Pricing priorities in waitlists

Filip Tokarski
Stanford GSB

January 5, 2025

Motivation

- Waitlists are a common **alternative to market mechanisms**
    - Used for affordable housing, daycare places, camping permits...
- Natural choice when we **do not want to extract revenue** from participants
- But using waitlists instead of prices causes **allocative inefficiency**...

Motivation

- We should consider intermediate options: waitlists with some partial pricing!

My question:

**How to optimally combine waitlists with prices
while recognizing that charging participants is undesirable?**

Literature

- **Mechanisms without money** (Hylland and Zeckhauser, 1979; Budish, 2011)
  - **This paper**: money allowed but transfers undesirable
- **Wasteful screening** (Hartline and Roughgarden, 2008; Yang, 2021)
  - **This paper**: combining wasteful and non-wasteful screening
- **Wait times 'acting as prices'** (Barzel, 1974; Leshno, 2022; Ashlagi et al., 2022)
  - **This paper**: but waiting screens only on relative values…
  - …while money screens on absolute values

Model

Agents

- The designer distributes two kinds of goods, $\boldsymbol{A}$ and $\boldsymbol{B}$
- Agents' values for A and B given by two-dimensional types $\boldsymbol{(a,b)} \in [0,1]^2$
- A type-(a,b) agent who gets a good, pays $\boldsymbol{p}$ and waits $\boldsymbol{t}$ gets utility:

$$\begin{aligned} e^{-\rho \cdot t}(a-p) &\quad \text{if she gets } A, \\ e^{-\rho \cdot t}(b-p) &\quad \text{if she gets } B. \end{aligned}$$

- NB: waiting **delays receipt** $\Rightarrow$ waiting cost **multiplies value** for the good!

Arrivals

- At every time $\tau \in \mathbb{R}$, flow masses $\boldsymbol{\mu_A, \mu_B} > 0$ of goods A and B arrive
  - Unit flow mass of goods arrives in total: $\mu_A + \mu_B = 1$
- At every time $\tau \in \mathbb{R}$, a unit flow mass of agents with types $(a, b) \sim \boldsymbol{F}$ arrives
  - F has full support and a differentiable pdf f
- Total good arrival rate $=$ agent arrival rate

Waitlists

- Separate first-come-first-serve waitlists for goods A and B

- The designer chooses:

  1. **Prices for joining** the two waitlists

  2. A **menu of pay-to-skip options** for each waitlist

- Arriving agents choose:

  1. **At most one waitlist** to join

  2. Whether they want some **pay-to-skip** option from their waitlist's menu

Steady state

- We will consider **steady states** of the waitlists
- In SS, **all agents of the same type make the same choices**
- Thus, the designer chooses **steady state allocations** of:
  1. Payments $p : [0, 1]^2 \to \mathbb{R}_+$
  2. Wait-times $t : [0, 1]^2 \to \mathbb{R}_+$
  3. Goods: $x : [0, 1]^2 \to \{A, B, \varnothing\}$

Designer's constraints

- Designer chooses allocation (p, t, x) subject to **IC, IR** and **supply** constraints:

$$\text{for every } (a, b), (a', b') \in [0,1]^2, \quad U[a, b, (p, t, x)(a, b)] \geq U[a, b, (p, t, x)(a', b')] \tag{IC}$$

$$\text{for every } (a, b) \in [0,1]^2, \quad U[a, b, (p, t, x)(a, b)] \geq 0 \tag{IR}$$

$$\int \mathbb{1}_{x(a,b)=A} \, \mathrm{d}F(a, b) \;\leq\; \mu_a, \qquad \int \mathbb{1}_{x(a,b)=B} \, \mathrm{d}F(a, b) \;\leq\; \mu_b \tag{S}$$

Designer's objective

- In SS, objective can be written in terms of **flows**. Choose (p, t, x) to maximize:

$$\gamma \cdot R + W$$

- $\gamma \in [0, 1]$ is the weight on revenue R:

$$R \;=\; \int p(a, b)\, \mathrm{d}F(a, b)$$

- W is the value for goods (net of payments) for agents getting them:

$$W \;=\; \int \underbrace{\mathbb{1}_{x(a,b)=A}(a - p(a, b))}_{\text{Agents getting good } A} \;+\; \underbrace{\mathbb{1}_{x(a,b)=B}(b - p(a, b))}_{\text{Agents getting good } B} \;\mathrm{d}F(a, b)$$

Designer's objective

$$W = \int \underbrace{\mathbb{1}_{x(a,b)=A}\big(a - p(a,b)\big)}_{\text{Agents getting good } A} + \underbrace{\mathbb{1}_{x(a,b)=B}\big(b - p(a,b)\big)}_{\text{Agents getting good } B} \, \mathrm{d}F(a,b)$$

- When allocating, designer cares about agents' values, but **not** when they arrived
- Counterintuitive implication: **no wait times** in the objective!
  - Indeed, an agent's utility is $e^{-\rho \cdot t}(a - p)$ not $a - p(a,b)\ldots$
  - $\ldots$ but giving it to her earlier **pushes someone else back**
- Also means the designer is **indifferent about some types skipping ahead!**

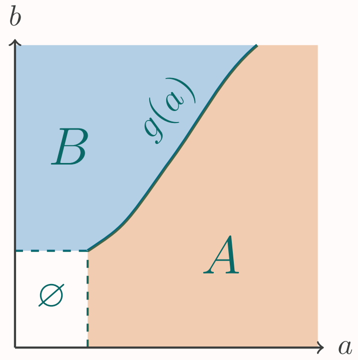
Feasible mechanisms

Who chooses which waitlist?

- When joining both waitlists costs money, some types do not participate ($\varnothing$)

- Types on the **boundary $\boldsymbol{g}$** indifferent between their best options in both waitlists

- Types below g pick some option in A, types above g pick some option in B



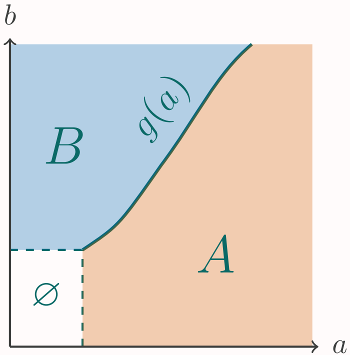
Who chooses which waitlist?

- Offering different **pay-to-skip** options alters the shape of boundary g

- Indeed, the designer is indifferent about some types skipping ahead…

- …and offers pay-to-skip options precisely to **deform the boundary $\boldsymbol{g}$**…

- …that is, to encourage certain types to join one or the other waitlist



Role of payments:

Two extreme cases

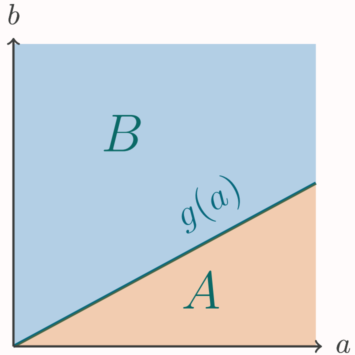
No payment benchmark

No payment benchmark

- Suppose joining is free and there are no pay-to-skip options
- Then everyone joins and wait-times ‘clear the market’
- Type (a, b) chooses A if:

$$e^{-\rho \cdot t_A} \cdot a > e^{-\rho \cdot t_B} \cdot b$$

- **Ratio** $\frac{a}{b}$ determines choice of waitlist



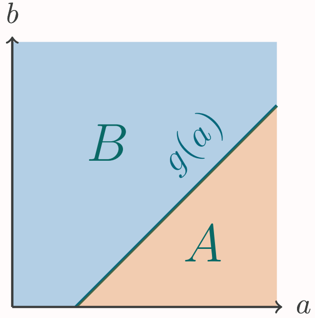
Non-wasteful payments ($\gamma = 1$)

Non-wasteful payments ($\gamma = 1$)

Proposition 1

If payments are not wasteful ($\gamma = 1$), the optimal mechanism offers **no pay-to-skip** options and prices entry to only one waitlist. The price is chosen to **equate wait-times** in both waitlists.

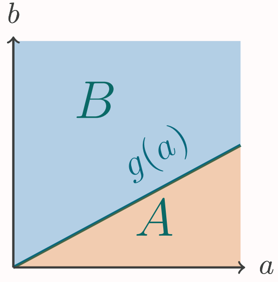
- This achieves the first-best!
- A-goods go to those with **highest** $\boldsymbol{a - b}$



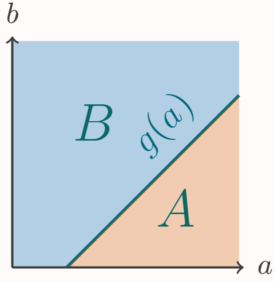
Role of payments: intuition

- Without payments, agents self-select only based on **relative values**

- Payments are wasteful, but let us screen on agents' **absolute values**



No payments



Payments + equal wait-times

- In general, payments create a **better allocation** but **are wasteful**

General case ($\gamma \in [0, 1]$)

General case

- **Assumption**: consider piece-wise continuously diff'able wait-time allocation rules

Theorem 1

The optimal mechanism prices entry to only one waitlist and offers finitely many pay-to-skip options.

Conjecture 1

For *sufficiently well-behaved distributions*, the optimal mechanism prices entry to only one waitlist and offers **no pay-to-skip options**.

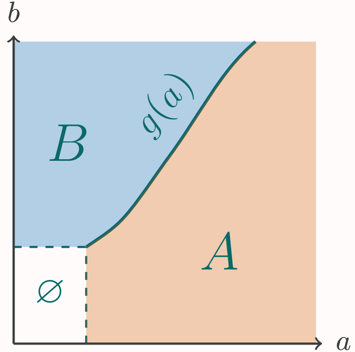
- Conjecture 1 holds in simulations for uniform, normal, Beta, etc...

Proof intuition

Indirect utilities

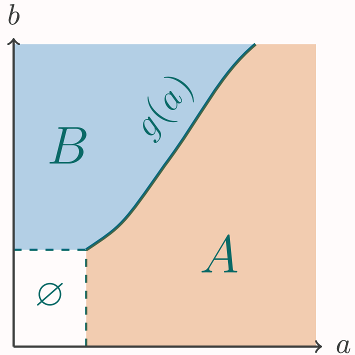
- Agents in the A-region choose some pay-to-skip options in waitlist A...

- ...and agents in the B-region choose one in waitlist B



Indirect utilities

- Agents in the A-region choose some pay-to-skip options in waitlist A...
- ...and agents in the B-region choose one in waitlist B
- Moreover, agents' pay-to-skip choice in their waitlist **does not depend** on their value for the other good!
- Therefore, we can write indirect utilities cond. on joining waitlists A and B as $\boldsymbol{U_A(a), U_B(b)}$



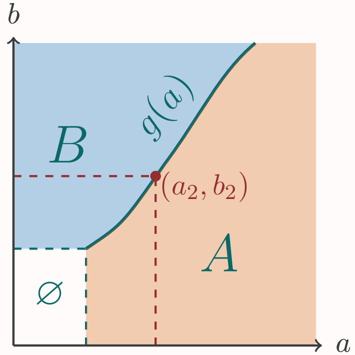
Indirect utilities

- $U_A(a), U_B(b)$ are the indirect utilities cond. on joining waitlists A and B

- We thus have two 1D screening problems (one for each wailtist)...

- ...connected by the boundary types' indifference conditions:

$$U_A(a_2) = U_B(g(a_2))$$



Proof strategy

1. Rewrite the problem in terms of U_A, U_B, and the g they induce
2. Fix any g and find the optimal U_A, U_B that implement it
3. Find the optimal g among optimally implemented boundaries

Objective in terms of U_A and U_B

1. Recall the objective is:

$$W = \int \underbrace{\mathbb{1}_{x(a,b)=A}(p(a,b)\cdot\gamma + a - p(a,b))}_{\text{Agents getting } A} + \underbrace{\ldots}_{\text{Agents getting } B} \mathrm{d}F(a,b)$$

2. To express $p(a,b)$ using $U_A(a)$, notice that:

$$U_A(a) = e^{-\rho\cdot t(a,b)}(a - p(a,b)) \quad \text{and} \quad U_A'(a) = e^{-\rho\cdot t(a,b)}$$

3. This gives $\frac{U_A(a)}{U_A'(a)} = a - p(a,b)$ and thus:

$$W = \int \mathbb{1}_{x(a,b)=A}\left(a\cdot\gamma + (1-\gamma)\cdot\frac{U_A(a)}{U_A'(a)}\right) + \underbrace{\ldots}_{\text{Agents getting } B} \mathrm{d}F(a,b)$$

Optimal U_A, U_B inducing a given boundary

- Fix some boundary g
- Pick indirect utilities U_A, U_B to maximize the objective:

$$W = \int \mathbb{1}_{x(a,b)=A} \left(a \cdot \gamma + (1-\gamma) \frac{U_A(a)}{U'_A(a)} \right) + \mathbb{1}_{x(a,b)=A} \left(b \cdot \gamma + (1-\gamma) \frac{U_B(b)}{U'_B(b)} \right) \mathrm{d}F(a,b)$$

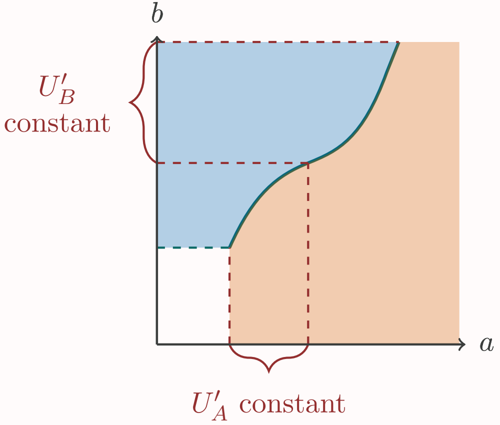
- Subject to:
  1. U_A, U_B being convex, increasing, and Lipschitz
  2. U_A and U_B being 0 for lowest participating types
  3. Agents at the boundary being indifferent: $U_A(a) = U_B(g(a))$
- Can see that **'more convex' $\boldsymbol{U_A, U_B}$ bad for objective**

How flat can we make U_A and U_B?

- U_A more convex $\rightarrow$ different wait-times $\rightarrow$ larger payments!
- Role of payments: **affect area split** by deforming the boundary
- So **pointless to charge more** than required **for particular area split**!
- How flat can we make U_A and U_B? Diff'ing boundary indifference gives:

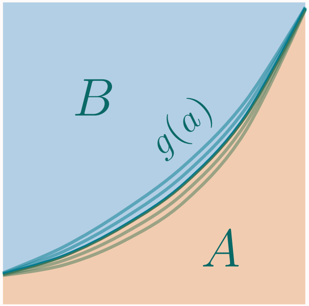
$$U_A'(a) = U_B'(g(a)) \cdot g'(a)$$

- Wherever $g(a)$ convex, set $U_A'(a)$ **constant** and $U_B'(b(a)) \propto g'(a)$!

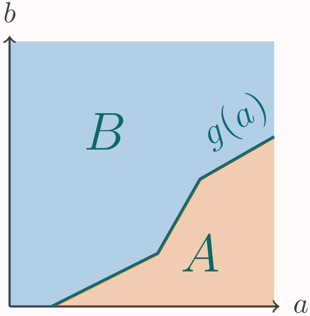


Picking the optimal boundary

- These conditions tell us **how to optimally implement** each boundary g
- Now, look at any convex region of g...
- ...and find necessary conditions for the **optimal shape** of g on it
- Turns out the optimal g has to be **linear on every such region**!
- Else, there is an improving perturbation

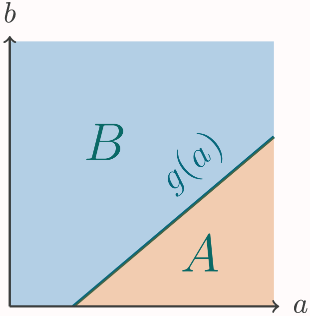


Optimal boundaries



Theorem 1

Entry price for only one waitlist
Finite pay-to-skip options



Conjecture 1

Entry price for only one waitlist
No pay-to-skip options

Conclusions

Conclusions

- The literature notes wait-times can to some extent **'act like prices'**
- Distinguish **waitlists** (waiting delays receipt) and **queues** (waiting wastes time)
- For waitlists, **wait-times only screen on relative preferences**
- Payments screen on **absolute preferences**, and could be useful even when wasteful

Thank you!