Towards Multi-Objective Optimization for UI Design

Anna Maria Feit
Aalto University
anna.feit@aalto.fi

Myroslav Bachynskyi
Max Planck Institute for Informatics
mbachyns@mpi-inf.mpg.de

Srinath Sridhar
Max Planck Institute for Informatics
ssridhar@mpi-inf.mpg.de

Abstract

In recent years computational optimization has been applied to the problem of finding good designs for user interfaces with huge design spaces. There, designers are struggling to integrate many different objectives into the design process, such as ergonomics, learnability or performance. However, most computationally designed interfaces are optimized with respect to only one objective. In this paper we argue that *multi-objective* optimization is needed to improve over manual designs. We identify 8 categories that cover design principles from UI design and usability engineering. We propose a multi-objective function in form of a linear combination of these factors and discuss benefits and pitfalls of multi-objective optimization.

Author Keywords

Optimization; multiple objectives; user interface design

Introduction

The design of user interfaces often follows a large number of design principles, taking many different factors into account: from ergonomics and performance over technological and economical constraints to aesthetic pleasure or fun. But is it feasible for a single person or group of designers to explore a huge space of different designs (*e.g.* about 10^{24} in the problem of arranging 20 menu items) and identify a single best one with respect to all the design factors?

Computational user interface design tries to support the designer in this task by making use of algorithmic optimization methods. The designer defines optimization goals and constraints by providing models about the relevant factors, such as user behavior or implementation cost. This approach has been successfully used in recent years. However, it almost exclusively concentrated on optimizing for performance [2, 11]. Only a few attempts have been made to include more than one factor.

In order to support designers we need to advance from optimizing for performance to offering multi-objective frameworks that can handle several different design factors. But what are the design factors that we should consider? To answer this question, we review design principles from user interface design to identify 8 objective categories. Related work in UI optimization can be classified in terms of these categories. We argue that multi-objective optimization of all listed criteria leads to better interface design than previous approaches when considering amortized usage. Finally, we discuss potential pitfalls and problems.

CHI'15, April 18–23, 2015, Seoul, Korea.
Workshop on Principles, Techniques and Perspectives on Optimization and HCI

Performance e.g. efficiency, responsiveness,
Human Error e.g. forgiveness, recovery
Ergonomics e.g. fatigue, muscle stress
Mental Workload e.g. predictability, simplicity
Learnability e.g. familiarity, memorability
Accessibility e.g. perceptibility, operability
Subjective experience e.g. fun, aesthetics
Technological Error e.g. recognition accuracy

Table 1: 8 categories that cover principles and objectives from user interface design and usability engineering. A carefully designed user interface should consider many of them.

What should we optimize for?

Reviewing guidelines from user interface design and usability engineering [7], as well as related work in the design and optimization of interactive systems, we identified 8 broad categories of design objectives, summarized in Table 1.

Performance refers to criteria such as efficiency of actions or responsiveness of the system. Many design problems center around this criterion. For example, keyboard layouts are optimized for shorter hand-travel distance or more hand alternation [11]. Based on models of visual search and selection time, performance is also optimized in menu design [2].

Human error during the operation of a user interface can have serious consequences such as accidents and death. A good design should be able to prevent serious mistakes and easily recover from unavoidable errors [7]. Moreover, performance and user experience can be improved if the system can account for easy flaws. For example, in virtual keyboard design, a mixture of touch and language models [8] is used to improve the typing accuracy.

Ergonomics represents total physiological cost of interacting with an interface, including stress, injury probability, energy expenditure and fatigue. Thus far, only few papers have considered optimization of physical ergonomics factors. Virtual keyboard optimization work has considered multiple ergonomics criteria including fatigue [13]. Size and location of a keyboard in mid-air and a menu on public displays was optimized according to energy expenditure, accuracy and performance [1].

Mental workload is determined by the consistency of a user interface, its predictability and simplicity [7]. Workload may be correlated with performance. For example, semantic relationships in menus reduce the complexity of the visual search and decrease search time [9]. The right mental workload is important. Consistently high effort leads to stress and anxiety, while a low workload may cause boredom. In both cases performance may be affected [14].

Learnability refers to how easily a novice user can perform a task, *i.e.* obviousness of the UI, but also how memorable it is after some time of absence [7]. An interface that is hard to learn will not be adapted by users and may fail completely. Keyboards have been jointly optimized for performance and familiarity of the layout [15]. The PianoText project showed how the transfer of expertise from another domain can shorten the learning time of a new input method [5].

Accessibility denotes the design of a system to be usable by different groups of people, which not only concentrates on those with special needs. Characteristics include the perceptibility and operability of the UI. SUPPLE [6] is an example that considers accessibility in interface generation.

User satisfaction is a key factor for the success of every interface. Design guidelines recommend an interface to be aesthetic and personalizable [7]. Those factors are hard to quantify and highly subjective. To our knowledge no effort has been made so far to computationally optimize them, but increasing interest can be seen towards gamification of everyday tasks and the integration of fun and enjoyment into the design process [3].

Technological error is not part of the traditional user-centered design. However, it often implicitly underlies the design process. For example, empirical observations show [12] that the Leap Motion sensor[1] is capable of

[1] `https://www.leapmotion.com/`

$$\begin{aligned} f = & w_P \times Performance + \\ & w_{HE} \times HumanError + \\ & w_E \times Ergonomics + \\ & w_{MW} \times MentalWorkload + \\ & w_L \times Learnability + \\ & w_A \times Accessibility + \\ & w_{SE} \times Subj.Experience + \\ & w_{TE} \times Tech.Error \end{aligned}$$

Equation 1: Multi-objective function combining individual functions of each category, where w_i are weights (importance) of each category defined by designer. In presence of competing categories different optimal solutions will exist for each distinct weight assignment. These solutions are points on Pareto Front illustrated in Figure 1.

tracking only a certain subset of motions while other gestures result in undefined behavior. Such limitations will have to be taken into account when optimizing gestures. For instance, if a tracker is unable to track a *thumbs up* gesture, then the optimizer should not include this gesture in the solution.

This categorization is certainly not complete. However, we propose it to emphasize the large variability of factors influencing the design of user interfaces. Recently, attempts have been made to optimize text entry methods for 2 or 3 factors simultaneously. This work can be characterized in terms of the proposed categories. For example, Dunlop and Levine [4] optimized a touchscreen keyboard for speed (performance), familiarity (learnability) and improved spell checking (human error). Sridhar *et al.* [12] optimized multi-finger gestures for mid-air text entry with respect to performance, anatomical constraints (ergonomics), learnability (cognitive effort) and mnemonics (learnability). While these are first attempts to optimize for a small subset of the proposed objectives, we argue that future research should develop new optimization methods for UI design that include many of the categories.

Benefits of Multi-Objective Optimization

The above mentioned categories can be linearly combined into a single objective function, as shown in Equation 1. We can not assume independence between categories, their cooperation or linearity. Thus, a user interface optimized for multiple possibly competing criteria is not optimal with respect to each individual criterion. However, we claim that in the long term interfaces created with multi-objective optimization of all listed categories would outperform the interfaces created by single-objective optimization or designed manually. Here are three arguments supporting our claim:

1. The designer treats importance of the categories by assigning corresponding weights in the objective function. This process is clearer than subjective guessing or a designer's heuristics applied to the non-optimized criteria, and leads to quantitatively better solution.

2. When optimizing for a single category, small improvements in it could lead to exponential losses in competing categories, for example degradation of learnability of an interface when optimizing for performance only. Multi-objective optimization avoids this problem if weights of all criteria are positive numbers.

3. Single-objective optimization creates interfaces that are optimal with respect to one category and ignores all related costs in other categories. Such interfaces can only serve a small target group. For example, high typing speed of stenographers is reached at the high cost in learnability, cognitive effort and user satisfaction. Instead, multi-objective optimization identifies an interface balanced in all categories. Such interfaces can satisfy more application scenarios, which could lead to broader adoption.

Optimization: Challenges and Pitfalls

We have discussed how UI design can be formulated as an optimization problem. But finding solutions to the optimization problem may be equally challenging. In this last section we discuss some considerations, challenges, and pitfalls in finding solutions.

The first challenge is the choice of the optimization algorithm. Linear objective functions with linear constraints can be solved by well known methods (*e.g.* simplex and interior point methods) in polynomial time. However, typical UI problems like keyboard layout optimization are known to be quadratic [10]. For such problems, a global optimum cannot be found in

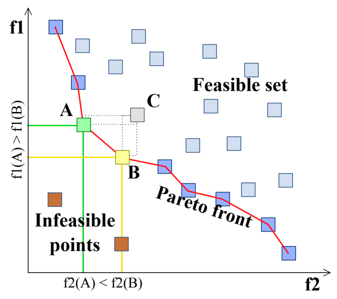

Figure 1: Pareto front for optimization of two competing objectives: f1 and f2. Blue points belong to feasible set. Pareto front is subset of feasible set and contains all points which are not dominated by any other point with respect to all objective functions (here f1 and f2). Points belonging to Pareto Front are equivalent among themselves, and each of them is optimal solution depending on the weight distribution between f1 and f2 in objective function.

polynomial time but approximate algorithms must be used to find locally optimal solutions [10].

A potential pitfall in optimizing for multiple objectives is when they compete with one another. For instance, when optimizing the position of a command in a menu, a *familiarity* objective might prefer to have the *Save* command as the first item because users expect it there. However, a *frequency* objective may find that the *Save* command is the least frequently used and thus favor placing it at the end. The designer might find that the *Save* command is actually placed somewhere in the middle as a result of these two competing objectives. Such subtle effects may lead to sub-optimal menus when compared to a manual designs.

Finally, the choice of weights for different terms in the objective function affects solutions. The standard, expensive solution to assigning weights is to use grid search *i.e.* enumerate all possible weight combinations and perform optimization over them to find the combination that produces the best outcome. However, this might be prohibitively expensive when the solution space is large (*e.g.* keyboard layout optimization). An alternative approach is to restrict the possible range of the weights, say, to $[0, 100]$. Subsequently, each weight can be assigned as a 'percentage' score. For example, a menu can be optimized for 50% familiarity and 50% speed. This approach was adopted for mid-air text entry by [12]. In spite of these solutions, choosing weights remains a hard problem.

References

[1] Bachynskyi, M., Palmas, G., Oulasvirta, A., and Weinkauf, T. Informing the design of novel input methods with muscle coactivation clustering. *ACM Trans. Comput.-Hum. Interact. 21*, 6 (Jan. 2015), 30:1–30:25.

[2] Bailly, G., Oulasvirta, A., Kötzing, T., and Hoppe, S. Menuoptimizer: Interactive optimization of menu systems. In *Proc. UIST*, UIST '13, ACM (New York, NY, USA, 2013), 331–342.

[3] Blythe, M. A., Overbeeke, K., Monk, A. F., and Wright, P. C. *Funology: from usability to enjoyment*, vol. 3. Springer, 2004.

[4] Dunlop, M., and Levine, J. Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In *Proc. CHI*, ACM (2012), 2669–2678.

[5] Feit, A. M., and Oulasvirta, A. Pianotext: Redesigning the piano keyboard for text entry. In *Proc. DIS*, DIS '14, ACM (New York, NY, USA, 2014), 1045–1054.

[6] Gajos, K., and Weld, D. S. Supple: automatically generating user interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces*, ACM (2004), 93–100.

[7] Galitz, W. O. *The essential guide to user interface design: an introduction to GUI design principles and techniques.* John Wiley & Sons, 2007.

[8] Goodman, J., Venolia, G., Steury, K., and Parker, C. Language modeling for soft keyboards. In *Proc. IUI*, ACM (2002), 194–195.

[9] Halverson, T., and Hornof, A. J. The effects of semantic grouping on visual search. In *Proc. CHI EA*, ACM (2008), 3471–3476.

[10] Karrenbauer, A., and Oulasvirta, A. Improvements to keyboard optimization with integer programming. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, ACM (2014), 621–626.

[11] Oulasvirta, A., Reichel, A., Li, W., Zhang, Y., Bachynskyi, M., Vertanen, K., and Kristensson, P. O. Improving two-thumb text entry on touchscreen devices. In *Proc. CHI*, ACM (2013), 2765–2774.

[12] Sridhar, S., Feit, A. M., Theobalt, C., and Oulasvirta, A. Investigating the dexterity of multi-finger input for mid-air text entry. In *Proc. of CHI*, ACM (2015).

[13] Wagner, M. O., Yannou, B., Kehl, S., Feillet, D., and Eggers, J. Ergonomic modelling and optimization of the keyboard arrangement with an ant colony algorithm. *Journal of Engineering Design 14*, 2 (2003), 187–208.

[14] Yerkes, R. M., and Dodson, J. D. The relation of strength of stimulus to rapidity of habit-formation. *Journal of comparative neurology and psychology 18*, 5 (1908), 459–482.

[15] Zhai, S., and Smith, B. A. Alphabetically biased virtual keyboards are easier to use: layout does matter. In *Proc. CHI*, ACM (2001), 321–322.