis, I., and Panchal, J. H., 2020. "Descriptive models of sequential decisions in engineering design: An experimental study". *Journal of Mechanical Design,* ***142***(8).

[18] Shamsaei, N., Yadollahi, A., Bian, L., and Thompson, S. M., 2015. "An overview of direct laser deposition for additive manufacturing; part ii: Mechanical behavior, process parameter optimization and control". *Additive Manufacturing,* ***8***, pp. 12–35.

[19] Pednault, E., Abe, N., and Zadrozny, B., 2002. "Sequential cost-sensitive decision making with reinforcement learning". In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 259–268.

[20] Barto, A. G., and Dietterich, T. G., 2004. "Reinforcement learning and its relationship to supervised learning". *Handbook of learning and approximate dynamic programming,* ***10***, p. 9780470544785.

[21] Reverdy, P. B., Srivastava, V., and Leonard, N. E., 2014. "Modeling human decision making in generalized gaussian multiarmed bandits". *Proceedings of the IEEE,* ***102***(4), pp. 544–571.

[22] Wu, C. M., Schulz, E., Speekenbrink, M., Nelson, J. D., and Meder, B., 2018. "Generalization guides human exploration in vast decision spaces". *Nature human behaviour,* ***2***(12), pp. 915–924.

[23] Schulz, E., Franklin, N. T., and Gershman, S. J., 2020. "Finding structure in multi-armed bandits". *Cognitive psychology,* ***119***, p. 101261.

[24] Griffiths, T. L., Lucas, C. G., Williams, J. J., and Kalish, M. L., 2008. "Modeling human function learning with gaussian processes". In Proceedings of the 21st International Conference on Neural Information Processing Systems, pp. 553–560.

[25] Lucas, C. G., Griffiths, T. L., Williams, J. J., and Kalish, M. L., 2015. "A rational model of function learning". *Psychonomic bulletin & review,* ***22***(5), pp. 1193–1215.

[26] Auer, P., 2002. "Using confidence bounds for exploitation-exploration trade-offs". *Journal of Machine Learning Research,* ***3***(Nov), pp. 397–422.

[27] Jones, D. R., Schonlau, M., and Welch, W. J., 1998. "Efficient global optimization of expensive black-box functions". *Journal of Global optimization,* ***13***(4), pp. 455–492.

[28] Carroll, J. D., 1963. "Functional learning: The learning of continuous functional mappings relating stimulus and response continua". *ETS Research Bulletin Series,* ***1963***(2), pp. i–144.

[29] Busemeyer, J. R., Byun, E., Delosh, E. L., and McDaniel, M. A., 1997. "Learning functional relations based on ex-

perience with input-output pairs by humans and artificial neural networks.".

[30] Mcdaniel, M. A., and Busemeyer, J. R., 2005. "The conceptual basis of function learning and extrapolation: Comparison of rule-based and associative-based models". *Psychonomic bulletin & review,* ***12***(1), pp. 24–42.

[31] Jones, A., Schulz, E., Meder, B., and Ruggeri, A., 2018. "Active function learning". *Biorxiv*, p. 262394.

[32] Bishop, C. M., 2006. *Pattern recognition and machine learning*. springer.

[33] Pitt, M. A., and Myung, I. J., 2002. "When a good fit can be bad". *Trends in cognitive sciences,* ***6***(10), pp. 421–425.

[34] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., 2011. "Scikit-learn: Machine learning in Python". *Journal of Machine Learning Research,* ***12***, pp. 2825–2830.

[35] Rasmussen, C. E., and Williams, C. K., 2006. *Gaussian processes for machine learning*. The MIT Press.

[36] Kopeček, P., 2014. "Selected heuristic methods used in industrial engineering". *Procedia Engineering,* ***69***, pp. 622–629.

[37] Sutton, R. S., and Barto, A. G., 2018. *Reinforcement learning: An introduction*. MIT press.