Motivation

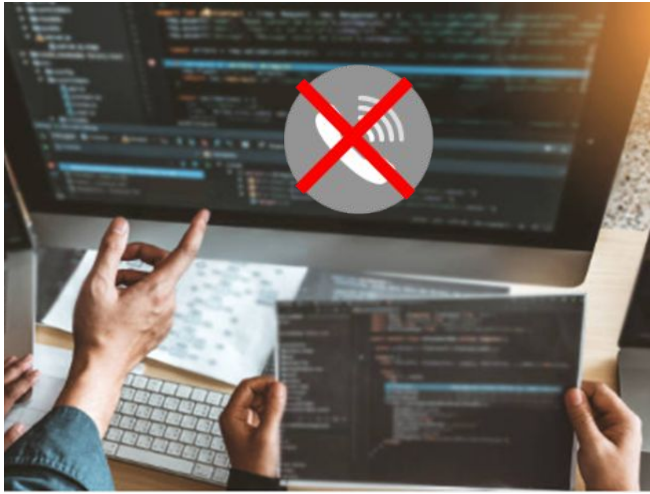
Where would you place a button to answer a call?

Where would you place a button to answer a call?

In sight, close to the hand

Avoid overlap with the environment

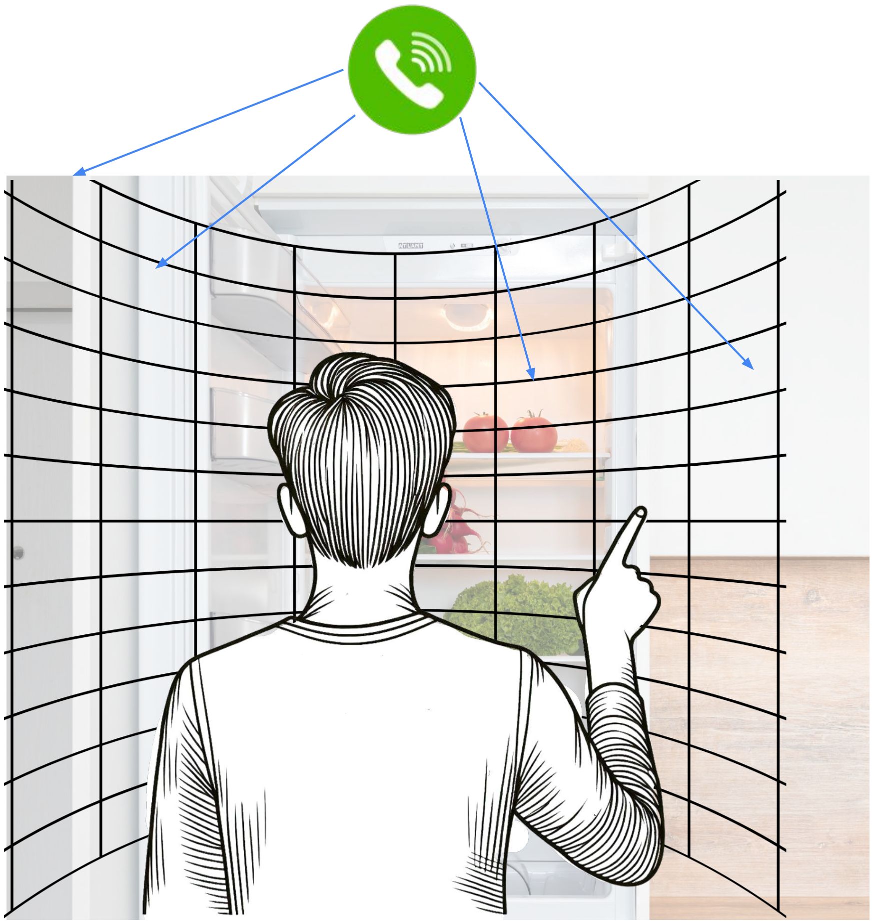
Do not interrupt user



Let's do this more generally

- Given a 2D/3D grid
- A users' position
- The 3D geometry of the environment
- and a UI element

Which criteria determine the goodness of the placement?

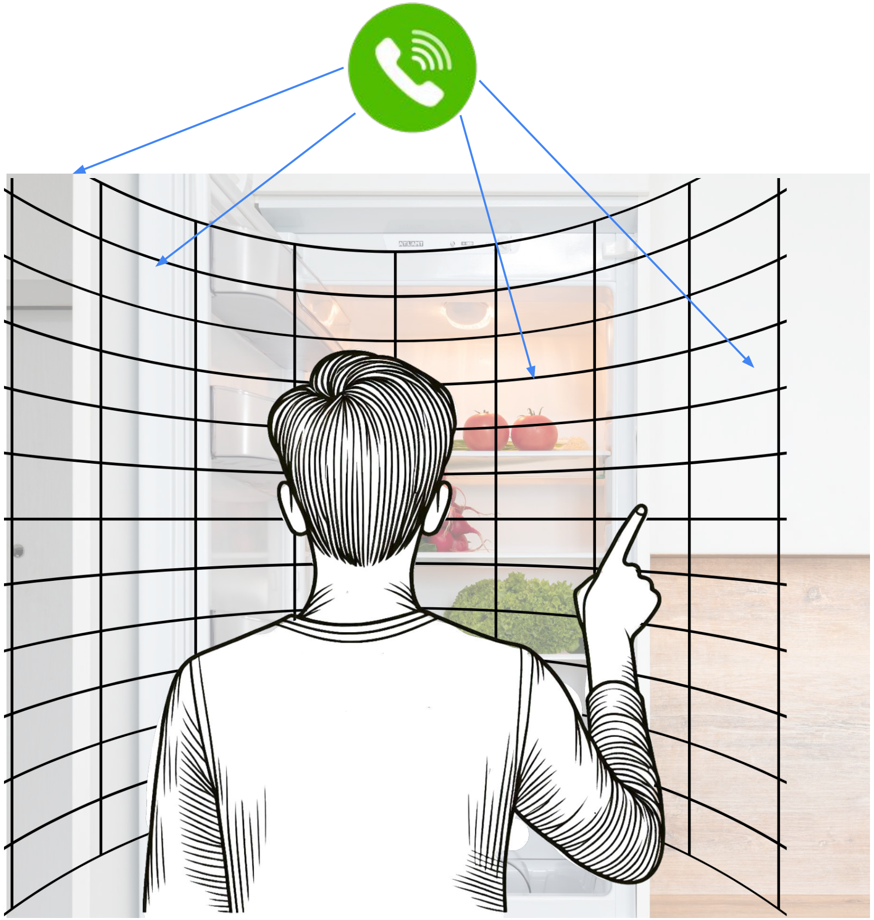


Let's do this more generally

- Given a 2D/3D grid
- And a UI element

Which criteria determine the goodness of the placement?

- Reachability
- **Performance**
- **Field of View**
- Occlusion
- Ergonomics
- …

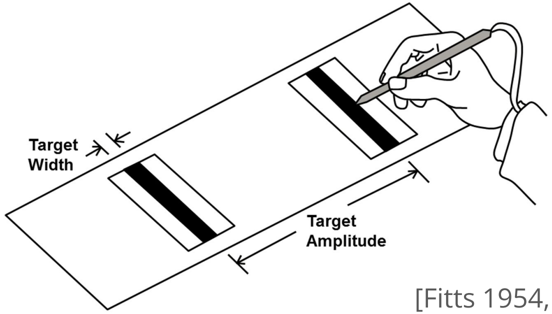


See also: Belo et al. AUIT – the Adaptive User Interfaces Toolkit for Designing XR Applications. UIST 2022. https://dl.acm.org/doi/10.1145/3526113.3545651

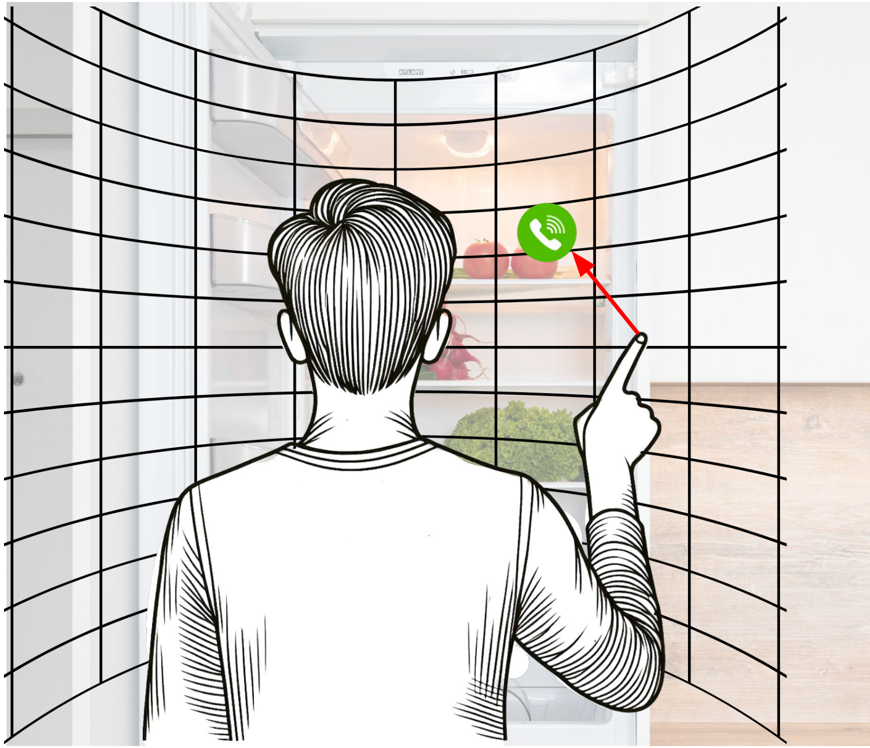
How can we quantify these two criteria?

Performance

$$MT = a + b \cdot log_2(\frac{A}{W} + 1)$$



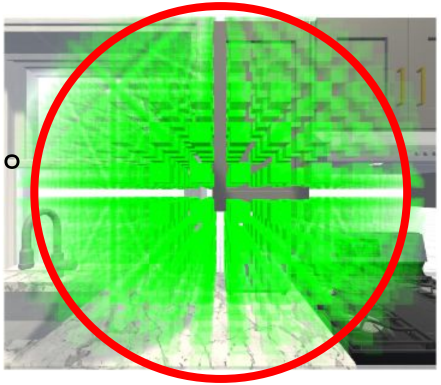
[Fitts 1954, MacKenzie 1992]

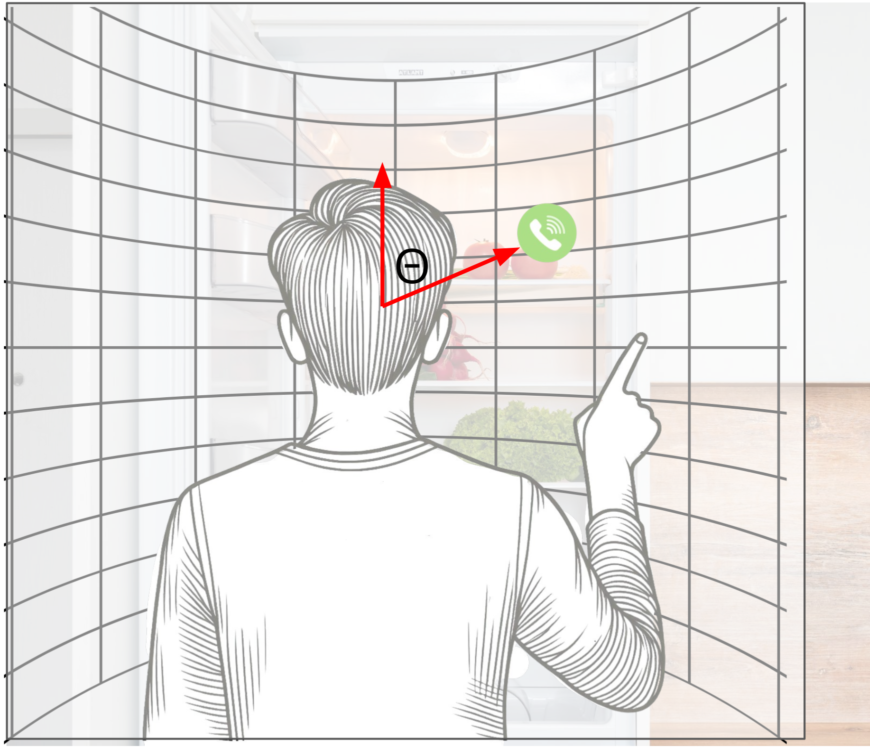


How can we quantify these two criteria?

$$c_{FoV}(\Theta) = \frac{\Theta}{\Theta_{max}}$$

Θ_{max}= 45°





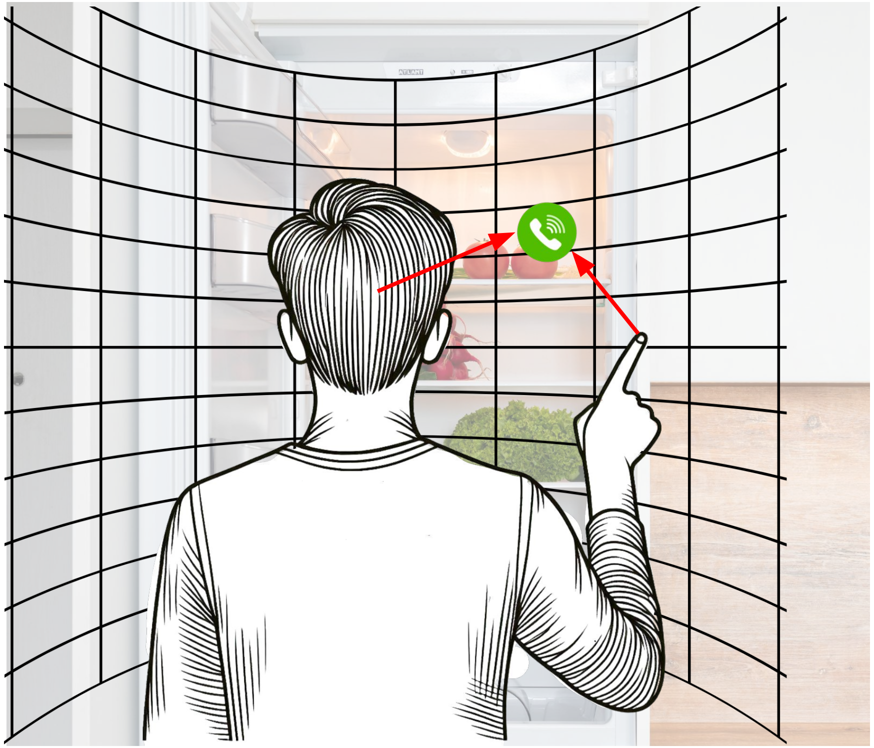
See also: Belo et al. AUIT – the Adaptive User Interfaces Toolkit for Designing XR Applications. UIST 2022. https://dl.acm.org/doi/10.1145/3526113.3545651

How good is this placement?

$$MT = a + b \cdot log_2(\frac{A}{W} + 1)$$

+

$$c_{FoV}(\Theta) = \begin{cases} 0 & \text{if } \Theta \leq \Theta_{\text{in}}, \\ \frac{\Theta - \Theta_{\text{in}}}{\Theta_{\text{out}} - \Theta_{\text{in}}} & \text{if } \Theta_{\text{in}} < \Theta \leq \Theta_{\text{out}}, \\ 1 & \text{if } \Theta > \Theta_{\text{out}}. \end{cases}$$



How good is this placement?

$$MT = a + b \cdot log_2(\frac{A}{W} + 1)$$

We can use **models** to evaluate a user interface

$$c_{FoV}(\Theta) = \begin{cases} 0 & \text{if } \Theta \leq \Theta_{\text{in}}, \\ \frac{\Theta - \Theta_{\text{in}}}{\Theta_{\text{out}} - \Theta_{\text{in}}} & \text{if } \Theta_{\text{in}} < \Theta \leq \Theta_{\text{out}}, \\ 1 & \text{if } \Theta > \Theta_{\text{out}}. \end{cases}$$

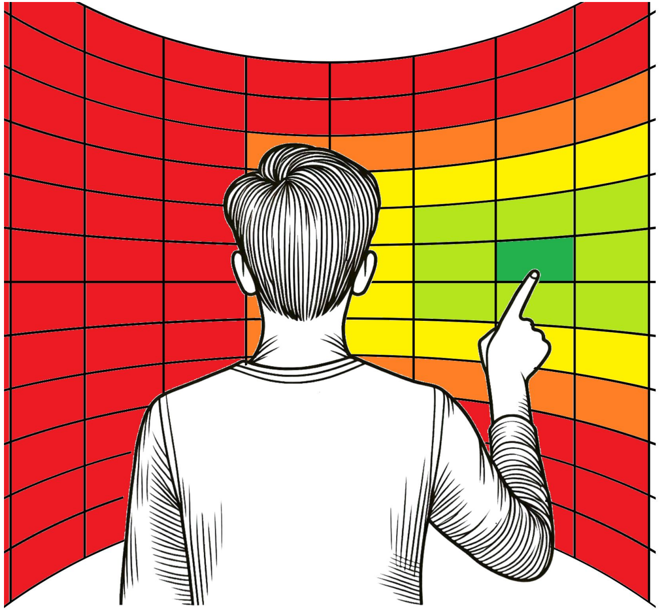
How do we find the best position?

Exhaustive search: place the element at each possible solution and compute the cost

See also: Belo et al. XRgonomics: Facilitating the Creation of Ergonomic 3D Interfaces. CHI 2021.
https://doi.org/10.1145/3411764.3445349

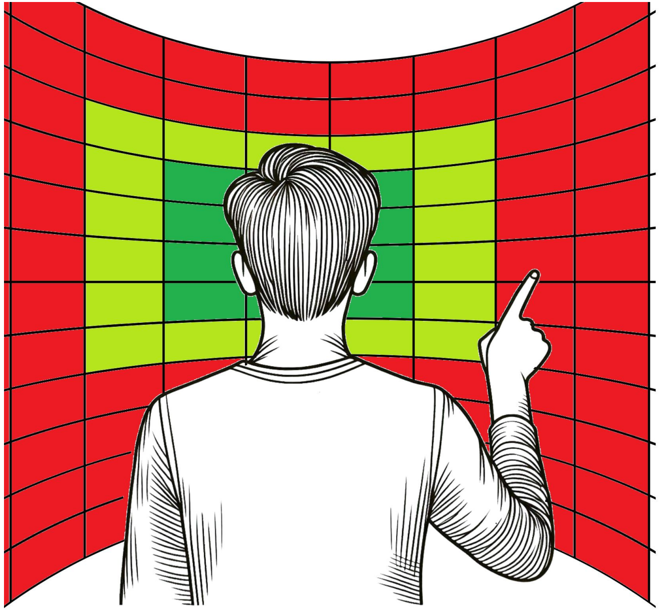
How do we find the best position?

Exhaustive search: place the element at each possible solution and compute the cost



Pointing

+

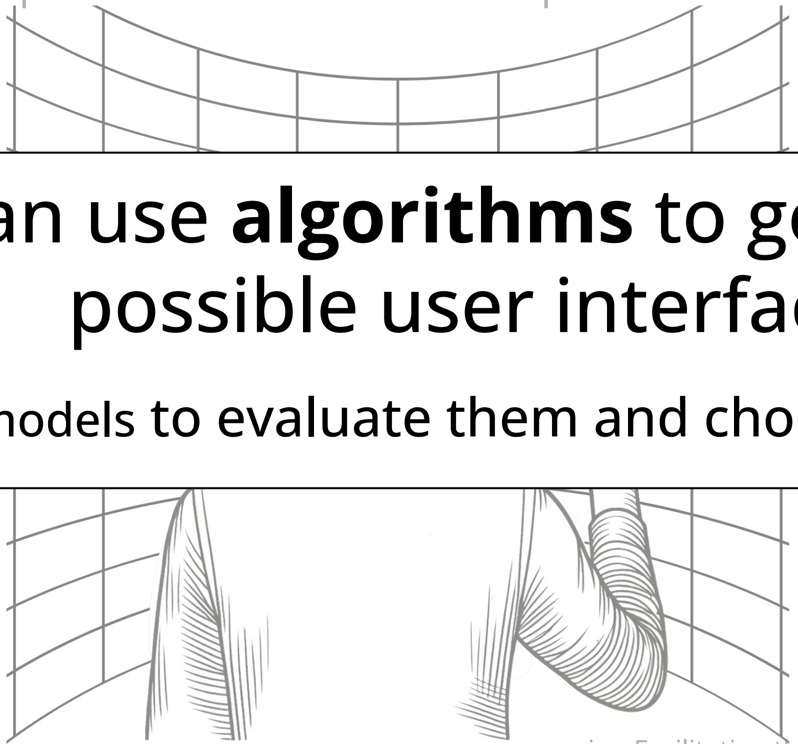


Field of view

See also: Belo et al. XRgonomics: Facilitating the Creation of Ergonomic 3D Interfaces. CHI 2021.
https://doi.org/10.1145/3411764.3445349

How do we find the best position?

Exhaustive search: place the element at each possible solution and compute the cost



We can use **algorithms** to generate all possible user interfaces

and use models to evaluate them and choose the best one

See also: Belo et al. XRgonomics: Facilitating the Creation of Ergonomic 3D Interfaces. CHI 2021. https://doi.org/10.1145/3411764.3445349

Now let's look at optimization a bit more systematically and implement this problem in code.

Switch to the notebook.

Summary and Discussion

What have you learned today?

- How to formulate optimization problems
    - Decision variables
    - Constraints
    - Objective functions
- How to solve them
    - Exhaustive search
    - Genetic Algorithms
- Multi-objective optimization
    - Weighted sum
    - Pareto frontier

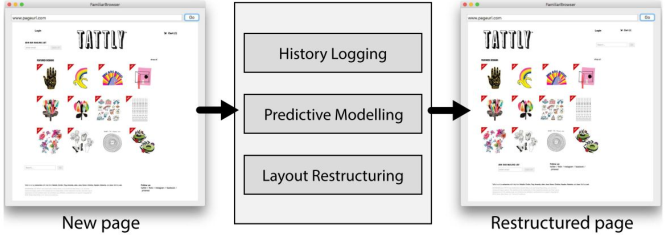
Optimization for UI Design vs. Adaptation+Personalization

Keyboard layouts, e.g. [Feit, 2021]

Multi-device, multi-user interfaces, [Park, 2018]

Interactive optimization of menus, [Bailly 2013]

Functionality selection, [Oulasvirta 2017]



Familiarisation [Todi, 2019]

Cognitive load [Lindlbauer, 2019]

Adapting XR interfaces, [Belo 2022]

Interactive adaptation with multiple objectives, [Johns 203]

Optimization for UI Design versus Adaptation

- Computational complexity
  - One-shot optimization, vs
  - continuous real-time optimization

  Affects objective function, problem formulation, choice of solver, ...

- Data source
  - static user models and simulations, offline data collection, etc. vs.
  - real-time data, e.g. for online parameter inference, about user's context, etc.

- Interaction with designers/developers versus end-users
  - Optimization as part of an iterative and user-centred design process [Feit 2021]
  - Additional considerations such as *when* the optimization is triggered, *how* the UI transitions to a new state, etc. see [Belo 2022], what is the adaptation cost for users

Optimization can be used for more

Interaction Design, e.g. gestural interaction, haptic feedback

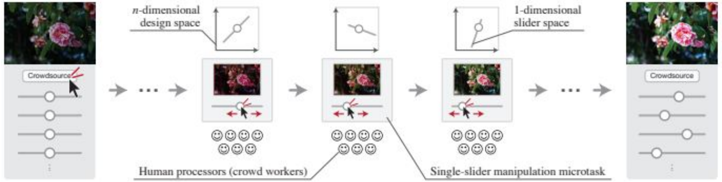
Interactive human-in-the loop optimization for photo editing, music, etc.

Optimization of data plots

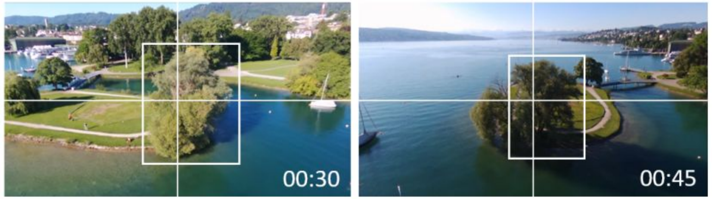
Providing design suggestions, helping in design ideation, or suggesting alternative designs

Drone videography and interactive path planning

Fabrication for personalization or compliance

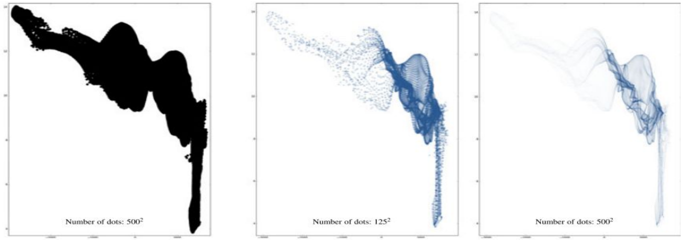


Koyama 2017



Gebhardt , 2021

Hofmann, 2023



Micallef, 2017

Some optimization methods / solvers

Method	Gradient	Solution	Variables	Implementation
Exhaustive Search	zero-order	exact	discrete	See the notebook
Branch and Bound	zero-order	exact	discrete	Python-MIP
Simulated Annealing	zero-order	approximate	cont.+discr.	Scipy
Genetic Algorithms	zero-order	approximate	cont.+discr.	Pymoo
Gradient Descent	first-order	approximate	continuous	Scikit learn
Interior Point optimization	second-order	approximate	cont.+discr.	Ipopt (python bindings)
Bayesian Optimization	zero-order	approximate	mostly cont.	BOtorch

Critical for success

- The formulation of the optimization problem
  - There are many types of standard problems (e.g. assignment problem) in operations research. Casting your problem as a standard problem gives you access to suitable algorithms
- The data and models used as objectives (more on modeling in the coming days)
- Suitable trade-offs between multiple objectives

Benefits of using optimization in HCI

- *Explicitness:* uncovers assumptions, trade-offs, design choices, …
- *Data -driven:* data and observations directly influence design and steer adaptation
- *Quality:* Can give guarantees on the goodness of designs, can cover the whole design space
- *Automation:* reduces manual effort and design time, designer can focus on creative aspects and problem formulation
- *Flexibility:* enables interfaces to optimally adapt to changing context, users, devices, preferences, etc.
- *Knowledge transfer:* Design knowledge can be transferred to other devices or contexts

Challenges in using optimization

- *Ramp-up costs:* formal definition and development of search methods can be laborious and risky
- *Hard problems:* real-world cases can be hard to model and solve, collaborate with optimization experts
- *Reliance on data and models:* defining meaningful objective functions (e.g., aesthetics, learnability) and gathering data or developing models to quantify them may require support from domain experts, cognitive scientists, machine learning experts etc.

We are always happy to collaborate. Reach out if you are interested in a specific problem

Research opportunities

- Computational design and adaptation tools for standard design problems
- "Participatory optimization": optimization tools that
  - Integrate into an iterative user-centred design process
  - That can be used by non-experts (w.r.t. optimization)
  - Carrying out diagnostics and visualizations to understand the results, trade-offs, decisions
  - Interaction techniques for end-users to participate in an adaptation process
- Models and simulations of user behavior
- Online learning and parameter inference
- Empirical evaluation and testing of adaptation methods
- ...

Further reading

More on integer programming and the assignment problem: Feit, A.(2019). Integer Programming for UI Optimization, Lecture notes from 5th Summer School on Computational Interaction, New York

An introduction to combinatorial optimization: Oulasvirta, A.(2019). Introduction to Combinatorial Optimization, Lecture notes from 5th Summer School on Computational Interaction, New York

More research papers:

Oulasvirta., A. & Karrenbauer, A. (2108). Combinatorial Optimization for Interface Design, in Computational Interaction. Oxford University Press.

Feit, A. M. (2018). Assignment Problems for Optimizing Text Input. Aalto University publication series Doctoral Dissertations.

Oulasvirta, A., Dayama, N. R., Shiripour, M., John, M., & Karrenbauer, A. (2020). Combinatorial Optimization of Graphical User Interface Designs. Proceedings of the IEEE

Lindlbauer, D., Feit, A. M., & Hilliges, O. (2019). Context-aware Online Adaptation of Mixed Reality Interfaces. In Proc UIST'19

Evangelista Belo, J. M., Lystbæk, M. N., Feit, A. M., Pfeuffer, K., Kán, P., Oulasvirta, A., & Grønbæk, K. (2022). AUIT–the Adaptive User Interfaces Toolkit for Designing XR Applications. In Proc. UIST'22