Improvements to Keyboard Optimization with Integer Programming

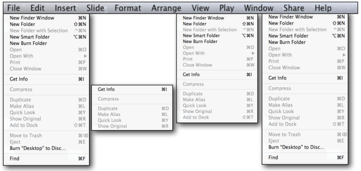
Andreas Karrenbauer[1] Antti Oulasvirta[1,2]

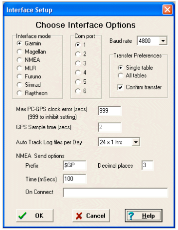
[1]Max Planck Center for Visual Computing and Communication, Saarbrücken, Germany

[2]Aalto University, Helsinki, Finland

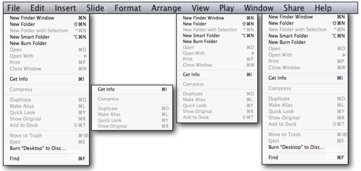
ACM UIST 2014

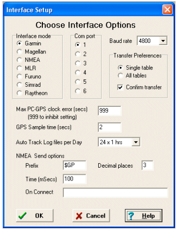
User Interface Optimization





User Interface Optimization

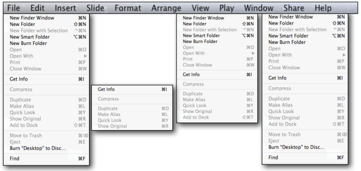


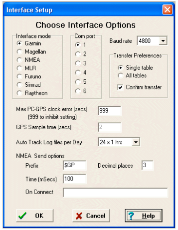


Brute Force

- Explicit enumeration
- Nonpractical for UI
  (immense design spaces $> 10^{26}$)
- No guarantees when stopped prematurely

User Interface Optimization





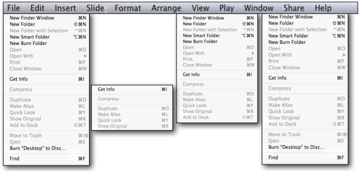
Brute Force

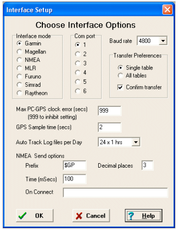
- Explicit enumeration
- Nonpractical for UI (immense design spaces $> 10^{26}$)
- No guarantees when stopped prematurely

Random Search Heuristics

- Random Enumeration
- Practical
- But by chance and no guarantees

User Interface Optimization





Brute Force

- Explicit enumeration
- Nonpractical for UI
  (immense design spaces $> 10^{26}$)
- No guarantees when stopped prematurely

?

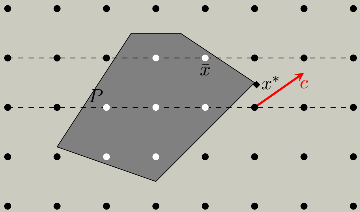
Random Search Heuristics

- Random Enumeration
- Practical
- But by chance and no guarantees

Integer Programming

Given a set of constraints, decide whether an integer point satisfies all of them.

Benefits

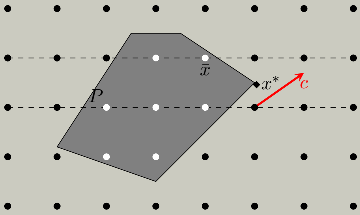


Integer Programming

Given a set of constraints, decide whether an integer point satisfies all of them.

Benefits

1. Great modeling power
2. Efficient general purpose solvers

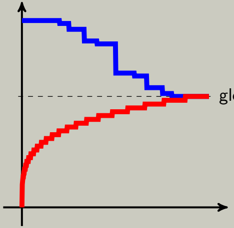


Integer Programming

Given a set of constraints, decide whether an integer point satisfies all of them.

Benefits

1. Great modeling power
2. Efficient general purpose solvers
3. Exact Methods finding the global optimum in finite time

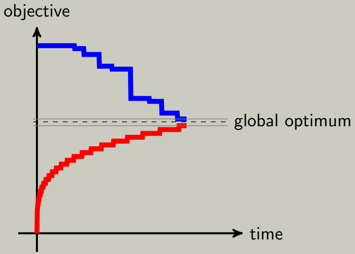


Integer Programming

Given a set of constraints, decide whether an integer point satisfies all of them.

Benefits

1. Great modeling power
2. Efficient general purpose solvers
3. Exact Methods finding the global optimum in finite time
4. Rigorous bounds for the optimum
   even when interrupted prematurely

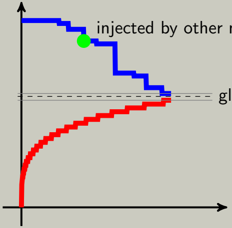


Integer Programming

Given a set of constraints, decide whether an integer point satisfies all of them.

Benefits

1. Great modeling power
2. Efficient general purpose solvers
3. Exact Methods finding the global optimum in finite time
4. Rigorous bounds for the optimum
   even when interrupted prematurely
5. Complementary to other methods (e.g., Simulated Annealing)

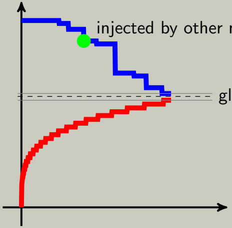


Integer Programming

Given a set of constraints, decide whether an integer point satisfies all of them.

Benefits

1. Great modeling power
2. Efficient general purpose solvers
3. Exact Methods finding the global optimum in finite time
4. Rigorous bounds for the optimum
   even when interrupted prematurely
5. Complementary to other methods (e.g., Simulated Annealing)
6. Practical



Overview of the Approach

Steps

1. Definition of constraints and objective
2. Formalization with decision variables
3. Integer Programming Model
4. Refinement by reformulations and linearizations
5. IP-Solver

The Letter Assignment Problem – Constraints and Objective

Constraints: find an assignment (1-to-1 correspondence)

$$\mathcal{L} = \left\{ \begin{matrix} ABCDEFGHI \\ JKLMNOPQR \\ STUVWXYZ \end{matrix} \right\}$$

$$f : \mathcal{L} \rightarrow \mathcal{K}$$

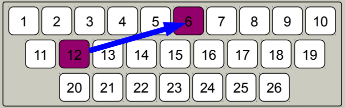
$$= \mathcal{K}$$

The Letter Assignment Problem – Constraints and Objective

Constraints: find an assignment (1-to-1 correspondence)

$$\mathcal{L} = \left\{ \begin{matrix} ABCDEFGHI \\ JKLMNOPQR \\ {\color{red}S}TUVWX{\color{red}Y}Z \end{matrix} \right\} \qquad f : \mathcal{L} \to \mathcal{K}$$

${\color{red}p_{SY}}$



$= \mathcal{K}$

Objective: maximize typing speed, i.e., minimize the expected inter-key intervals

$$\sum_{k,\ell \in \mathcal{L}} {\color{red}p_{k,\ell}} \cdot {\color{blue}t_{f(k),f(\ell)}}$$

bigram probabilities $p_{k,\ell}$

inter-key intervals $t_{i,j}$

The Letter Assignment Problem – Decision Variables

Model assignment by binary decision variables

$$x_{\ell j} = \begin{cases} 1 & \text{if letter } \ell \text{ is assigned to keyslot } j\text{, i.e., } f(\ell) = j \\ 0 & \text{otherwise} \end{cases}$$

The Letter Assignment Problem – Decision Variables

Model assignment by binary decision variables

$$x_{\ell j} = \begin{cases} 1 & \text{if letter } \ell \text{ is assigned to keyslot } j \text{, i.e., } f(\ell) = j \\ 0 & \text{otherwise} \end{cases}$$

Constraints

- Each letter ℓ is assigned to exactly one keyslot $\rightsquigarrow$ $\forall \ell \in \mathcal{L} : \sum_{j \in \mathcal{K}} x_{\ell,j} = 1$
- Each keyslot j contains exactly one letter $\rightsquigarrow$ $\forall j \in \mathcal{K} : \sum_{\ell \in \mathcal{L}} x_{\ell,j} = 1$

The Letter Assignment Problem – Decision Variables

Model assignment by binary decision variables

$$x_{\ell j} = \begin{cases} 1 & \text{if letter } \ell \text{ is assigned to keyslot } j\text{, i.e., } f(\ell) = j \\ 0 & \text{otherwise} \end{cases}$$

Constraints

- Each letter ℓ is assigned to exactly one keyslot $\rightsquigarrow$ $\forall \ell \in \mathcal{L} : \sum_{j \in \mathcal{K}} x_{\ell,j} = 1$
- Each keyslot j contains exactly one letter $\rightsquigarrow$ $\forall j \in \mathcal{K} : \sum_{\ell \in \mathcal{L}} x_{\ell,j} = 1$

Objective

expected inter-key interval $\sum_{k,\ell \in \mathcal{L}} p_{k,\ell} \cdot t_{f(k),f(\ell)}$ $\rightsquigarrow$ $\sum_{i,j \in \mathcal{K}} \sum_{k,\ell \in \mathcal{L}} p_{k,\ell} \cdot t_{i,j} \cdot x_{k,i} \cdot x_{\ell,j}$

The Letter Assignment Problem – IP Model

Quadratic Assignment Problem

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{\ell=1}^{n} p_{k\ell} t_{ij} x_{ki} x_{\ell j}$$

$$\text{s.t.} \sum_{\ell=1}^{n} x_{\ell j} = 1 \quad \forall j \in \{1,\dots,n\}$$

$$\sum_{j=1}^{n} x_{\ell j} = 1 \quad \forall \ell \in \{1,\dots,n\}$$

$$x_{\ell j} \in \{0,1\} \quad \forall \ell, j \in \{1,\dots,n\}$$

The Letter Assignment Problem – IP Model $\rightarrow$ IP-Solver

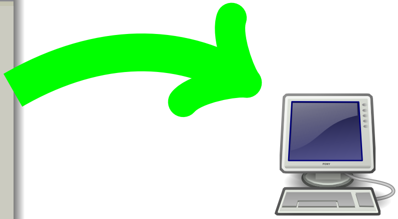
Quadratic Assignment Problem

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{\ell=1}^{n} p_{k\ell} t_{ij} x_{ki} x_{\ell j}$$

$$\text{s.t.} \sum_{\ell=1}^{n} x_{\ell j} = 1 \quad \forall j \in \{1,\ldots,n\}$$

$$\sum_{j=1}^{n} x_{\ell j} = 1 \quad \forall \ell \in \{1,\ldots,n\}$$

$$x_{\ell j} \in \{0,1\} \quad \forall \ell, j \in \{1,\ldots,n\}$$



The Letter Assignment Problem – IP Model $\leftrightarrow$ IP-Solver

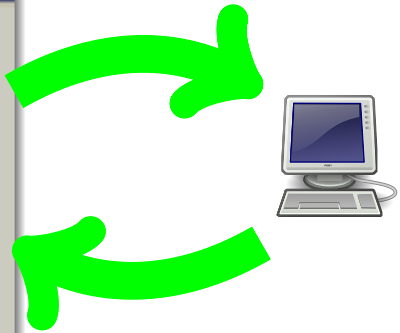
Quadratic Assignment Problem

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{\ell=1}^{n} p_{k\ell} t_{ij} x_{ki} x_{\ell j}$$

$$\text{s.t.} \sum_{\ell=1}^{n} x_{\ell j} = 1 \quad \forall j \in \{1, \ldots, n\}$$

$$\sum_{j=1}^{n} x_{\ell j} = 1 \quad \forall \ell \in \{1, \ldots, n\}$$

$$x_{\ell j} \in \{0, 1\} \quad \forall \ell, j \in \{1, \ldots, n\}$$



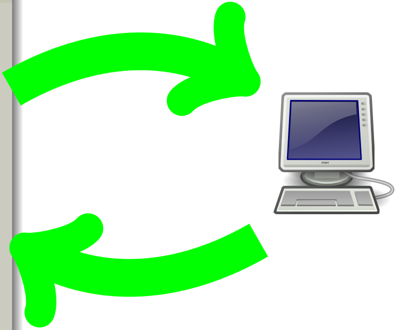
The Letter Assignment Problem – IP Models $\leftrightarrow$ IP-Solver

Reformulations

$$
\begin{aligned}
\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{\ell=1}^{n} & p_{k\ell} t_{ij} x_{ki} x_{\ell j} \\
\text{s.t.} \sum_{\ell=1}^{n} x_{\ell j} &= 1 \quad \forall j \in \{1,\dots,n\} \\
\sum_{j=1}^{n} x_{\ell j} &= 1 \quad \forall \ell \in \{1,\dots,n\} \\
x_{\ell j} &\in \{0,1\} \quad \forall \ell, j \in \{1,\dots,n\}
\end{aligned}
$$

$$
\begin{aligned}
\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{\ell=1}^{n} & p_{k\ell} t_{ij} y_{ijk\ell} \\
\text{s.t.} \sum_{\ell=1}^{n} x_{\ell j} &= 1 \quad \forall j \in \{1,\dots,n\} \\
\sum_{j=1}^{n} x_{\ell j} &= 1 \quad \forall \ell \in \{1,\dots,n\} \\
x_{\ell j} &\in \{0,1\} \quad \forall \ell, j \in \{1,\dots,n\} \\
\sum_{k=1}^{n} y_{ijk\ell} &= x_{\ell j} \quad \forall i, j, \ell \in \{1,\dots,n\} \\
\sum_{i=1}^{n} y_{ijk\ell} &= x_{\ell j} \quad \forall j, k, \ell \in \{1,\dots,n\} \\
0 \leq y_{ijk\ell} &= y_{ji\ell k} \quad \forall i, j, k, \ell \in \{1,\dots,n\}
\end{aligned}
$$

$$
\begin{aligned}
\min \sum_{\ell=1}^{n}\sum_{j=1}^{n} & w_{ij} \\
\text{s.t.} \sum_{\ell=1}^{n} x_{\ell j} &= 1 \quad \forall j \in \{1,\dots,n\} \\
\sum_{j=1}^{n} x_{\ell j} &= 1 \quad \forall \ell \in \{1,\dots,n\} \\
x_{\ell j} &\in \{0,1\} \quad \forall \ell, j \in \{1,\dots,n\} \\
w_{\ell j} &= x_{\ell j} \sum_{k=1}^{n}\sum_{i=1}^{n} p_{k\ell} t_{ij} x_{ki} \quad \forall \ell, j \in \{1,\dots,n\} \\
w_{\ell j} &\geq 0 \quad \forall \ell, j \in \{1,\dots,n\} \\
w_{\ell j} &\geq \sum_{k=1}^{n}\sum_{i=1}^{n} p_{k\ell} t_{ij} x_{ki} - a_{ij} \cdot (1 - x_{\ell j}) \quad \forall \ell, j \in \{1,\dots,n\}
\end{aligned}
$$



The Letter Assignment Problem – IP Models $\leftrightarrow$ IP-Solver

Reformulations

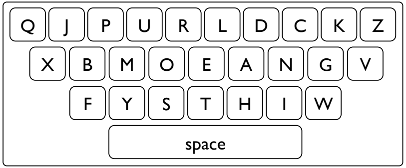
$$
\begin{aligned}
\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{\ell=1}^{n} & p_{k\ell} t_{ij} x_{ki} x_{\ell j} \\
\text{s.t.} \sum_{\ell=1}^{n} x_{\ell j} &= 1 \quad \forall j \in \{1,\dots,n\} \\
\sum_{j=1}^{n} x_{\ell j} &= 1 \quad \forall \ell \in \{1,\dots,n\} \\
x_{\ell j} &\in \{0,1\} \quad \forall \ell, j \in \{1,\dots,n\}
\end{aligned}
$$

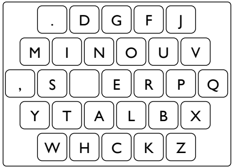
$$
\begin{aligned}
\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{\ell=1}^{n} & p_{k\ell} t_{ij} y_{ijk\ell} \\
\text{s.t.} \sum_{\ell=1}^{n} x_{\ell j} &= 1 \quad \forall j \in \{1,\dots,n\} \\
\sum_{j=1}^{n} x_{\ell j} &= 1 \quad \forall \ell \in \{1,\dots,n\} \\
x_{\ell j} &\in \{0,1\} \quad \forall \ell, j \in \{1,\dots,n\} \\
\sum_{k=1}^{n} y_{ijk\ell} &= x_{\ell j} \quad \forall i, j, \ell \in \{1,\dots,n\} \\
\sum_{i=1}^{n} y_{ijk\ell} &= x_{\ell j} \quad \forall j, k, \ell \in \{1,\dots,n\} \\
0 \le y_{ijk\ell} &= y_{ji\ell k} \quad \forall i, j, k, \ell \in \{1,\dots,n\}
\end{aligned}
$$

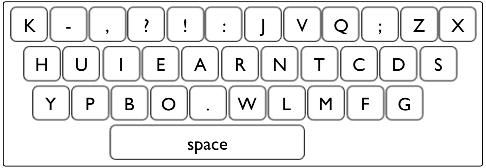
$$
\begin{aligned}
\min \sum_{\ell=1}^{n}\sum_{j=1}^{n} & w_{ij} \\
\text{s.t.} \sum_{\ell=1}^{n} x_{\ell j} &= 1 \quad \forall j \in \{1,\dots,n\} \\
\sum_{j=1}^{n} x_{\ell j} &= 1 \quad \forall \ell \in \{1,\dots,n\} \\
x_{\ell j} &\in \{0,1\} \quad \forall \ell, j \in \{1,\dots,n\} \\
w_{\ell j} &= x_{\ell j} \sum_{k=1}^{n}\sum_{i=1}^{n} p_{k\ell} t_{ij} x_{ki} \quad \forall \ell, j \in \{1,\dots,n\} \\
w_{\ell j} &\ge 0 \quad \forall \ell, j \in \{1,\dots,n\} \\
w_{\ell j} &\ge \sum_{k=1}^{n}\sum_{i=1}^{n} p_{k\ell} t_{ij} x_{ki} - a_{ij} \cdot (1 - x_{\ell j}) \quad \forall \ell, j \in \{1,\dots,n\}
\end{aligned}
$$

- Models can be solved in parallel
- Complementary to other approaches such as random search heuristics

Results

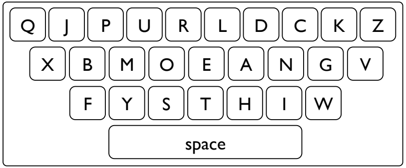


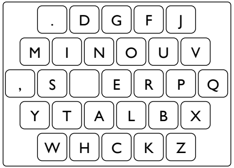


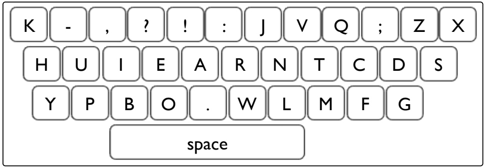


Virtual 3-row trapezoid	Virtual hexagonal	Physical 3-row standard
26 keys	29 keys	34 keys
Fitts' model	Fitts' model	Hiraga et al. 10-finger model

Results





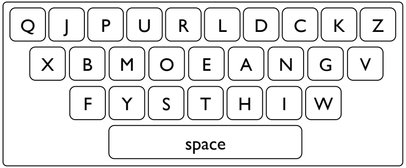


Virtual 3-row trapezoid	Virtual hexagonal	Physical 3-row standard
26 keys	29 keys	34 keys
Fitts' model	Fitts' model	Hiraga et al. 10-finger model

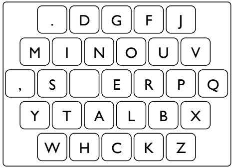
Keyboards (wpm)

qwerty (33.65)	metropolis (42.04)	qwerty (73.21)
DSK (33.97)	justhci (42.38)	DSK (76.59)
This (43.10)	This (42.65)	This (78.36)

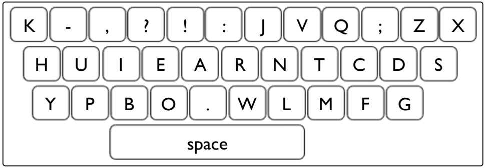
Results



Virtual 3-row trapezoid
26 keys
Fitts' model



Virtual hexagonal
29 keys
Fitts' model



Physical 3-row standard
34 keys
Hiraga et al. 10-finger model

Keyboards (wpm)		
qwerty (33.65)	metropolis (42.04)	qwerty (73.21)
DSK (33.97)	justhci (42.38)	DSK (76.59)
This (43.10)	This (42.65)	This (78.36)
any (< 44.82)	any (< 45.55)	any (< 79.95)

Summary

Take-home messages

- Integer Programming proved useful for UI Optimization
- IP provides rigorous mathematical guarantees for the optimum
- IP is complementary to other methods (e.g., random search heuristics)

`http://resources.mpi-inf.mpg.de/keyboardoptimization/`

(by Oct 15)

Summary

Take-home messages

- Integer Programming proved useful for UI Optimization
- IP provides rigorous mathematical guarantees for the optimum
- IP is complementary to other methods (e.g., random search heuristics)

Thank you for your attention!

http://resources.mpi-inf.mpg.de/keyboardoptimization/

(by Oct 15)