Cheatsheet

Combinatorial Optimization

1) Understanding of uses and assumptions of computational interaction and design
2) Ability to cast simple design problems as combinatorial optimization tasks, including design space, objectives, constraints.

Computational interaction applies computational methods to explain, enhance, and learn from interaction with a human.

1) Relies on a formal description of the problem.
2) Relies on data and algorithms.

Data influenced a model, which is used by an algorithm to do the design work.

The **combinatorial design problem** finds an optimal design d from design space D measured by objective function g given constraints θ

$$\max_{d \in D} g(d, \theta).$$

- **Design task**. The description of the design task.
- **Objective function** g. Measures the goodness of the design.
- **Design space** D. In combinatorial design problems, the size of the design space grows into the factorial of the number of elements, $O(n!)$.
- **Task instance** θ. Constraints of the design task.

Example:

- Design task: Letter assignment to a keyboard.
- Objective function: Fitt's law.
- Design space: All possible different letter assignments.
- Task Instance: The type of keyboard.
- Optimization can be done using an optimizer such as simulated annealing.

Perception and Attention

1) Windows of visibility
2) Rosenholtz' clutter model
3) Ability to predict how bottom-up (saliency) and top-down attention would proceed for a given layout.

Human visual system (HSV)

1) Sensation (1-100 ms)
2) Detection (30-300 ms)
3) Organization (30-500 ms)
4) Selection (200-400 ms)
5) Adaptation (seconds to years)

Windows of visibility: Limits to HVS

1) Wavelength (380-780 nm)
2) Field of view (190 degrees horizontal, 125 degrees vertical)
3) Trichromaticity (perception of blue, green and red wavelengths)
4) Luminance (100 max/min)
5) Spatial frequency
6) Local contrast
7) Fixation

Clutter as feature congestion:

- HVS has evolved to spot unusual items in scenes
- Clutter is the state in which excess items, or their representation or organization, lead to a degratation of performance at some task.

Rosenholz' clutter model: If a feature vector is an outlier to the local distribution of feature vectors, then that feature is **salient**.

Control

1) Ability to predict movement with Fitts' law and steering law when parameters are given
2) Ability to model (block diagram) a pointing gesture using control theory, in particular, a block diagram implementing 2OL or similar model

---

Fitts' law: Predicts pointing movement time as a function of distance and width of target.

$$MT = a + b \log_2 \left(\frac{D}{W} + 1 \right)$$

where D is the distance of the target, W is the width of the target, and a and b are parameters obtained by fitting the model into data.

Steering law: Predicts steering movement time as a function of distance and width of target.

$$MT = a + b \frac{A}{W}$$

where A is the distance of the steering line, W is the width of the steering line, and a and b are parameters obtained by fitting the model into data.

Control Theory:

- 0OL – Position control
- 1OL – Velocity control
- 2OL – Acceleration control

Input

1) Ability to tell what kinds of filtering are needed for different issues in raw sensor data
2) Understanding of operating principles of a filter (e.g. 1€ filter) and a recognizer (e.g. 1$ recognizer)
3) Ability to construct a decoder for single or sequential input

---

Filtering is required due to **noise** in signal. Noise is unwanted disturbance (or) fluctuation in an electric signal. Types of sensor noise:

1) *Noise* – Continuous random variations in the measured position.
2) *Dropout* – Complete loss of measurement or tracking.
3) *Glitches* – Random spikes of sensing that are not due to intentional movement.

Filtering Techniques: Trade-off between *jitter* and *lag*.

1) *Moving average*

$$\hat{X} = \sum_{i=t-n}^{t} X_i s$$

where $\hat{X}$ is filtered value, X_i value at time i, t current time and n window size.

- n increase: more lag, less jitter
- n decrease: less lag, more jitter

2) *Low-pass filter* (single exponential)

$$\hat{X}_i = \alpha X_i + (1-\alpha)\hat{X}_{i-1}$$

where $\hat{X}_i$ filtered values at time i, X_i sensor value at time i and $\alpha \in [0,1]$ smoothing factor.

- α increase: less lag, more jitter
- α decrease: more lag, less jitter

3) *1€-filter*: A low-pass filter where the value of α is dependent on the velocity. The idea is that at low speeds jitter is a problem and at high speeds lag is a problem. Because α depends on speed it adjusts to this.

$$\alpha = \frac{1}{1+\frac{\tau}{T_e}}$$

$$\tau = \frac{1}{2\pi f_C}$$

$$f_C = f_{C_{min}} + \beta|\dot{\hat{X}}_i|$$

Bayesian human-in-the-loop optimization

Understanding of core concepts in Bayesian optimization, including surrogate model, prior update, acquisition function

---

Bayesian optimization: Find the minimum of a function $f(x)$ withing some bounded domain $X \in \mathbb{R}^D$

$$x^* = \operatorname{argmin}_{x \in X} f(x)$$

- f is a black box that can be 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

Want to find the minimum with small number of evaluations of f

1) Construct a tractable **statistical surrogate model** g of f.
    - Gaussian processes
2) Turn the optimization problem into **a sequence of easier problems**.
    - Choose next x to evaluate f using **guided exploration** by maximizing an **acquisition function** $\alpha(x; D_{t-1})$

$$x_t = \operatorname{argmax}_x \alpha(x; D_{t-1}).$$

Integer Programming

Ability to formulate a menu and keyboard design problem as a mixed integer linear program.

Linear menu assignment problem: The cost c_{ij} for assigning an item i to a position j is defined by the expected time to select the item: $c_{ij} = p_i \cdot d_j \cdot r$. Thus the problem can be formulated as:

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} p_i \cdot d_j \cdot r \cdot x_{ij}$$

subject to

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad \forall j = 1..N$$

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad \forall i = 1..N$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j = 1..N$$

- x_{ij} denotes if an item i is assigned to position j.
- p_i is the frequency distribution of the menu items. There are two conditions that must hold for a the distribution: $\sum_{i=1}^{N} p_i = 1$ and $p_i \geq 0$.
- $d_j \geq 0$ is the distance from the start of the menu.
- $r > 0$ is the constant reading cost.

Biomechanics

Ability to evaluate the fatiguability of a given posture or movement using the Consumed Endurance model (when parameter values are given).

Strength is defined

$$S(T_{shoulder}) = 100 \cdot \frac{T_{shoulder}}{T_{max}}. \tag{1}$$

Endurance is defined

$$E(T_{shoulder}) = \frac{1236.5}{(S(T_{shoulder}) - 15)^{0.618}} - 72.5. \tag{2}$$

The magnitude of the torque for static arm is

$$\begin{aligned} T_{shoulder} &= \|\mathbf{T}_{shoulder}\| \\ &= \|\mathbf{r} \times m\mathbf{g}\| \quad , \\ &= m r_x g \end{aligned} \tag{3}$$

where

- $\mathbf{r} = [r_x, r_y]$ is a vector pointing to the *center of mass* of the arm.
- m the total mass of the arm.
- $\mathbf{g} = [0, g]$ is gravitation vector where $g = 9.81$ is the magnitude of the gravitational acceleration.

Formal Methods

1) Ability to draw a finite state diagram for simple interactive devices
2) Ability to interpret a simple verification statement expressed with temporal logic.

Create the graph of the finite state machine, then use graph algorithms to prove statements.

Cognitive Models

Ability to formulate an information foraging diagram (patch model) for a given application case.

- Patch and diet model.
- Can be used to calculate how much time should be used for *foraging* information on a single patch until moving to the next one.

Bandits

1) Understanding the bandit problem
2) Understanding how exploration/exploitation is solved
3) Understanding bandits

Reinforcement Learning

1) Ability to formulate a navigation or decision-making task in interaction as a reinforcement learning problem, including the Markov decision process (MDP).
2) Understanding of difference between POMDP and MDP.