Cooperative Multi-Objective Bayesian Design Optimization

GEORGE MO, University of Cambridge, United Kingdom
JOHN DUDLEY, University of Cambridge, United Kingdom
LIWEI CHAN, National Yang Ming Chiao Tung University, Taiwan
YI-CHI LIAO, Aalto University, Finland
ANTTI OULASVIRTA, Aalto University, Finland
PER OLA KRISTENSSON, University of Cambridge, United Kingdom

Computational methods can potentially facilitate user interface design by complementing designer intuition, prior experience, and personal preference. Framing a user interface design task as a multi-objective optimization problem can help with operationalizing and structuring this process at the expense of designer agency and experience. While offering a systematic means of exploring the design space, the optimization process cannot typically leverage the designer's expertise in quickly identifying that a given 'bad' design is not worth evaluating. We here examine a cooperative approach where both the designer and optimization process share a common goal, and work in partnership by establishing a shared understanding of the design space. We tackle the research question: how can we foster cooperation between the designer and a systematic optimization process in order to best leverage their combined strength? We introduce and present an evaluation of a cooperative approach that allows the user to express their design insight and work in concert with a multi-objective design process. We find that the cooperative approach successfully encourages designers to explore more widely in the design space than when they are working without assistance from an optimization process. The cooperative approach also delivers design outcomes that are comparable to an optimization process run without any direct designer input, but achieves this with greater efficiency and substantially higher designer engagement levels.

CCS Concepts: • **Human-centered computing → Interaction design process and methods**.

Additional Key Words and Phrases: Interaction Technique; Interface Design; Bayesian optimization

1 INTRODUCTION

Design can be framed as a multi-objective optimization problem in which design parameters are selected to maximize user outcomes [25]. Framing design in this way can help to systematize the process of arriving at a final design configuration but eliminating the role of the human designer is not without cost. Removing the human designer may have detrimental consequences in terms of reduced engagement, satisfaction, and deskilling [8]. Furthermore, humans are powerful synthesizers and can leverage extensive bodies of knowledge and prior experience to inform their decisions [37]. Human-in-the-loop approaches have shown significant promise in many applications of machine learning [39]. The human-in-the-loop concept is particularly suited to the interaction design context, given that performance and design objectives can be very difficult to articulate *a priori*. Designers working in-the-loop can also potentially improve the efficiency of the design optimization

Authors' addresses: George Mo, University of Cambridge, United Kingdom; John Dudley, University of Cambridge, United Kingdom; Liwei Chan, National Yang Ming Chiao Tung University, Taiwan; Yi-Chi Liao, Aalto University, Finland; Antti Oulasvirta, Aalto University, Finland; Per Ola Kristensson, University of Cambridge, United Kingdom.

ACM 2160-6463/2024/6-ART
https://doi.org/10.1145/3657643

Table 1. Characteristics of Cooperative AI (adapted from [11]) and their alignment with the features of our Cooperative interface.

Characteristic	*Description*	*Cooperative MOBO Interface Features*
Shared Understanding	The ability to take into account the consequences of actions, to predict another's behavior, and the implications of another's beliefs and preferences.	• Intuitive visualization of evaluated designs. • Ability to share developed intuition and understanding regarding portions of the design space to avoid. • Consideration of previously evaluated designs when proposing new designs.
Communication	The ability to explicitly and credibly share information with others relevant to understanding behavior, intentions, and preferences.	• Ability for the user to express regions of the design space that MOBO should avoid. • Input of a confidence value for the forbidden regions and ranges. • Visualization of design space coverage.
Commitment	The ability to make credible promises when needed for cooperation.	• Forbidden regions prevent MOBO from pursuing designs in that region to enable credible user steering of MOBO.

process by leveraging their evolving intuition and prior experience to quickly ascertain whether a given design configuration has promise, or should be avoided.

In this paper, we explore a cooperative approach that leverages the benefits and capabilities of each alternative approach by allowing designers to work in concert with the systematic optimization process. This conceptual framing of the approach is consistent with the established definition of 'cooperative' in the *Oxford English Dictionary*: "that works together, or with another or others, towards the same end, purpose, or effect" [30]. Fostering cooperation, however, requires the forging of trust and shared understanding, established through some form of dialogue. Dafoe et al. [11] outline three key characteristics of Cooperative AI: shared understanding, communication, and commitment. The challenge then for cooperative design optimization, and the focus of this paper, is how to construct an effective interface between the designer and optimization process that embodies these characteristics.

An effective interface for cooperative design optimization should allow for the underlying optimization process to be changed at will. In practice, however, different optimization methods exhibit different qualities and opportunities for interactivity. Therefore, the specific choice of optimization process underlying the cooperative approach is a secondary but nevertheless important consideration. Multi-objective Bayesian optimization (MOBO) has emerged as an effective method for systematically examining an unfamiliar design problem in order to efficiently obtain Pareto optimal configurations [2]. We employ MOBO in this study as the systematic design technique given its efficiency and suitability for HCI design problems where measures of user outcomes are potentially noisy and uncertain. There has also been limited work examining how a designer can work in unison with MOBO in order to improve outcomes [8].

To produce an interface allowing designers to work in concert with MOBO, we aimed to support the three key characteristics of good cooperation espoused by Dafoe et al. [11]. Table 1 lists the implemented features (detailed later in Section 3) of our cooperative MOBO approach that align with these characteristics. Most importantly, we allow the designer to express their emerging understanding of the design space to the MOBO process as well as promote a shared understanding through enhanced visualization of state and history. To effectively incorporate the designer's developing intuition of the design space, we also introduce a distinction between a complete evaluation of a particular design and evaluations that a designer might choose to abandon early due to parameter choices quickly assessed to be unsuitable. Throughout this paper, we refer to a *heuristic* evaluation [14, p. 324] as a trial evaluation where the designer can obtain a quick but less rigorous appreciation of the quality of the design, and a *formal* evaluation as a full evaluation where a design is thoroughly assessed through multiple different conditions to obtain metrics of performance. To give a concrete example of this distinction, consider a designer who selects some parameters for an interaction technique and then performs a short self-experiment to test the efficiency and accuracy of the design. This *heuristic* evaluation can quickly inform the designer of whether that particular design has promise or should be discarded. By contrast, a *formal* evaluation in this example might involve the designer running a controlled user study with a sample of users to more precisely assess the quality of the design.

We investigate the potential of our cooperative MOBO approach in two user studies. Study 1 is designed to examine the relative merits of the cooperative approach compared with an entirely designer-led process or an entirely optimizer-led process. We therefore compare three operating conditions: Designer, where the design process is entirely controlled by the designer; Optimizer, where the design process is entirely controlled by MOBO; and Cooperative where the designer can exercise high-level command over the MOBO procedure. This investigation is non-trivial given potential nuisance factors associated with learning that prevent a protocol in which participants complete the same design task in each condition. Instead, we employ three distinct design tasks but simulated them such that we can control for difficulty and complexity. This enables a within-subjects design protocol that allows participants to offer rich comparative feedback on the experience of designing in each condition.

Study 2 is an expert evaluation of the Cooperative approach as applied to designers' own real-world design problems. This study serves to capture expert opinion on the advantages and disadvantages of the cooperative MOBO procedure when used in practice. To facilitate this investigation, we implemented the Cooperative procedure as a web application and provide stub code and documentation to facilitate integration. Study 2, therefore, captures feedback both on the suitability of the Cooperative approach as well as on the ease with which our design tool can be integrated and used.

In summary, this paper makes three main contributions:

(1) It introduces a Cooperative approach that enables greater designer control over the MOBO procedure as applied to interaction design problems.
(2) It presents an empirical evaluation of the Cooperative approach in direct comparison with the two opposing ends of the spectrum: the Designer and Optimizer approaches.
(3) It contributes an implementation of the Cooperative approach as a web-based design tool, including integration stub code, to facilitate rapid adoption by the HCI community.

2 RELATED WORK

Interaction with human-in-the-loop optimization has recently gained the attention of the HCI community [8, 19, 20, 24, 29, 36]. Bayesian optimization has emerged as a promising technique for supporting this paradigm thanks to its suitability for interactive settings. In comparison to other optimization methods like reinforcement learning or genetic algorithms, Bayesian optimization is sample-efficient. This efficiency means that less input is required from users, which is critical for applications involving experts. It also offers a principled way to handle variance and noise inherent in human input. Consequently, the method has been investigated across a variety of applications, including: personalizing user interfaces [21], optimizing interaction techniques [8], tuning visual designs [6, 24], making design recommendations [17], and adapting feedback to users [19], among others (see Dudley and Kristensson [16]).

However, the question remains how to offer effective controls and feedback during optimization. A recent study by Chan et al. [8] investigated designers led by a Bayesian optimizer to optimize an interaction technique. Chan et al. found that optimizer-led designers explore larger areas of design spaces, and were better able to overcome design fixation; however their reported levels of agency dropped dramatically. They felt that the optimizer was 'holding their hand' and diminishing their ownership of the final outcome. Chan et al.'s work highlights the need for interfaces and interactions that improve the experience of a designer working in the loop with an optimizer by addressing aspects of controllability, transparency, and explainability. This is critical for improving the efficacy and acceptability of this method among practitioners.

In this brief review, we focus on emerging research on interaction techniques and tools for steering optimization processes and Bayesian optimizers specifically. For a review of Bayesian optimization from a machine learning perspective, we point the reader to Shahriari et al. [34].

2.1 Conventional Human-in-the-Loop Optimization

Bayesian optimization is a machine learning method for optimization of non-transparent systems. This means that it is suitable for optimization tasks where no efficient mathematical representation of an objective function exists. All optimization methods for non-transparent systems operate by querying samples from the objective function and deciding where to query next. Thanks to the use of a *surrogate model*, Bayesian approaches are well suited to applications where the objective function is expensive or difficult to evaluate [34]. The surrogate model, often a Gaussian process model, is updated based on samples and used to select the next query point optimally. An *acquisition function* is a principled way to use a surrogate model to decide where to explore (look at regions of high uncertainty in the design space) and when to exploit (search in the local vicinity of a promising candidate). However, updating the surrogate model is expensive in itself. With an increasing number of design parameters, objectives, or samples, the cost of an update can increase dramatically. At the moment, Bayesian optimization is best suited for low-dimensional problems [5].

In *human-in-the-loop optimization*, the human 'is' the non-transparent system. The optimization process is indirectly guided by human input through feedback and observed human behavior in response to a set of input parameters. Feedback can be either explicit (asking the user for ratings) or implicit (via measurements taken during the actual use of the design). In standard uses of Bayesian optimization in human-in-the-loop design, the user has no control over which samples are queried. For example, Khajah et al. [21] used Bayesian optimization to maximize gamers' engagement by tuning game mechanics. Kadner et al. [19] customized font designs for individuals to maximize reading speed. Dudley et al. [17] used task completion time measured in a crowdsourced task as an objective to refine the design parameters of simple user interfaces.

2.2 Visualization of Multidimensional and Multi-Objective Design Spaces

Until recently, applications of Bayesian optimization in HCI were limited to the optimization of a single objective [8]. However, most HCI problems are characterized by a complex interplay among multiple and often competing objectives (e.g., speed versus accuracy in text entry). As there is no longer one defined optimum for multiple objectives, the concept of Pareto optimality is important. A design is considered to be Pareto optimal if no individual objective can be enhanced by changing the design parameters without resulting in at least one individual objective being worse off. To address this gap, researchers have turned towards multi-objective approaches for Bayesian optimization. These methods can produce a *Pareto frontier* that shows the candidates that strike unique and optimal trade-offs among the objectives. A *Pareto frontier display* is a visualization of the Pareto set for a user. While this approach can produce informative outcomes to pick from, it does not provide a method for steering the optimizer.

Within the visualization community, there has been significant work examining how to support users with inspecting and developing an understanding of high-dimensional design spaces. Sedlmair et al. [32] consider a related problem to ours and seek to provide a framework for the analysis of the parameter space around data visualizations. Various visualization methods have also been proposed to help highlight the link between parameters and design objectives in higher dimensions. Spence et al. [35] describe the Influence Explorer, a visual tool utilizing a Parallel Coordinates Plot (PCP) with additional visualization of the instantiated designs. This tool was designed to facilitate the quick exploration of parameters and their influence on performance. Spence et al. [35] also introduce the Prosecution Matrix which provides a relatively compact visualization of the influence of pairs of parameters. Paraglide [4] is another GUI tool for examining parameter spaces of multi-dimensional simulation models while Torsney-Weir et al. [38] presents a method for visualizing higher dimensional shapes by capturing 2D slices. These various methods can potentially enhance user understanding of the design space but do not in themselves facilitate direct interaction with the optimization process.

2.3 Preference Galleries

Brochu et al. [7] and Koyama et al. [23, 24] demonstrated how Bayesian optimization can be used in concert with preference galleries. A *preference gallery* is a display of several design candidates, from which the user can choose the most interesting one(s). The authors showed that visual designs can be tuned more effectively using this approach. Brochu et al. [7] demonstrated a technique for allowing designers to quickly determine appropriate values for smoke animation while Koyama et al. [23, 24] sought to streamline user editing of photographs to achieve a desired visual appearance. The benefit of preference galleries is that it increases the user's agency. By expressing 'this is interesting', the user can steer the optimizer toward more relevant designs. However, all options are still provided by the optimizer and the user has no possibility to express other than like/dislike.

2.4 Bayesian Optimization Libraries

The recognized value of Bayesian optimization as a general-purpose optimization tool has prompted the development of a range of software facilitating its use. Single objective Bayesian optimization is now available within established libraries such as SCIKIT-OPTIMIZE for Python and the Statistics and Machine Learning Toolbox for MATLAB. Given the added complexity involved in multi-objective Bayesian optimization, current (at the time of writing) packaged implementations vary in their level of maturity and capabilities. BoTorch [2] is a relatively full-featured and actively developed library implementing a particular variant of multi-objective Bayesian optimization [13]. Other packages supporting multi-objective Bayesian optimization include GPFlowOpt [22] and MOBOpt [18]. These various projects chiefly target developers familiar with the techniques and correspondingly offer good configurability. They offer limited or no interaction techniques for visualizing or controlling the optimization process.

2.5 Design Tools using Optimizers

Outside of Bayesian optimization, human-in-the-loop optimization has been extensively applied to design tasks in HCI. In MenuOptimizer [1], the designer is assisted during the task of combinatorial optimization of menus. In DesignScape [29], layout suggestions are given for position, scale, and alignment of elements. Other design tools that have a human-in-the-loop aspect include Sketchplore [36] where real-time design optimization is integrated into a sketching tool; Forte [9], in which designers can directly iterate on fabrication shape design through topology optimization; in Kapoor et al. [20], where the behavior of classification systems can be iteratively refined by designers; and in Lomas et al. [26], where game elements are iteratively adjusted to increase user performance. At the time of writing, we are not aware of a design tool specifically developed for applying Bayesian optimization. We focus on addressing this gap by presenting and evaluating an interface enabling a cooperative design process leveraging Bayesian optimization.

2.6 Summary

To sum up, while previous methods have shown the potential of Bayesian optimization in assisting designers with the task of exploring a design spaces, there is a need to develop principled approaches that allow experts to: (i) better express their knowledge, (ii) understand the design space, and (iii) guide the optimizer. We seek to address this observed gap by exploring a cooperative approach to design optimization that facilitates shared understanding, communication and commitment as per the characteristics described in Table 1.

3 DESIGN OF THE COOPERATIVE MOBO INTERFACE

The review of prior work in the previous section highlights various deficiencies in the conventional application of Bayesian optimization to user interface design. Most importantly, conventional approaches do not allow the designer to fully leverage their domain expertise or prior experience, and may inhibit the designer from gaining a comprehensive understanding of the design space. Therefore, our design of the Cooperative MOBO interface has the following specifications:

(1) The interface should allow the designer to specify and evaluate any design instance within the design space and facilitate inspection of how the different parameters impact the corresponding objectives through interactive visualizations.
(2) The interface should allow the designer to choose between performing a heuristic and formal evaluation.
(3) The interface needs to support functions giving the designer the ability to guide the design search in a mixed-initiative workflow using information from both heuristic and formal evaluations.

These three motivating points are intended to allow designers to actively engage in the design process and express their intuition. The mixed-initiative workflow allows designers to both use their own judgment as well as to receive guidance from MOBO as they search for Pareto optimal designs. The final design of the Cooperative MOBO interface is illustrated in Figure 1. Below, we detail the various features of the Cooperative MOBO interface and how they are integrated.

3.1 Expressing and Visualizing the Design Parameterization

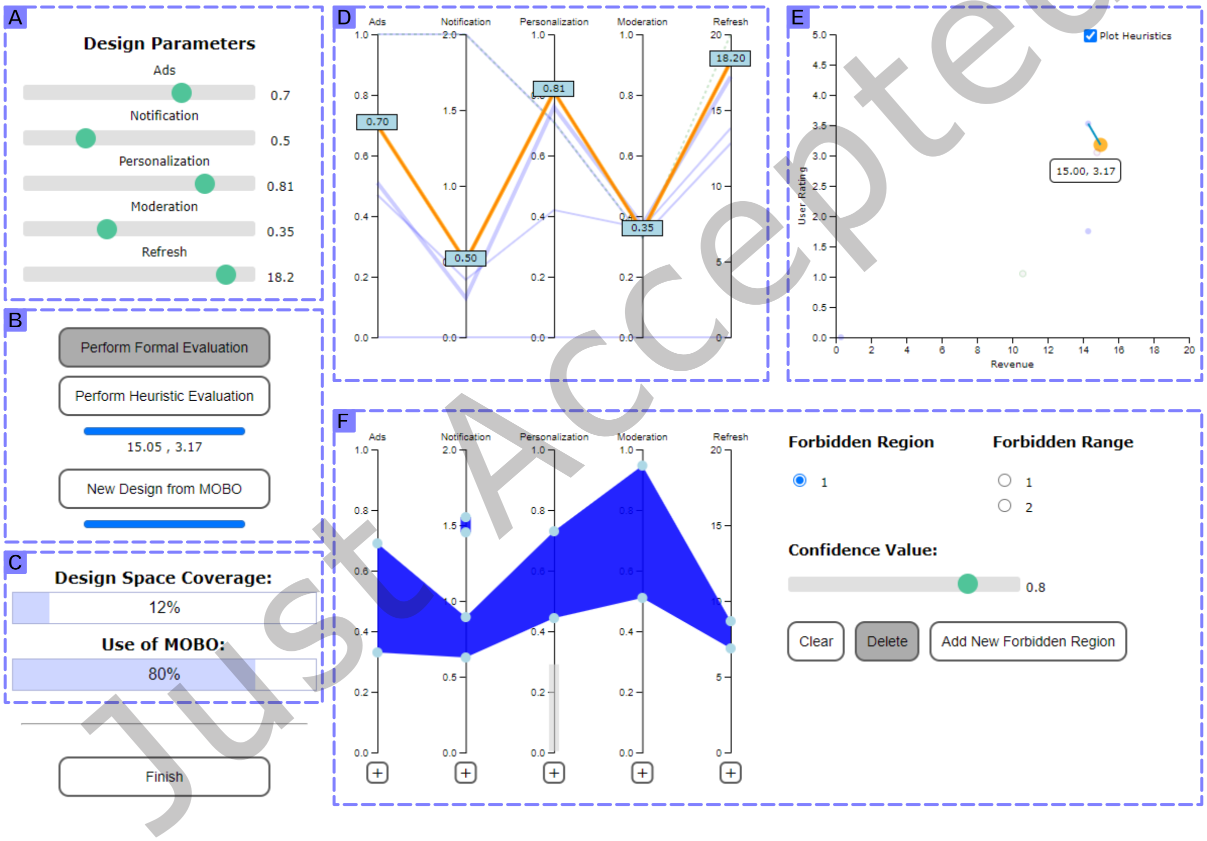
First, the designer must input the design parameters and the corresponding design objectives to be optimized. The design parameters should be continuous or have values that can be expressed in a continuous range with a lower and upper bound. The design objectives should also be continuous and should be expressed in a way that they are maximized in the design process. It is recommended that the number of design parameters be fewer than 10 as per guidance in the Bayesian Optimization literature [27], and currently, the application is limited to two competing objectives to facilitate clear visualization in the interface.

During the design process, the designer can adjust the sliders shown in region A of Figure 1 to set the desired values for each of the design parameters. In the study described later in Section 5, any adjustment of these sliders was immediately reflected in the interfaces shown in Figure 3. The designer can then decide to perform heuristic or formal evaluations for the specified design. We support interactive visualizations to help the designer in the design process. Region D of Figure 1 shows the parallel coordinates plot (PCP) used to concisely represent the previously evaluated designs as well as the currently selected design. The previously evaluated designs are also displayed on the objectives chart (region E). The PCP shows both the heuristic and formally evaluated designs, with the heuristic designs represented by dotted lines and the formal designs represented by solid lines. In the objectives chart, the heuristic designs are shown with non-filled dots whereas the formal designs are filled. When the user hovers over a particular design in the PCP or the objectives plot, the application highlights that particular design and the corresponding objective values in both the PCB and objectives chart with a yellow color, and displays the numerical values. The user can also toggle the *Plot Heuristics* checkbox, to hide or show the heuristic designs in the interface.

The visualizations also integrate interactive features with the design parameter sliders. When the sliders of region A are moved, a red line on the PCP of region D updates to indicate the current design parameter slider values. This visualization aids the designer in understanding where the currently selected design parameter values sit with respect to previously evaluated points in the design space. The most recent design evaluated is shown in yellow for formal evaluations and green for heuristic evaluations [28]. In addition, when the user clicks on a particular design in the PCP or the objectives chart, it sets those values in the sliders. This can be helpful if the user wishes to return to a previously evaluated design and examine variations around that point.

3.2 Using MOBO to Obtain a New Design

To enable the mixed-initiative aspect of the application, the interface features a *New Design from MOBO* button as shown in region B of Figure 1. When this button is clicked, the application executes the algorithm detailed



Design Parameters

Ads 0.7

Notification 0.5

Personalization 0.81

Moderation 0.35

Refresh 18.2

15.05 , 3.17

Design Space Coverage: 12%

Use of MOBO: 80%

Forbidden Region ● 1

Forbidden Range ○ 1 ○ 2

Confidence Value: 0.8

Fig. 1. The Cooperative MOBO Interface. The top left (A) shows the sliders for adjusting the design parameters. Below these sliders (B) are the buttons to perform a formal or heuristic evaluation as well as to obtain a design recommendation from MOBO. Towards the bottom left (C) are the metrics showing the design space coverage and use of MOBO. The visualizations at the top represent the PCP (D) and corresponding objectives chart (E) of the designs previously evaluated. In the bottom half (F), there is the interactive visualization for inputting the forbidden regions and ranges. Note that in this figure the red line in the PCP that corresponds to the slider values is overlaid with the yellow line which represents the most recently evaluated design.

later in Section 4 and searches for a new promising design candidate to recommend to the user. After execution, the design parameter sliders are updated to the recommended design candidate values. The user can then decide to either formally or heuristically evaluate this recommended design.

3.3 Monitoring Progress and History

To aid the designer in understanding the design process history, we show two metrics in region C of the interface to the designer: *Design Space Coverage* and *Use of MOBO*. Design Space Coverage is defined as follows. Suppose that the design parameter space is of dimension d, which we divide into 2^d hypercubes. This represents each dimension being split into two halves, such that if $d = 2$, there would be four corners of the design space that could potentially be 'covered'. We chose to split each dimension into two since as d increases, using a larger split number would result in the design coverage metric tending to zero. We count the number of these hypercubes that contain a formally evaluated design, which is represented by k. The Design Space Coverage is then defined as $\frac{k}{2^d}$, which represents a proxy for the percentage of the design space that is formally evaluated. This metric is included to encourage users to explore more of the design space so as to avoid design fixation.

The Use of MOBO Percentage is calculated as follows. Suppose that at the time of display, the *Perform Formal Evaluation* button has been clicked on T times. Note that we only consider the execution of formal evaluations in this metric since formal evaluations are significantly more time-consuming than heuristic evaluations and therefore, chiefly dictate the efficiency of the design process. Now suppose that the user has clicked the *New Design from MOBO* button to aid in the design process t times immediately prior to clicking the *Perform Formal Evaluation* button. Then the Use of MOBO Percentage is defined as $\frac{t}{T}$, hence representing the proportion of formal evaluations for which the user used MOBO to aid in the design process. This metric was included in the interface to foster a cooperative mode of work. Visibility of the metric provides awareness of the relative frequency of use of MOBO, and, it is hoped, may encourage a balance between leveraging suggested designs and manually selecting designs.

3.4 Expressing Forbidden Regions and Ranges

We also introduce forbidden regions and ranges to provide the designer with the ability to guide the design search using MOBO. Conceptually, these inputted regions and ranges are portions of the design space that the user does not want MOBO to continue proposing design candidates from.

We implemented an interactive chart for inputting forbidden regions and ranges, as shown in region F of Figure 1. To input a forbidden region, the user can click on *Add New Forbidden Region* which inputs a forbidden region in the chart centered around the design parameter values currently set in the sliders. The upper and lower boundaries for the forbidden region are initially set as ±5% of the parameter range for each parameter. The user can change the upper and lower boundaries of the forbidden region by dragging the light blue circles shown in Figure 1. The user can also change the confidence value of the forbidden region, with 1 representing that the user has full confidence in excluding designs from that forbidden region and 0 representing no confidence. Selecting *Clear* clears the selection of the forbidden region and selecting *Delete* deletes the selected forbidden region. A forbidden region can be selected by directly clicking on it in the interactive chart.

We also support the input of forbidden ranges. Suppose a designer has figured out that for one particular parameter, designs with that parameter having a value in a particular range should not be further explored. Then, inputting a forbidden range for that parameter with that particular range prevents MOBO from further proposing designs with a parameter value in that range. In the interactive chart, an adjustable forbidden range can be inputted for a parameter by clicking the '+' button in the appropriate axis. The interactive features for the forbidden range are the same as those for the forbidden regions.

3.5 Integrating with the Web Interface

We believe that the interface presented in this section has good generalizability and can be used widely in multi-objective design problems encountered in HCI. To accommodate such use, the interface itself is implemented as a web application with computations offloaded to the server. This allows, as described later in Section 6, for the rapid integration of the cooperative optimization process within designers' existing prototype applications. As an initial demonstration of the swift integration concept, we make available[1] a Unity integration package that can be imported from GitHub via the native Unity package manager. This package includes detailed step-by-step instructions, sample code, and a toy design problem (implemented as a Prefab game object), illustrating the integration procedure. This allows the developer to manage the optimization process using the web interface: new designs are sent for evaluation to the Unity application and evaluation results are automatically returned and displayed in the interface. A minimal integration with a custom design task in Unity involves only two steps:

(1) Listening to one action that is called when new design parameters are received from the web interface:
`CODWebInterface.OnDesignParametersUpdated(List<float> parameterValues)`

(2) Calling one method when the design evaluation (either formal or heuristic) is complete:
`CODWebInterface.EvaluationComplete(List<float> objectiveValues, bool formal = true)`

4 TECHNICAL APPROACH

In this section, we present the technical details underlying the integration of the forbidden regions and ranges when acquiring a new design proposal from MOBO. Our goal is to allow designers to steer the MOBO acquisition process in order to explore new regions or avoid undesirable regions of the design space. As a concrete example, consider an interaction design task for a selection technique where different target distances and orientations are evaluated to obtain the average speed and accuracy for a given parameter configuration. These parameter and performance values can then be fed into the MOBO procedure. However, when performing this task we may intuitively know within the first few trials whether a design is very good or very bad. In such circumstances, it may be useful if the evaluation can be halted early. The problem is that the average speed and accuracy calculated for those few initial trials cannot be fed into MOBO because it will likely yield a biased result since it reflects only a subset of the operational task space. Hence, we need to find a method that can allow designers to early stop in what we call a heuristic trial but at the same time incorporate that heuristic trial into MOBO so it can use this additional information.

A solution to this problem is to introduce what we call forbidden regions and ranges. The idea is that after a heuristic trial, if the designer wants to early stop because the design is bad, then the designer can input a forbidden region around this design point to prevent MOBO from further proposing a new design similar to this bad design. In addition to this, we added a feature to input a forbidden range in which we can exclude a whole interval of a particular design parameter (for example, a design parameter being smaller than a threshold is expected to always yield a bad design). We also let the designer tune the confidence of the forbidden region or range which allows the designer to choose how strongly a particular region or range is to be avoided by MOBO.

Although our original motivation was to include early stopping in evaluations, this actually translates to a wider application in cases where we have heuristic and formal evaluations. The approach also allows the designer to nudge the optimization process towards wider design exploration by, for example, inputting a forbidden region in an area of the design space that is being overly exploited.

[1] https://github.com/Jojadud/MOBODesignerPackage

4.1 Formalization of the Approach

Suppose we have a design space of dimension d, $\mathcal{X} \in \mathbb{R}^d$, and an objective space of dimension k, $\mathcal{Y} \in \mathbb{R}^k$. In practice, $\mathcal{X} = [0,1]^d$ and $\mathcal{Y} = [-1,1]^k$ after normalization. We also denote here, $V_M = 2^k$ as the maximum hypervolume in this problem statement.

In our multi-objective optimization scenario, we have a function representing a non-transparent system $f : \mathcal{X} \rightarrow \mathcal{Y}$. Suppose we have collected data $\mathcal{D}$. In MOBO, we have an acquisition function, expected improvement in Pareto hypervolume ($EIPV$), from which we maximize to obtain the next point to sample: $x^* = \arg\max_{x \in \mathcal{X}} EIPV(x|\mathcal{D})$.

In our forbidden region framework, we essentially have a list of forbidden regions $\{\mathcal{R}_j\}_{j=1}^J$ that are all of the form $\{[l_1, u_1], ..., [l_d, u_d]\}$. In practice, let us say we have a heuristic point a that we want to wrap around a forbidden region of width $2w$. Then the corresponding forbidden region would be $\{[a_1 - w, a_1 + w], ..., [a_d - w, a_d + w]\}$, where we have set by default $w = 0.05$.

Essentially, we want to penalize everything inside the forbidden region with a penalty function, that is, we introduce a penalty term of $\frac{\alpha}{d(x,\mathcal{R}_j)^2}$, where $d(x, \mathcal{R}_j)$ is the closest distance between a point x and the forbidden region. Here, α is a penalty scaling factor which we set to $\alpha = 0.01$ by default based on observed behavior when applied on synthetic functions. Hence, the acquisition function we have at the moment is:

$$ACQ(x|\mathcal{D}) = EIPV(x|\mathcal{D}) - \sum_{j=1}^{J} \frac{\alpha}{d(x,\mathcal{R}_j)^2} \tag{1}$$

However, with this form of the penalty function we run into the problem of when $x \in R_j$ yielding a $d(x, \mathcal{R}_j) = 0$, and a numerical explosion in the penalty. To account for this, we set a bound on the penalty function by V_M, the theoretical maximum hypervolume, as this is the upper bound, albeit loose, of $EIPV(x|\mathcal{D})$. Hence, our modified acquisition function becomes:

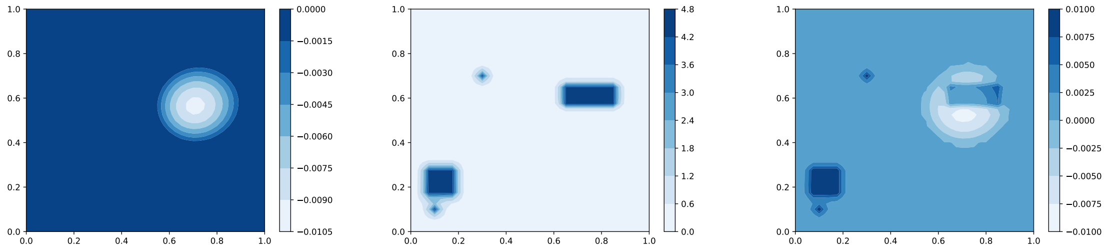
$$ACQ(x|\mathcal{D}) = EIPV(x|\mathcal{D}) - \sum_{j=1}^{J} \min(V_M, \frac{\alpha}{d(x,\mathcal{R}_j)^2}) \tag{2}$$

However, this gives rise to another problem—it is entirely possible that V_M is not the same order of magnitude as $EIPV(x|\mathcal{D})$, yielding very large penalties and rendering the entire acquisition function dominated by penalties and not by the expected increase in Pareto hypervolume. To counter this, we attach another normalization constant to penalty functions to make them the same order of magnitude as $EIPV(x|\mathcal{D})$. Hence, the modified acquisition function is:

$$ACQ(x|\mathcal{D}) = EIPV(x|\mathcal{D}) - Z\sum_{j=1}^{J} \min(V_M, \frac{\alpha}{d(x,\mathcal{R}_j)^2}) \tag{3}$$

where we set $Z = \frac{\max_{x \in \mathcal{X}} EIPV(x|\mathcal{D})}{V_M}$ to enforce the two terms to have the same order of magnitude. To illustrate this, consider Figure 2, which shows the $EIPV$ acquisition function, the acquisition function with penalizing factors without normalization, and finally the acquisition function with penalizing factors with normalization. Note that without normalization, the forbidden regions are clearly seen but completely overwhelm the $EIPV$ values. However, with normalization the $EIPV$ values can have a significant effect.

To introduce the confidence of including each forbidden region, we introduce a factor called the confidence factor $\beta_j \in [0,1]$ to each confidence region. If $\beta_j = 0$, it means we are completely uncertain of including the forbidden region $\mathcal{R}_j$ and hence it has no effect, but if $\beta_j = 1$ then we are fully confident in including the forbidden region and hence it has the intended penalizing effect. This factor is set by the designer for each forbidden region. Hence, the final acquisition function we use in our studies is:



(a) (b) (c)

Fig. 2. (a) shows the base *EIPV* acquisition function; (b) shows the acquisition function with penalty functions without normalization, where only the penalty region effects can be seen; (c) shows that after normalization the effect of the base acquisition function *EIPV* can be seen, in addition to the penalty regions.

$$ACQ(x|\mathcal{D}) = EIPV(x|\mathcal{D}) - Z\sum_{j=1}^{J}\beta_j \min(V_M, \frac{\alpha}{d(x,\mathcal{R}_j)^2}) \tag{4}$$

Given all the forbidden regions and the confidence values corresponding to each region, we use the following procedure to find the next point of acquisition:

(1) Use a gradient ascent algorithm (e.g., L-BFGS-B) with repeated trials to find $\max_{x\in\mathcal{X}} EIPV(x|\mathcal{D})$, and hence evaluate $Z = \frac{\max_{x\in\mathcal{X}} EIPV(x|\mathcal{D})}{V_M}$.
(2) Use gradient ascent again with multiple repeated trials to maximize $ACQ(x|\mathcal{D})$ to find x^* which is the next design proposal.

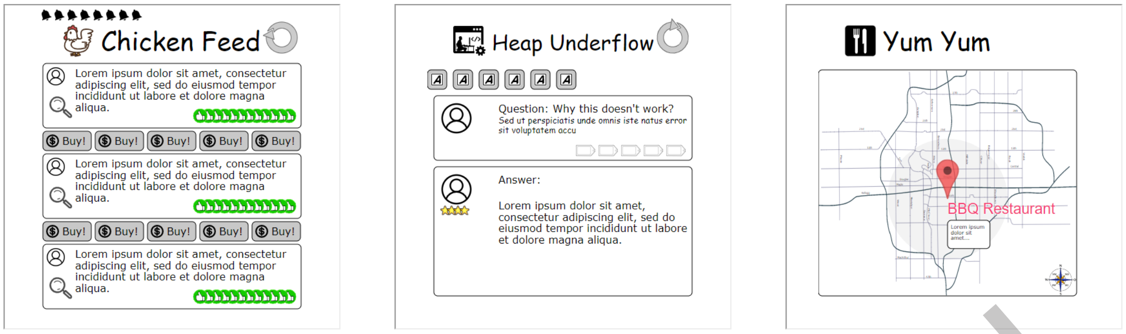
4.2 Including Forbidden Ranges

As previously mentioned, sometimes we want to exclude a whole region of the design space based on one parameter. For example, we might want to exclude all designs with a parameter having a value in a particular range. Suppose we want to exclude all designs within the design dimension i with values between $[l_i, u_i]$. Then, the corresponding forbidden region would be: $\{[0,1], ..., [l_i, u_i], ..., [0,1]\}$. Afterwards, we proceed the same way as detailed in the previous subsection by treating the forbidden range as a special form of a forbidden region.

5 STUDY 1: DESIGN OPTIMIZATION TASK

The purpose of Study 1 is to examine the advantages and disadvantages of the COOPERATIVE method over the conventional alternative approaches of entirely relying on the designer (DESIGNER) or entirely relying on the optimizer (OPTIMIZER). The DESIGNER method is where the designer has the capacity to perform both formal and heuristic evaluations but only has access to the sliders to change the design parameters and the PCP and objectives chart. The OPTIMIZER method is where the designer can only use the *New Design from MOBO* button to obtain a new design but has no ability to change the design parameter sliders and can only perform formal evaluations. This is to reflect the scenario where the designer would only use MOBO to obtain new designs for formal evaluation as in the original MOBO algorithm—it does not support the inclusion of information from the heuristic evaluations. This is also consistent with a recent study by Chan et al. [8] that compared the DESIGNER and OPTIMIZER methods.

There are three hypotheses that we aim to examine through this experiment.


Chicken Feed
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Buy!
Heap Underflow
Question: Why this doesn't work?
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accu
Answer:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Yum Yum
BBQ Restaurant

(a) App 1: Social microblogging.

(b) App 2: Q&A message board.

(c) App 3: Restaurant map.

Fig. 3. The three design applications presented to participants in Study 1. Any adjustment of the parameter sliders (shown in Figure 1 (A)) is directly reflected by visual changes in the interface. For example, in App 1, increasing the 'Ads' displays more of the 'Buy!' icons between each post. Increasing the 'Notification' rate displays more 'bell' icons at the top of the interface. Increasing 'Personalization' adds more 'like' icons to each post. Increasing 'Moderation' increases the size of the 'magnifying glass' icon. Finally, increasing the 'Refresh' rate causes the spinning arrow in the top right of the interface to turn faster. The appearance of all three application interfaces at both extremes of the parameter ranges is shown in Figure 7 in Appendix A.

- $\mathcal{H}1$: The COOPERATIVE method leads to similar performance, as indicated by the relative hypervolume of the derived Pareto-optimal designs, to the performance of the OPTIMIZER method.
- $\mathcal{H}2$: The COOPERATIVE method requires fewer formal design evaluations than the OPTIMIZER method to complete the design optimization task.
- $\mathcal{H}3$: The COOPERATIVE method results in higher agency and engagement for designers compared to the OPTIMIZER method.

When designing this study, we considered both between-subjects and within-subjects study designs. We ultimately pursued a within-subjects design for two key reasons. First, the between-subjects study design with three conditions would have required a large number of participants (likely 48 or more participants). Second, the between-subjects study design would not have allowed participants to reflect on and comment on the relative advantages and disadvantages of the different conditions. However, to compare each method in a within-subjects design we require a distinct design problem for each condition as a control for learning effects. To tackle this issue, we designed three design problems of matched difficulty. We developed three design problems inspired by three typical web applications (the applications in Figure 3), thereby preserving a consistent theme. To ensure these design problems were comparable in terms of difficulty, we employed multi-objective test functions of a similar form (detailed in the next section) to relate the design parameters to their hypothetical performance. In other words, these test functions represent a synthetic but controllable replacement for actual end-user evaluations of these various applications.

5.1 Test Functions for Study 1

In Study 1, we employed various multi-objective test functions that were all five dimensional, with two objectives. Although the design parameters and outputted objective values shown in the interface might be of varying ranges, they are first respectively normalized to be in the range [0, 1] and [-1, 1]. Hence, the test functions implemented have a domain of $[0, 1]^5$ and a range of $[-1, 1]^2$. The specification that we sought to meet for the test functions was that the difficulty in finding the optimal trade-off designs should be approximately consistent. As a result, we designed the functions to have similar forms and to have the same final Pareto hypervolume.

For a realistic design scenario, we expect that the objective function is roughly convex and that multiple modes of optimal designs are quite rare. This can be justified with the observed objective function behaviors for the interactive design tasks in prior work [17]. As a result, we designed each of the objectives $j = 1, 2$ to be quadratic in form with the optimum value c_j, position of the optimum to be $\mathbf{a}_j = [a_{j1}, a_{j2}, a_{j3}, a_{j4}, a_{j5}]$ and the scaling factors to be $\mathbf{b}_j = [b_{j1}, b_{j2}, b_{j3}, b_{j4}, b_{j5}]$:

$$[f_1(\mathbf{x}), f_2(\mathbf{x})] = [c_1 - \sum_{i=1}^{5} b_{1i}(x_i - a_{1i})^2, c_2 - \sum_{i=1}^{5} b_{2i}(x_i - a_{2i})^2]$$

Note that if we perform the permutation of $(1, 2, 3, 4, 5) \rightarrow (\sigma(1), \sigma(2), \sigma(3), \sigma(4), \sigma(5))$, the form of f is preserved but the parameter axes are swapped. In addition, note that for the domain of $\mathbf{x} \in [0, 1]^5$, if we apply the transformation of $(a_{1i}, a_{2i}) \rightarrow (1 - a_{1i}, 1 - a_{2i})$, then we also preserve the final hypervolume but with now the optimum at $\mathbf{x}_j = [1 - a_{j1}, 1 - a_{j2}, 1 - a_{j3}, 1 - a_{j4}, 1 - a_{j5}]$ for each objective function. Swapping the j's (i.e. $[1, 2] \rightarrow [2, 1]$) also preserves the form of the overall objective function. We call these functions isomorphic to each other. To exploit these properties to ensure that the three methods have applications of relatively the same difficulty to find the optimal designs, each of the test functions has the dimensions being permutations or mirror reflections of each other and hence they are all isomorphic (i.e. same maximum hypervolume and optimum designs up to reflection and permutation).

For the study, we have three applications with isomorphic objective synthetic functions and a separate tutorial task with two design parameters and two objectives. Table 2 shows the parameters and objectives for each test function and what they represent in the three applications and the tutorial task, with the corresponding ranges. Figure 3 shows the visual interfaces for the three design applications. Figure 7 in Appendix A illustrates these same three design applications at the extreme parameter values. Note that adjusting each of the design parameters in the sliders triggers a change in the visual interface of the application. For example, increasing the size of the icon in the restaurant map application will result in the icon in the visualization increasing in size. The visual feedback allows the designers to develop an intuitive understanding of the effect of the design change. Table 3 lists the parameters for the application and tutorial objective functions after normalization. These parameters were selected so that the Pareto optimal values correspond to some sort of intuitive behavior (i.e. increasing the density of ads increases daily revenue).

After querying the functions we add noise to generate the synthetic performance results. The amount of noise added differs for formal and heuristic evaluations. For a formal evaluation, we add a uniformly distributed noise from $Unif(-0.05, 0.05)$, and for a heuristic evaluation, the noise is drawn from $Unif(-0.25, 0.25)$, independently for each of the two objectives. For a formal evaluation, to simulate the longer time it takes to obtain the result, it takes 20 seconds to obtain the result, but for a heuristic evaluation, it only takes 3 seconds. These times were selected to introduce a trade-off between the accuracy of the result and the time it takes to obtain it via the heuristic and formal evaluations. The unnormalized objective values are then displayed in the interface.

5.2 Participants and Setup

We recruited $n = 18$ participants (11 male, 7 female, aged between 20 and 36 and with an average age of 26.1) from our institution through emailing lists for this first study. The study was approved by the Research Ethics Committee in the Department of Engineering at the University of Cambridge. The participants were mainly from engineering backgrounds, with 12 of them with engineering education (e.g., civil engineering, computer science, aerospace engineering) and other participants varying in fields from theoretical linguistics and biochemistry. 8 participants also reported having had experience with computational design. This demographic aligns well with the target users of the Cooperative MOBO interface as the features are intended for individuals who have some experience with design, and may therefore feel comfortable incorporating personal intuition. For the study, the

Table 2. Design parameters, objectives and ranges for each of the applications and Tutorial (*T*).

App	x_1	x_2	x_3	x_4	x_5	f_1	f_2
1	Density of ads - [0, 1]	Notification frequency - [0, 2] per hour	Personalization rate of content - [0, 1]	Moderation rate of content - [0, 1]	Refresh time of content - [0, 20] minutes	Daily revenue - [0, 20] thousands USD	User rating - [0, 5]
2	# Question categories - [5, 50]	Refresh time of content - [0, 1000]	Length of question preview - [0, 500] characters	Max number of question tags - [1, 10]	Threshold activity rating for user to answer questions - [0, 5]	Answering rate of questions - [0, 2] per minute	% Questions Answered - [0, 100]
3	Location icon transparency - [0.5, 1]	Cursor distance for restaurant to show - [5, 50]	Location icon size - [1, 10]	Description box size - [10, 50]	Restaurant name text size - [10, 30]	Average speed to find restaurants - [0, 2] per minute	Accuracy in finding all restaurants - [0, 100]
T	Force to register contact on screen - [10, 100] N	Area to register contact on screen - [0.5, 3.0] cm^2	-	-	-	Average target hit speed - [0, 3] per second	Accuracy of hitting targets - [0, 100] %

Table 3. Parameters $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{b}_1$, $\mathbf{b}_2$, c_1, c_2 for the synthetic functions used in the applications and Tutorial (*T*).

App	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{b}_1$	$\mathbf{b}_2$	c_1	c_2
1	[0.9,0.3,0.8,0.25,0.25]	[0.3,0.35,1.1,0.75,0.3]	[0.9,0.4,1.3,0.7,0.4]	[1.0,0.6,1.2,0.5,0.4]	0.7	0.8
2	[-0.1,0.25,0.7,0.7,0.65]	[0.2,0.75,0.75,0.1,0.7]	[1.2,0.5,0.4,1.0,0.6]	[1.3,0.7,0.4,0.9,0.4]	0.8	0.7
3	[1.1,0.75,0.35,0.3,0.3]	[0.8,0.25,0.3,0.9,0.25]	[1.2,0.5,0.6,1.0,0.4]	[1.3,0.7,0.4,0.9,0.4]	0.7	0.8
T	[0.3,0.35]	[0.7,0.65]	[1.0,0.8]	[1.2,0.9]	0.7	0.8

participant had access to a monitor and a mouse. For each participant, the whole study lasted approximately 2 hours. All participants received a £20 voucher as a token of appreciation for their time.

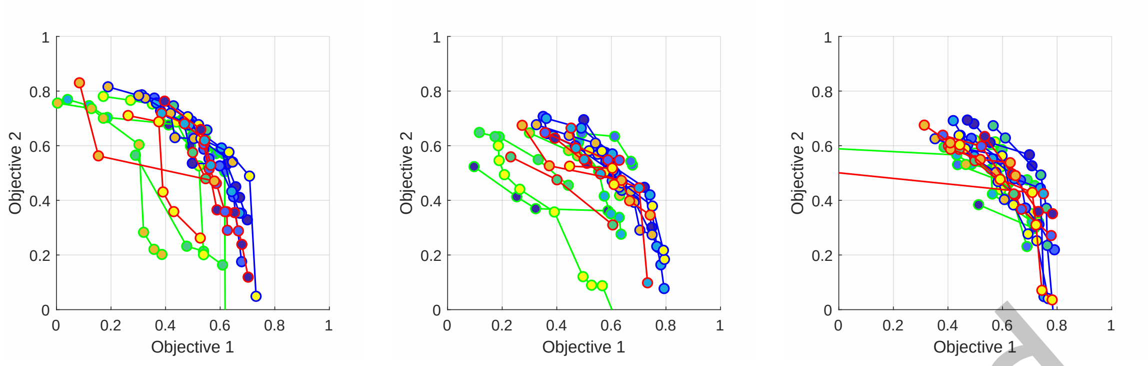
5.3 Experimental Method

As described previously, we employ a within-subjects study design. The interface conditions and design applications were counterbalanced as described in Appendix B.

5.4 Procedure

The steps of the study procedure were as follows:

(1) Before introducing the three applications, we gave the participant a detailed tutorial on design optimization with multiple objectives and the idea of Pareto optimal designs. This was to familiarize the participants with the context of multi-objective interaction design.



(a) App 1.

(b) App 2.

(c) App 3.

Fig. 4. Combined Pareto front for the three applications. Green: Designer, Red: Cooperative, Blue: Optimizer. The marker fill color indicates the Pareto front for a single individual. In simple terms, a Pareto front producing a curve closer to the top-right of the plot can be interpreted as capturing a better set of good designs. Although there is substantial variation between users, we can observe that the Pareto fronts obtained in the Optimizer condition (blue) are generally grouped towards the top right while the Pareto fronts obtained in the Designer condition (green) are generally further from the top right (most prominent in App 1 and 2). The Pareto fronts obtained in the Cooperative condition (red) generally sit between the other two conditions.

(2) We then gave the participant a video tutorial on the interactive features that they will interact with in the web application. This video tutorial showed only the interactive features that the participant would immediately interact with in the task—for example, if the first application was using the Designer condition, the participant would only be introduced to the PCP, the objectives chart, and the sliders. However, if the first task was using the Cooperative condition, the forbidden regions and ranges, and the MOBO button would also be introduced.

(3) For each of the three applications, we gave the participant 5 minutes to interact with the tutorial task with two design parameters and two objectives. If the participant did not interact with a particular interactive feature (such as inputting forbidden regions or ranges), we explicitly showed the participant what the feature did. The objective of this step was to ensure participants were familiar with all the features of the interface.

(4) The main application design task was then presented to the participant with the instruction to find three designs for each application that optimally trade-off the two objectives by maximizing the two objectives. We gave the participant the context of designing a web interface for a technology company. The design parameters and objectives were described in detail with their corresponding ranges, and the interactive actions that the participant could use in this task were listed (e.g., manually tuning with sliders, using the MOBO button). The two different types of evaluations were also described with the trade-offs in accuracy and time to execute outlined. Finally, participants were informed of the following constraints for completing the task—the participant could decide to finish the task after a minimum of 15 minutes, but could continue to work on the task up to a maximum of 20 minutes.

(5) After all three condition-application pairings were completed, we presented the participant with a final questionnaire comparing the three interfaces. The main goal of the questionnaire was to understand the user's engagement and sense of agency in the optimization process and the perceived confidence in the

results. As there was no existing questionnaire that precisely served our needs, we adapted questions from the Creativity Support Index [10] and System Usability Scale [3] questionnaires.

5.5 Results

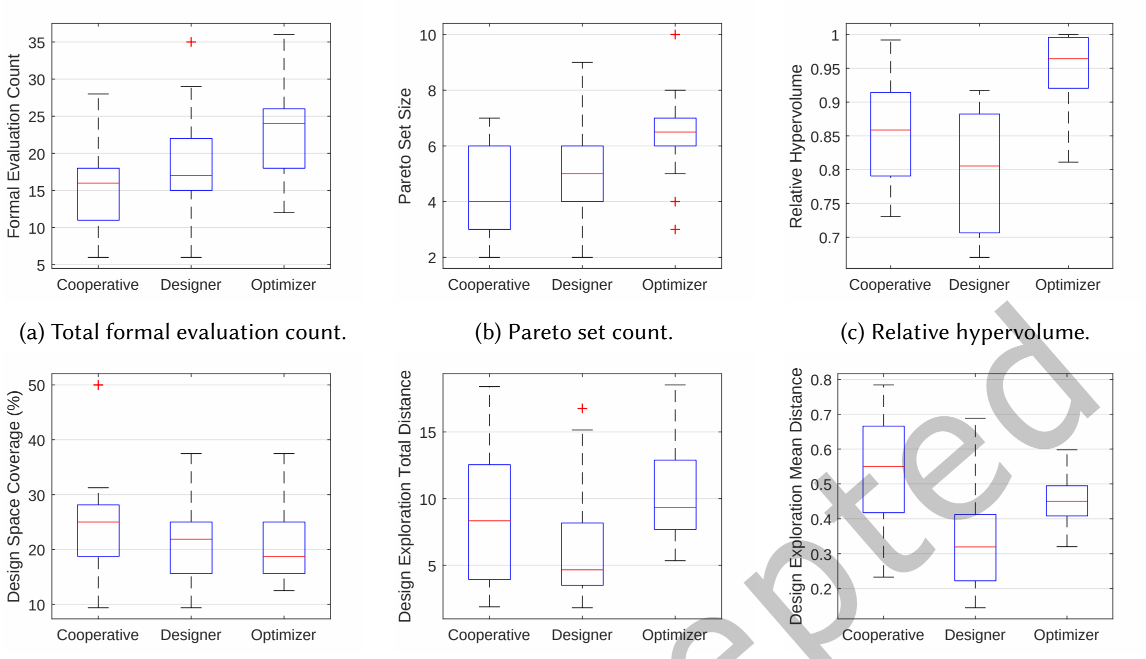
5.5.1 Pareto Set Discovery. Figure 4 plots the various Pareto fronts obtained in the different conditions for the three applications. Note that there are six fronts for each condition in each plot as this represents the number of participants who completed that particular condition-application pairing. Visual inspection suggests that generally the Optimizer condition delivers Pareto fronts with high hypervolume and good consistency between participants. The Designer condition appears to show visually higher levels of inter-participant variability. The Cooperative condition produces Pareto fronts that appear to sit, very approximately, between the other two conditions.

We make this comparison more concrete by computing several key metrics indicative of the efficiency and quality of the combined Pareto sets in each condition. Boxplots of these various metrics are shown in Figure 5. In subsequent analysis, we test for a significant condition effect using Friedman's test since the metric distributions are skewed. We perform multiple comparisons using Tukey's honestly significant difference procedure. As a check for nuisance effects associated with application or condition-application interaction, we also run a repeated measures ANOVA on the metrics shown in Figure 5. This reveals no significant application effect or condition-application interaction. This provides confidence that our efforts to produce applications of comparable difficulty and our counterbalanced experiment design were effective in limiting these nuisance effects.

Figure 5a shows a boxplot of the total number of formal evaluations performed by each participant in each condition. The mean counts are 15.17, 18.33, and 23.11 for Cooperative, Designer and Optimizer respectively. We find a significant effect for condition ($\chi^2(2) = 9.942$, $p = 0.007$), with the Cooperative condition exhibiting significantly fewer formal evaluations than Optimizer (p = 0.0047). There were no significant differences for the other pairwise comparisons. This result indicates that participants performed significantly fewer formal evaluations in the Cooperative condition than in the Optimizer condition. This is as expected since participants could only perform formal evaluations in the Optimizer condition whereas they could choose between a heuristic or formal evaluation in the Cooperative condition.

Figure 5b shows a boxplot of the total number of Pareto optimal designs obtained. The mean counts are 4.22, 5.17, and 6.44 for Cooperative, Designer and Optimizer respectively. We find a significant effect for condition ($\chi^2(2) = 9.164$, $p = 0.010$) with the Cooperative condition yielding significantly fewer designs than Optimizer (p = 0.0071). The other pairwise comparisons showed no significant differences. This finding indicates that the Cooperative condition produces significantly fewer Pareto optimal designs than the Optimizer condition. Although the difference is not significant, the Cooperative condition also delivers fewer designs than the Designer condition. This is likely due to the observed behavior of participants in the Designer condition of fixating on a particular design and evaluating minor variations around this point in the parameter space. This hypothesis also aligns with the observed significant differences in mean travel distance results reported later in this section.

We compute the relative hypervolume obtained in each condition for each participant by normalizing it with respect to the maximum hypervolume obtained in the corresponding application. Boxplots of the relative hypervolume obtained by participants in each condition are shown in Figure 5c. The mean relative hypervolume obtained is 0.85, 0.80, and 0.95 for Cooperative, Designer and Optimizer respectively. We find a significant effect for condition ($\chi^2(2) = 18.778$, $p < 0.001$) with Designer yielding significantly smaller hypervolume than Optimizer (p < 0.001). There are no significant differences for the other pairwise comparisons. This result matches with the visual observations that can be made in Figure 4 where the hypervolumes enclosed by the Designer



(a) Total formal evaluation count.

(b) Pareto set count.

(c) Relative hypervolume.

(d) Design space coverage.

(e) Design space travel distance.

(f) Mean travel distance.

Fig. 5. Key metrics indicative of the efficiency and quality of the Pareto set obtained in each condition. The Cooperative condition resulted in significantly fewer formal evaluations than the Optimizer condition (Figure 5a). This then produced significantly fewer designs in the Pareto set for Cooperative compared to Optimizer (Figure 5b) but with no significant difference between these two conditions in terms of the relative hypervolume (Figure 5c). There was no significant condition effect for design space coverage but Cooperative did deliver the highest mean design space coverage at 24.0% (Figure 5d). These results suggest that designers in the Cooperative condition were more efficient in their design space exploration than in the Optimizer condition and yielded comparable final outcomes in terms of the performance objectives of the designs in the Pareto set. Contrasting against the Designer condition, both the Optimizer and Cooperative conditions resulted in greater variation between successive design instances evaluated (Figures 5e and 5f).

Pareto fronts appear generally smaller than the Optimizer Pareto fronts, and with the Cooperative Pareto fronts sitting somewhere in between.

Another informative metric is the design space coverage as introduced in Section 3.3. The mean design space coverage is 24.0%, 21.0%, and 21.2% for Cooperative, Designer, and Optimizer respectively. This result suggests slightly greater design space coverage in the Cooperative condition but we find no significant effect for the condition.

Finally, there are two related metrics that provide a proxy measure of the degree of broad exploration of the design space versus local ‘fixation’. These are the total design space travel distance and the mean design space travel distance. The total design space travel distance is the sum of the Euclidean distance between subsequent formally evaluated points in the design space. The mean design space travel distance is the total divided by the total number of evaluations performed. Boxplots of these two metrics for the participant group in each condition are shown in Figures 5e and Figures 5f. The total design space travel distance reveals a significant effect for condition ($\chi^2(2) = 8.111$, $p = 0.017$). The mean total distances are 8.59, 6.47, and 10.44 for Cooperative,

Table 4. Median [min, max] responses on a seven-point Likert scale from 1—strongly disagree to 7—strongly agree for the seven statements presented in the post-condition questionnaire of Study 1.

	Statement	*Designer*	*Cooperative*	*Optimizer*
1	I was able to grasp the impact of each design parameter on the output of the design.	5.5 [2, 7]	5.0 [1, 7]	4.0 [1, 7]
2	The system allowed me to explore different design parameters to see what its impact on the design output is.	6.0 [4, 7]	6.0 [1, 7]	3.0 [1, 6]
3	I felt that I had control over searching different areas of the design space.	6.0 [3, 7]	5.5 [3, 7]	1.5 [1, 6]
4	I have an intuitive sense of the design space from using this system.	5.5 [2, 7]	5.0 [3, 7]	4.0 [1, 7]
5	I felt that the optimal designs I have obtained make sense.	6.0 [3, 7]	6.0 [4, 7]	5.0 [1, 7]
6	I felt that I was formally evaluating designs that aligned with my intuition of the design.	5.0 [2, 7]	5.5 [3, 7]	3.5 [1, 7]
7	I am confident that I found the optimal designs in the design space.	5.0 [3, 7]	5.0 [2, 7]	5.0 [1, 7]

Designer and Optimizer respectively. In post hoc pairwise comparisons, the Optimizer condition is found to be significantly higher than the Designer condition (p = 0.0128). When normalized by the number of evaluations performed, the mean travel distances are 0.536, 0.352, and 0.452 for Cooperative, Designer, and Optimizer respectively. There is a significant effect for condition ($\chi^2(2) = 11.444$, $p = 0.003$) with the mean distance traveled in the Cooperative condition significantly higher than the Designer condition. This result suggests that the Cooperative condition leads to more variation in design parameters between subsequent evaluations than in the Designer condition.

5.5.2 Subjective Experience and Preference. At the conclusion of the design exercise in each condition, participants responded to the seven statements listed in Table 4. The median responses (also presented in Table 4) highlight a comparable experience and intuition for participants in the Designer and Cooperative conditions. The Optimizer condition, however, received distinctly lower scores for aspects of the experience related to agency and understanding of the design space. We find significant differences for statements 1 to 6 (respectively: $\chi^2(2) = 9.750$, $p = 0.008$, $\chi^2(2) = 17.207$, $p < 0.001$, $\chi^2(2) = 24.133$, $p < 0.001$, $\chi^2(2) = 10.226$, $p = 0.006$, $\chi^2(2) = 6.145$, $p = 0.046$ $\chi^2(2) = 12.033$, $p = 0.002$) with the Cooperative condition yielding significantly higher scores than the Optimizer condition for statements 2 to 5. There are no significant differences between the Cooperative and Designer conditions.

Finally, participants were asked to express their overall preference among the three interfaces used. The majority of participants (12/18) indicated Cooperative as their top preference, followed by Designer (4/18), and the least favored interface was Optimizer (2/18). Participants also responded to the statement, "I would like to use the <method> again." on a seven-point Likert scale from 1—strongly disagree to 7—strongly agree. The median scores were 6.0, 5.0, and 3.0 for Cooperative, Designer and Optimizer respectively. The median score for the six participants who did not select Cooperative as their top preference was 5.0, indicating that the experience of the Cooperative condition for this subset of participants was still generally positive.

5.5.3 Summary. We briefly summarize the interpretation of these findings. Responding to $\mathcal{H}1$, Cooperative produces comparable (no statistical difference) relative hypervolume to Optimizer. Furthermore, responding to $\mathcal{H}2$, Cooperative delivers this hypervolume with fewer (significant) formal evaluations, suggesting greater efficiency over Optimizer. A downside is that there are fewer (significant) designs in the Pareto set for Cooperative than Optimizer. The fact that Cooperative traveled to fewer design instances than Optimizer but still yielded only marginally lower relative hypervolume suggests that the reduced resolution of the Pareto set may be tolerable if the captured designs are representative of the true Pareto optimal designs. The design space coverage is comparable (no statistical difference) for all methods. The mean travel distance (an indicator of the degree of local exploration versus broader exploration) is lowest for Designer and highest for Cooperative, with this difference being significant. This suggests Cooperative may help to offset the fixations issues encountered with Designer. The subjective experience of using Optimizer was much worse than Designer/Cooperative (which are themselves about the same). This suggests Cooperative maintains the benefits of Optimizer in terms of efficiency and hypervolume while avoiding many of the detriments of the exclusively systematic approach. Examining the subjective experience ($\mathcal{H}3$), the results of the questionnaire suggests that Cooperative promotes greater engagement in the task compared to Optimizer. Finally, 12 out of 18 participants indicated that Cooperative was their most preferred interface.

6 STUDY 2: EXPERT EVALUATION

Study 1 demonstrated the benefits of the cooperative optimization approach on simulated design tasks, compared to the designer-led and the optimizer-led approaches. We observed that the Cooperative condition produced Pareto sets of comparable quality with the Optimizer condition but with fewer evaluations and without the degraded user experience observed for the Optimizer condition. This highlights the significant promise of the cooperative optimization interface and approach presented. This second study aims to learn about experts' viewpoints and experiences working on their custom optimization problem using the cooperative approach. We investigate both the experience of designing with the cooperative optimization approach as well as the ease or difficulty associated with integrating the design tool into their prototype application. This investigation leverages the web interface integration capability described in Section 3.5 and illustrated in Figure 6. There are thus two main hypotheses that we aim to test in this study.

- $\mathcal{H}1$: The Cooperative method can be applied to common forms of interaction design problems.
- $\mathcal{H}2$: The users perceive high usability when operating the interface and the system of the Cooperative method to address an existing interaction design problem.

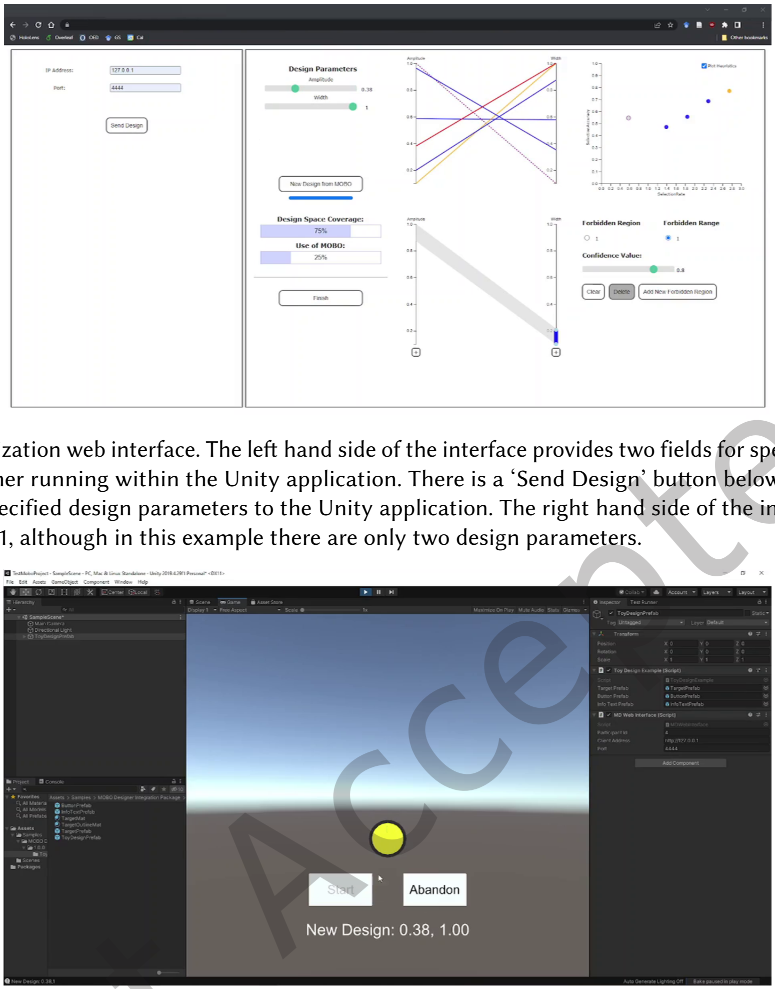
As in the simulated design tasks in Study 1, the cooperative web interface allows the expert participants to actively explore the design space by adjusting the parameter sliders, to access suggestions from MOBO whenever they need, as well as the use of the forbidden region and range features.

6.1 Participants

We recruited three interaction researchers (all PhD students in Human-Computer Interaction) aged 24 to 26. Each participant brought their own design problem and prototype implementation to the study. We constrained recruitment to participants with a prototype implementation in Unity since this is the current target of the supporting integration package. Participants were also required to have basic development skills since they are asked to independently integrate with the web API.

6.2 Study Method

The study was divided into four stages:


IP Address: 127.0.0.1
Port: 4444
Design Space Coverage: 75%
Use of MOBO: 25%
Confidence Value: 0.8
New Design: 0.38, 1.00

(a) Cooperative optimization web interface. The left hand side of the interface provides two fields for specifying the IP address and Port for the listener running within the Unity application. There is a 'Send Design' button below these fields which is clicked to send the specified design parameters to the Unity application. The right hand side of the interface is the same as that shown in Figure 1, although in this example there are only two design parameters.

(b) Unity demonstration application. The tutorial Unity application shown presents a reciprocal tapping task. The target button is the yellow circle with black outline. The user begins an evaluation by clicking on the 'Start' button, and then clicks on the button targets in sequence with their cursor. At any point, the user may click on the 'Abandon' button, which is then treated as a heuristic evaluation. In this tutorial, the design is parameterized by the size of the button and the offset between subsequent target buttons.

Fig. 6. Integration of web interface with the Unity demonstration application.

(1) The pre-study was conducted via email, and we asked expert participants to think about a few design problems they currently face. Participants were then instructed to reflect on the following questions: What are the parameters that govern the behavior and characteristics of this design? and How would they measure the quality of a given design configuration (i.e. what performance objectives does it seek to satisfy)?

(2) A preparatory interview was then conducted prior to the main design exercise. We interviewed participants about their design problem, the main parameters of the design, the main objectives relevant to the design, as well as their usual practice in determining suitable design parameter values given these objectives. We then introduced the Cooperative MOBO web interface, i.e., the tool, and Unity integration package with a tutorial video. We asked the participants to consider whether they are able to reframe the design problem into a simplified user task that takes less than two minutes to complete for a given parameter configuration. Overall, this preparatory step served to introduce the web interface as a tool for facilitating the optimization of their design problem and to help the participants prepare their design problem before entering the design exercise. The preparatory interview took approximately 30 minutes. Participants were given at least one evening break before starting the design exercise. This allowed participants to reflect on the discussion and make any necessary changes to their prototype application.

(3) The design exercise proper ran for approximately two hours. The experimenter initially confirmed with the participants about the framing of their design problem and design task, provided them with a link to the GitHub page of the Unity package, and then had them follow the integration instructions indicated in the GitHub repository README. As part of the tutorial, the Unity package includes a toy example of a simple button design task for target acquisition using a cursor. This task interface is illustrated in Figure 6b. When a new design is received, the size and offset of the button (yellow with black outline in Figure 6b) are updated. The user can then press ‘Start’ to initiate a 2D reciprocal target acquisition task based on the current design configuration. During the task, the user may choose to press the ‘Abandon’ button which suspends the current task and treats the observed performance as a heuristic evaluation. Otherwise, the user completes the reciprocal tapping task by clicking on each button in sequence with their cursor.

    In this tutorial, the interaction behaviors are already integrated with the web interface and no code modification is necessary. The tutorial provides participants with hands-on experience in using the Cooperative optimization procedure and allows them to refer to an example of how to complete the integration code. Once they had familiarized themselves with the procedure and integration, participants started importing the package into their custom program and working on the integration. The experimenter would sit beside the participant and observe any hurdles encountered and offer minimal assistance only when necessary. Having carried out the integration, participants were then instructed to utilize the Cooperative MOBO web interface and identify at least three optimal designs that yield optimal trade-offs between the two objectives. The participants directly evaluated the interface they were designing in order to obtain real measures of performance for the defined objectives. We recorded the full log of interactions and design evaluation results.

(4) Once the design exercise was complete, participants were asked to fill out the System Usability Scale (SUS) questionnaire and a post-exercise questionnaire. Finally, we conducted a post-exercise interview, where we asked participants about their impression of working with the Cooperative interface compared to their usual practice in design optimization, how the results produced by the tool compared to those obtained from their usual practice, what strategies they applied when using the tool, and what features of the design tool they found useful.

6.3 Results

The three participants examined three distinct interaction problems relevant to their research. All involved a prototype application developed in Unity and there was a common thread in that all design problems related to an interaction for use in augmented or virtual reality.

Table 5. Median responses on a seven point Likert scale from 1—very difficult to 7—very easy.

Statement	*P1*	*P2*	*P3*
How straightforward was the process of integrating the design tool into your application code?	6.0	6.0	7.0
How straightforward was the process of identifying suitable design parameters?	5.0	5.0	6.0
How straightforward was the process of identifying suitable design objectives?	5.0	7.0	6.0
How straightforward was the process of reframing your design problem as a simplified user task?	7.0	4.0	6.0

Table 6. Median responses on a seven-point Likert scale from 1—strongly disagree to 7—strongly agree.

Statement	*P1*	*P2*	*P3*
I found the "New Design from MOBO" button to be useful.	7.0	6.0	7.0
I found the ability to specify a "Forbidden Range" to be useful.	7.0	5.0	7.0
I found the ability to specify a "Forbidden Region" to be useful.	7.0	3.0	4.0
I was able to grasp the impact of each design parameter on the design objectives.	7.0	5.0	5.0
The tool allowed me to explore different design parameters to see how they impacted the design objectives.	6.0	7.0	7.0
I felt that the optimal designs I have obtained make sense.	7.0	5.0	6.0
I am confident that I found the optimal designs in the design space.	7.0	5.0	5.0

The three participants completed their design exercises using the Cooperative interface. The proportion of use of the 'New Design from MOBO' button was 61%, 29% and 50% for *P1–3* respectively. All participants obtained a set of Pareto optimal designs but the nature of the different design problems was such that each participant obtained a different number of unique Pareto optimal designs: *P1* found 4, *P2* found 3 and *P1* found 2. All participants commented that suggestions provided by the 'New Design from MOBO' button encouraged them to explore regions of the design space that went against their intuition but which in fact turned out to deliver promising design candidates in terms of performance. These findings provide evidence in support of $\mathcal{H}1$.

After completing the design exercise, participants responded to the statements summarized in Tables 5 and 6. The responses in Table 5 generally indicate that the process of integrating and using the cooperative optimization interface is relatively straightforward. Table 5 covers the experience of using the various features presented in the interface. The feedback on most features was positive apart from the "Forbidden Region" which some participants found initially confusing to interpret. We believe that with increased experience, users would be more comfortable in inputting forbidden regions, as we observed in Study 1. Finally, the three participants completed the System Usability Scale questionnaire with the instruction to focus on the integration and use of the design tool. The usability ratings obtained were: 80, 100, and 94 which indicates very good usability [3]. These high SUS scores provide evidence in support of $\mathcal{H}2$.

7 DISCUSSION

Our focus in this paper is to propose and evaluate a mixed-initiative Cooperative method and interface which allows for more control during the design optimization process compared to using a conventional MOBO process with limited user input. Most significantly, our incorporation of forbidden regions and ranges into the

Cooperative method allows the designer to obtain a design space coverage and final Pareto hypervolume that is comparable to the Optimizer condition while allowing the designer to incorporate design intuition to guide the design search with heuristic evaluations. This translated into a more efficient search of the design space with less formal evaluations, as compared to both the Designer and Optimizer methods. Additionally, our findings in terms of the significant difference in mean travel distance suggest that the Cooperative condition may have promoted wider exploration and less potential design fixation than the Designer condition. When this result is viewed in combination with the higher relative hypervolume obtained in the Cooperative condition compared to the Designer condition, there is some evidence to suggest that the interface features provided encouraged better trade-offs between exploitation and exploration. It is important to note, however, that there is some risk that the Cooperative condition may be subject to the same good and bad biases of the designer encountered in the Designer condition. These negative biases may manifest as unproductive exploitation of sub-optimal regions that the designers think are promising or dismissal of optimal regions that the designers think are bad. The fact that the relative hypervolume in the Cooperative condition remains below that achieved in the Optimizer condition may indicate the persistence of these biases and motivates further work on how these biases might be constructively managed.

Although the Cooperative method focuses on a canonical problem in multi-dimensional computational design with heuristic and formal evaluations, it is still somewhat difficult to frame non-trivial design problems in such a way that it is suitable for use. Further guidance is required on the parameterization of certain design problems and how to tackle design problems involving categorical design parameters. Similarly, additional guidance on the selection of objectives, particularly in the context of correlated objectives, would be beneficial. Standard acquisition functions are generally robust to correlated objectives although there are more advanced functions that may be employed in such circumstances [33]. In addition, our software is currently limited to only two objectives. Extending this to three objectives and beyond would require exploration into different visualization techniques to allow efficient exploration of the design space, as well as more efficient implementations of the algorithms.

Methods in multi-objective design optimization yield a variety of final design parameters that are Pareto-optimal. However, in all three methods explored, there is little guidance offered on what to do once this Pareto-optimal set is obtained. The literature in design engineering has suggested that creative design involves the co-evolution of problem and solution spaces [15]. Applying this approach, the Pareto-optimal set could be treated as the new design space from which the designer could select the final design. Although it may be reasonable to defer to the designer when choosing the final design that satisfies an optimal trade-off, there could be guidance given in the software implementation to suggest the best design candidate given the priorities expressed by the designer.

In future work, it would be interesting to explore what other direct interaction mechanisms could support and incentivize designers to explore more of the design space while obtaining Pareto optimal designs in an efficient manner in terms of the number of evaluations. Possible future directions could involve the design of visualizations that allow designers to compare a set of promising new unexplored design candidates instead of proposing just a single new design point. It would also be interesting to examine what algorithmic developments in conjunction with visualizations can aid mixed-initiative efforts to further guide designers to new promising design regions.

From our first study, we observed that participants may lose trust in MOBO during the Cooperative condition in situations where the user employs MOBO too early and obtains poor suggestions. This early breakdown in trust typically led to less reliance on MOBO as assistance during the design process. This echoes an overarching problem of establishing and maintaining appropriate trust in human-AI interaction [12, 31]. Possible routes of future work to tackle this problem could involve giving more explicit guidance on when to use MOBO to propose a new design and giving the designer more transparency on how MOBO is proposing the new design through interactive visualizations. There may also be opportunities to develop new algorithms that explicitly allow designers to control the relative level of exploration of the design space so as to meet user expectations of the performance of the proposed design.

8 CONCLUSIONS

In this paper, we have proposed a cooperative multi-objective Bayesian design optimization approach that allows designers to perform multi-objective design in a mixed-initiative manner. We specifically designed a new optimization algorithm that takes in forbidden regions and ranges set by the designer so as to use information from both formal and heuristic evaluations. We compared this Cooperative method to both the Designer and Optimizer methods and showed that it is efficient in obtaining Pareto-optimal designs and promotes the incorporation of designer intuition and control in the design process. We also demonstrated the Cooperative method's application to several different bespoke interaction design tasks, illustrating its versatility and efficacy in practice with experts. Overall, we have introduced a cooperative multi-objective Bayesian design optimization method and interface that provides both performance and design experience benefits, effectively combining the advantages of conventional designer and optimizer-led methods.

ACKNOWLEDGMENTS

Antti Oulasvirta was supported by the Research Council of Finland (flagship program: Finnish Center for Artificial Intelligence, FCAI, grants 328400, 345604, 341763; Subjective Functions, grant 357578). Liwei Chan was supported by the National Science and Technology Council of Taiwan (NSTC 112-2221-E-A49-125). John J. Dudley and Per Ola Kristensson were supported by the EPSRC (grants EP/S027432/1 and EP/W02456X/1).

REFERENCES

[1] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. MenuOptimizer: interactive optimization of menu systems. In *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*. ACM Press, St. Andrews, Scotland, United Kingdom, 331–342. https://doi.org/10.1145/2501988.2502024

[2] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. 2020. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*. http://arxiv.org/abs/1910.06403

[3] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *J. Usability Studies* 4, 3 (may 2009), 114–123.

[4] Steven Bergner, Michael Sedlmair, Torsten Moller, Sareh Nabi Abdolyousefi, and Ahmed Saad. 2013. ParaGlide: Interactive Parameter Space Partitioning for Computer Simulations. *IEEE Transactions on Visualization and Computer Graphics* 19, 9 (2013), 1499–1512. https://doi.org/10.1109/TVCG.2013.61

[5] Ali Borji and Laurent Itti. 2013. Bayesian optimization explains human active search. In *Advances in Neural Information Processing Systems*, C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2013/file/a3f390d88e4c41f2747bfa2f1b5f87db-Paper.pdf

[6] Eric Brochu, Tyson Brochu, and Nando de Freitas. 2010. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '10)*. Eurographics Association, Goslar, DEU, 103–112.

[7] Eric Brochu, Tyson Brochu, and Nando de Freitas. 2010. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 103–112.

[8] Liwei Chan, Yi-Chi Liao, George B. Mo, John J. Dudley, Chun-Lien Cheng, Per Ola Kristensson, and Antti Oulasvirta. 2022. Investigating Positive and Negative Qualities of Human-in-the-Loop Optimization for Designing Interaction Techniques. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA. https://doi.org/10.1145/3491102.3501850

[9] Xiang 'Anthony' Chen, Ye Tao, Guanyun Wang, Runchang Kang, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2018. Forte: User-Driven Generative Design. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3174070 event-place: Montreal QC, Canada.

[10] Erin Cherry and Celine Latulipe. 2014. Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Trans. Comput.-Hum. Interact.* 21, 4, Article 21 (jun 2014), 25 pages. https://doi.org/10.1145/2617588

[11] Allan Dafoe, Yoram Bachrach, Gillian Hadfield, Eric Horvitz, Kate Larson, and Thore Graepel. 2021. Cooperative AI: machines must learn to find common ground. *Nature* 593, 7857 (May 2021), 33–36. https://doi.org/10.1038/d41586-021-01170-0 Bandiera_abtest: a Cg_type: Comment Number: 7857 Publisher: Nature Publishing Group Subject_term: Machine learning, Computer science, Society, Technology, Sociology, Human behaviour.

[12] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel. 2020. Open Problems in Cooperative AI. https://doi.org/10.48550/arXiv.2012.08630 arXiv:2012.08630 [cs].

[13] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. 2020. Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization. _eprint: 2006.05078.

[14] A. Dix, J. Finlay, G.D. Abowd, and R. Beale. 2003. *Human-computer Interaction.* Pearson/Prentice-Hall. https://books.google.co.uk/books?id=IuQxui8GHDcC

[15] Kees Dorst and Nigel Cross. 2001. Creativity in the design process: co-evolution of problem–solution. *Design Studies* 22, 5 (Sept. 2001), 425–437. https://doi.org/10.1016/S0142-694X(01)00009-6

[16] John Dudley and Per Ola Kristensson. 2022. Bayesian Optimisation of Interface Features. *Bayesian Methods for Interaction and Design* (2022), 259.

[17] John J. Dudley, Jason T. Jacques, and Per Ola Kristensson. 2019. Crowdsourcing Interface Feature Design with Bayesian Optimization. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19).* Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300482

[18] Paulo Paneque Galuzio, Emerson Hochsteiner [de Vasconcelos Segundo, Leandro dos Santos Coelho, and Viviana Cocco Mariani. 2020. MOBOpt — multi-objective Bayesian optimization. *SoftwareX* 12 (2020), 100520. https://doi.org/10.1016/j.softx.2020.100520

[19] Florian Kadner, Yannik Keller, and Constantin Rothkopf. 2021. AdaptiFont: Increasing Individuals' Reading Speed with a Generative Font Model and Bayesian Optimization. In *Proceedings of the 2021 Conference on Human Factors in Computing Systems (CHI'21).* ACM.

[20] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. 2010. Interactive Optimization for Steering Machine Classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10).* ACM, New York, NY, USA, 1343–1352. https://doi.org/10.1145/1753326.1753529

[21] Mohammad M. Khajah, Brett D. Roads, Robert V. Lindsey, Yun-En Liu, and Michael C. Mozer. 2016. Designing Engaging Games Using Bayesian Optimization. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16).* Association for Computing Machinery, New York, NY, USA, 5571–5582. https://doi.org/10.1145/2858036.2858253

[22] Nicolas Knudde, Joachim van der Herten, Tom Dhaene, and Ivo Couckuyt. 2017. GPflowOpt: A Bayesian Optimization Library using TensorFlow. *arXiv preprint – arXiv:1711.03845* (2017). https://arxiv.org/abs/1711.03845

[23] Yuki Koyama, Issei Sato, and Masataka Goto. 2020. Sequential gallery for interactive visual design optimization. *ACM Transactions on Graphics* 39, 4 (July 2020), 88:88:1–88:88:12. https://doi.org/10.1145/3386569.3392444

[24] Yuki Koyama, Issei Sato, Daisuke Sakamoto, and Takeo Igarashi. 2017. Sequential line search for efficient visual design optimization by crowds. *ACM Transactions on Graphics* 36, 4 (July 2017), 48:1–48:11. https://doi.org/10.1145/3072959.3073598

[25] Yi-Chi Liao, John J. Dudley, George B. Mo, Chun-Lien Cheng, Liwei Chan, Antti Oulasvirta, and Per Ola Kristensson. 2023. Interaction Design With Multi-Objective Bayesian Optimization. *IEEE Pervasive Computing* 22, 1 (2023), 29–38. https://doi.org/10.1109/MPRV.2022.3230597

[26] J. Derek Lomas, Jodi Forlizzi, Nikhil Poonwala, Nirmal Patel, Sharan Shodhan, Kishan Patel, Ken Koedinger, and Emma Brunskill. 2016. Interface Design Optimization As a Multi-Armed Bandit Problem. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16).* ACM, New York, NY, USA, 4142–4153. https://doi.org/10.1145/2858036.2858425 event-place: Santa Clara, California, USA.

[27] Riccardo Moriconi, Marc Peter Deisenroth, and K. S. Sesh Kumar. 2020. High-dimensional Bayesian optimization using low-dimensional feature spaces. *Machine Learning* 109, 9 (Sept. 2020), 1925–1943. https://doi.org/10.1007/s10994-020-05899-z

[28] Jakob Nielsen and Rolf Molich. 1990. Heuristic Evaluation of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '90).* Association for Computing Machinery, New York, NY, USA, 249–256. https://doi.org/10.1145/97243.97281

[29] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. 2015. DesignScape: Design with Interactive Layout Suggestions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15).* ACM, New York, NY, USA, 1221–1224. https://doi.org/10.1145/2702123.2702149

[30] OED Online. 2022. cooperative, adj. and n. https://www.oed.com/view/Entry/41038

[31] Max Schemmer, Niklas Kuehl, Carina Benz, Andrea Bartos, and Gerhard Satzger. 2023. Appropriate Reliance on AI Advice: Conceptualization and the Effect of Explanations. In *Proceedings of the 28th International Conference on Intelligent User Interfaces* (Sydney, NSW, Australia) *(IUI '23).* Association for Computing Machinery, New York, NY, USA, 410–422. https://doi.org/10.1145/3581641.3584066

[32] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Torsten Möller. 2014. Visual Parameter Space Analysis: A Conceptual Framework. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2161–2170. https://doi.org/10.1109/TVCG.2014.2346321

[33] Amar Shah and Zoubin Ghahramani. 2016. Pareto Frontier Learning with Expensive Correlated Objectives. In *International Conference on Machine Learning*. PMLR, 1919–1927. http://proceedings.mlr.press/v48/shahc16.html

[34] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* 104, 1 (Jan. 2016), 148–175. https://doi.org/10.1109/JPROC.2015.2494218

[35] B. Spence, L. Tweedie, H. Dawkes, and Hua Su. 1995. Visualisation for functional design. In *Proceedings of Visualization 1995 Conference*. 4–10. https://doi.org/10.1109/INFVIS.1995.528680

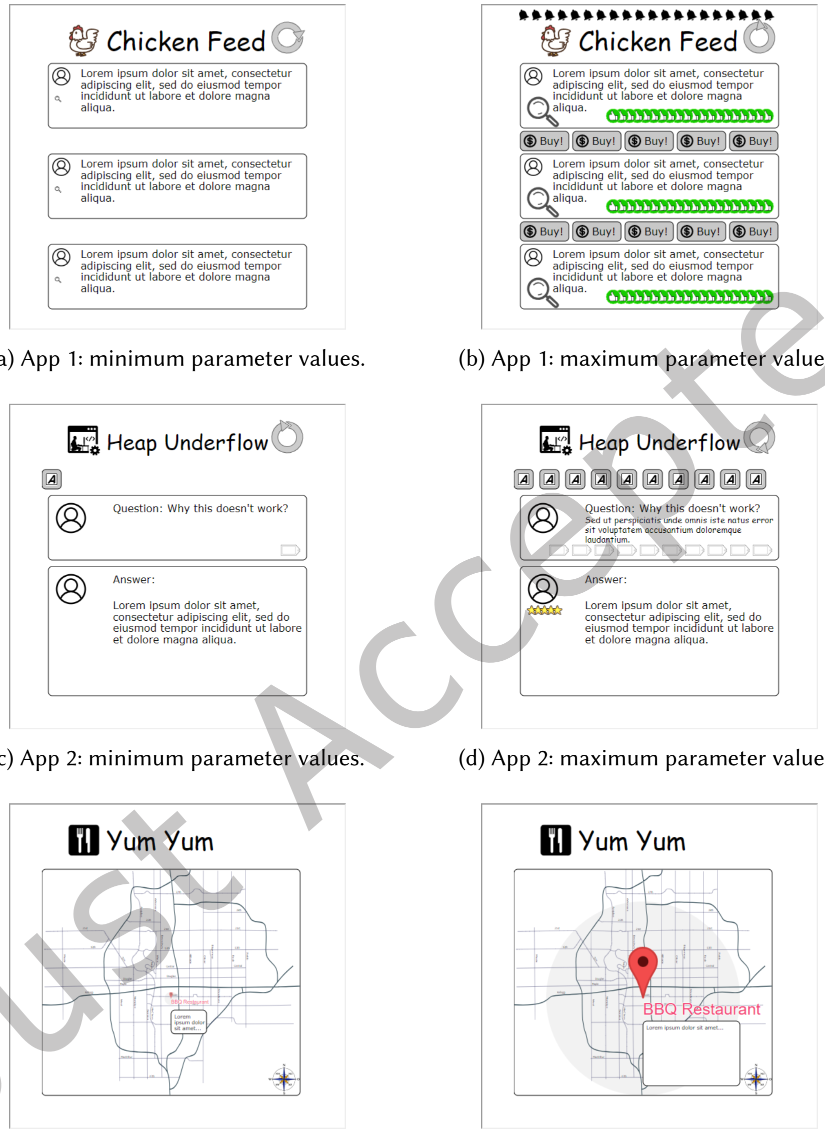
[36] Kashyap Todi, Daryl Weir, and Antti Oulasvirta. 2016. Sketchplore: Sketch and Explore with a Layout Optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS '16)*. ACM, New York, NY, USA, 543–555. https://doi.org/10.1145/2901790.2901817

[37] Alice Toniolo, Federico Cerutti, Timothy J. Norman, Nir Oren, John A. Allen, Mani Srivastava, and Paul Sullivan. 2023. Human-machine collaboration in intelligence analysis: An expert evaluation. *Intelligent Systems with Applications* 17 (2023), 200151. https://doi.org/10.1016/j.iswa.2022.200151

[38] T. Torsney-Weir, T. Möller, M. Sedlmair, and R. M. Kirby. 2018. Hypersliceplorer: Interactive visualization of shapes in multiple dimensions. *Computer Graphics Forum* 37, 3 (2018), 229–240. https://doi.org/10.1111/cgf.13415

[39] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems* 135 (2022), 364–381. https://doi.org/10.1016/j.future.2022.05.014

A DESIGN APPLICATION PARAMETER EXTREMES



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Question: Why this doesn't work?

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.

Answer:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet...

(a) App 1: minimum parameter values.

(b) App 1: maximum parameter values.

(c) App 2: minimum parameter values.

(d) App 2: maximum parameter values.

(e) App 3: minimum parameter values.

(f) App 3: maximum parameter values.

Fig. 7. Extreme parameter settings for the three design applications presented to participants in Study 1.

B CONDITIONS AND APPLICATION BALANCING IN STUDY 1

Table 7. Balancing of Condition and Application order in Study 1.

P	1st	2nd	3rd
1	Designer : App 1	Optimizer : App 2	Cooperative : App 3
2	Designer : App 1	Cooperative : App 2	Optimizer : App 3
3	Cooperative : App 1	Designer : App 2	Optimizer : App 3
4	Cooperative : App 1	Optimizer : App 2	Designer : App 3
5	Optimizer : App 1	Cooperative : App 2	Designer : App 3
6	Optimizer : App 1	Designer : App 2	Cooperative : App 3
7	Designer : App 2	Optimizer : App 3	Cooperative : App 1
8	Designer : App 2	Cooperative : App 3	Optimizer : App 1
9	Cooperative : App 2	Designer : App 3	Optimizer : App 1
10	Cooperative : App 2	Optimizer : App 3	Designer : App 1
11	Optimizer : App 2	Cooperative : App 3	Designer : App 1
12	Optimizer : App 2	Designer : App 3	Cooperative : App 1
13	Designer : App 3	Optimizer : App 1	Cooperative : App 2
14	Designer : App 3	Cooperative : App 1	Optimizer : App 2
15	Cooperative : App 3	Designer : App 1	Optimizer : App 2
16	Cooperative : App 3	Optimizer : App 1	Designer : App 2
17	Optimizer : App 3	Cooperative : App 1	Designer : App 2
18	Optimizer : App 3	Designer : App 1	Cooperative : App 2