Optimization in Interface Design

Xiaojun Bi
xiaojun@cs.stonybrook.edu

11/26/2019

Some materials are from 5th Summer School of Computational Interaction

Outline

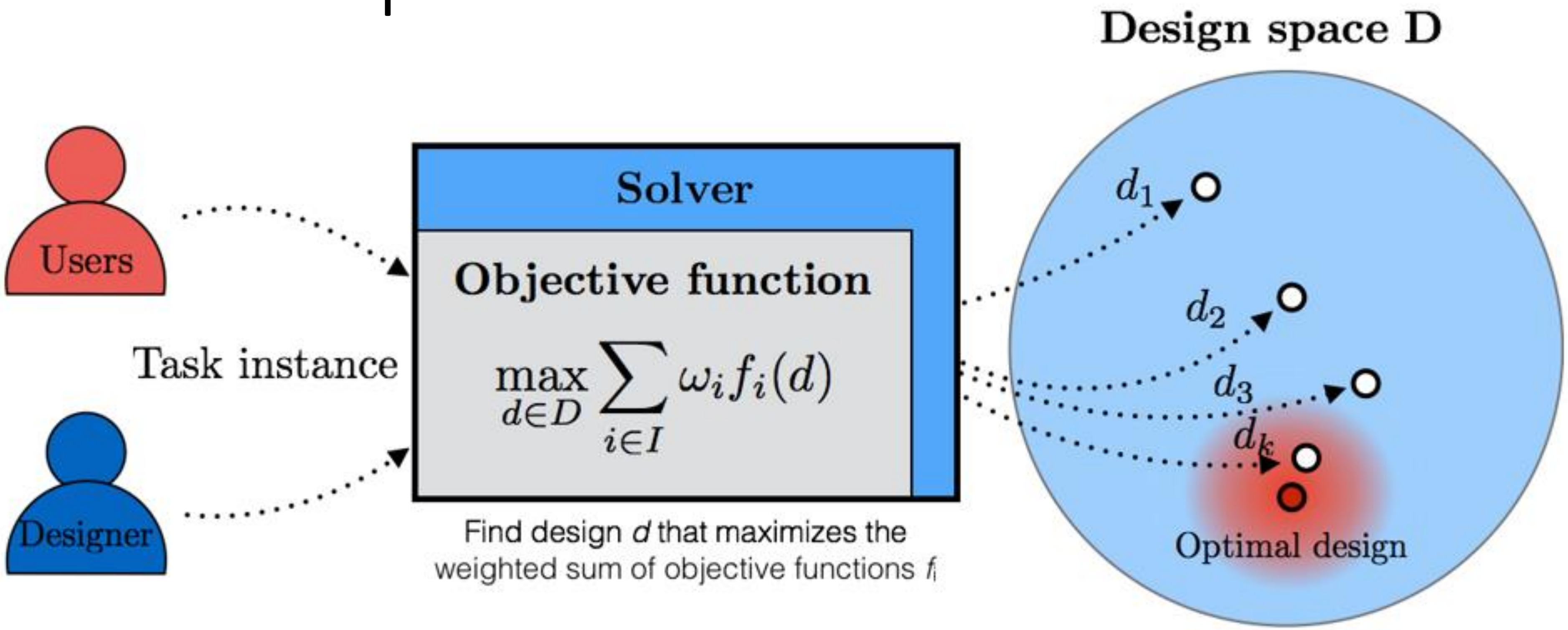
- Combinatorial Optimization
- Bayesian Optimization

Combinatorial Optimization in HCI

Combinatorial optimization refers to algorithmic search for combinations of design decisions that best meet stated design objectives.

Published applications in HCI include keyboards and panels, menu systems, graphical user interfaces, visualizations, and input methods.

Basic Concepts



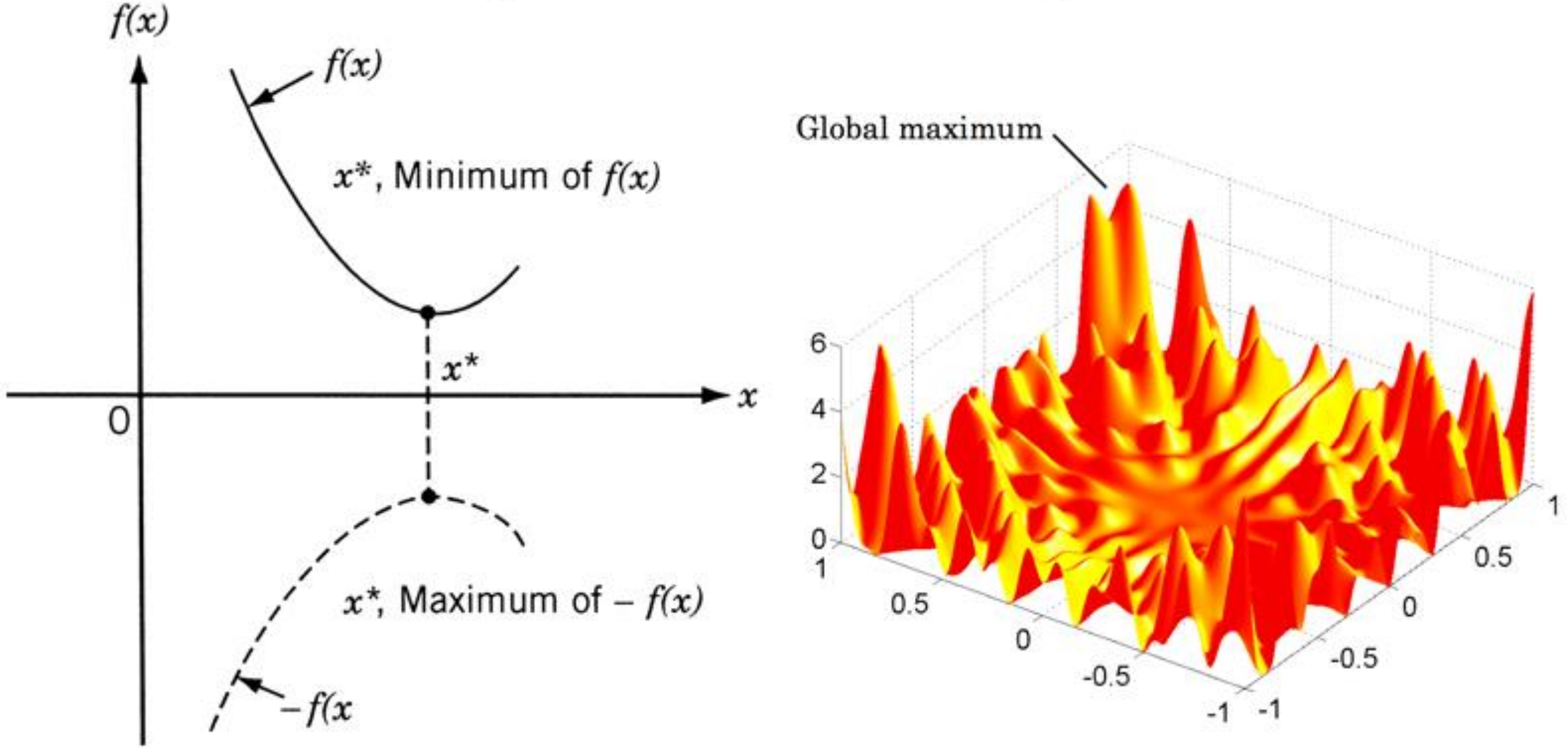
1.design space (search space; a.k.a. feasible set, candidate set): a finite set of alternative designs;

2.objective function: defines what you mean by 'good' or 'desirable design';

3.task instance: sets task-specific parameters.

Solver

Optimization landscapes



AdaM: Adapting Multi-User Interfaces for Collaborative Environments in Real-Time

Seonwook Park[1], Christoph Gebhardt[1,†], Roman Rädle[2,†], Anna Maria Feit[3], Hana Vrzakova[4], Niraj Ramesh Dayama[3], Hui-Shyong Yeo[5], Clemens N. Klokmose[2], Aaron Quigley[5], Antti Oulasvirta[3], Otmar Hilliges[1]

[1]ETH Zurich [2]Aarhus University [3]Aalto University [4]University of Eastern Finland [5]University of St Andrews

Figure 1. Given a graphical user interface (left), AdaM automatically decides which UI elements should be displayed on each device in real-time. Our optimization is designed for multi-user scenarios and considers user roles and preferences, device access restrictions and device characteristics.

ABSTRACT

Developing cross-device multi-user interfaces (UIs) is a challenging problem. There are numerous ways in which content and interactivity can be distributed. However, good solutions must consider multiple users, their roles, their preferences and access rights, as well as device capabilities. Manual and rule-based solutions are tedious to create and do not scale to larger problems nor do they adapt to dynamic changes, such as

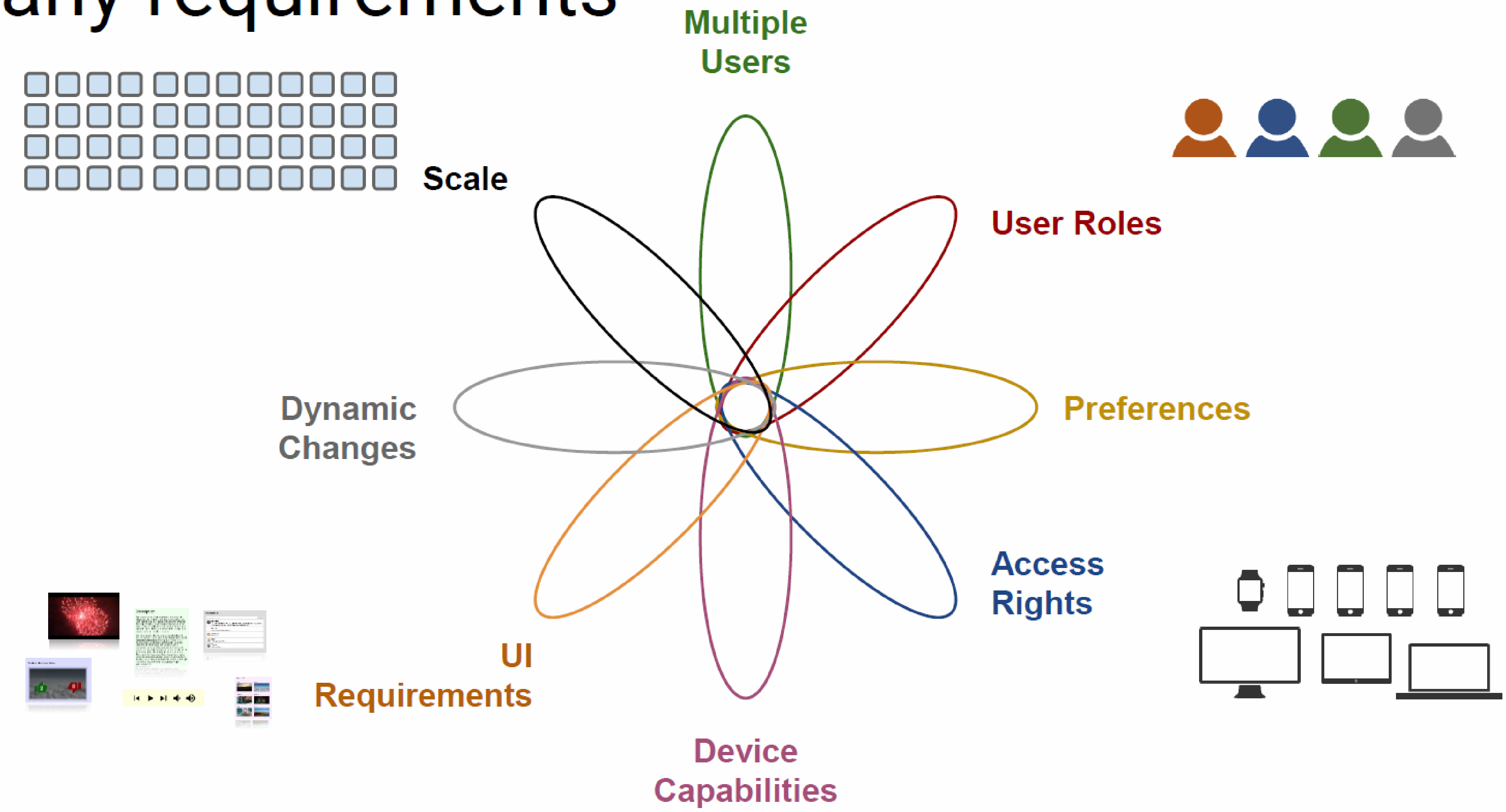
INTRODUCTION

Many users now carry not one, but several computing devices, such as laptops, smartphones or wearable devices. In addition, our environments are often populated with public and semi-public displays. In collaborative settings, such as at work or in education, many application scenarios could benefit from UIs that are distributed across available devices and potentially also across multiple users participating in a joint activity. How-


EMPLOYEE 1
EMPLOYEE 2
PRESENTER
SECRETARY
MANAGER
https://www.kpms.co.uk/wp-content/uploads/2017/03/PN-40TC1_huddle-meeting1_i522258271.jpg

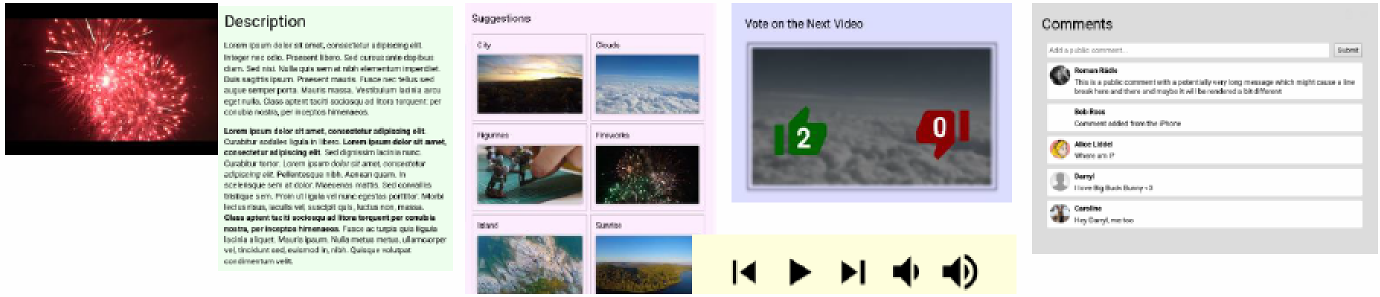
What to show on which screen?

Many requirements



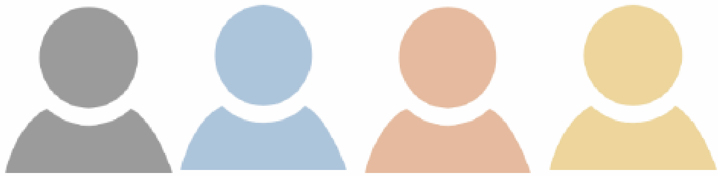
Design space

Elements
$e \in \mathcal{E}$



Devices
$d \in \mathcal{D}$

Users
$u \in \mathcal{U}$



$$x_{ed} = \begin{cases} 1, \text{if } e \text{ is assigned to } d \\ 0, \text{otherwise} \end{cases} \qquad o_{eu} = \begin{cases} 1, \text{if u has access to e} \\ 0, \text{otherwise} \end{cases} \qquad s_{ed} \in \mathbb{Z}^{+}$$

Constraints

- Device access

- Element permission

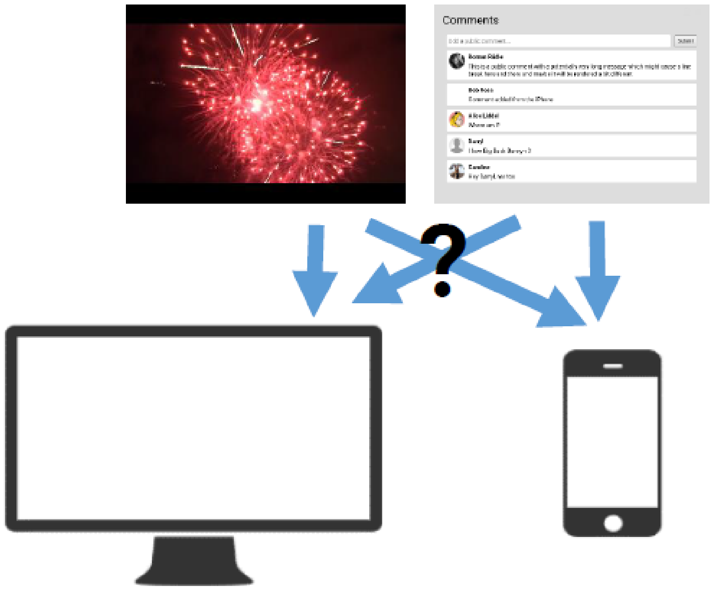
- Device capacity

Objectives

Assignment Quality (Usability)

More important elements should:

- better match element requirements and device characteristics
  (visual quality, text input, touch pointing, mouse pointing)
- be displayed larger



Objectives

Assignment Quality (Usability)

More important elements should:

- better match element requirements and device characteristics
  (visual quality, text input, touch pointing, mouse pointing)
- be displayed larger

Completeness (Usefulness)

Maximize

- the number of elements a user has access to and
- the access of users to elements they need (have permission to use)

(a) Initial configuration. It can be seen that elements respect element-device compatibilities in their assignment.

(b) Bob and Carol's preferences can both be respected. On the left, only Bob's preference of the "Suggested Videos" element is represented. On the right, Carol's preference for the "Description" element is also addressed by placing the element on the tablet shared with Bob.

(c)

Figure 8. A demonstration of our full system with optimization backend and distributed frontend. In this example, we can see 4 users and 7 devices in play with three user preferences represented. Our system quickly adapts to the changing setting with ease. (a) and (b) are explained in their own captions and (c) reflects Darryl's preference for reading comments.

Outline

- Combinatorial Optimization

- Bayesian Optimization

Problem

Find the minimum of a function $f(x)$ within some bounded domain $\mathcal{X} \subset \mathbb{R}^D$:

$$x^* = \arg\min_{x \in \mathcal{X}} f(x)$$

How would you solve this?

- Hand-tuning / trial and error,
- random search,
- grid search,
- gradient descend,
- evolutationary algorithms,
- different types of programming (linear, integer, etc.),
- ...

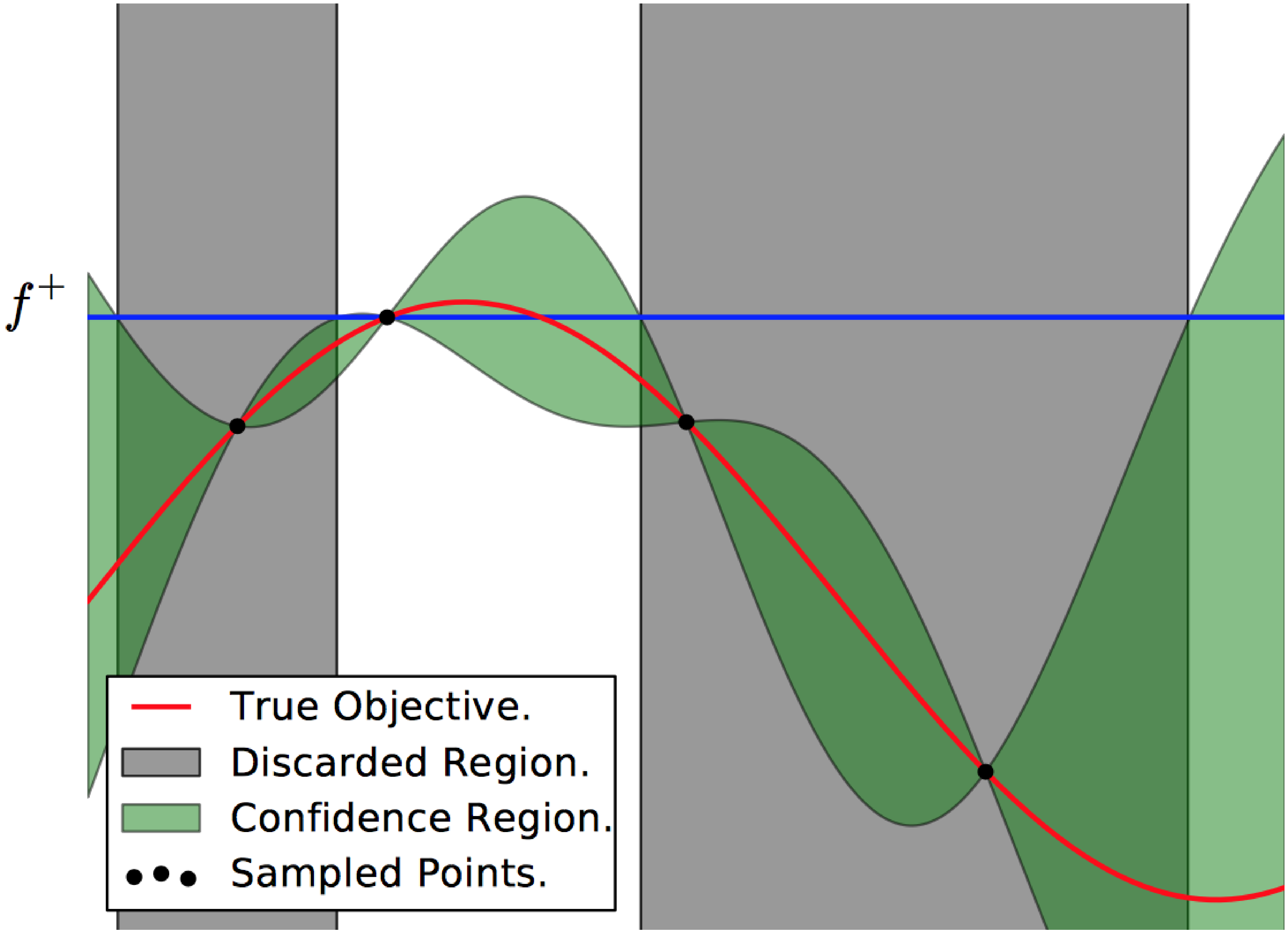
The computational problem

Problem: Find the minimum of a function $f(x)$ within some bounded domain $\mathcal{X} \subset \mathbb{R}^D$:

$$x^* = \arg\min_{x \in \mathcal{X}} f(x)$$

Challenges

- f is a black-box that we can only evaluate point-wise,
- f can be multi-modal,
- f is slow or expensive to evaluate,
- evaluations of f are noisy,
- f has no gradients available (can be used if available).



f^+
True Objective.
Discarded Region.
Confidence Region.
Sampled Points.

Application - A/B testing

But what if... f (x)

- can only be evaluated implicitly,
- with a lot of noise, and
- is costly to evaluate.

For example, optimize for click-through rate, retention time, or purchases, implicitly measuring user satisfaction, interest, or revenue of different version of a web site.

By choosing conditions wisely, Bayesian optimization can converge to a good design quicker and avoid contaminating users with potentially bad designs.

Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2010)
M. Otaduy and Z. Popovic (Editors)

A Bayesian Interactive Optimization Approach to Procedural Animation Design

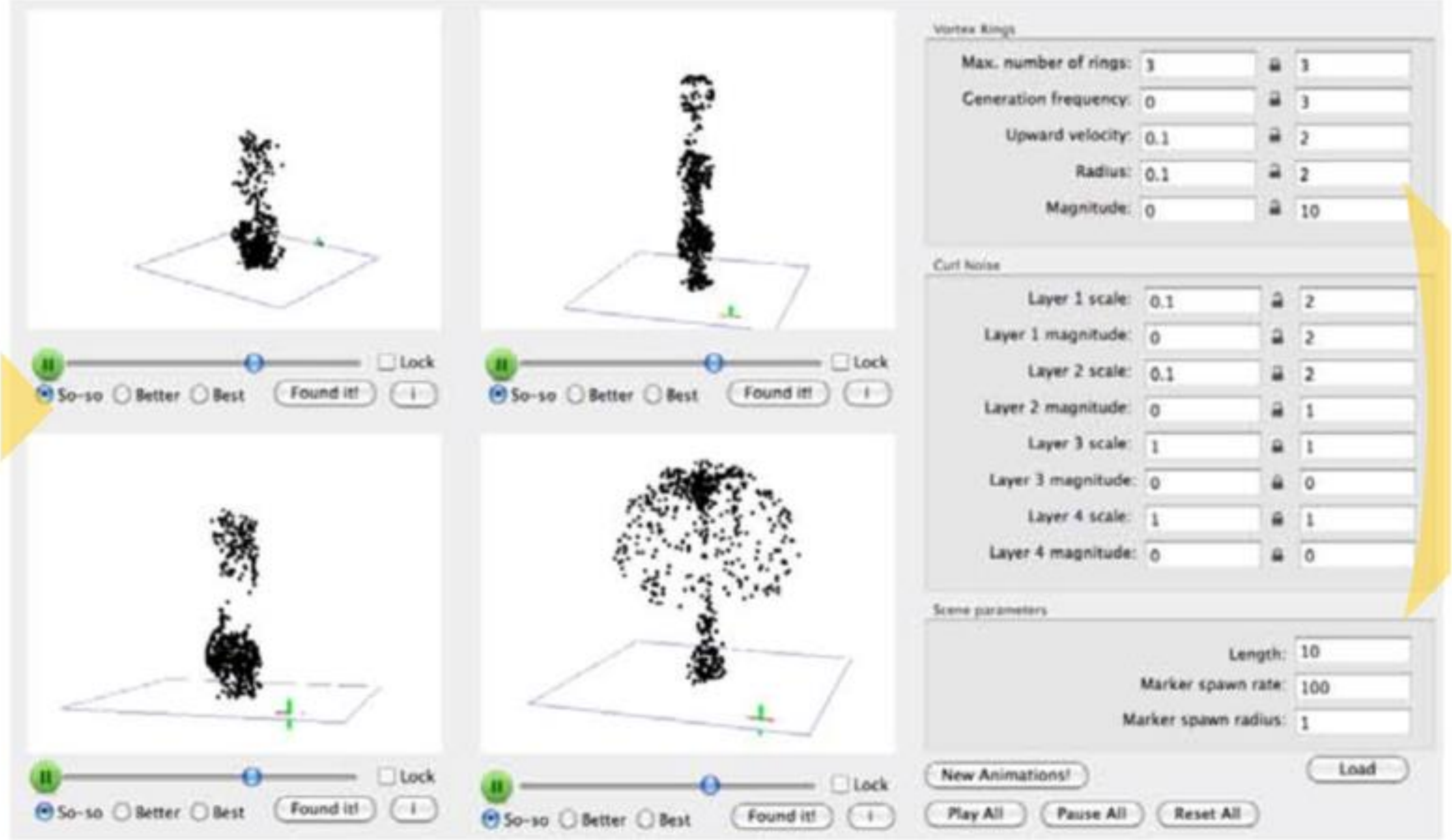
Eric Brochu Tyson Brochu Nando de Freitas

University of British Columbia

Abstract

The computer graphics and animation fields are filled with applications that require the setting of tricky parameters. In many cases, the models are complex and the parameters unintuitive for non-experts. In this paper, we present an optimization method for setting parameters of a procedural fluid animation system by showing the user examples of different parametrized animations and asking for feedback. Our method employs the Bayesian technique of bringing in "prior" belief based on previous runs of the system and/or expert knowledge, to assist users in finding good parameter settings in as few steps as possible. To do this, we introduce novel extensions to Bayesian optimization, which permit effective learning for parameter-based procedural animation applications. We show that even when users are trying to find a variety of different target animations, the system can learn and improve

User interface



Algorithm

Algorithm 1 Bayesian optimization for animation galleries

1: Let $n = 0$ and NG be the number of instances in the gallery interface, and choose an initial set of parameters, $\mathbf{x}_{1:NG}$.
2: **repeat**
3: Generate gallery of animation instances from the parameters.
4: Record k user preferences $\{\mathbf{r}_{1:k}, \mathbf{c}_{1:k}\}$ from the set $\{\mathbf{x}_{n+1:n+NG}\}$ and add to $\mathcal{D}$.
5: Compute the predictive distribution $\mu(\cdot), s^2(\cdot)$.
6: Let $n = n + NG$.
7: Compute a new set of NG parameters $\{\mathbf{x}_{n+1:n+NG}\}$ by iteratively maximizing the expected improvement function.
8: **until** Animator is satisfied