Fair Ride Allocation on a Line

Kaz Makino

RIMS, Kyoto University

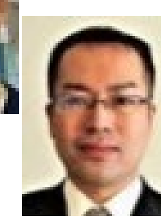
Joint work with

Yuki Amano (Kyoto Univ.)

Ayumi Igarashi (Univ. Tokyo)

Yasushi Kawase (Univ. Tokyo)

Hirotaka Ono (Nagoya Univ.)



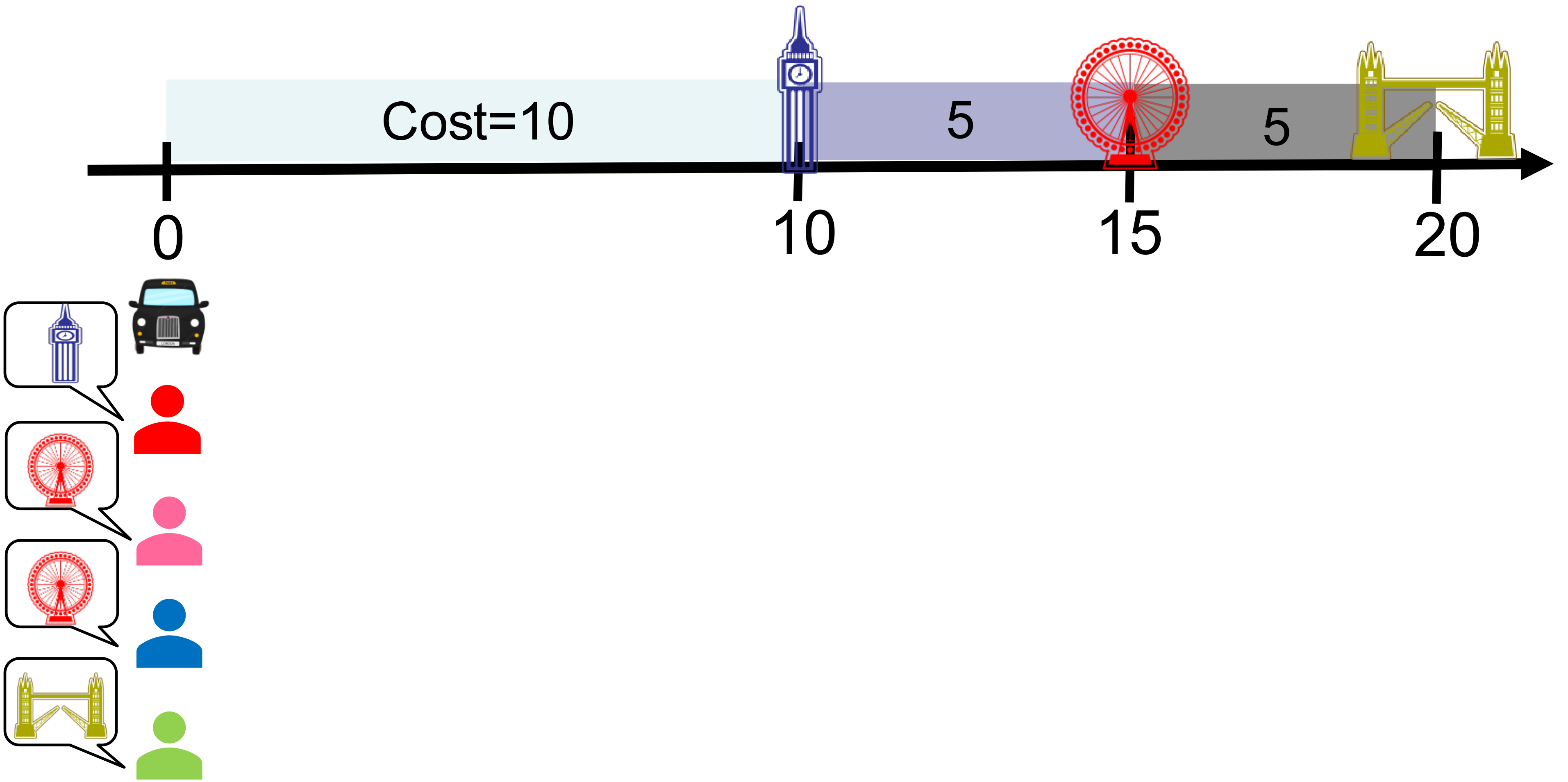
Outline of my talk

1. Airport game
2. Our Model
3. Solution Concepts
4. Relationships among concepts
5. Algorithmic results
6. Hardness results
7. Conclusion

Airport game [Littlechild and Owen, 1973]

- Imagine a group of agents sharing a taxi ride.
- **How can we divide the cost**？ **Total cost** ＝20

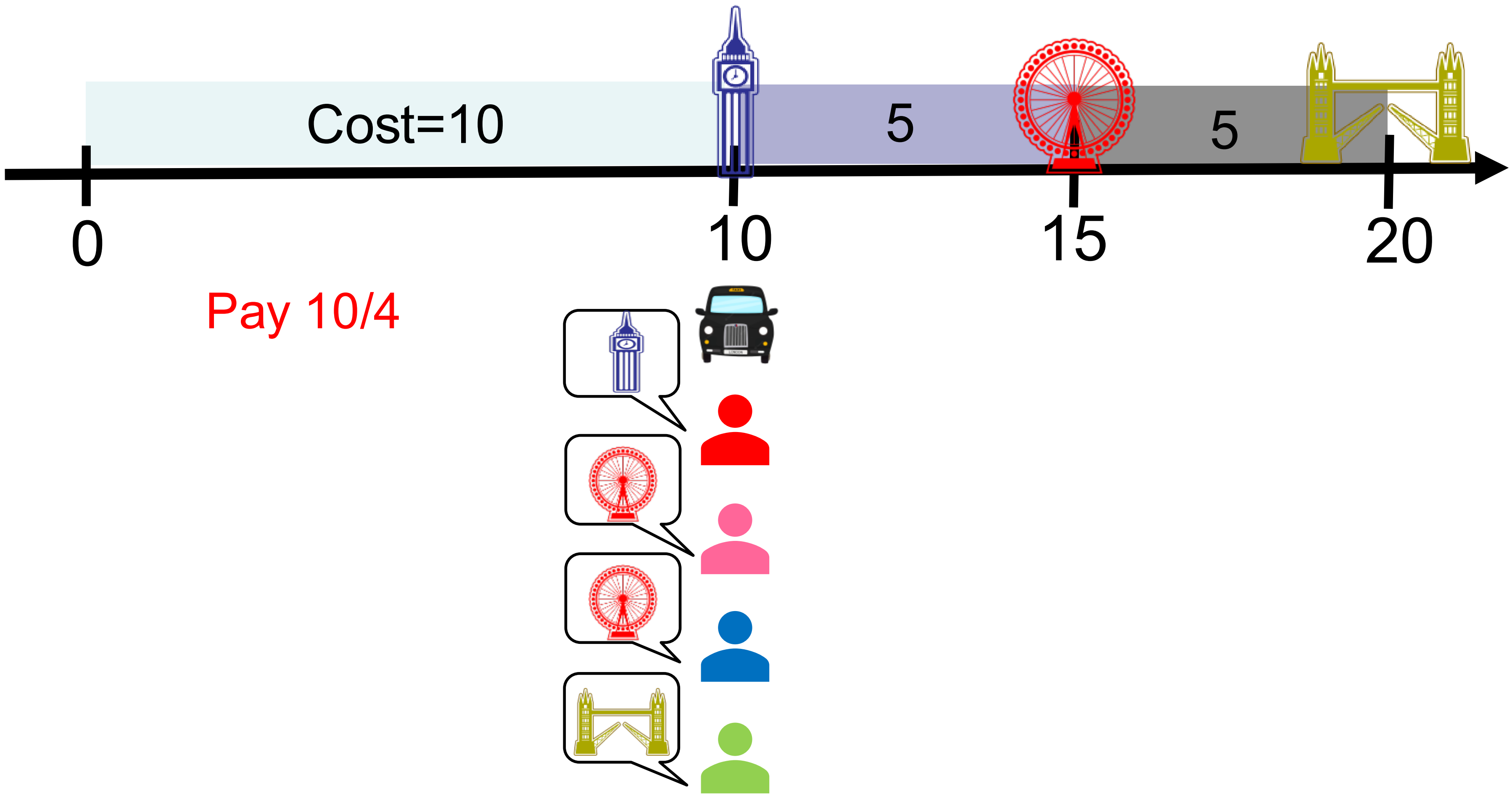
Divide the cost of each segment equally



Airport game [Littlechild and Owen, 1973]

- Imagine a group of agents sharing a taxi ride.
- **How can we divide the cost** ? **Total cost** ＝20

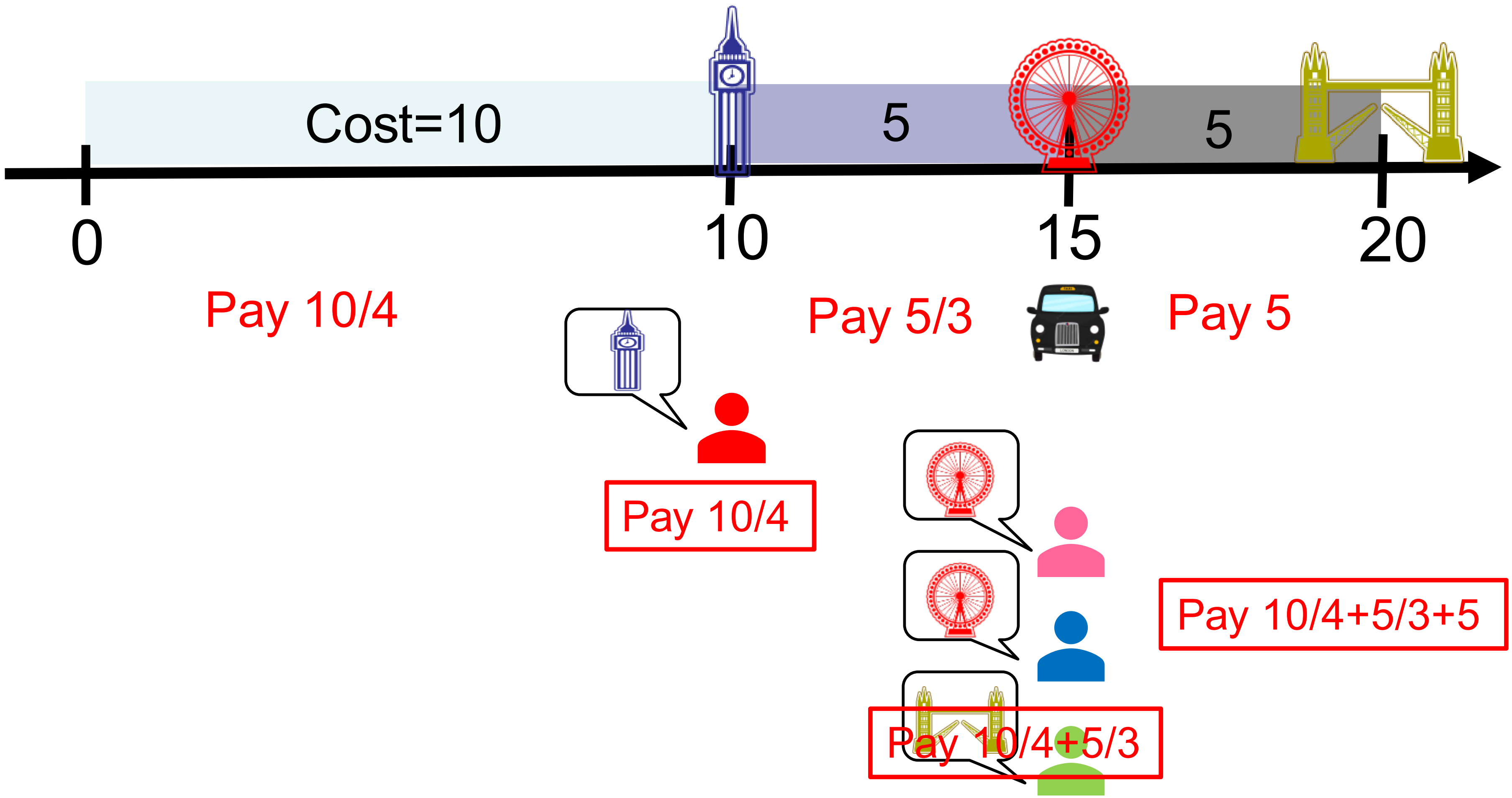
Divide the cost of each segment equally



Airport game [Littlechild and Owen, 1973]

- Imagine a group of agents sharing a taxi ride.
- **How can we divide the cost**？ **Total cost** ＝20

Divide the cost of each segment equally



Airport game [Littlechild and Owen, 1973]

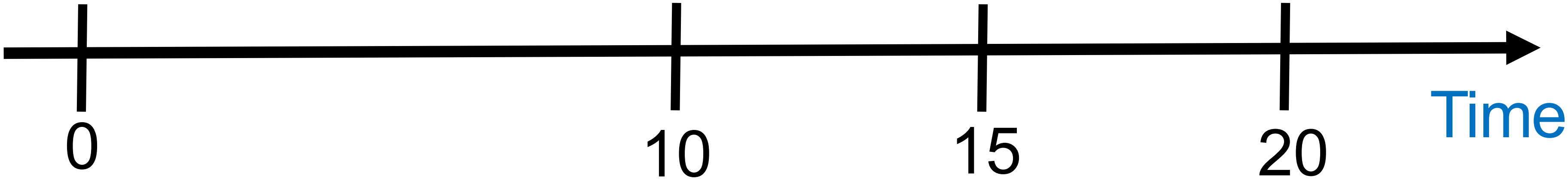
- Imagine a group of agents sharing a taxi ride.
- **How can we divide the cost** ? **Total cost** ＝20

Divide the cost of each segment equally

This sequential equal division = The Shapley value

Other applications

Sharing a facility over time for agents with different demands.

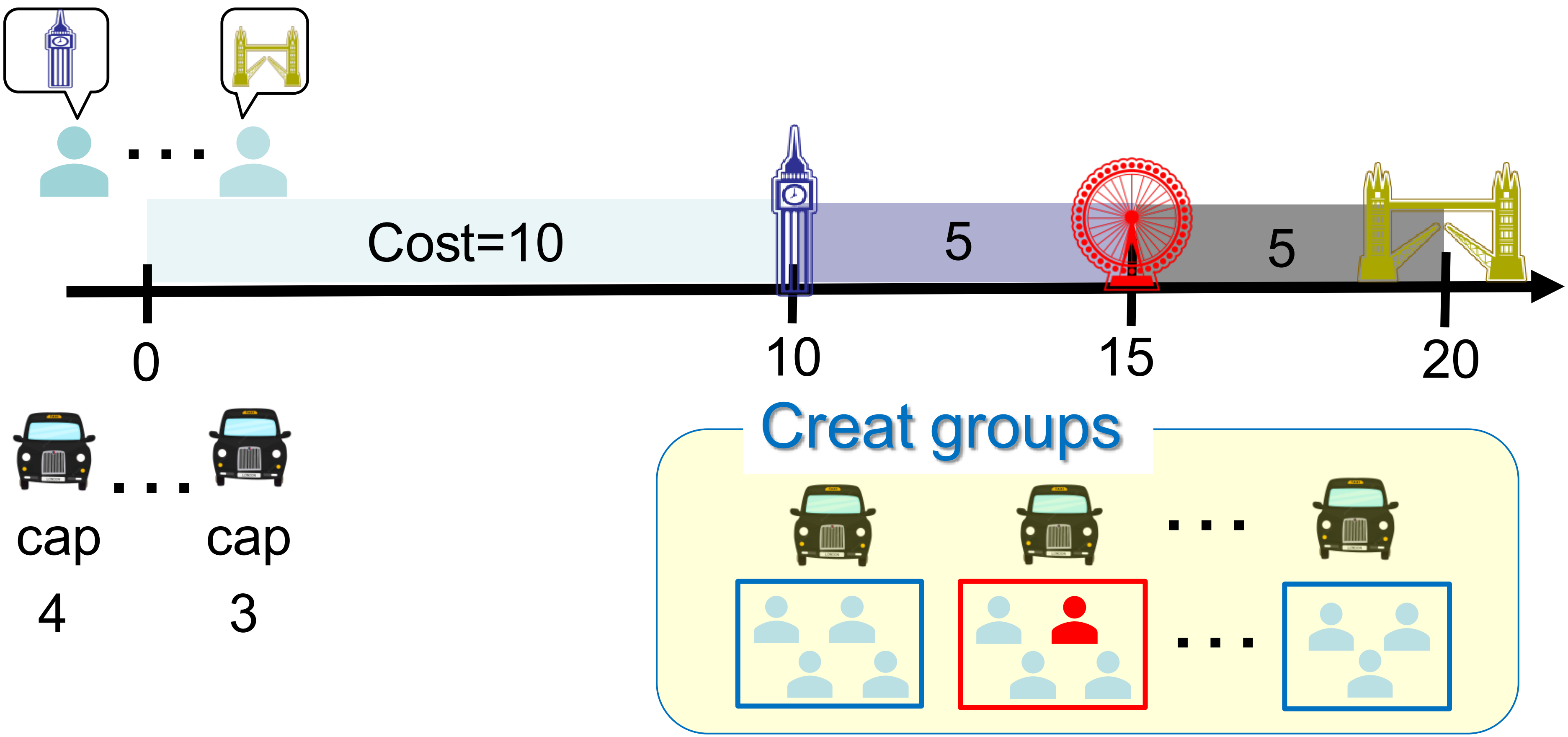


Outline of my talk

1. Airport game
2. **Our Model**
3. Solution Concepts
4. Relationships among concepts
5. Algorithmic results
6. Hardness results
7. Conclusion

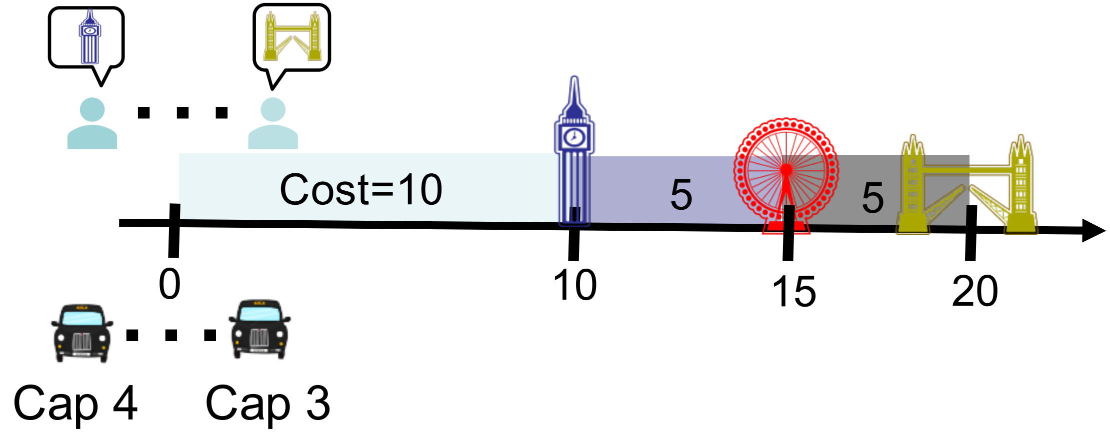
Our model

- A natural extension of airport game
- A single taxi ⇒ Multiple taxis with capacities



Payment: Shapley value of the group

Our model



- n agents with destinations $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathcal{R}_+$
- k taxis with capacities $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k \in \mathcal{Z}_+$
- All taxies have identical cost functins $c: \mathcal{R}_+ \rightarrow \mathcal{R}_+$
  - monotone $c(\mathbf{x}) \leq c(\mathbf{x}')$ if $\mathbf{x} \leq \mathbf{x}'$ — w.l.o.g. $c(\mathbf{x}) \equiv \mathbf{x}$
  - depends on the final destination
- Agents in the same taxi use Shapley value

Set of Agents who take taxi 1

Objective: Find a **fair** partition of agents $\mathcal{T}$)

Basic 1. On a line: river, highway, time

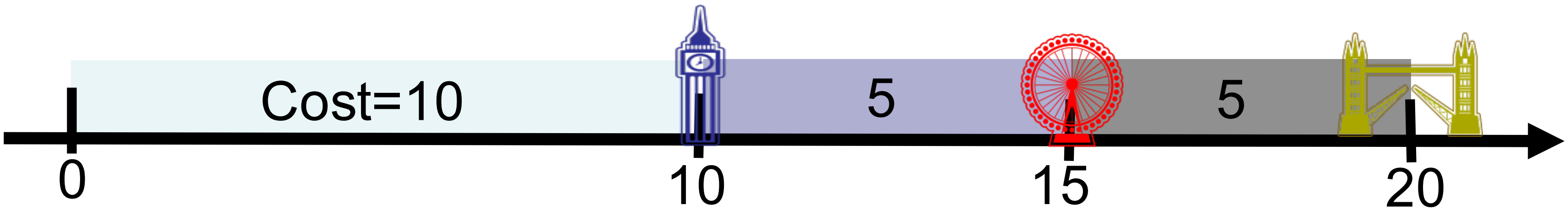
2. Idenitcal cost

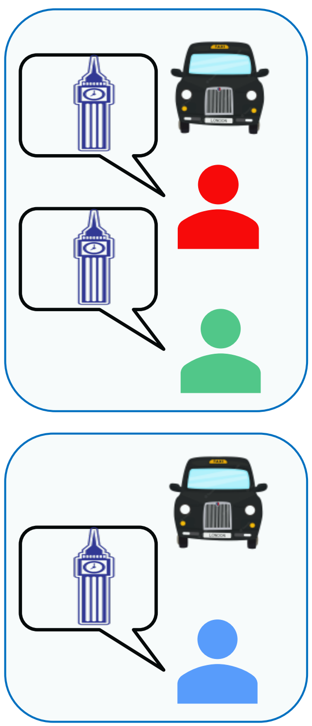
Outline of my talk

1. Airport game
2. Our Model
3. **Solution Concepts**
4. Relationships among concepts
5. Algorithmic results
6. Hardness results
7. Conclusion

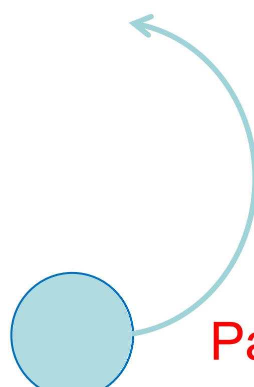
Solution Concepts

- **Envy-free**: no agent can reduce her cost by replacing herself with another agent





Pay 10/2=5



Pay 10

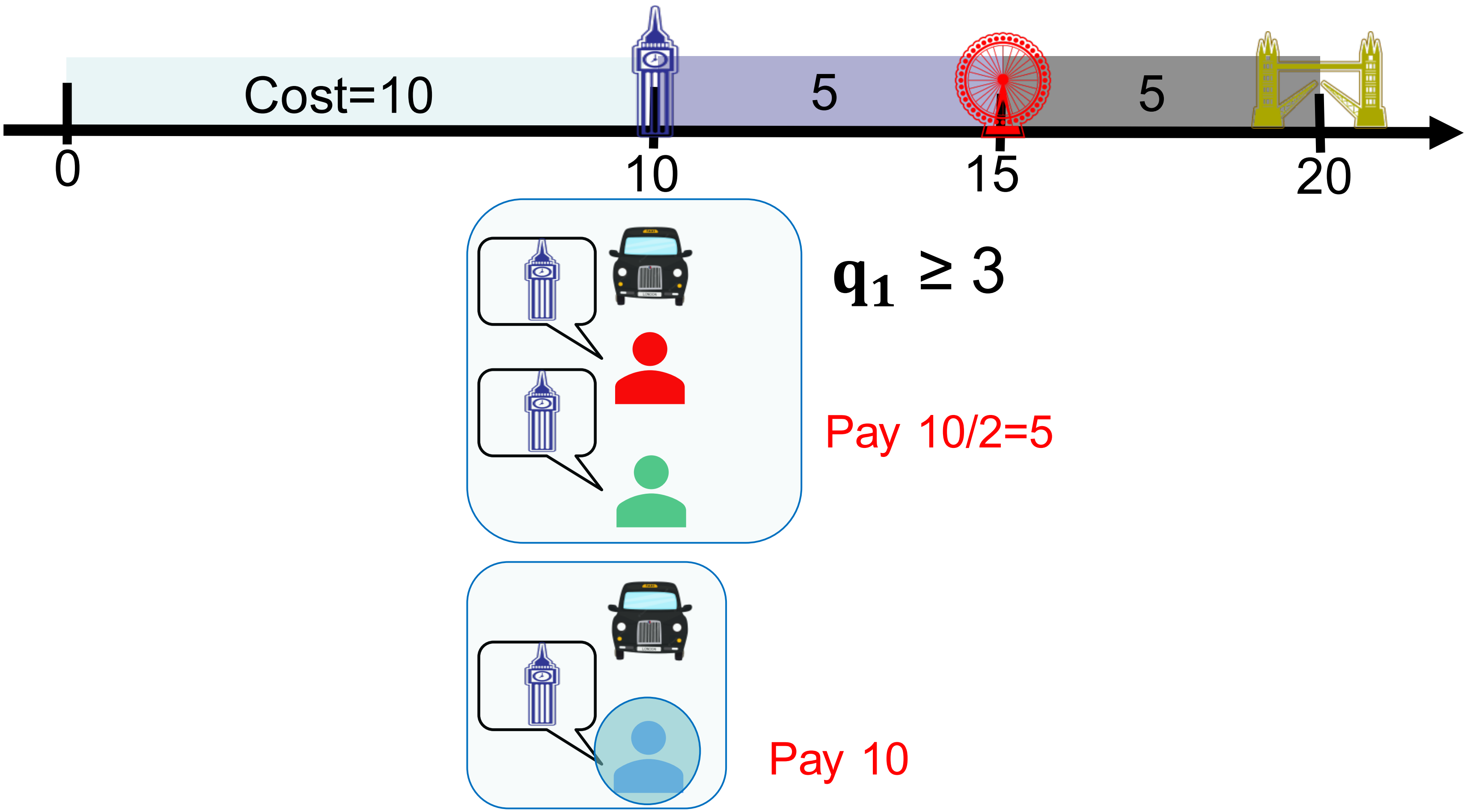
All agents have the same destination

⇩

All taxies carry the same number of agents

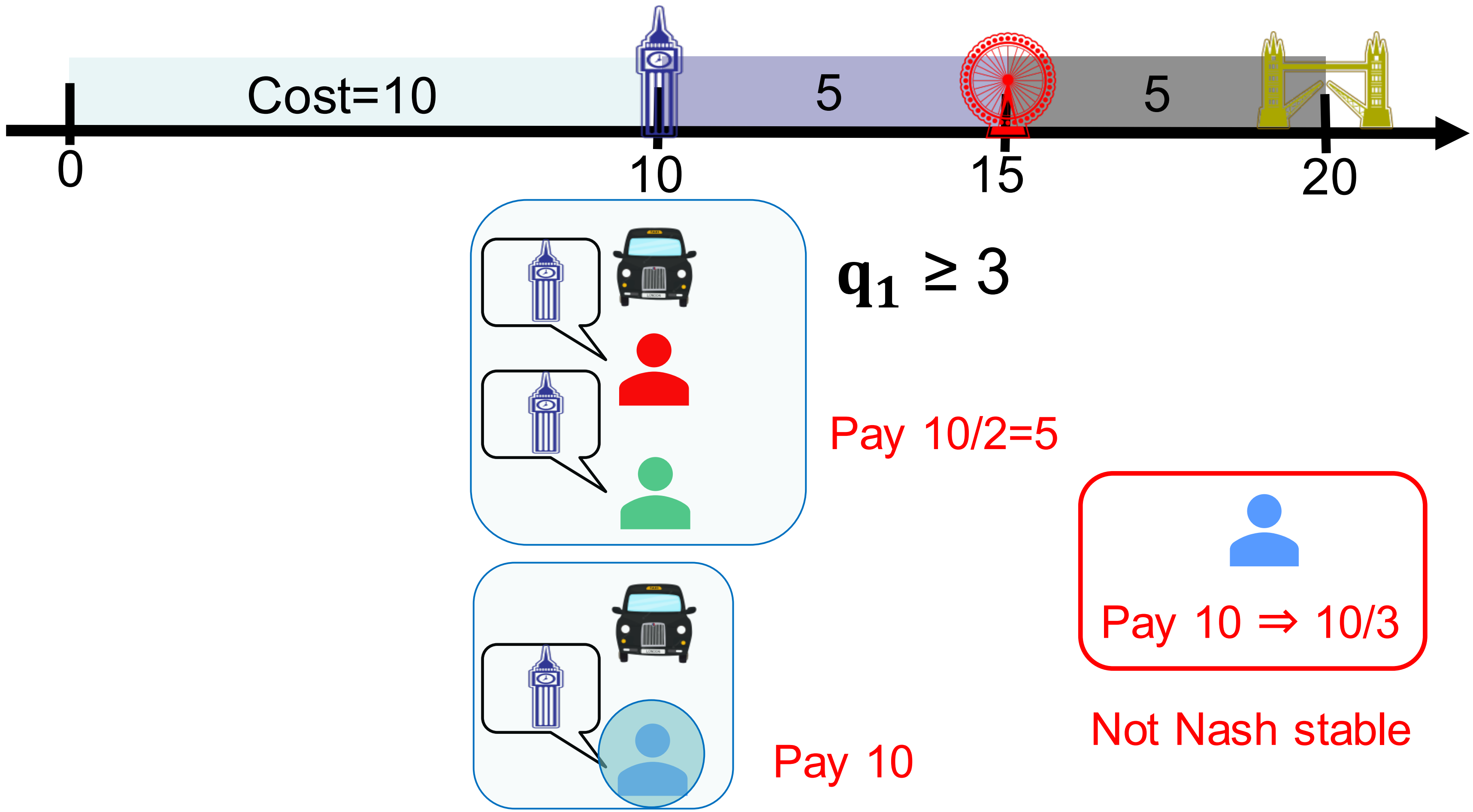
Solution Concepts

- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi



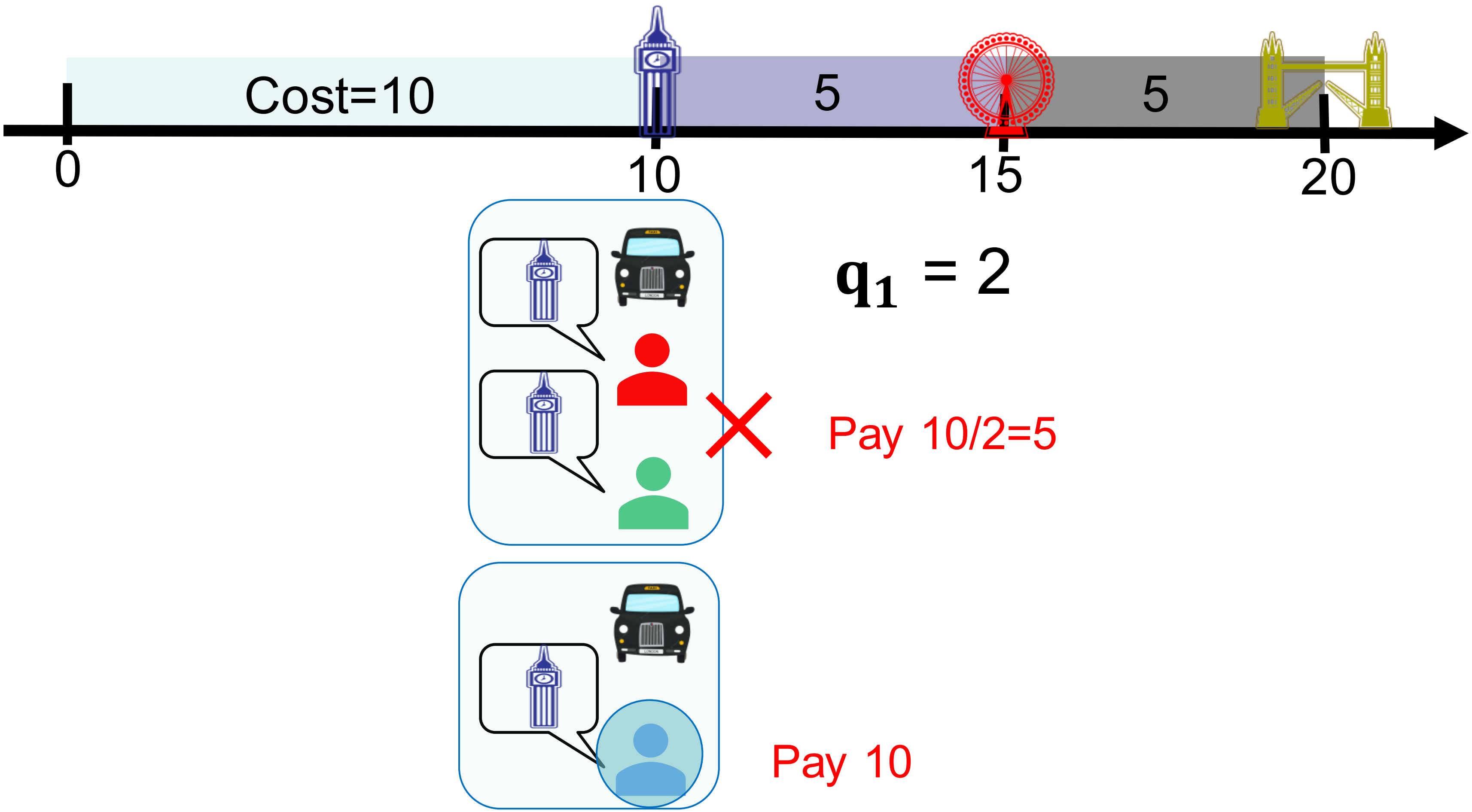
Solution Concepts

- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi



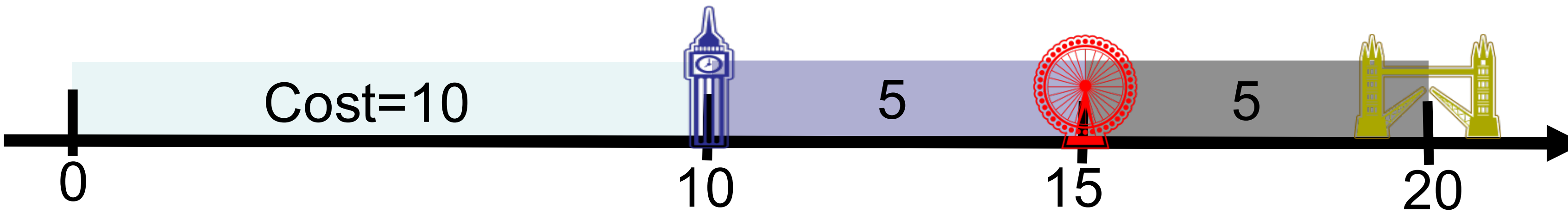
Solution Concepts

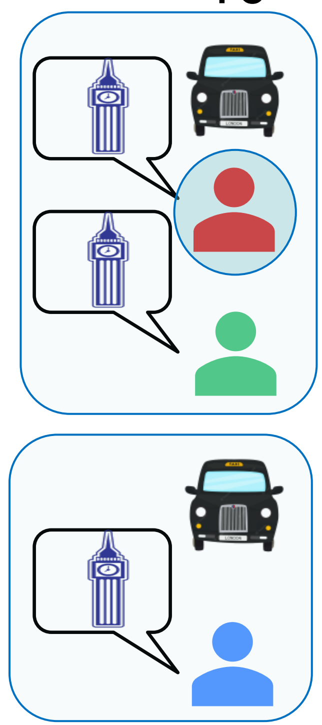
- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi



Solution Concepts

- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi





$\mathbf{q_1} = 2$

Pay 10/2=5

All agents have the same destination

↓

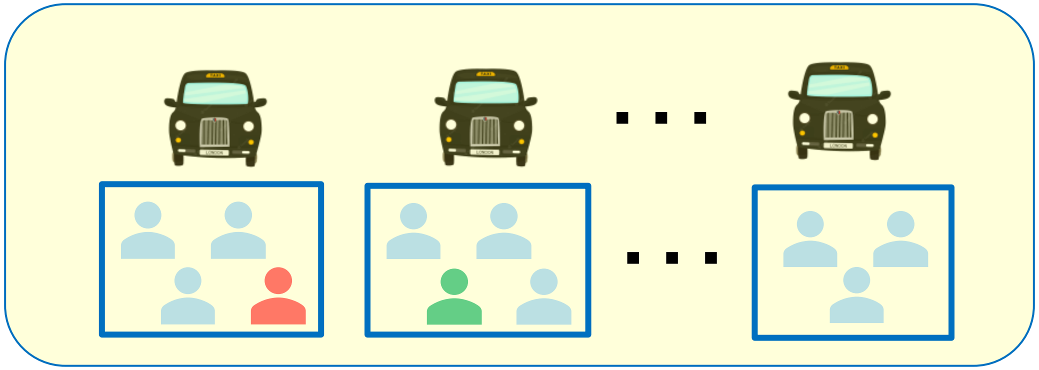
All taxies except 1 are full

Nash stable

Pay 10

Solution Concepts

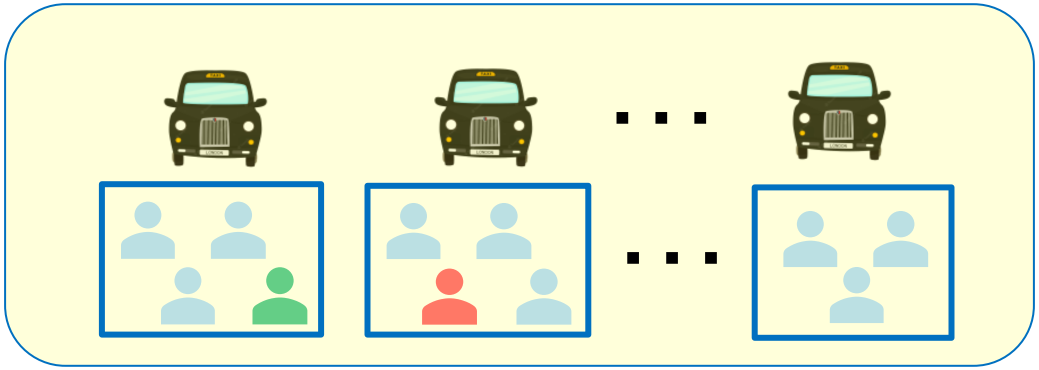
- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi
- **weakly swap-stable**: no two agents envy each other



Solution Concepts

- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi
- **weakly swap-stable**: no two agents envy each other
- **strongly swap-stable**: no two agents a and b s.t. a envies b & b can replace a

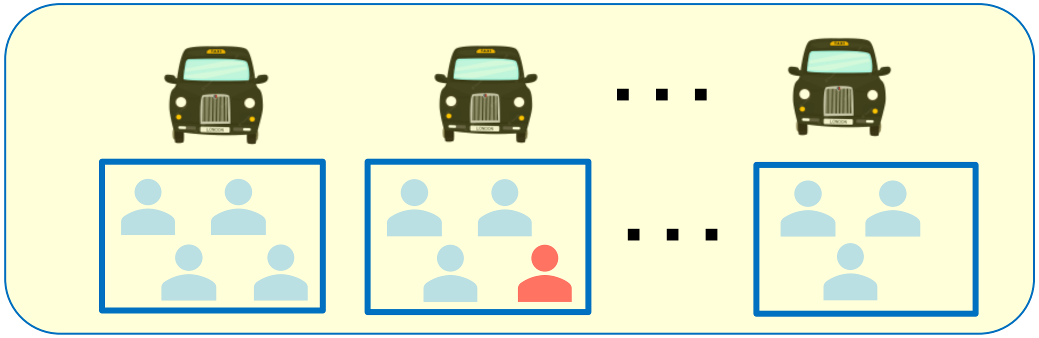
(at least as good as the current situation)



Solution Concepts

- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi
- **weakly swap-stable**: no two agents envy each other
- **strongly swap-stable**: no two agents a and b s.t. a envies b & b can replace a

  (at least as good as the current situation)
- **Contractually individually stable**: No agent can deviate without increasing the payment of some agent in her current taxi.

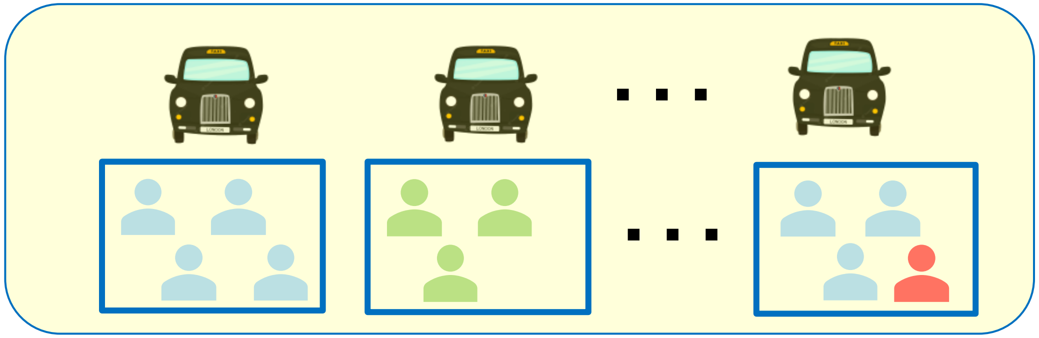


Solution Concepts

- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi
- **weakly swap-stable**: no two agents envy each other
- **strongly swap-stable**: no two agents a and b s.t. a envies b & b can replace a

  (at least as good as the current situation)

- **Contractually individually stable**: No agent can deviate without increasing the payment of some agent in her current taxi.

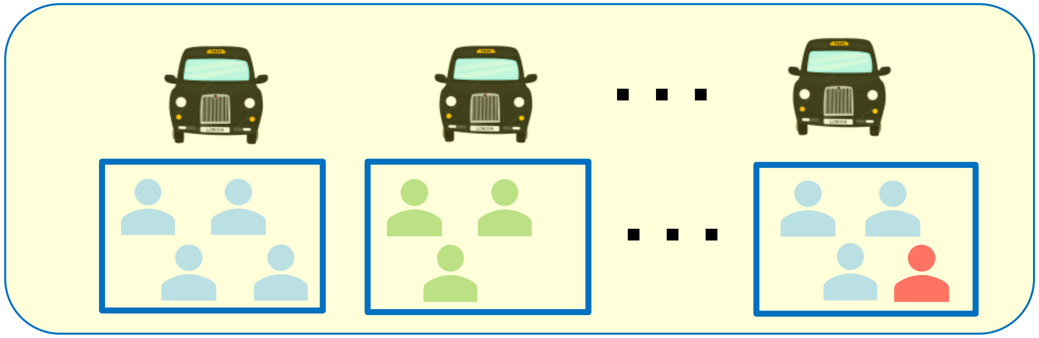


Solution Concepts

- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi
- **weakly swap-stable**: no two agents envy each other
- **strongly swap-stable**: no two agents a and b s.t. a envies b & b can replace a

  only 1 agent in her taxi

  (at least as good as the current situation)
- **Contractually individually stable**: No agent can deviate without increasing the payment of some agent in her current taxi.



Solution Concepts

- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi
- **weakly swap-stable**: no two agents envy each other
- **strongly swap-stable**: no two agents a and b s.t. a envies b & b can replace a
- **Contractually individually stable**: No agent can deviate without increasing the payment of some agent in her current taxi.
- **Socially optimal**: minimize the total payment
- **weakly Pareto optimal** $\mathcal{T}$:

  $\nexists S$ s.t. $\varphi_S(a) < \varphi_{\mathcal{T}}(a)$ for all $a \in A$
- **strongly Pareto optimal** $\mathcal{T}$:

  $\nexists S$ s.t. $\varphi_S(a) \le \varphi_{\mathcal{T}}(a)$ for all $a \in A$

  $\varphi_S(a) < \varphi_{\mathcal{T}}(a)$ for some $a \in A$

Payment of a under $\mathcal{T}$

Solution Concepts

- **Envy-free**: no agent can reduce her cost by replacing herself with another agent
- **Nash stable**: no agent can reduce her cost by deviating to another taxi
- **weakly swap-stable**: no two agents envy each other
- **strongly swap-stable**: no two agents a and b s.t. a envies b & b can replace a
- **Contractually individually stable**: No agent can deviate without increasing the payment of some agent in her current taxi.
- **Socially optimal**: minimize the total payment
- **weakly Pareto optimal** $\mathcal{T}$:

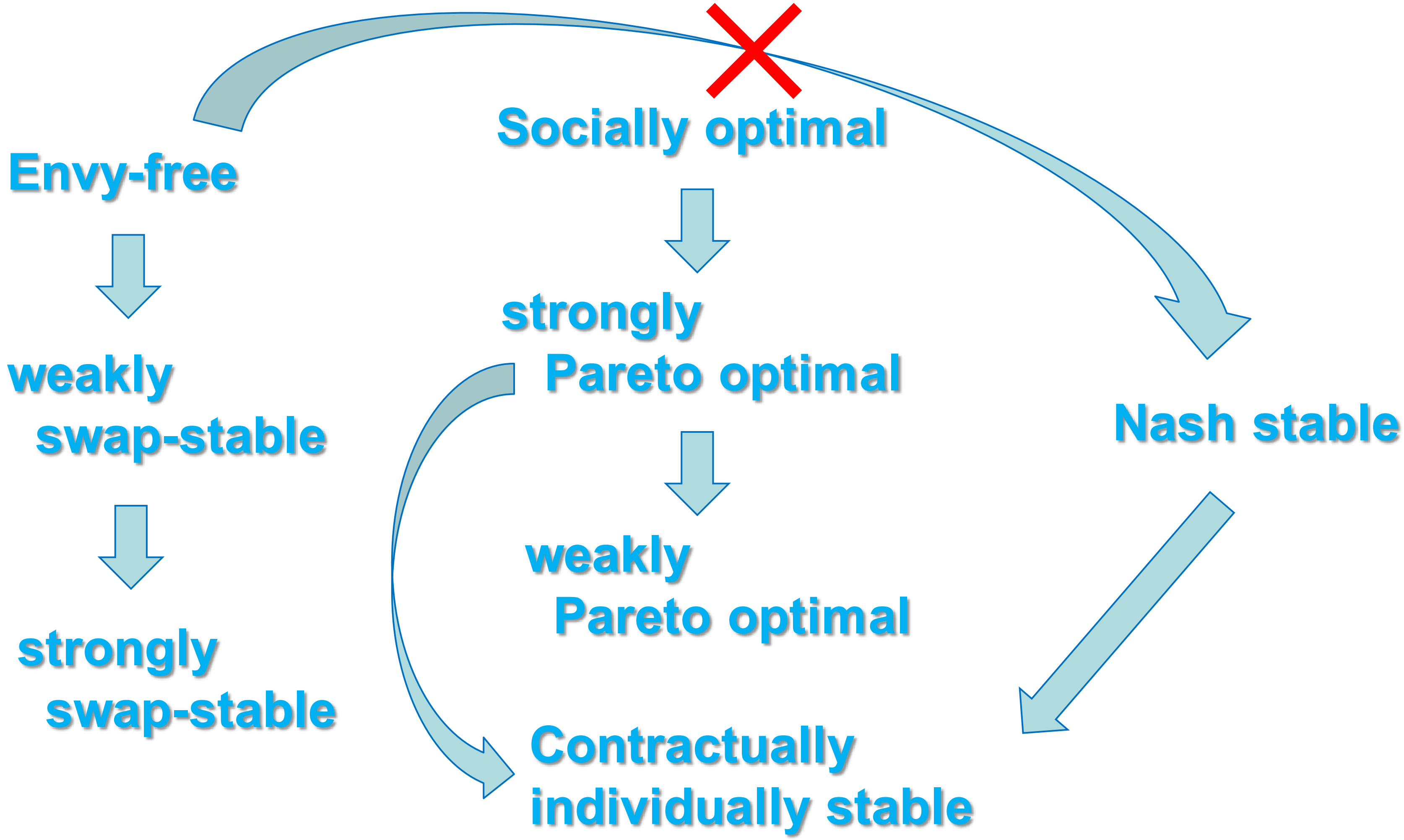
  $\nexists S$ s.t. $\varphi_S(a) < \varphi_{\mathcal{T}}(a)$ for all $a \in A$

  Payment of a under $\mathcal{T}$
- **strongly Pareto optimal** $\mathcal{T}$:

  $\nexists S$ s.t. $\varphi_S(a) \leq \varphi_{\mathcal{T}}(a)$ for all $a \in A$

  $\varphi_S(a) < \varphi_{\mathcal{T}}(a)$ for some $a \in A$

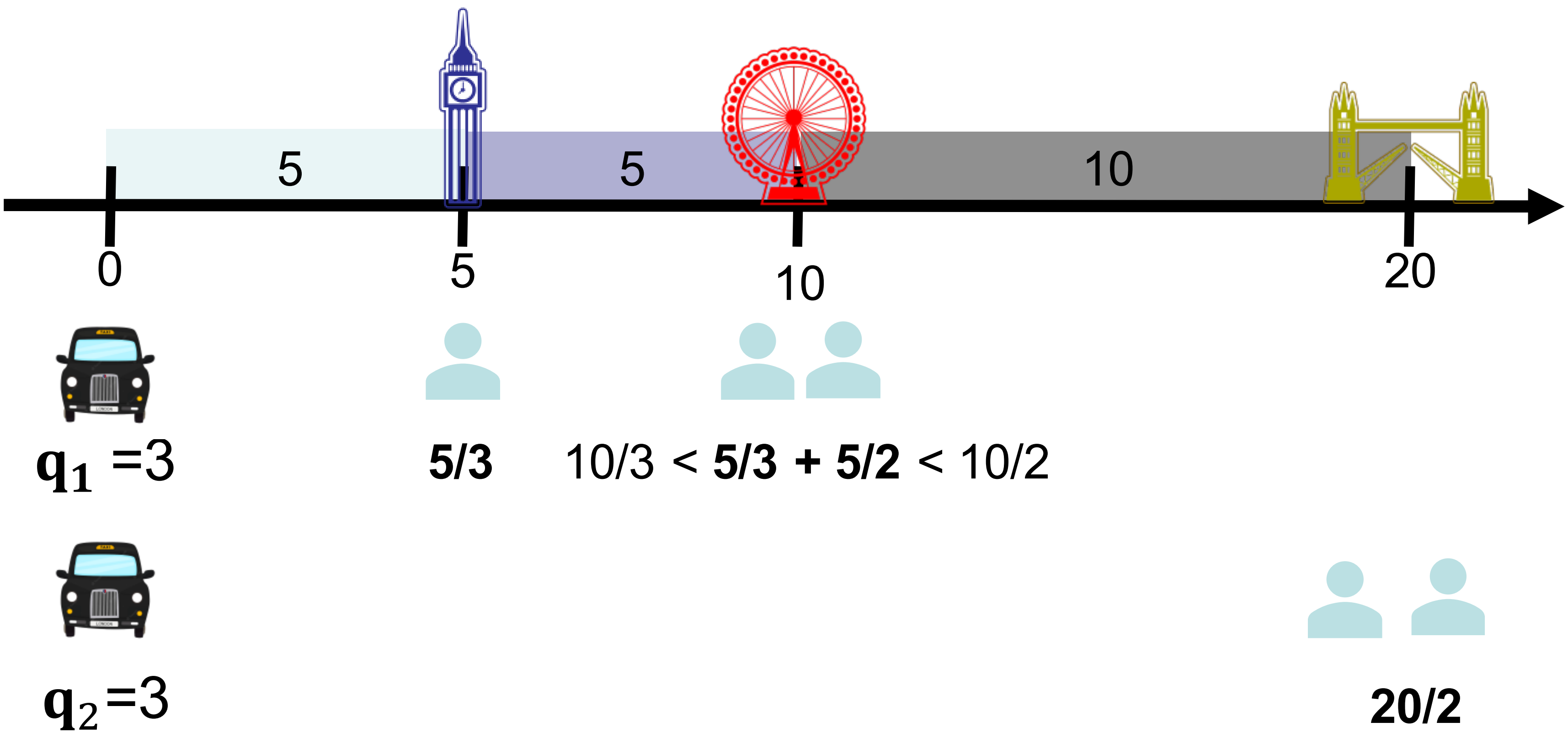
Relationships among concepts



No other implication exists

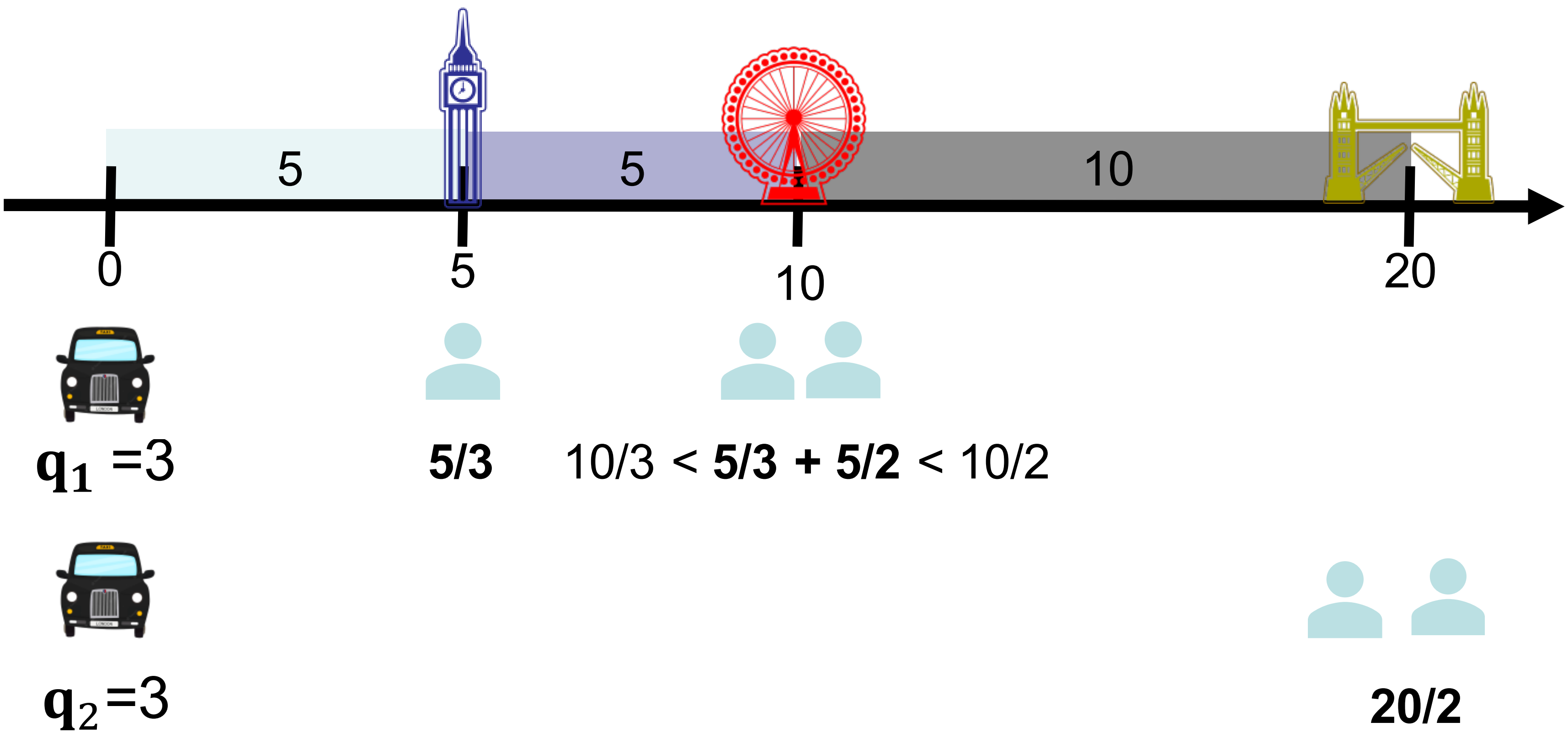
Relationships among concepts

Envy-freeness ⇏ Nash stability



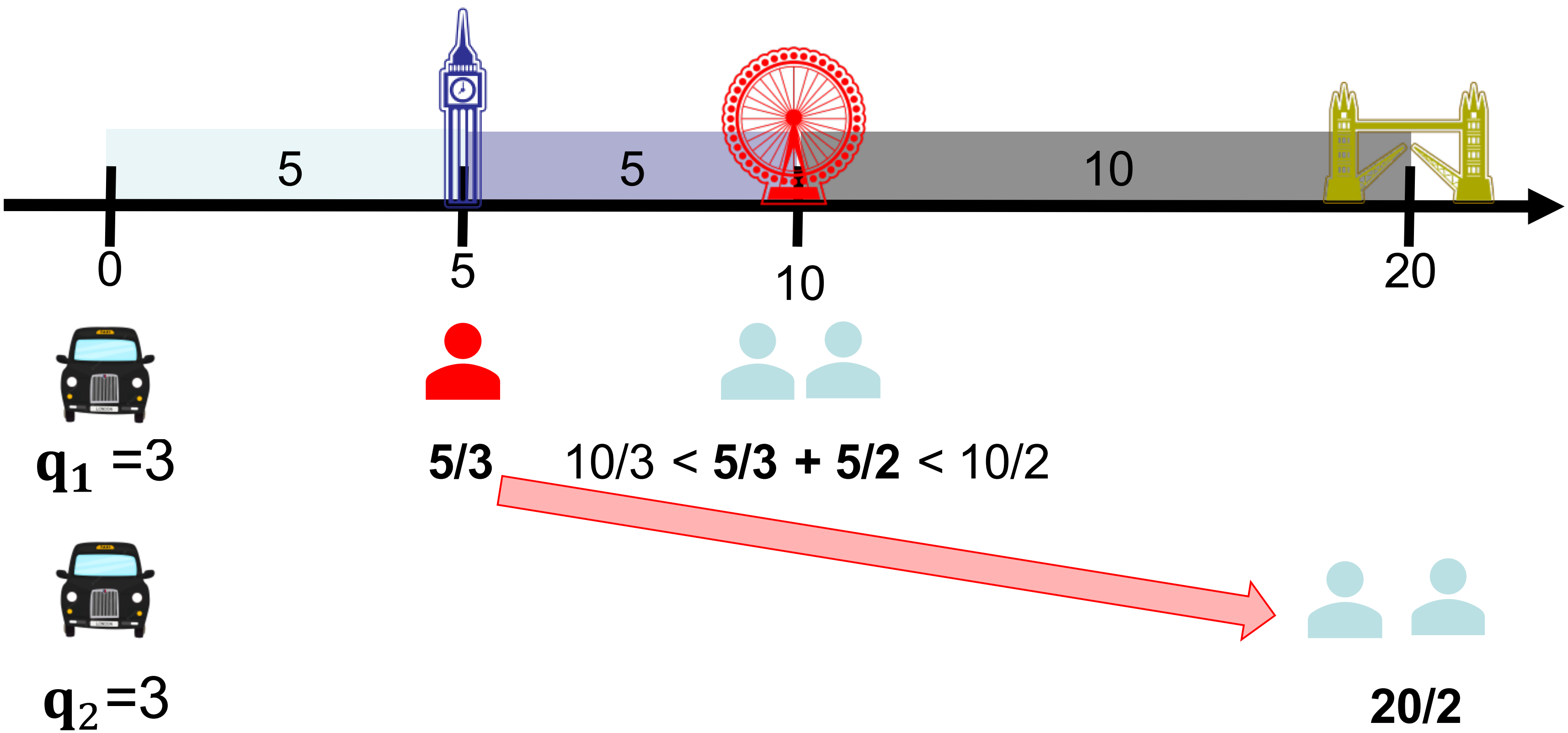
Relationships among concepts

Envy-freeness ⇏ Nash stability



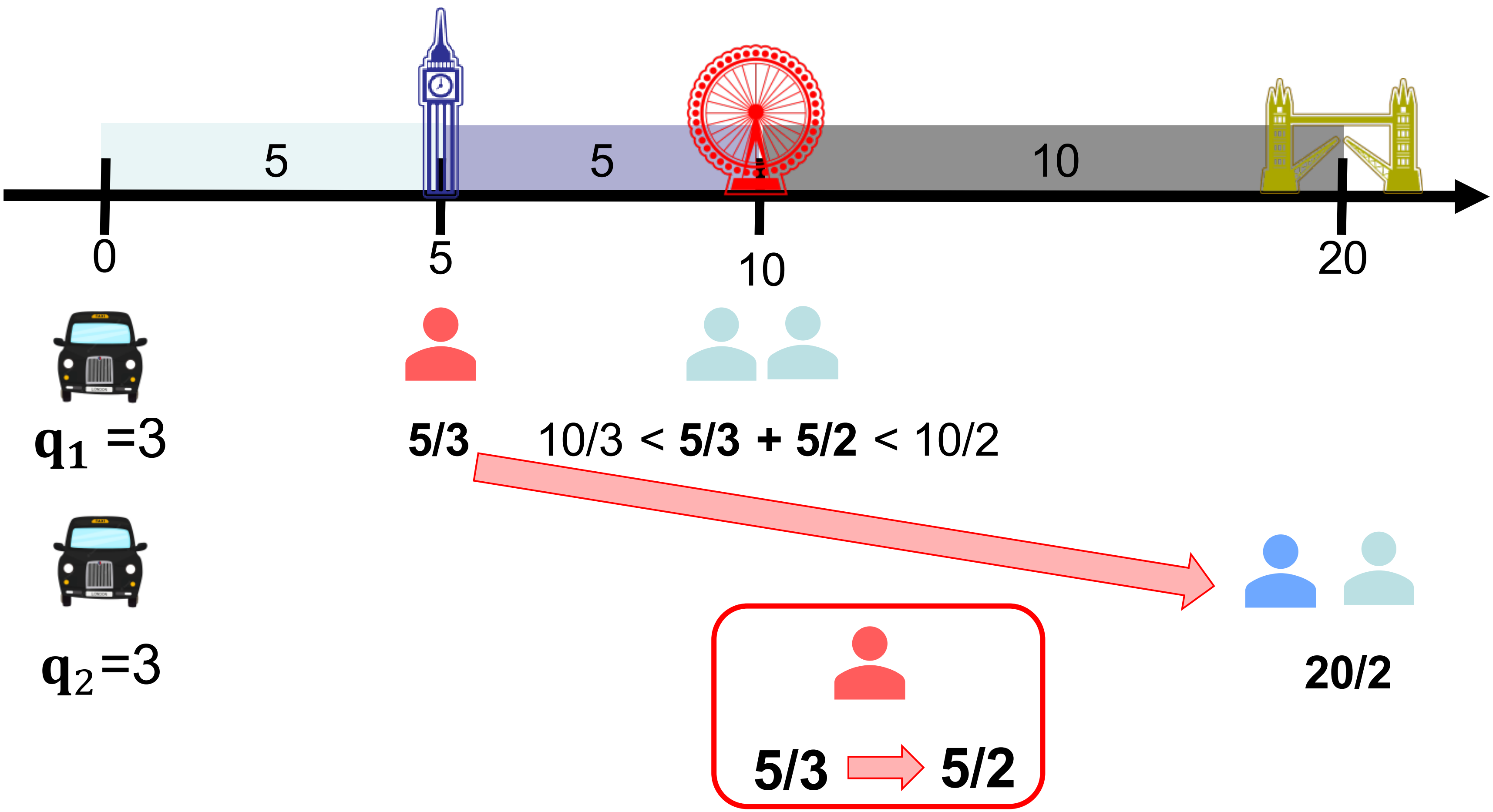
Relationships among concepts

Envy-freeness ⇏ Nash stability



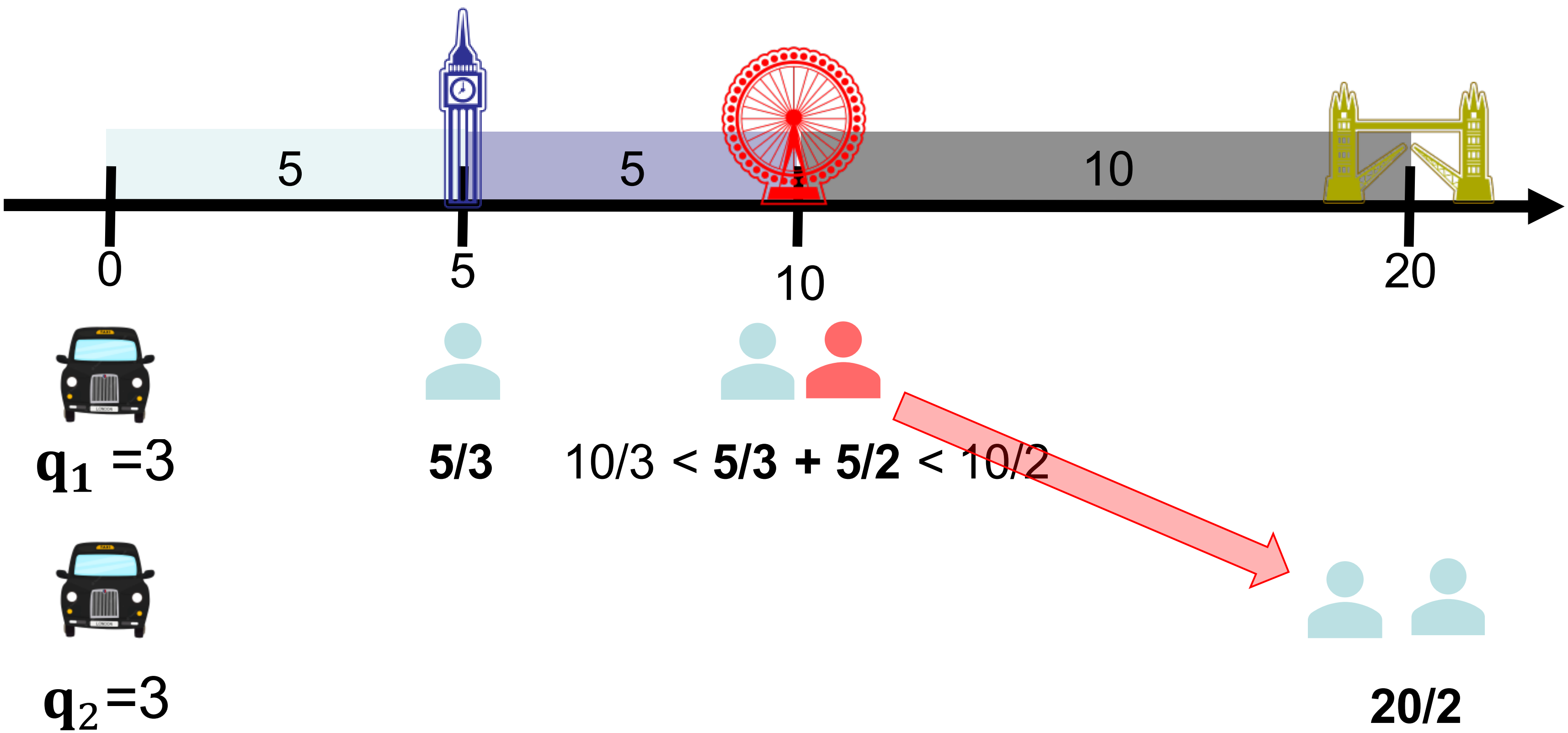
Relationships among concepts

Envy-freeness ⇏ Nash stability



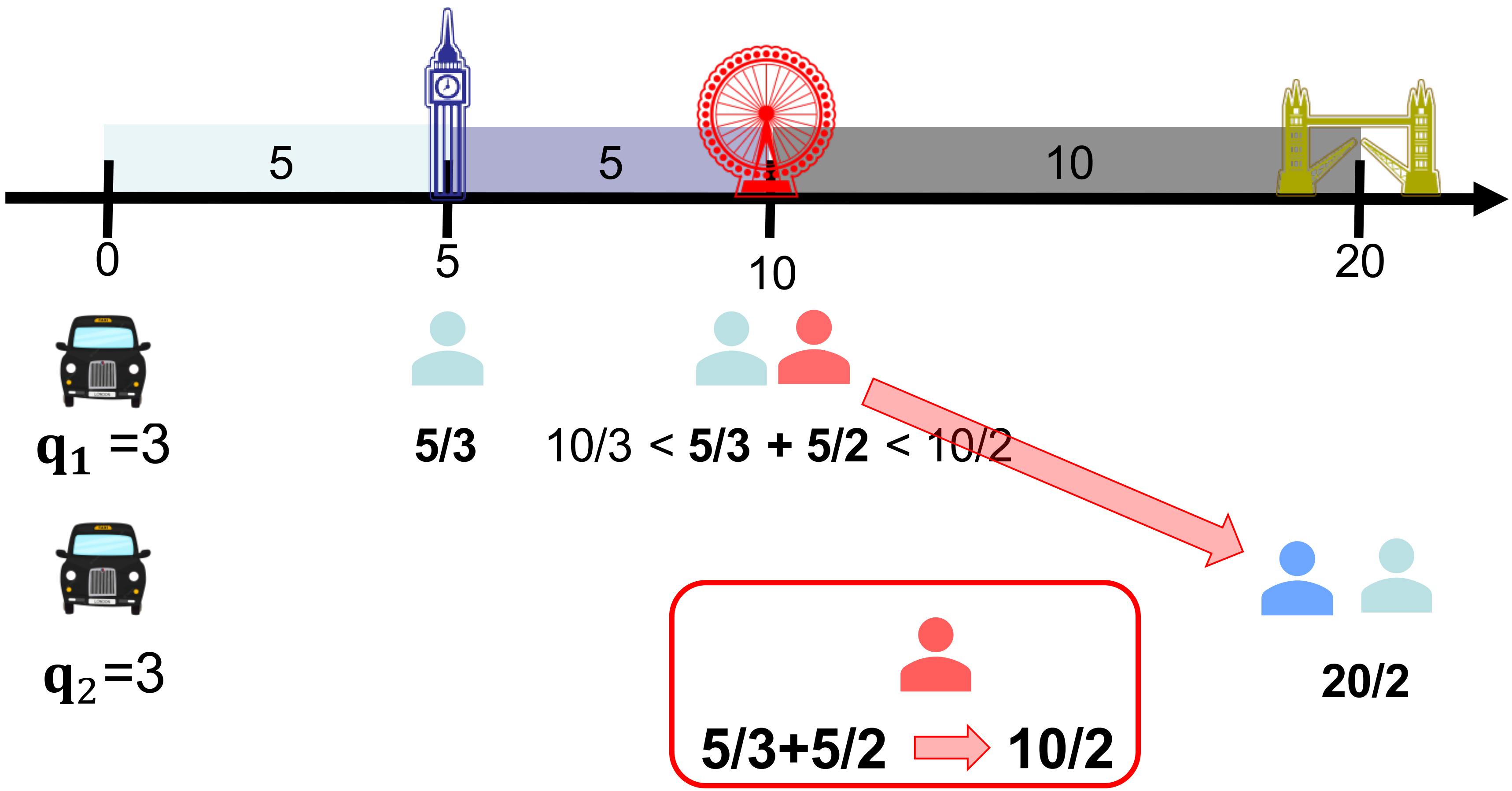
Relationships among concepts

Envy-freeness ⇏ Nash stability



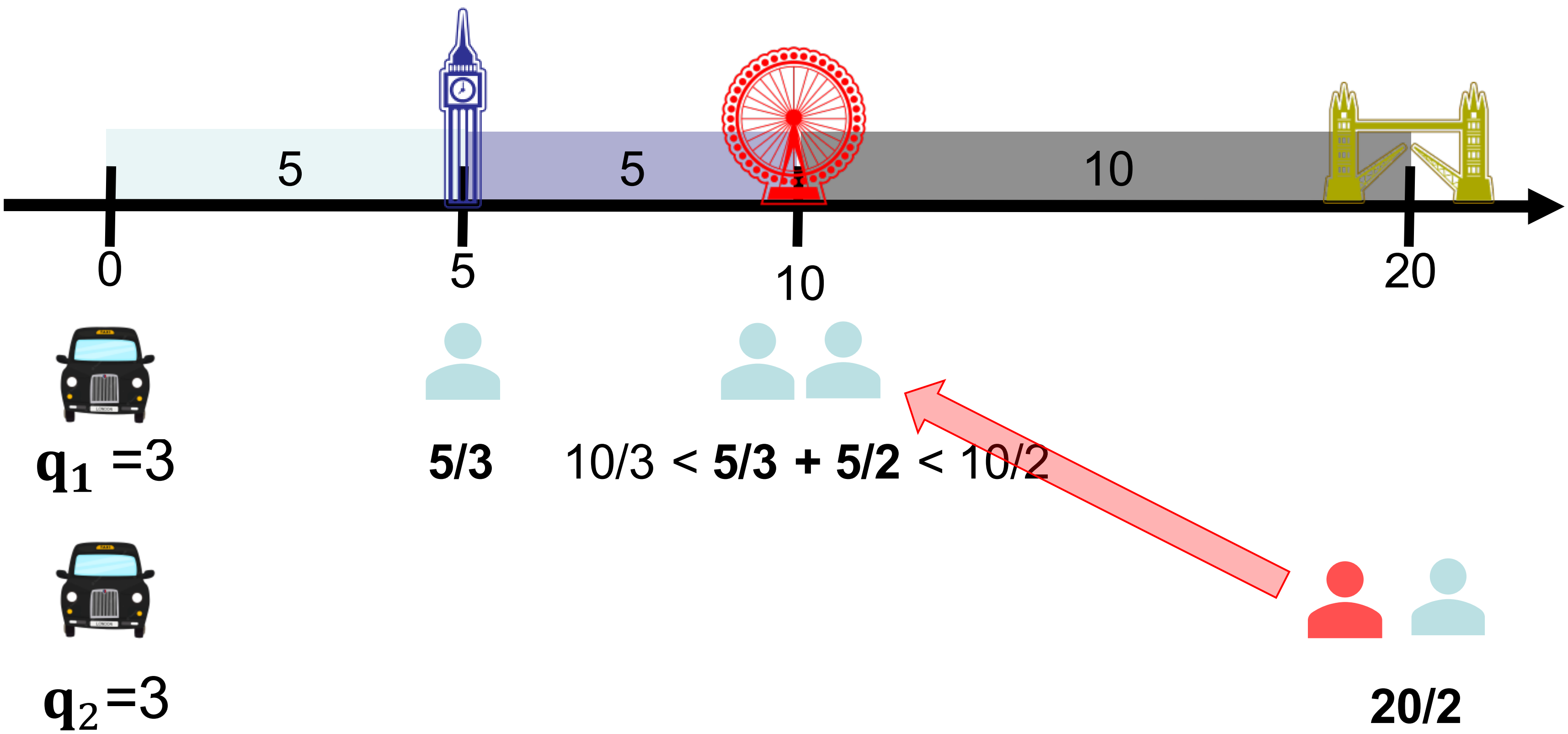
Relationships among concepts

Envy-freeness ⇏ Nash stability



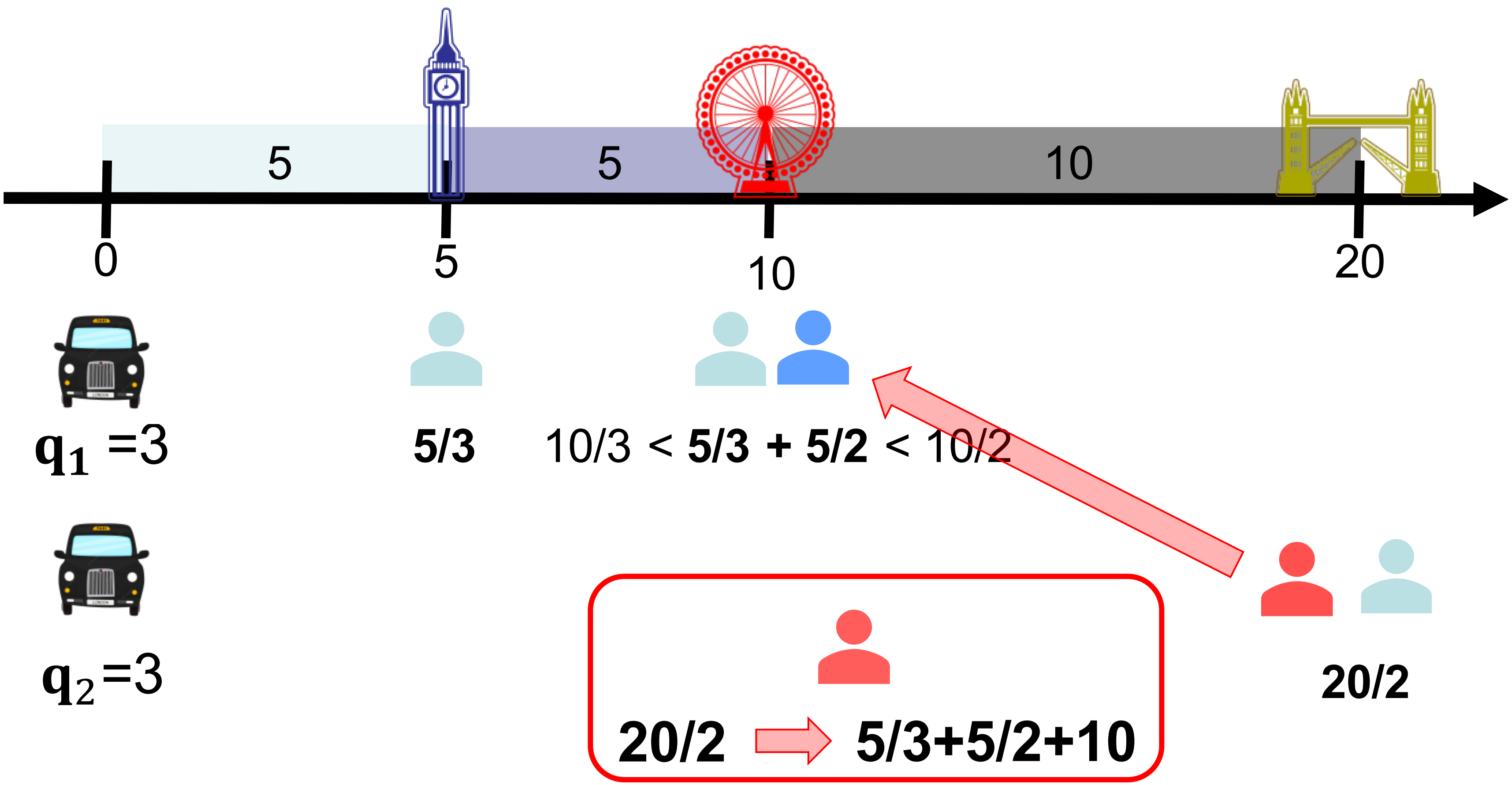
Relationships among concepts

Envy-freeness $\nRightarrow$ Nash stability



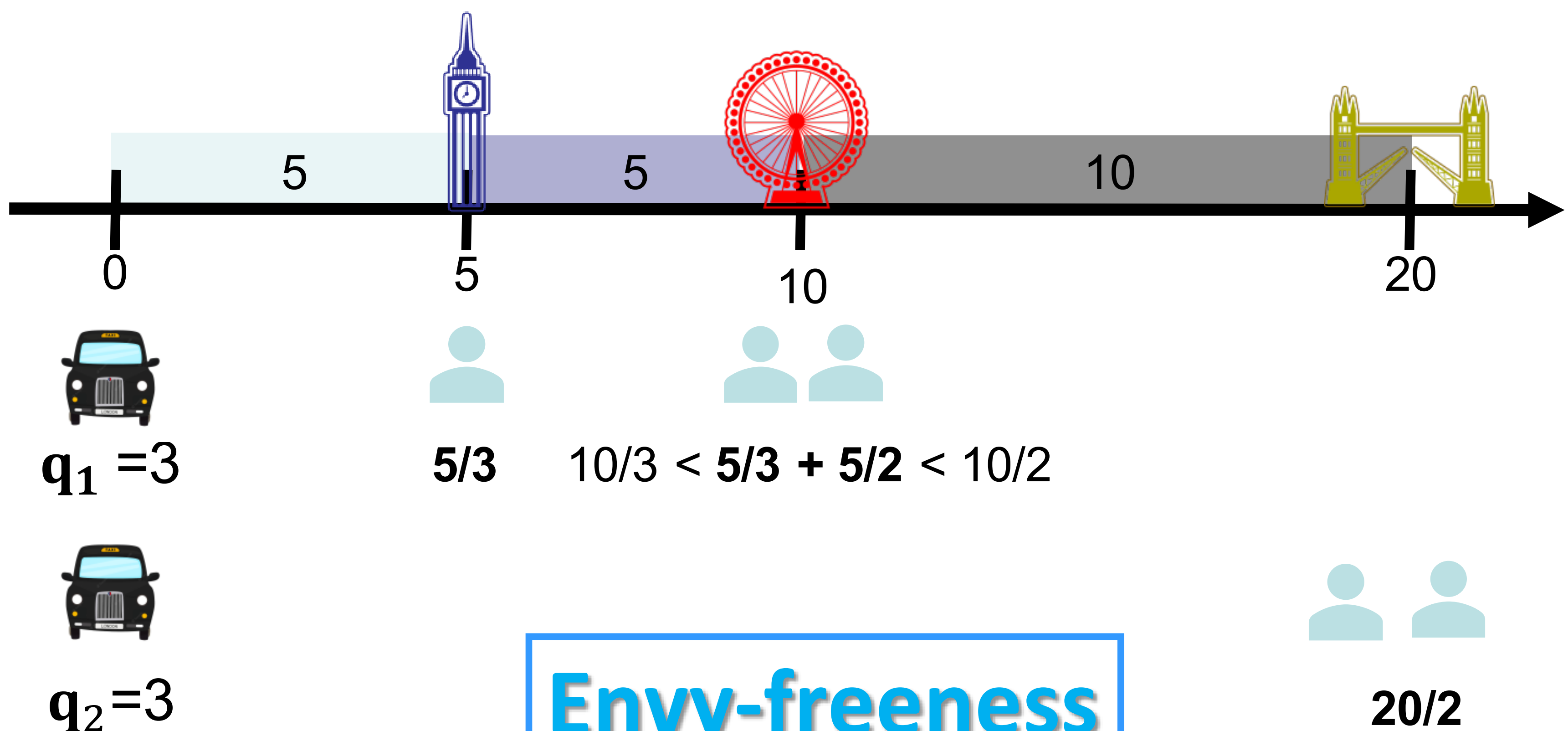
Relationships among concepts

Envy-freeness ⇏ Nash stability



Relationships among concepts

Envy-freeness $\not\Rightarrow$ Nash stability

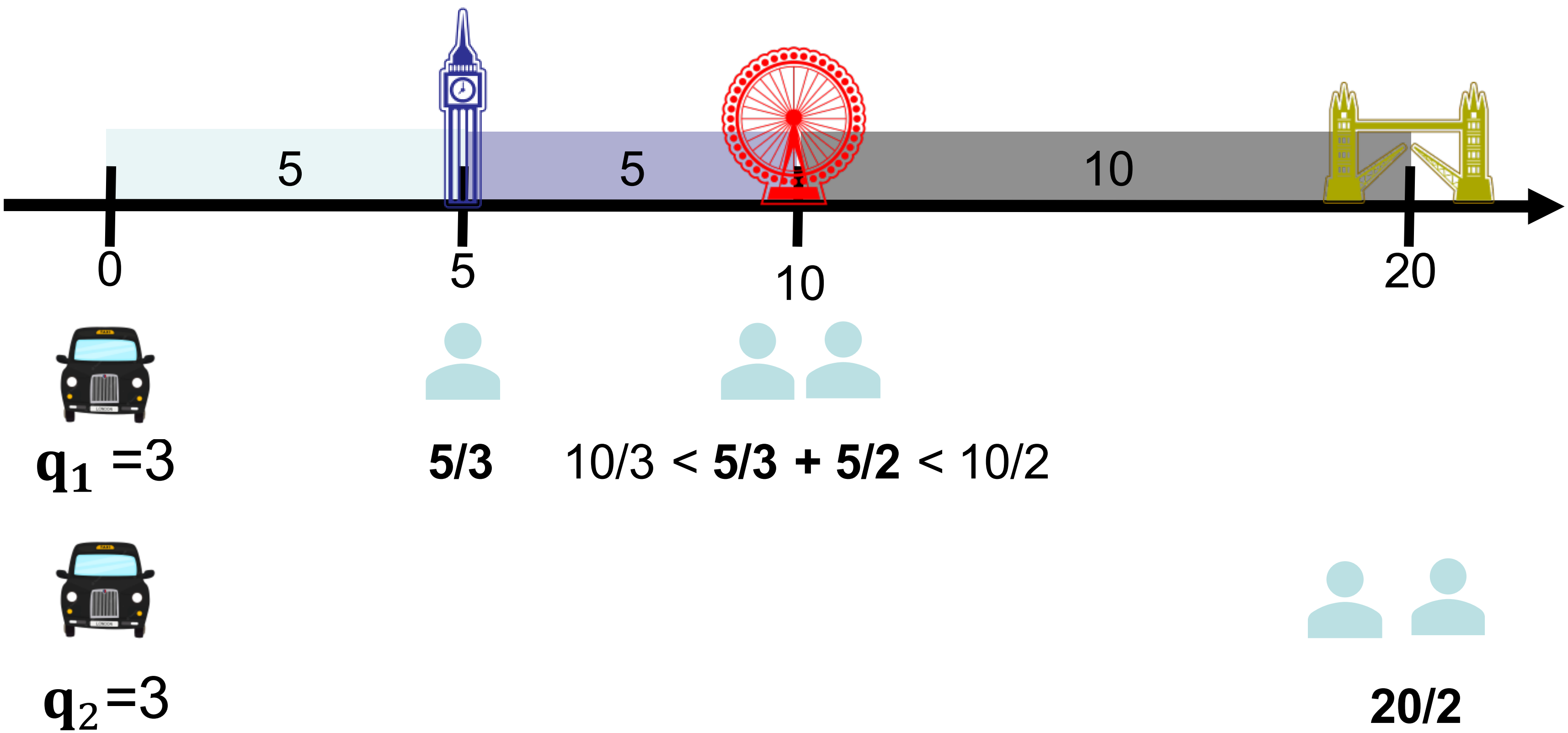


Envy-freeness

Relationships among concepts

Envy-freeness $\not\Rightarrow$ Nash stability

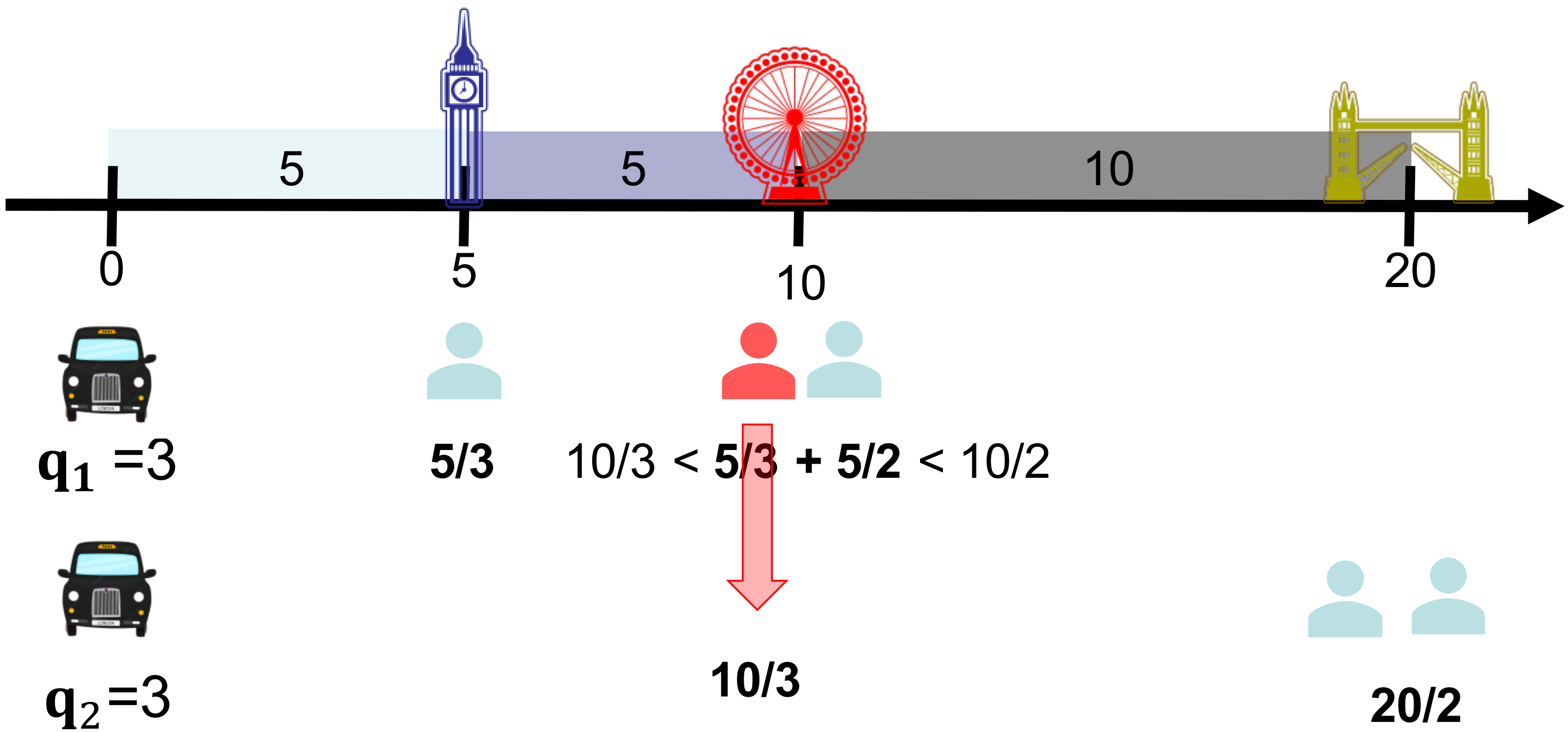
Not Nash stable



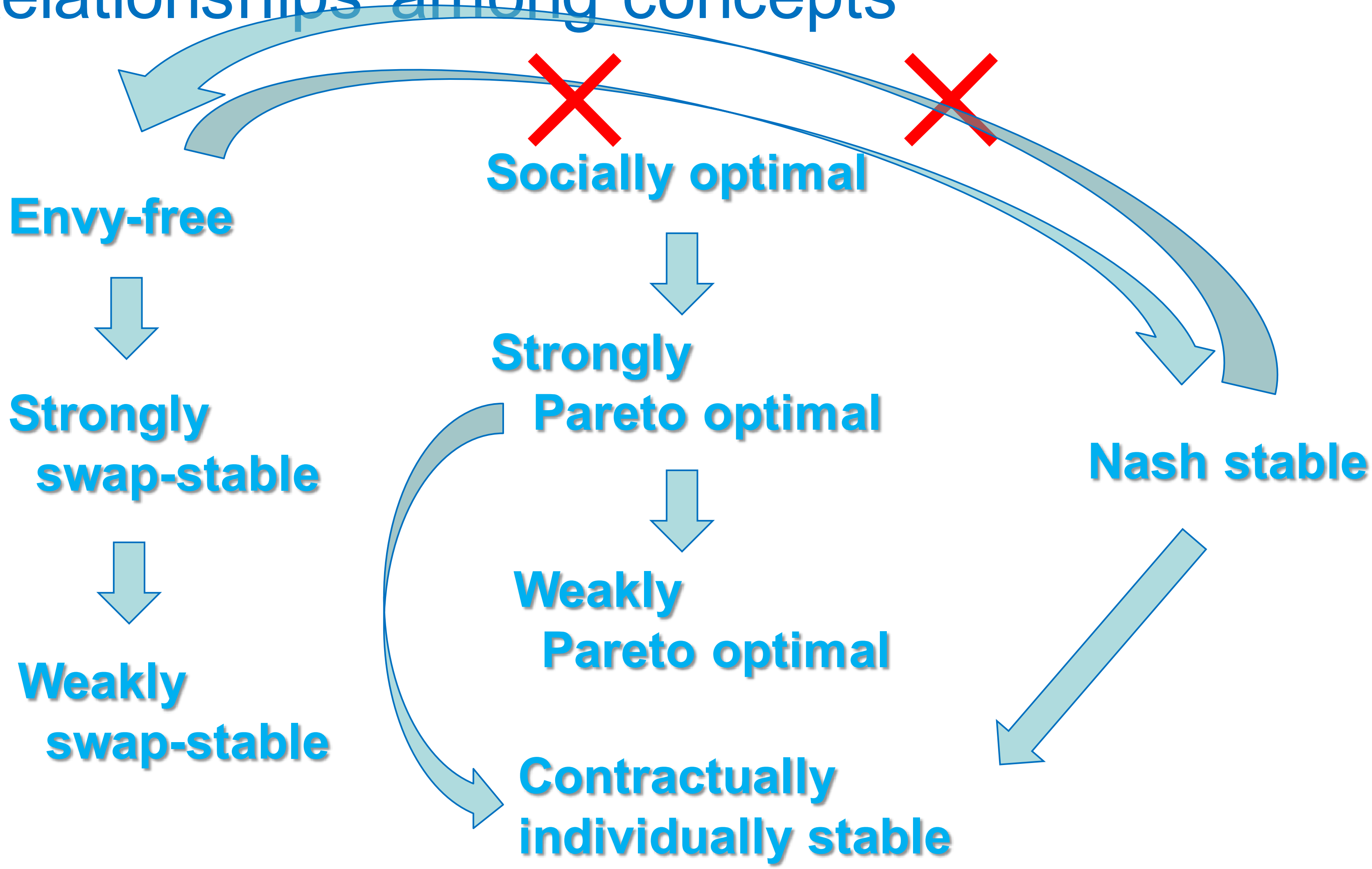
Relationships among concepts

Envy-freeness $\not\Rightarrow$ Nash stability

Not Nash stable



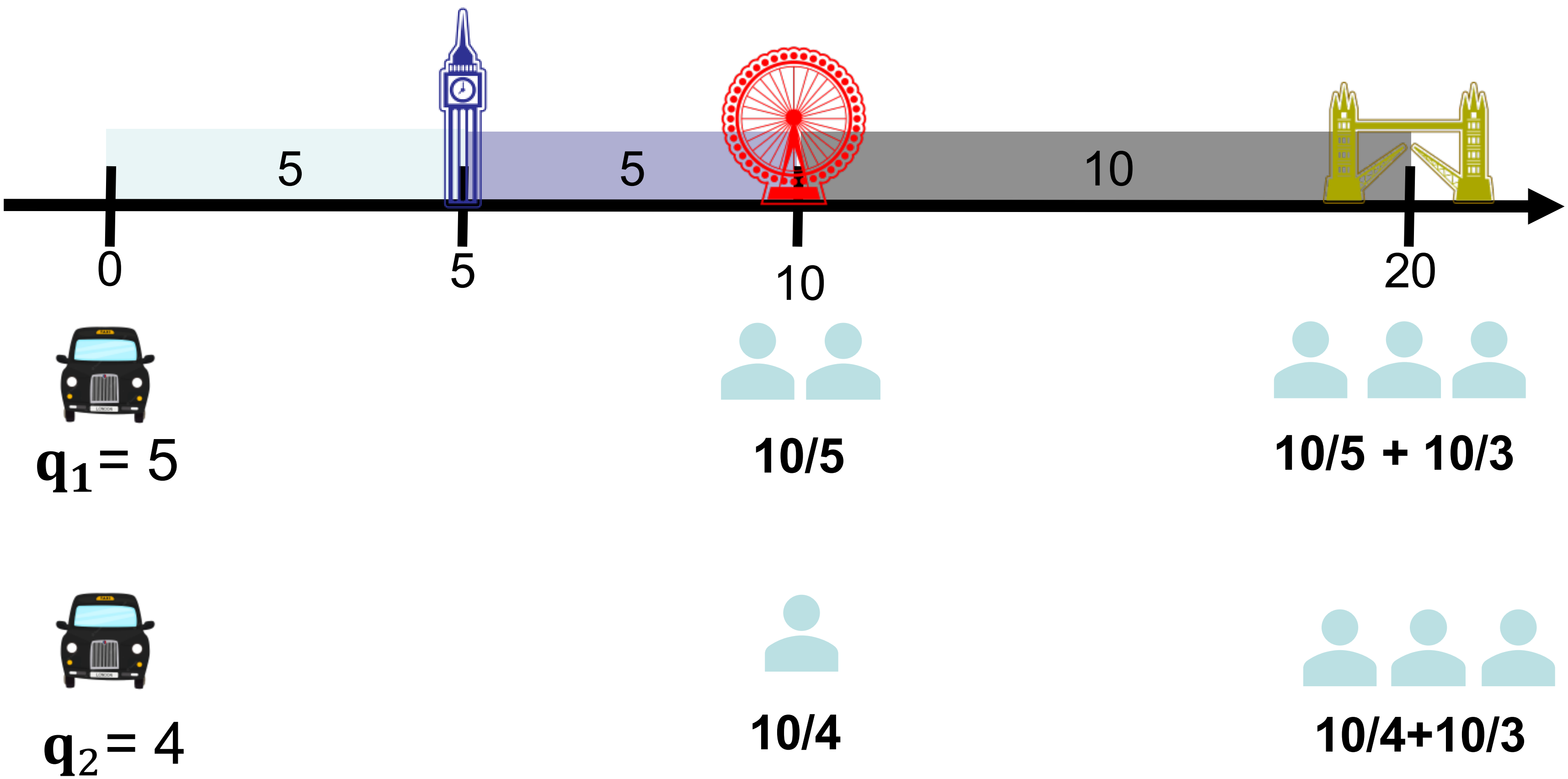
Relationships among concepts



No other implication exists

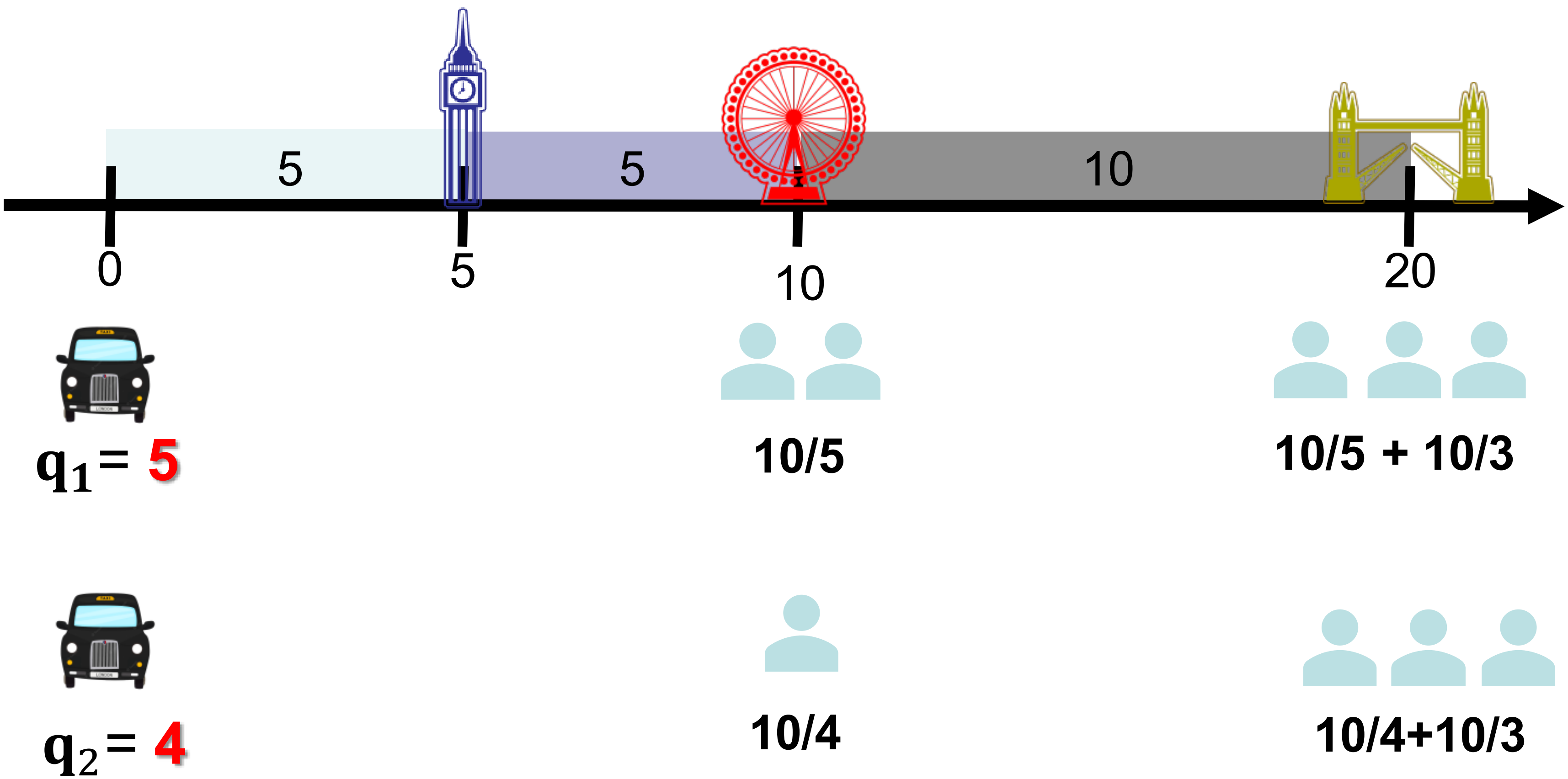
Relationships among concepts

Nash stability $\not\Rightarrow$ Envy-freeness



Relationships among concepts

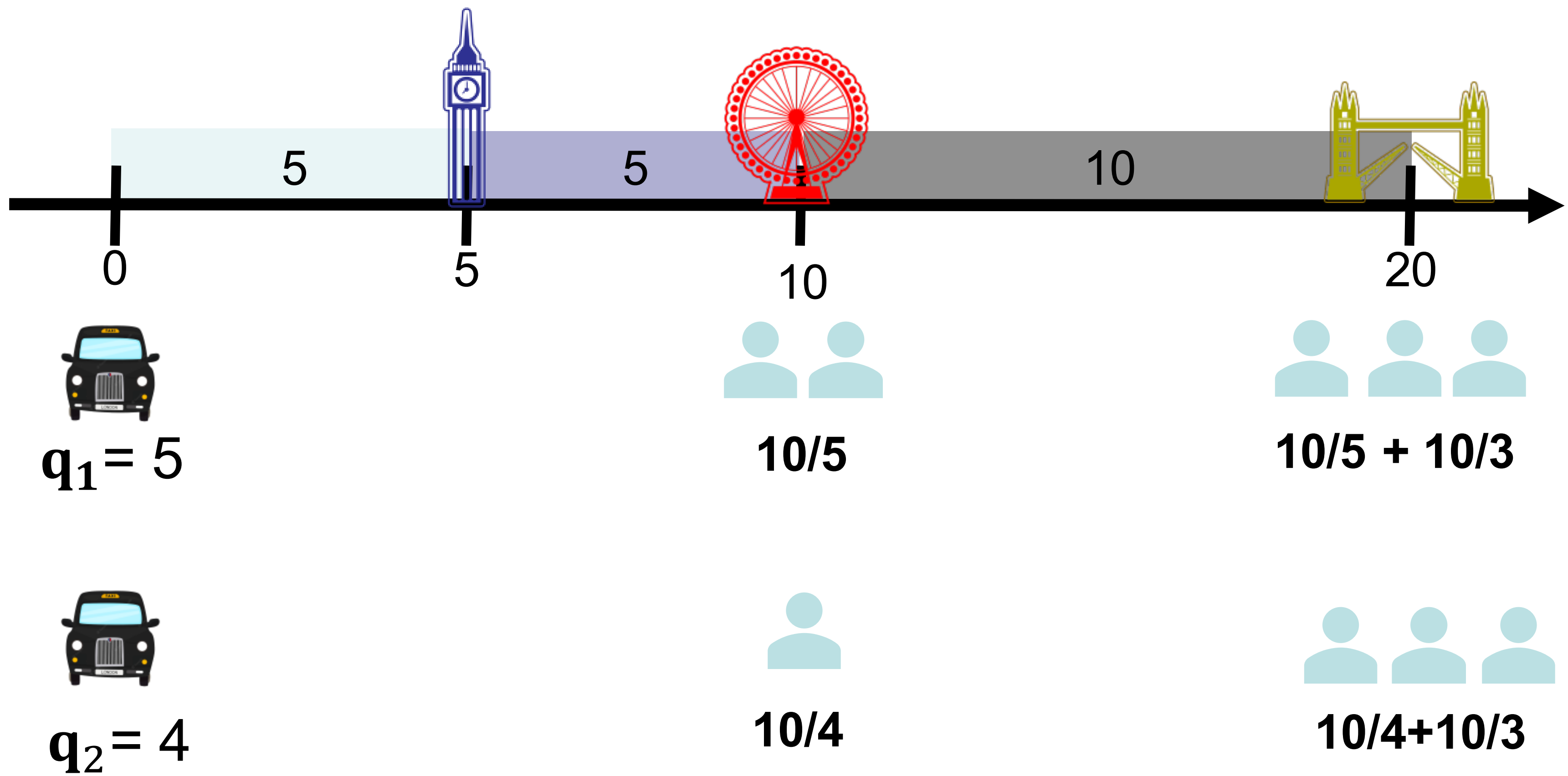
Nash stability $\not\Rightarrow$ Envy-freeness



Relationships among concepts

Nash stability $\not\Rightarrow$ Envy-freeness

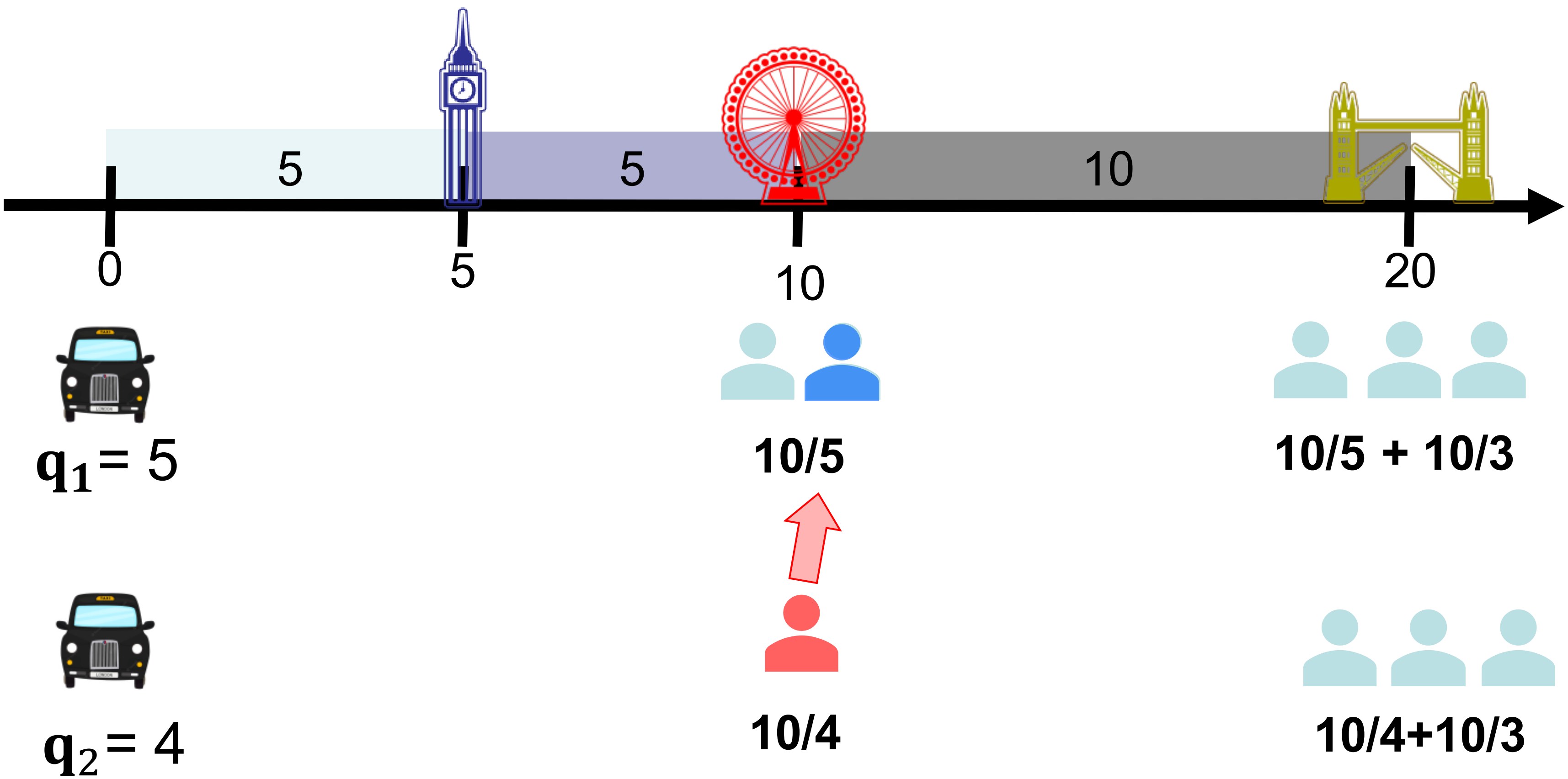
NOT Envy-freeness



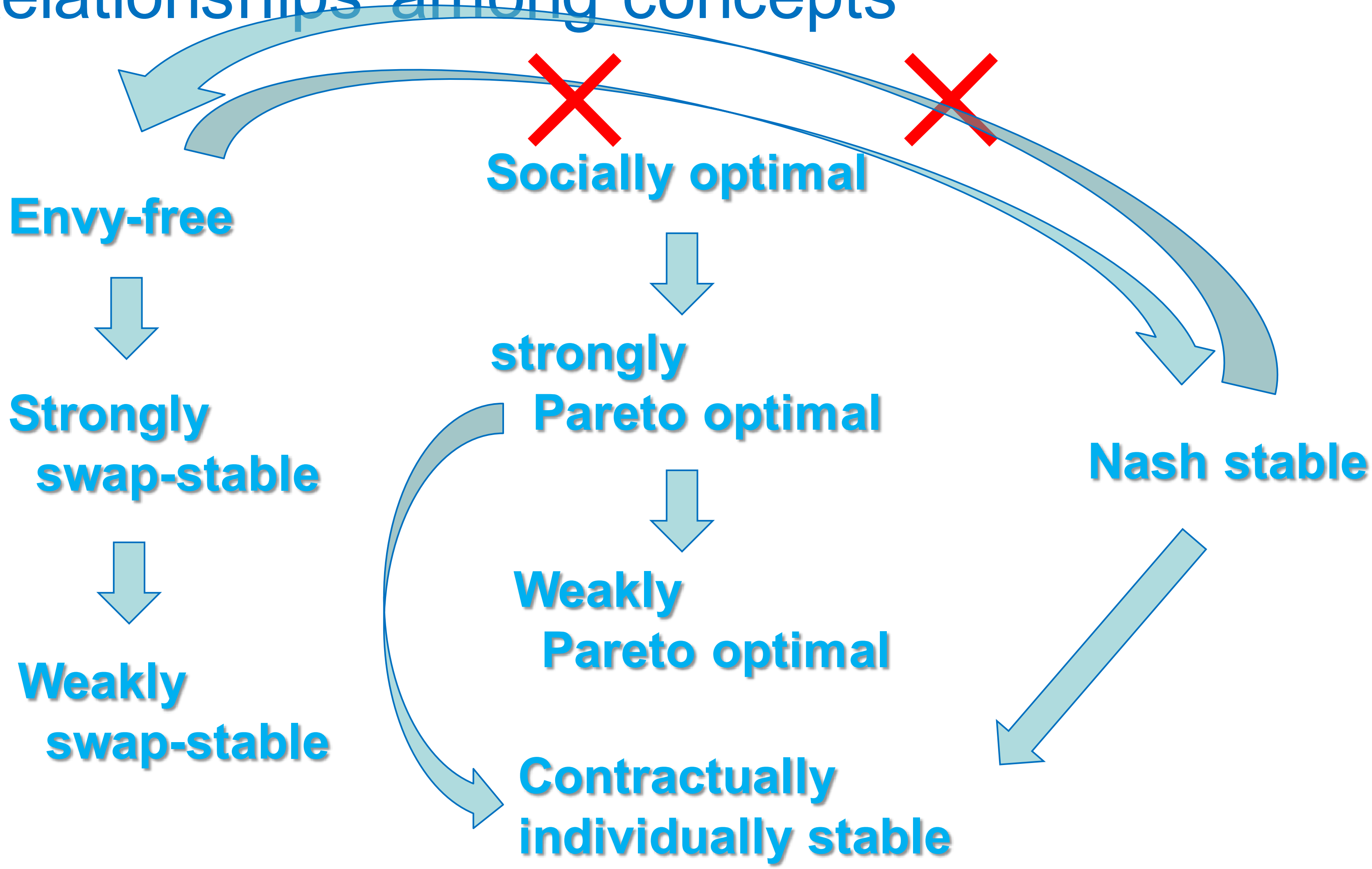
Relationships among concepts

Nash stability ⇏ Envy-freeness

NOT Envy-freeness



Relationships among concepts

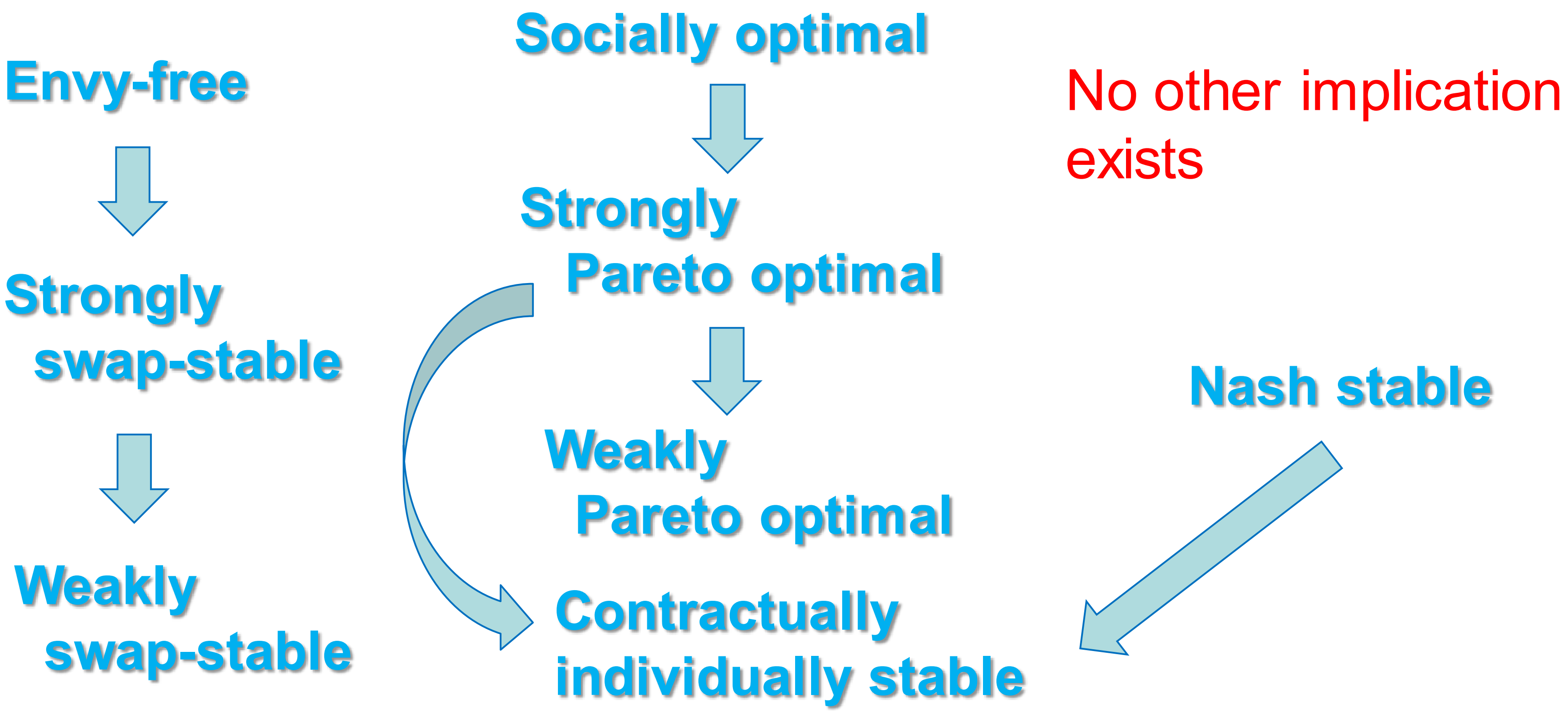


No other implication exists

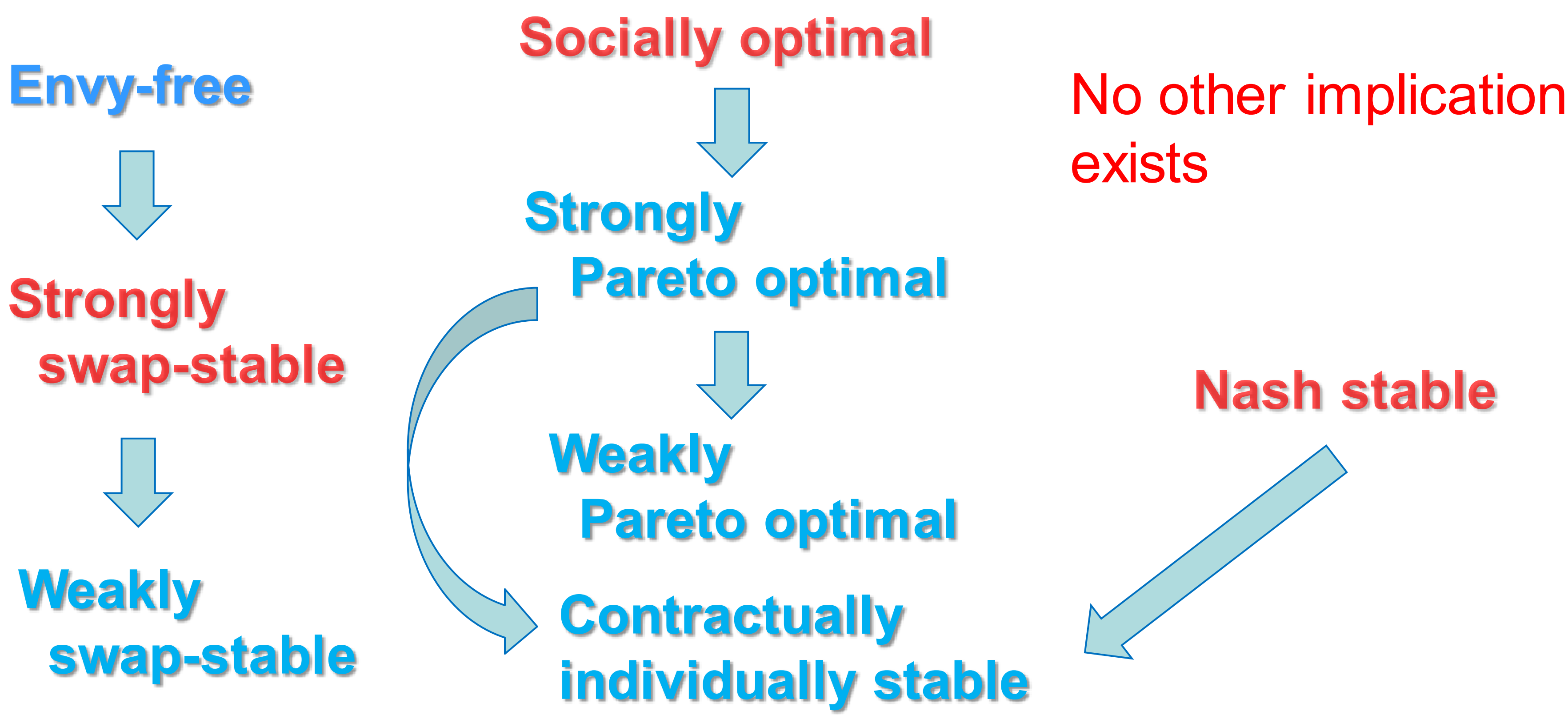
Outline of my talk

1. Airport game
2. Our Model
3. Solution Concepts
4. Relationships among concepts
5. **Algorithmic results**
6. Hardness results
7. Conclusion

Relationships among concepts



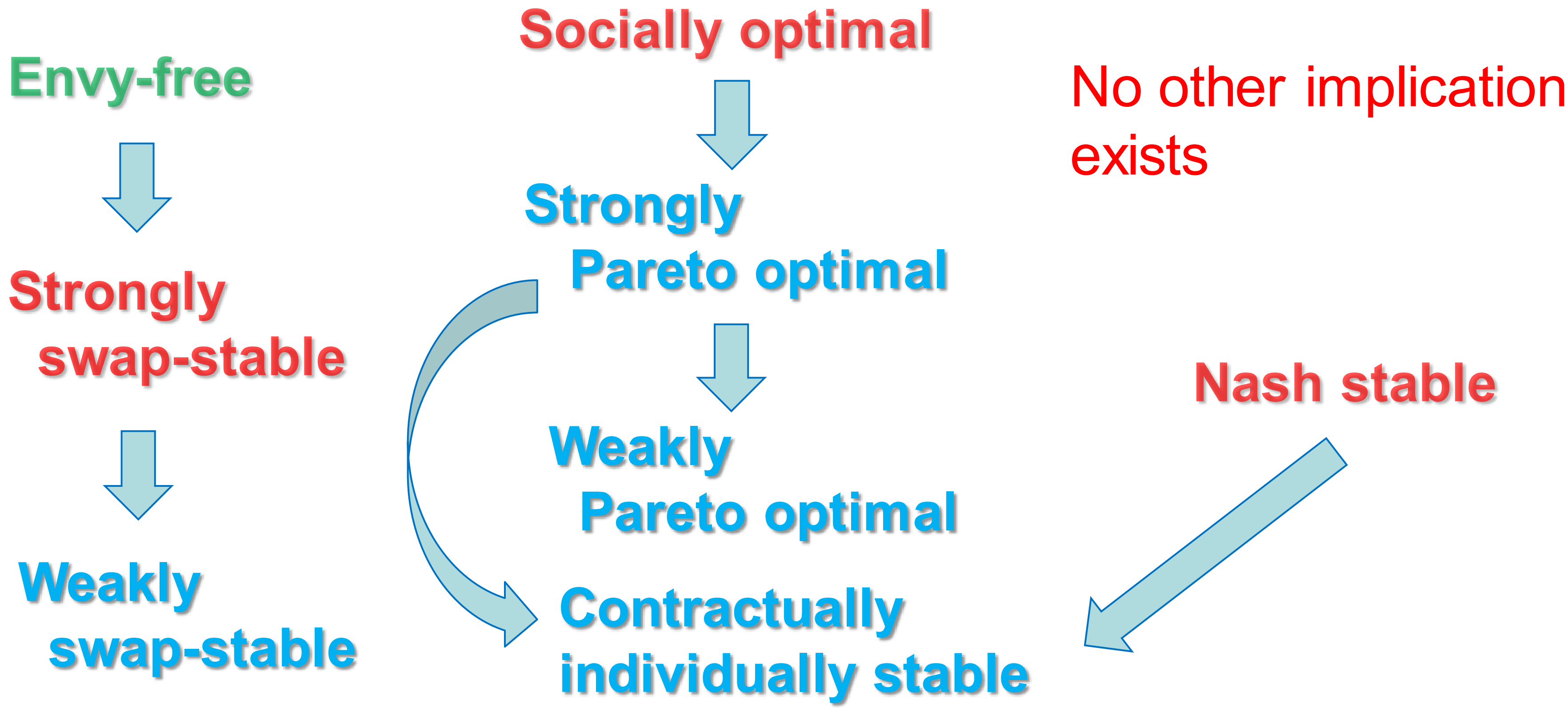
Relationships among concepts



$\exists\ \mathcal{T}$ s.t. **strongly swap-stable, socially optimal, Nash stable**

P-computable

Relationships among concepts



$\exists\ \mathcal{T}$ s.t. strongly **swap-stable**, socially optimal, Nash stable

P-computable

Envy-free $\mathcal{T}$ does **not** always exist

Structual properties and several P-computable cases

Algorithmic results

P-computable

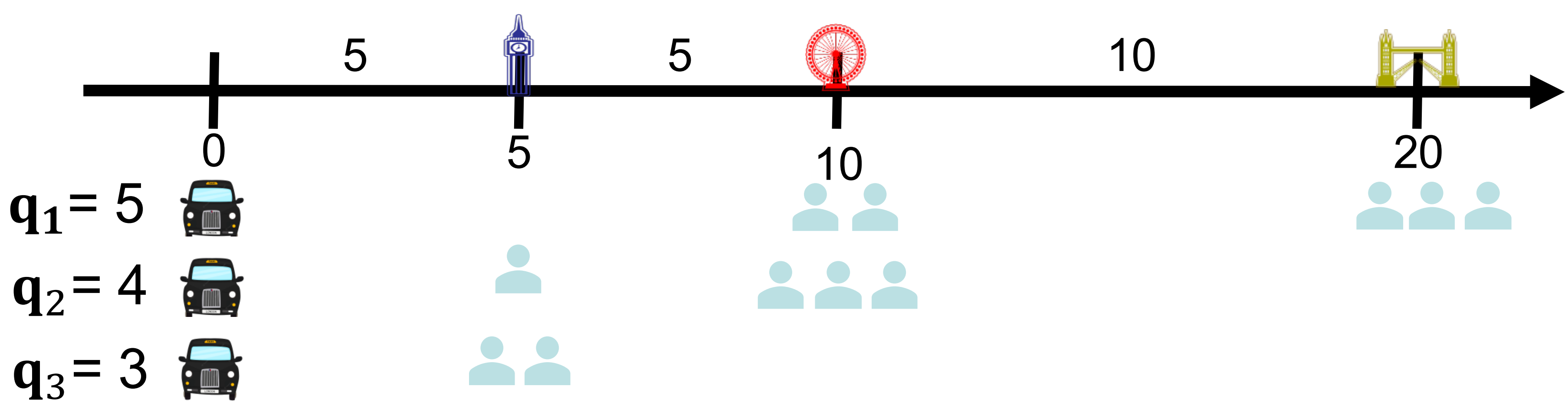
$\exists\ \mathcal{T}$ s.t. **strongly swap-stable**, **socially optimal, Nash stable**

strongly swap-stable: $\nexists$ no two agents a and b
s.t. a envies b & b can replace a

Socially optimal: minimize the total payment

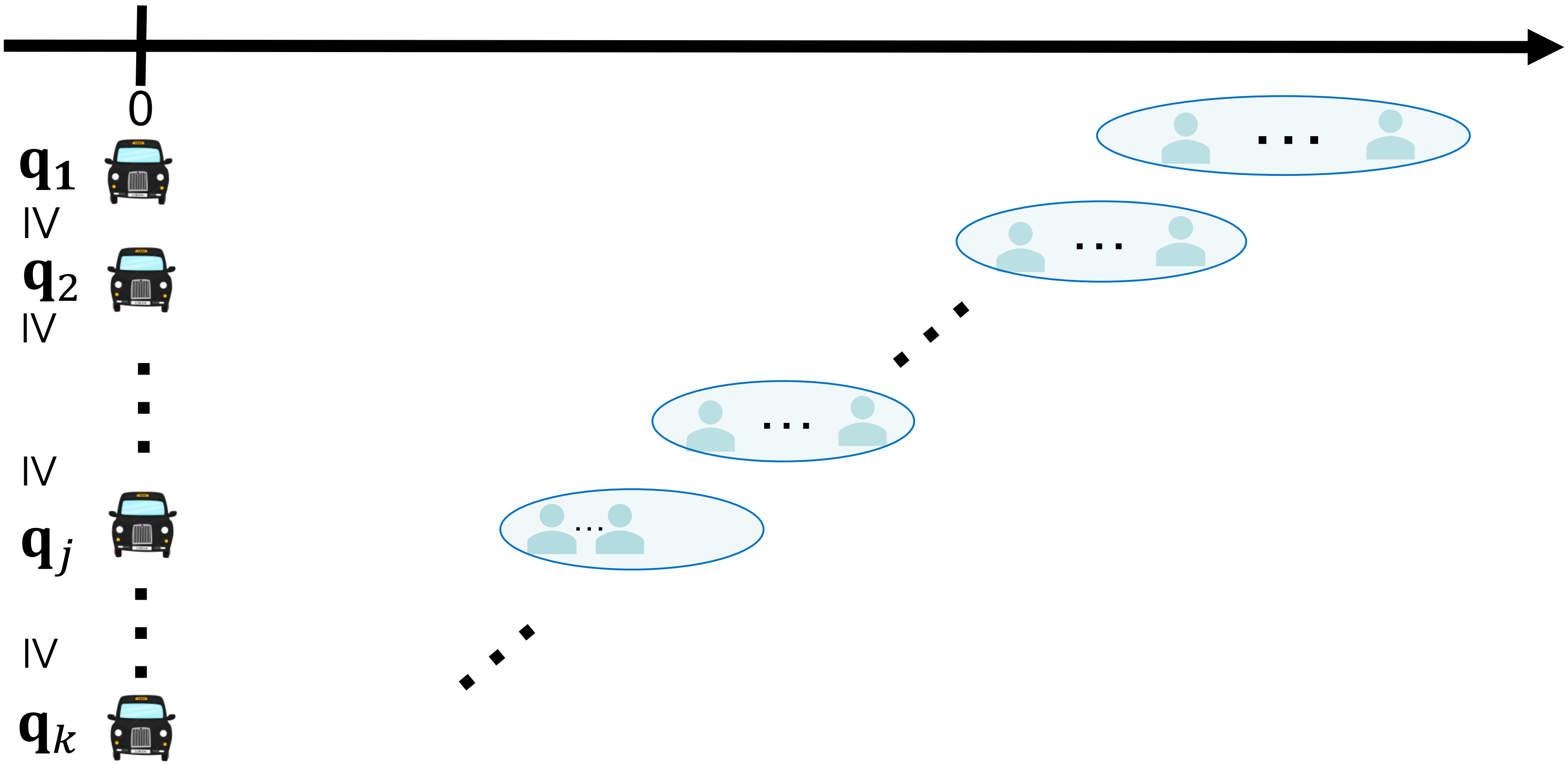
Nash stable: no agent can reduce her cost
by deviating to another taxi

Backward greedy: greedily add agents a from the furthest $\mathbf{x}_a$ to taxi T_i with smallest available i, where $\mathbf{q_1} \geq \mathbf{q_2} \geq \cdots \geq \mathbf{q_k}$



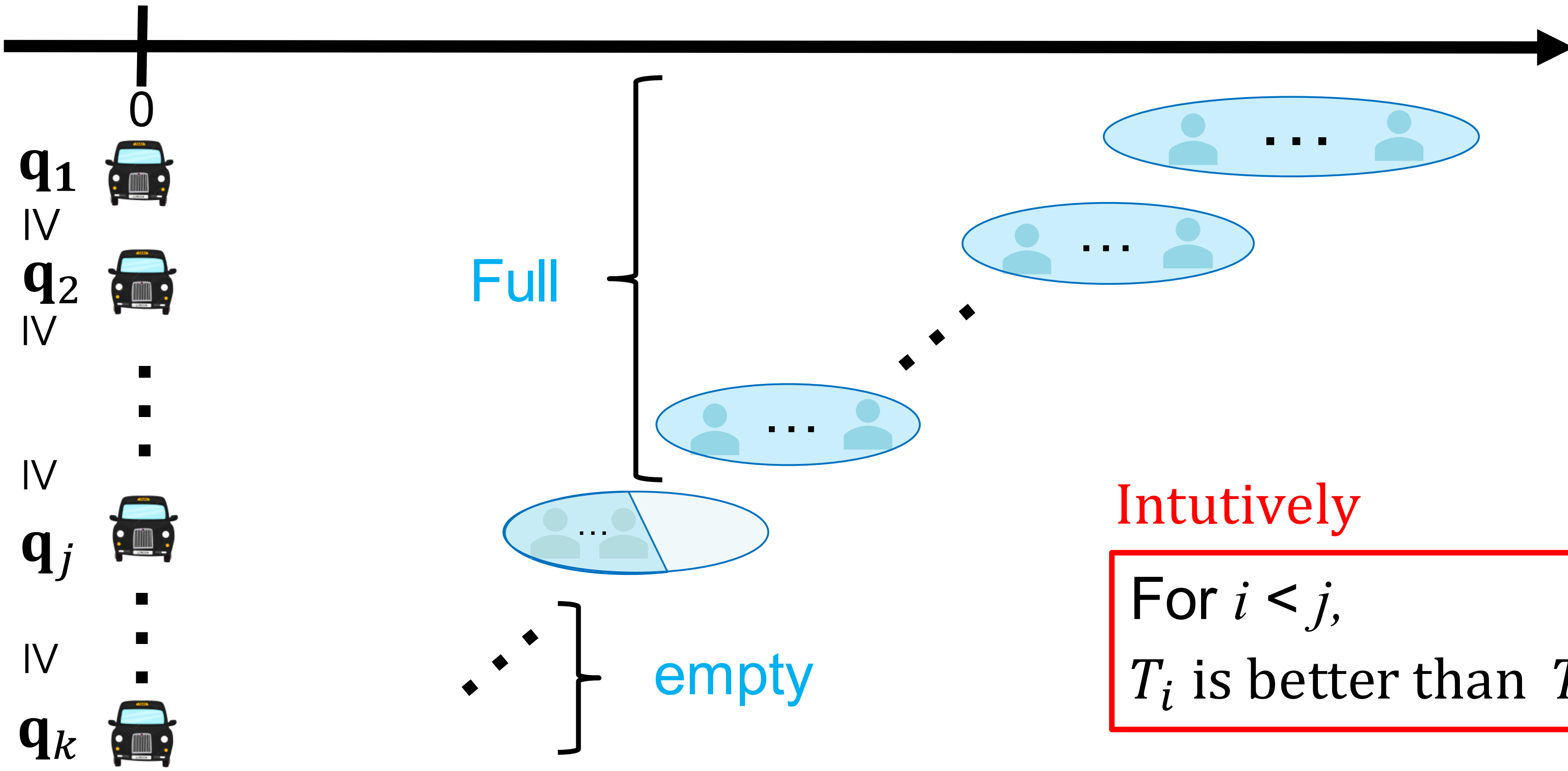
Algorithmic results

Backward greedy: greedily add agents a from the furthest $\mathbf{x}_a$ to taxi T_i with smallest available i, where $\mathbf{q_1} \geq \mathbf{q_2} \geq \cdots \geq \mathbf{q_k}$



Algorithmic results

Backward greedy: greedily add agents a from the furthest $\mathbf{x}_a$ to taxi T_i with smallest available i, where $\mathbf{q_1} \geq \mathbf{q_2} \geq \cdots \geq \mathbf{q_k}$

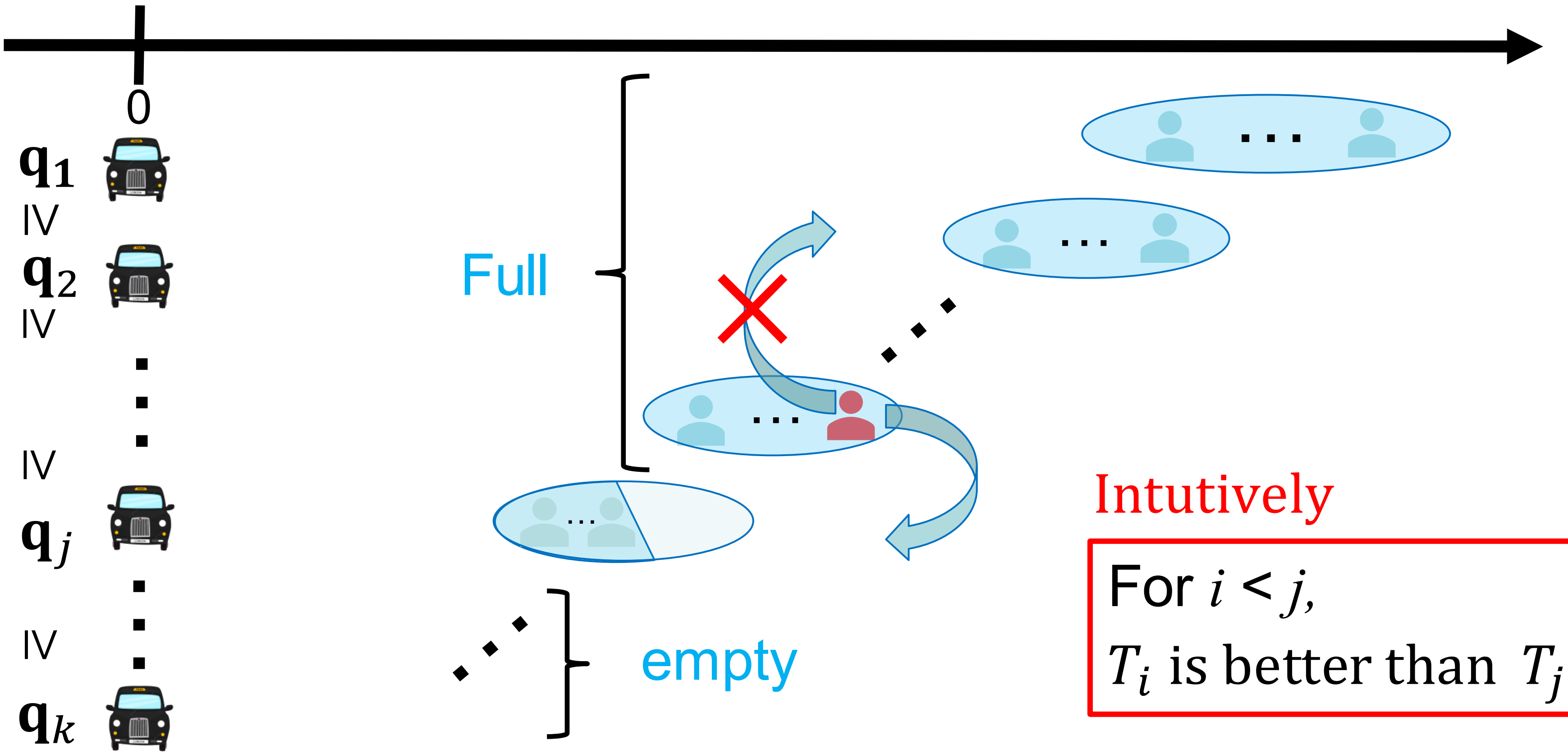


Intutively

For $i < j$,
T_i is better than T_j

Algorithmic results

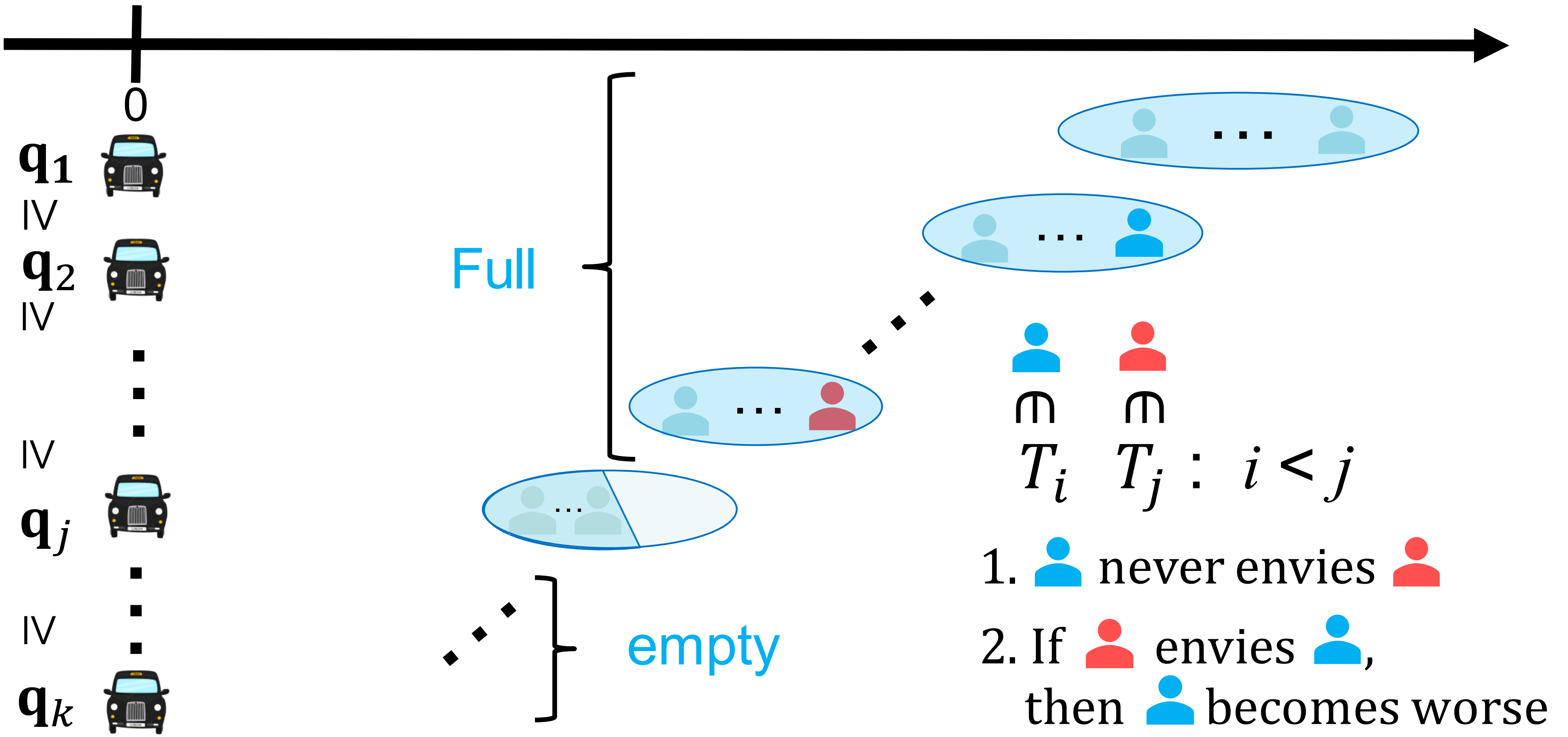
Backward greedy: greedily add agents a from the furthest $\mathbf{x}_a$ to taxi T_i with smallest available i, where $\mathbf{q_1} \geq \mathbf{q_2} \geq \cdots \geq \mathbf{q_k}$



Nash stable: no agent can reduce her cost by deviating to another taxi

Algorithmic results

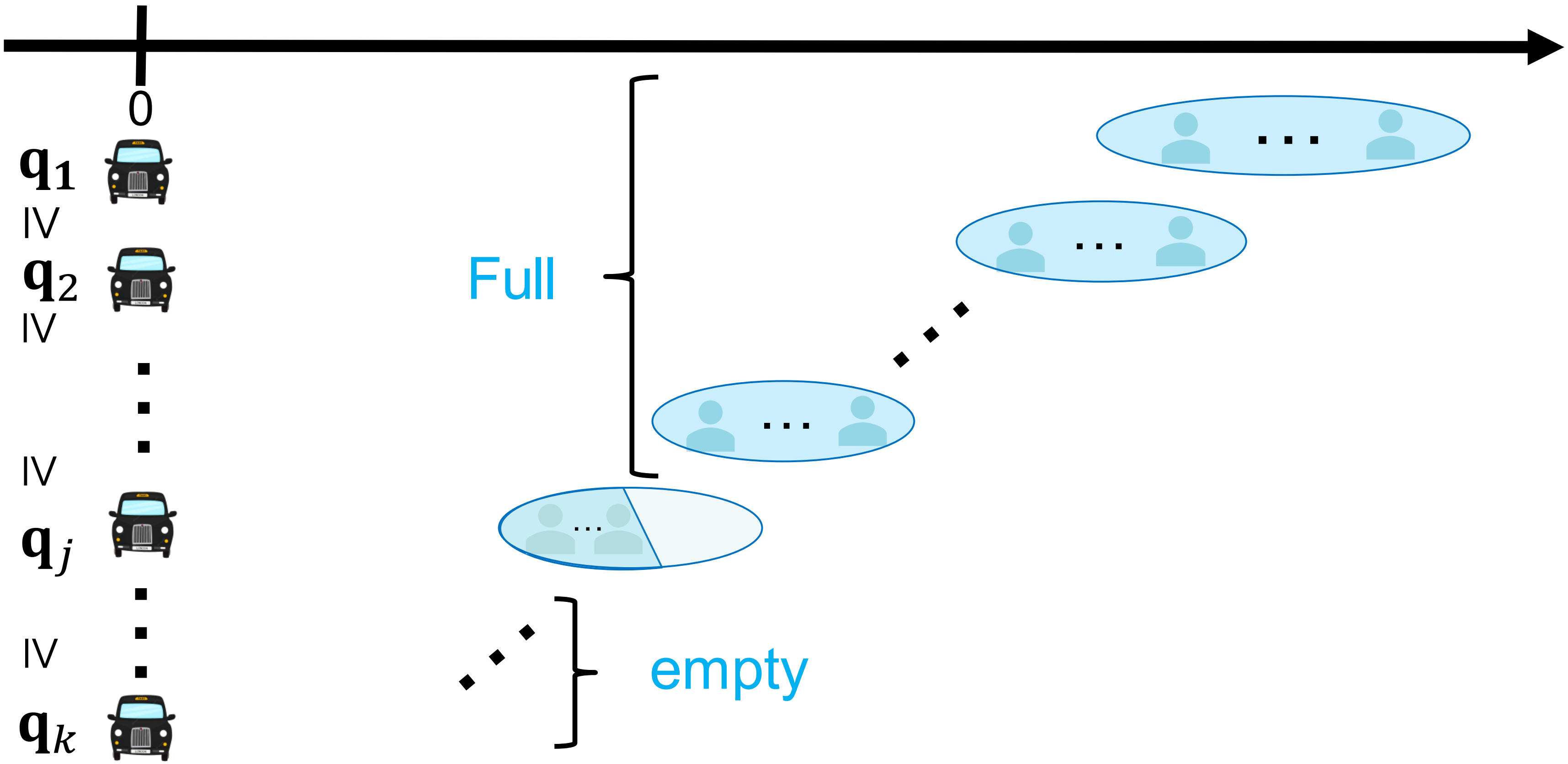
Backward greedy: greedily add agents a from the furthest $\mathbf{x}_a$ to taxi T_i with smallest available i, where $\mathbf{q_1} \geq \mathbf{q_2} \geq \cdots \geq \mathbf{q_k}$



strongly swap-stable: ∄ no two agents a and b s.t. a envies b & b can replace a

Algorithmic results

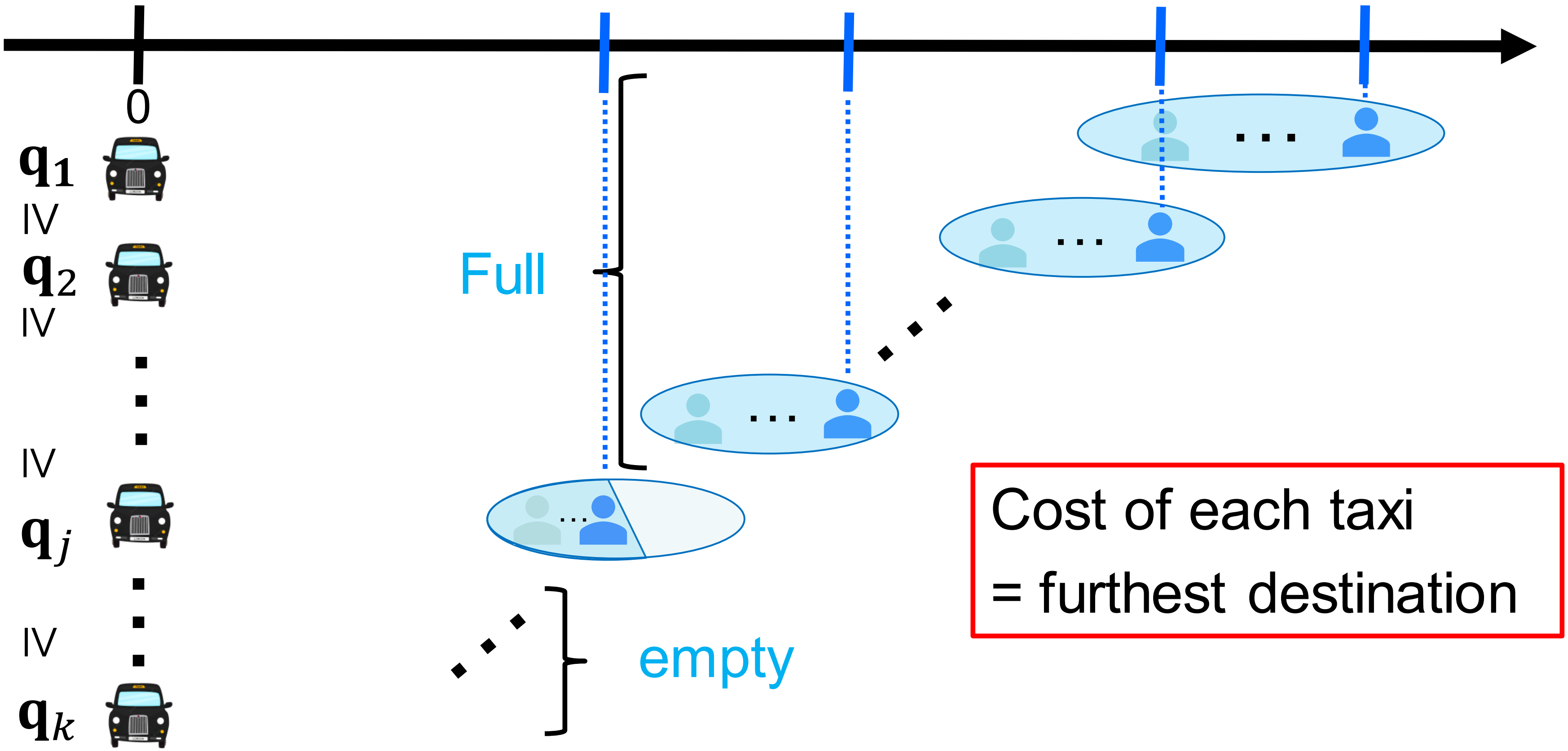
Backward greedy: greedily add agents a from the furthest $\mathbf{x}_a$ to taxi T_i with smallest available i, where $\mathbf{q_1} \geq \mathbf{q_2} \geq \cdots \geq \mathbf{q_k}$



Socially optimal: minimize the total payment

Algorithmic results

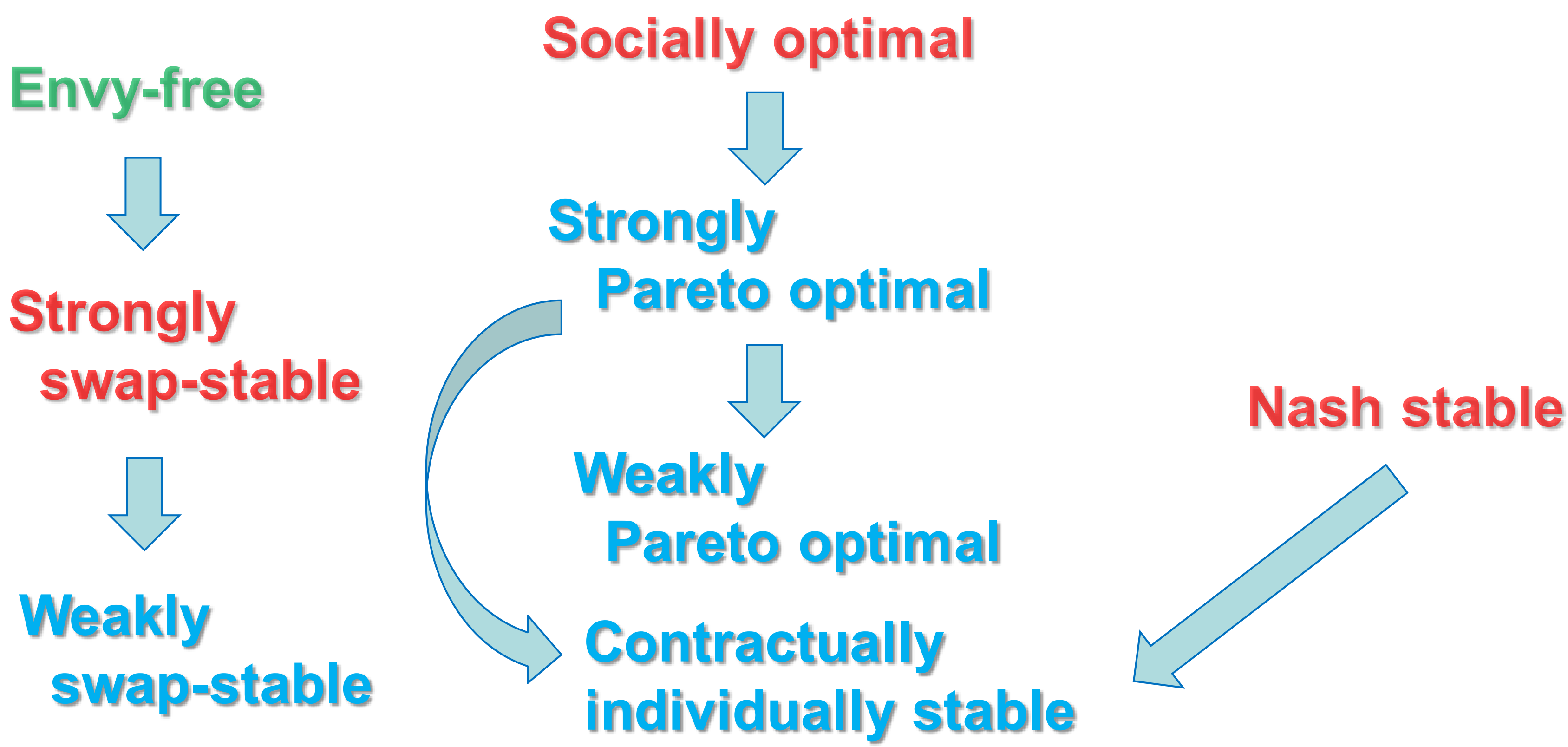
Backward greedy: greedily add agents a from the furthest $\mathbf{x}_a$ to taxi T_i with smallest available i, where $\mathbf{q_1} \geq \mathbf{q_2} \geq \cdots \geq \mathbf{q_k}$



Socially optimal: minimize the total payment

Algorithmic results

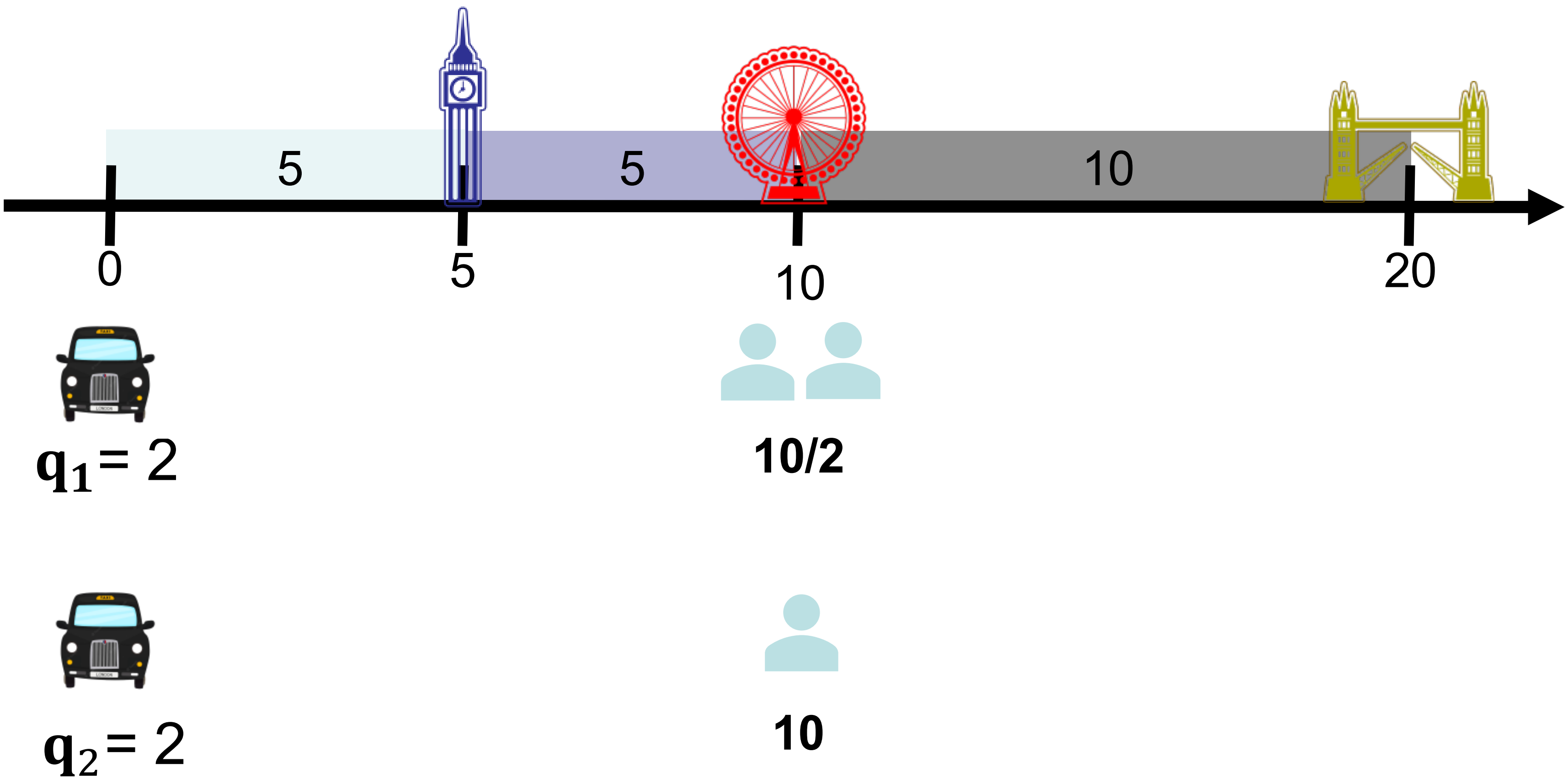
Th. Backward greedy efficiently computes an allocation which is **Nash stable**, **strongly swap-stable**, and **socially optimal**.



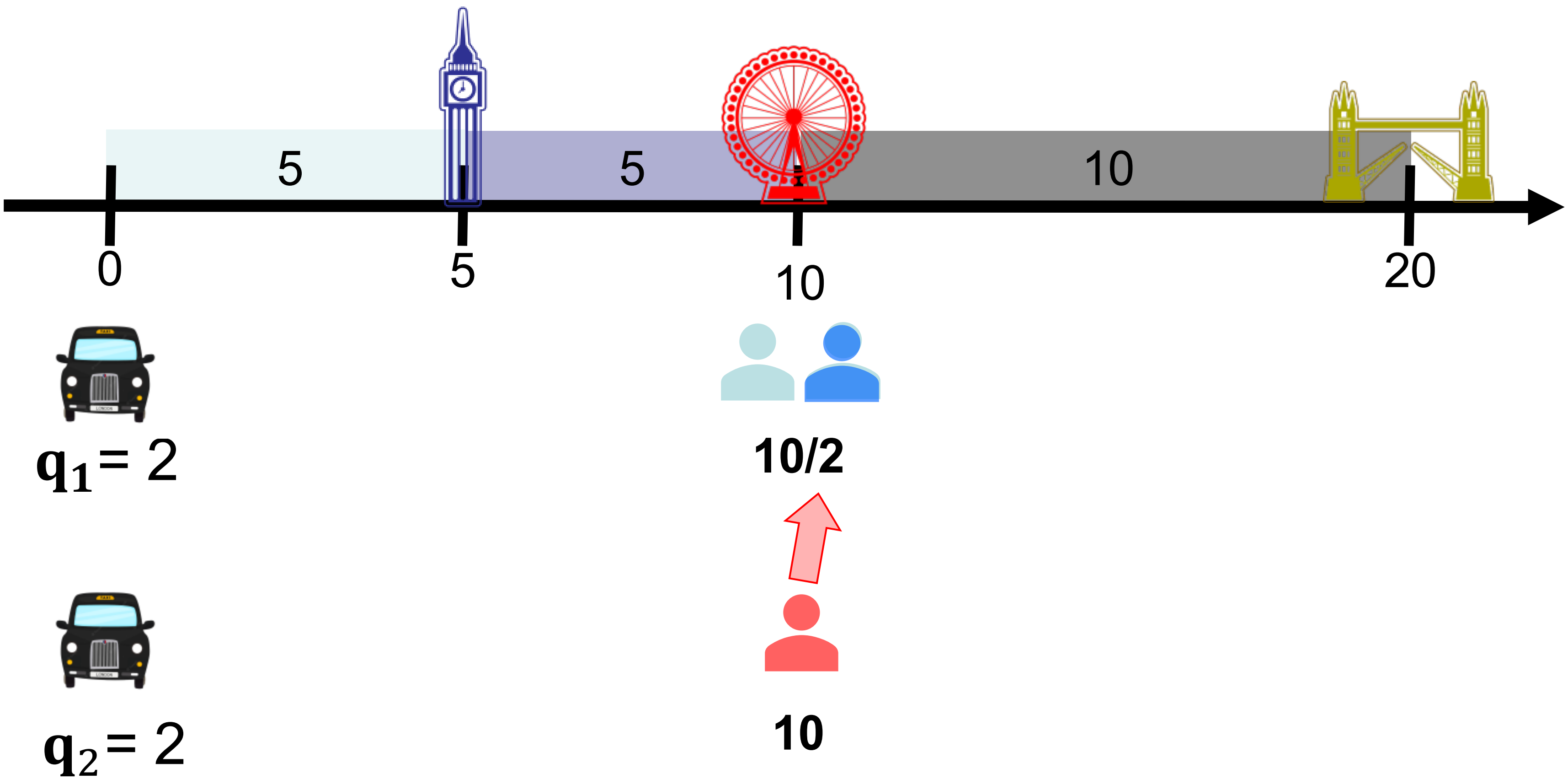
Envy-free:

- Envy-free allocations may not exist.
- Three structural properties:
  monotonicity, split property, and locality
- Efficient algorithms for envy-freeness when
  (1) the number of taxis is a constant,
  (2) the capacity of each taxi is at most four, or
  (3) the number of destination types is small.
- NP-hardness for two relaxations of envy-freeness

Envy-free allocations may not exist

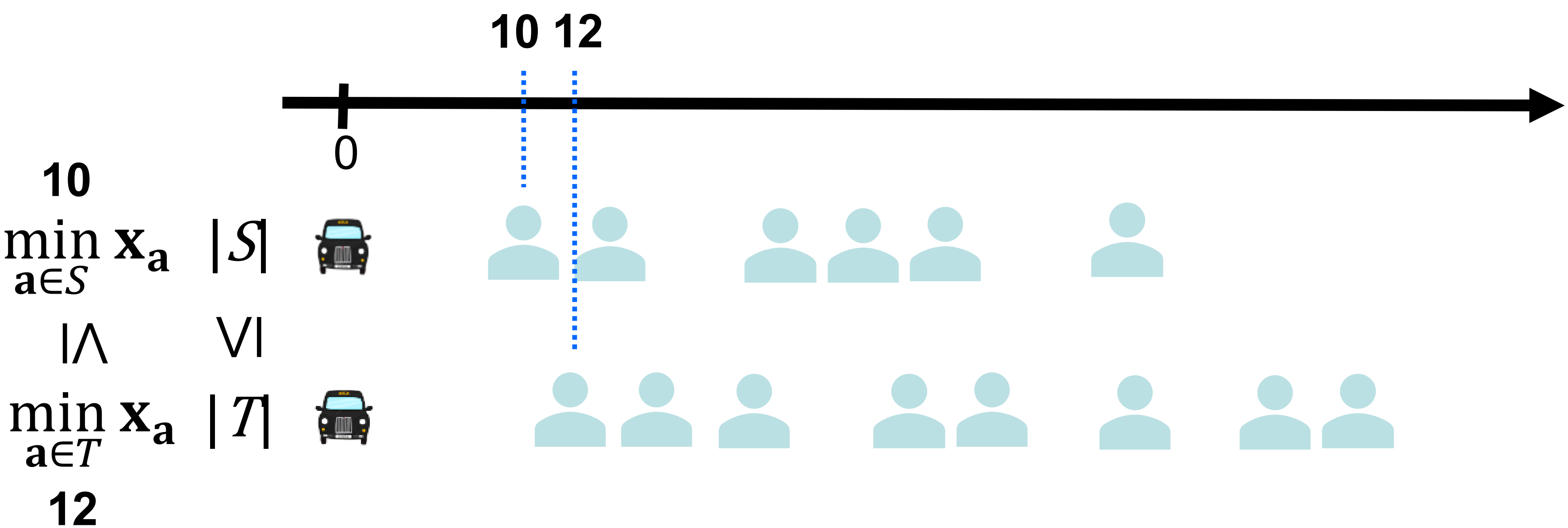


Envy-free allocations may not exist



Properties for envy-freeness:

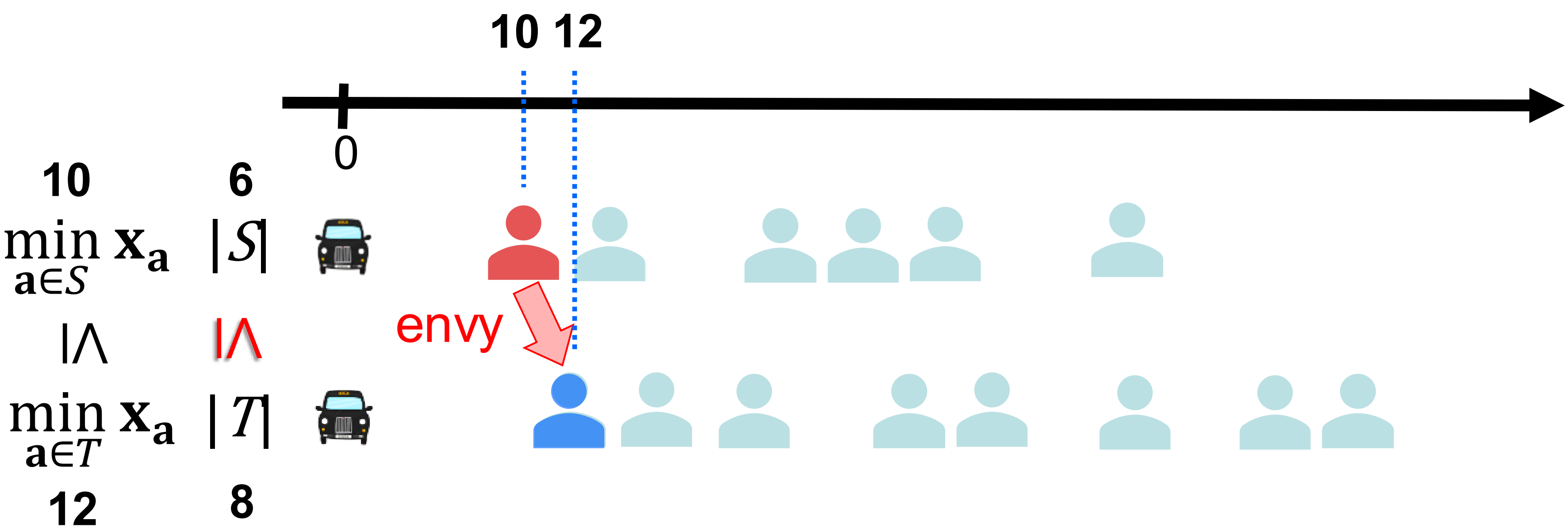
Monotonicity $\min_{\mathbf{a}\in S} \mathbf{x_a} \leq \min_{\mathbf{a}\in T} \mathbf{x_a} \Rightarrow |S| \geq |T|$



Properties for envy-freeness:

Monotonicity $\min_{\mathbf{a} \in S} \mathbf{x}_\mathbf{a} \leq \min_{\mathbf{a} \in T} \mathbf{x}_\mathbf{a} \Rightarrow |S| \geq |T|$

Suppose not

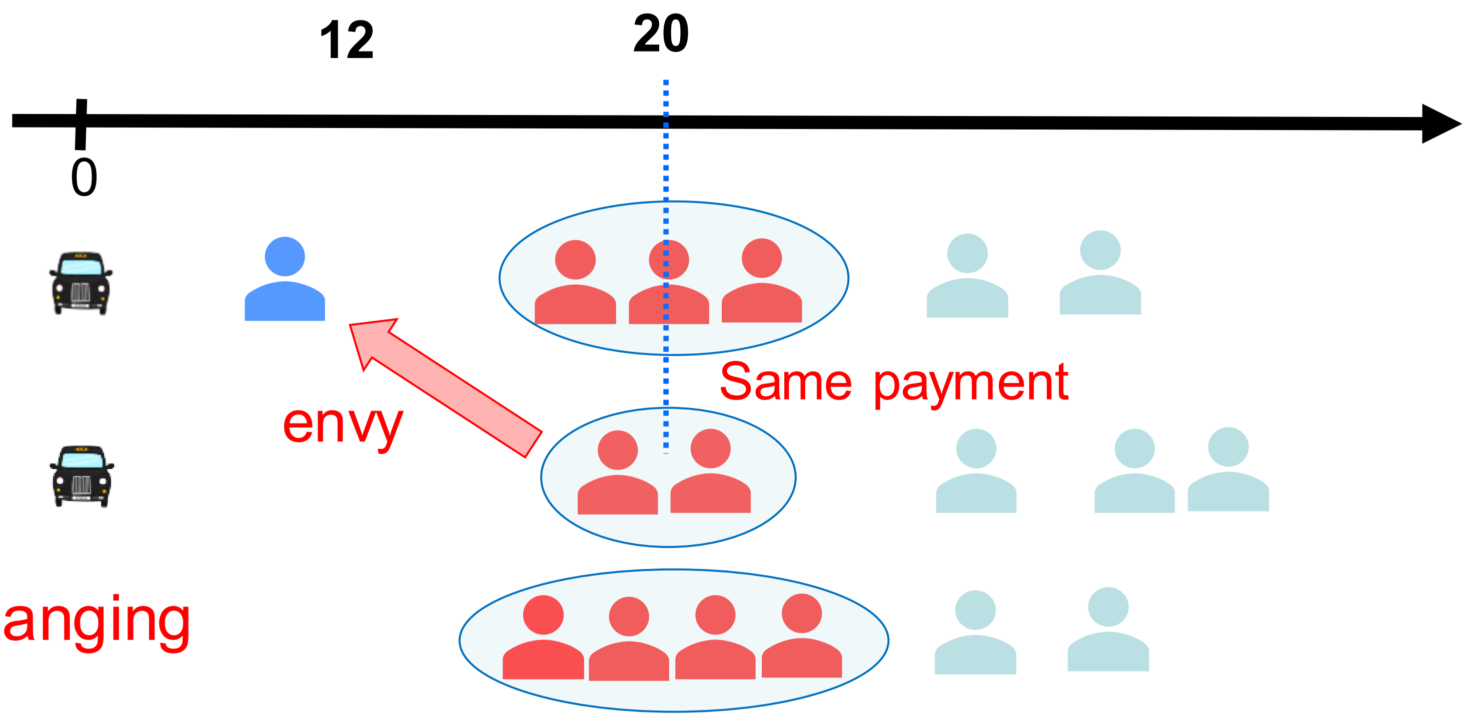


Properties for envy-freeness:

Monotonicity $\min_{\mathbf{a} \in S} \mathbf{x_a} \leq \min_{\mathbf{a} \in T} \mathbf{x_a} \Rightarrow |S| \geq |T|$

Split property If agents of the same destination ride different taxis, they are the first passengers to drop off.

Suppose not



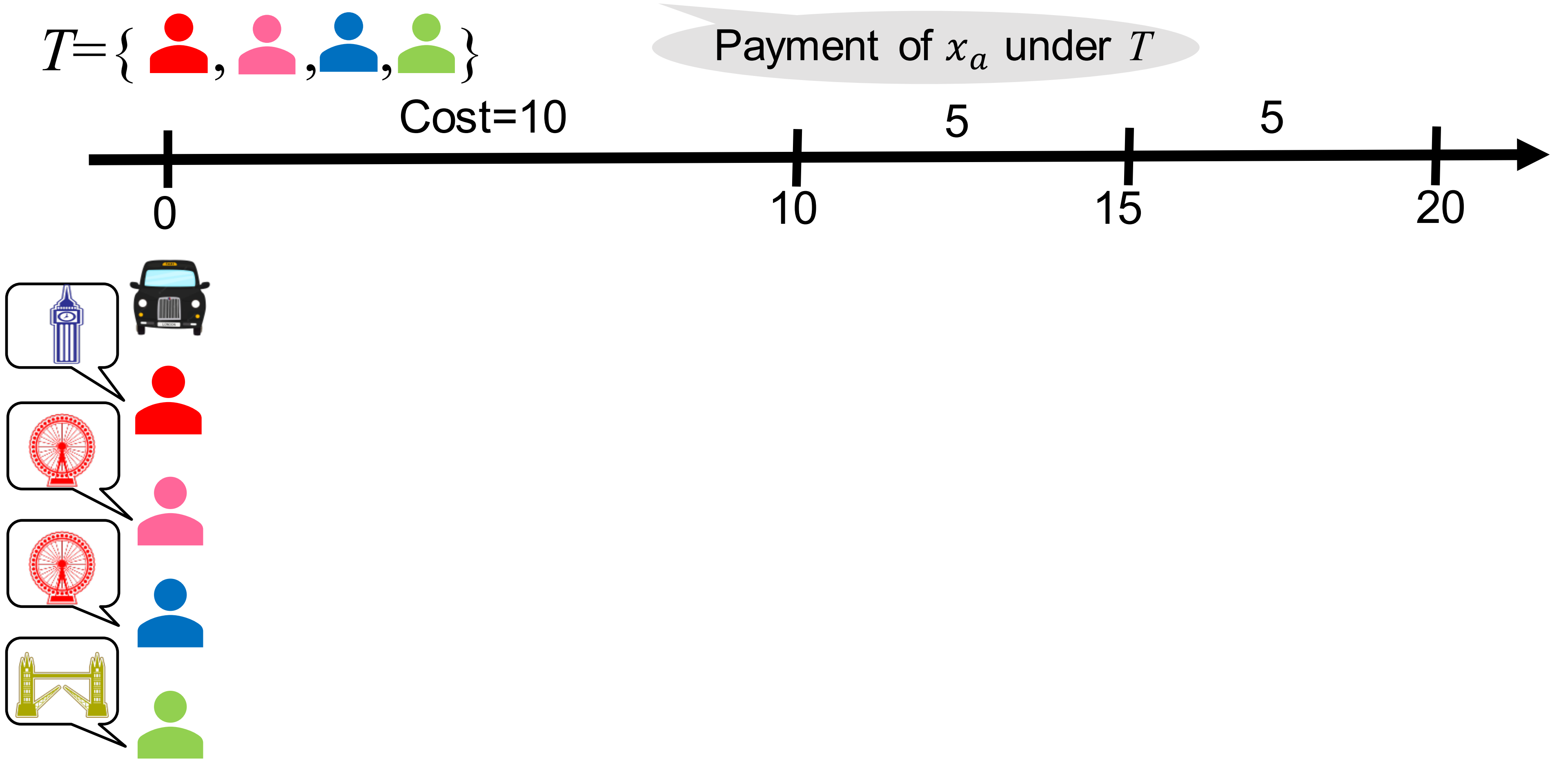
After exchanging

Properties for envy-freeness:

Monotonicity $\min_{\mathbf{a} \in S} \mathbf{x_a} \leq \min_{\mathbf{a} \in T} \mathbf{x_a} \Rightarrow |S| \geq |T|$

Split property If agents of the same destination ride different taxis, they are the first passengers to drop off.

Locality $\varphi(S, x_a) \leq \varphi(T, x_a)$ for all $a \in S$ *and* $T (\neq S)$

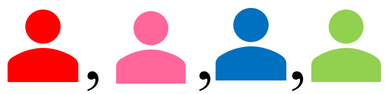


Properties for envy-freeness:

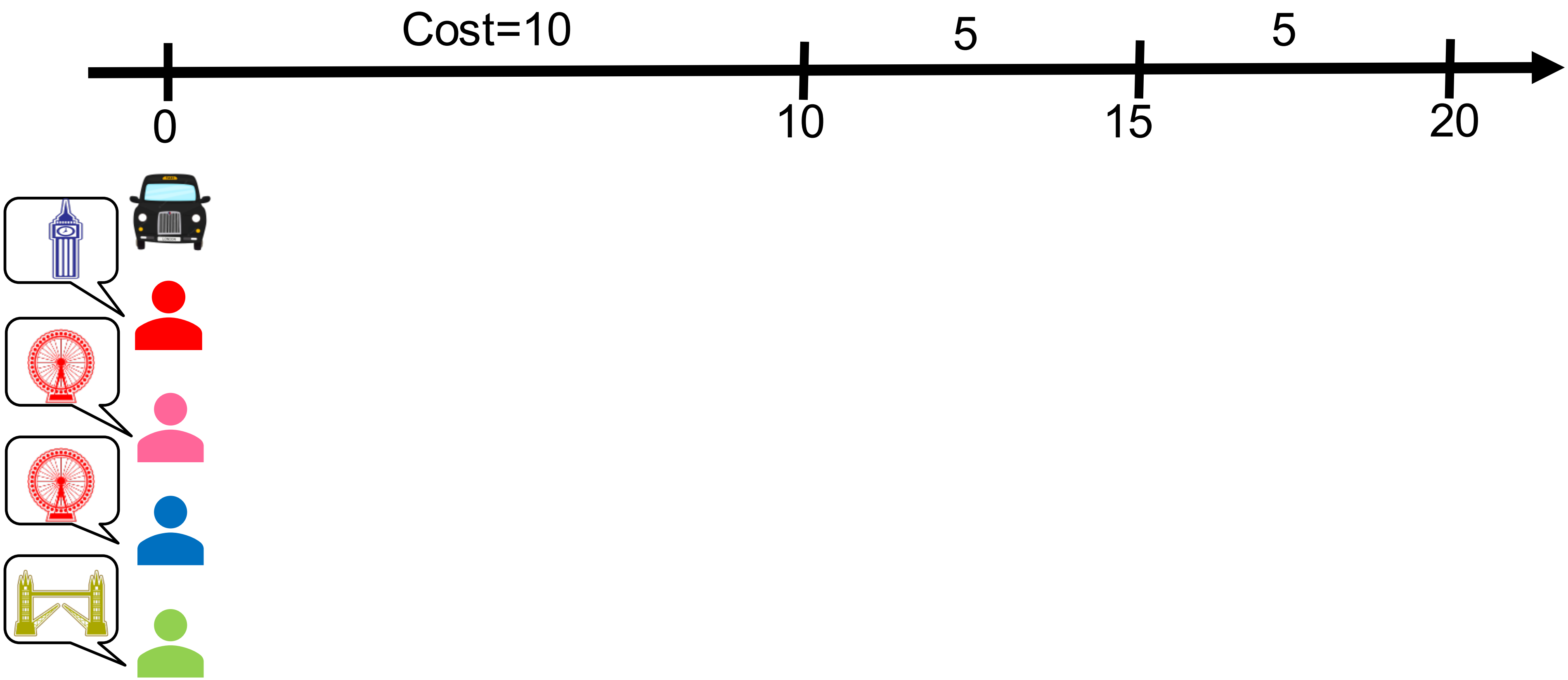
Monotonicity $\min_{\mathbf{a}\in S} \mathbf{x_a} \leq \min_{\mathbf{a}\in T} \mathbf{x_a} \Rightarrow |S| \geq |T|$

Split property If agents of the same destination ride different taxis, they are the first passengers to drop off.

Locality $\varphi(S, x_a) \leq \varphi(T, x_a)$ for all $a \in S$ *and* $T\,(\neq S)$

$T = \{$  $\}$

Payment of x_a under T

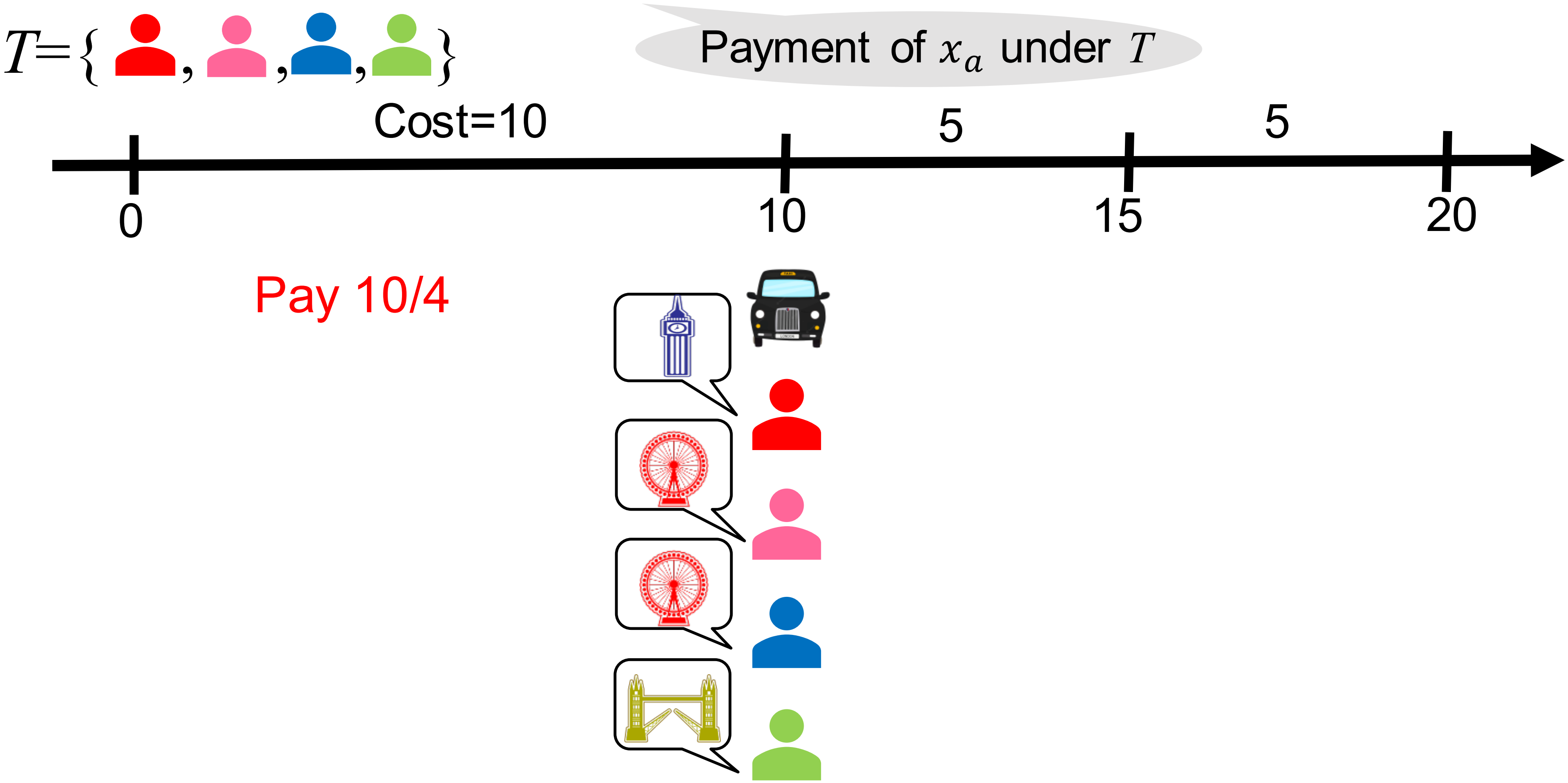


Properties for envy-freeness:

Monotonicity $\min_{\mathbf{a} \in S} \mathbf{x_a} \leq \min_{\mathbf{a} \in T} \mathbf{x_a} \Rightarrow |S| \geq |T|$

Split property If agents of the same destination ride different taxis, they are the first passengers to drop off.

Locality $\varphi(S, x_a) \leq \varphi(T, x_a)$ for all $a \in S$ and $T\ (\neq S)$

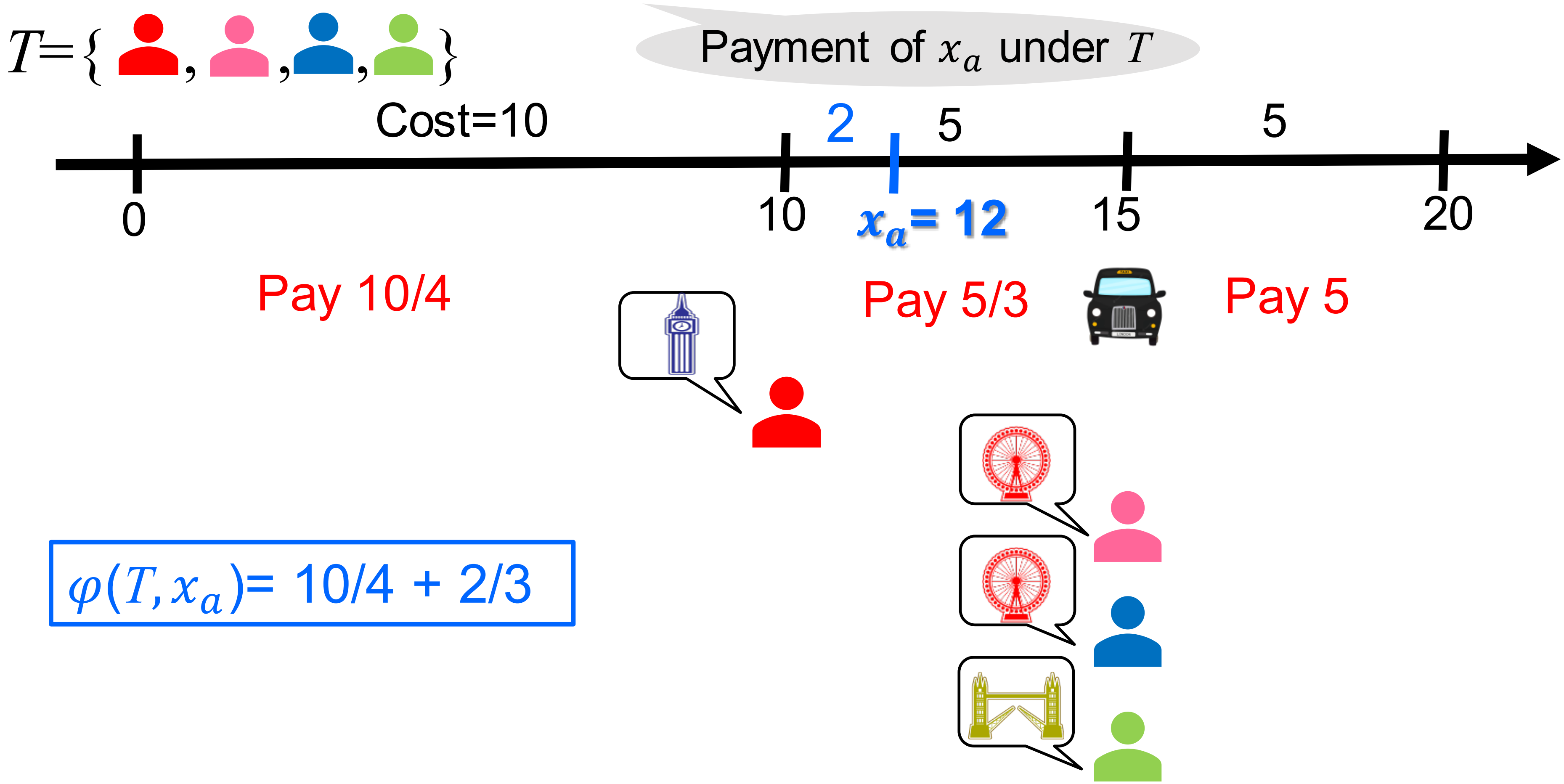


Properties for envy-freeness:

Monotonicity $\min_{\mathbf{a} \in S} \mathbf{x_a} \leq \min_{\mathbf{a} \in T} \mathbf{x_a} \Rightarrow |S| \geq |T|$

Split property If agents of the same destination ride different taxis, they are the first passengers to drop off.

Locality $\varphi(S, x_a) \leq \varphi(T, x_a)$ for all $a \in S$ and $T (\neq S)$



$\varphi(T, x_a)$= 10/4 + 2/3

Envy-free:

- Envy-free allocations may not exist.
- Three structural properties:
  monotonicity, split property, and locality
- **Efficient algorithms for envy-freeness when**
  (1) the number of taxis is a constant,
  (2) the capacity of each taxi is at most four, or
  (3) the number of destination types is small.
- NP-hardness for two relaxations of envy-freeness

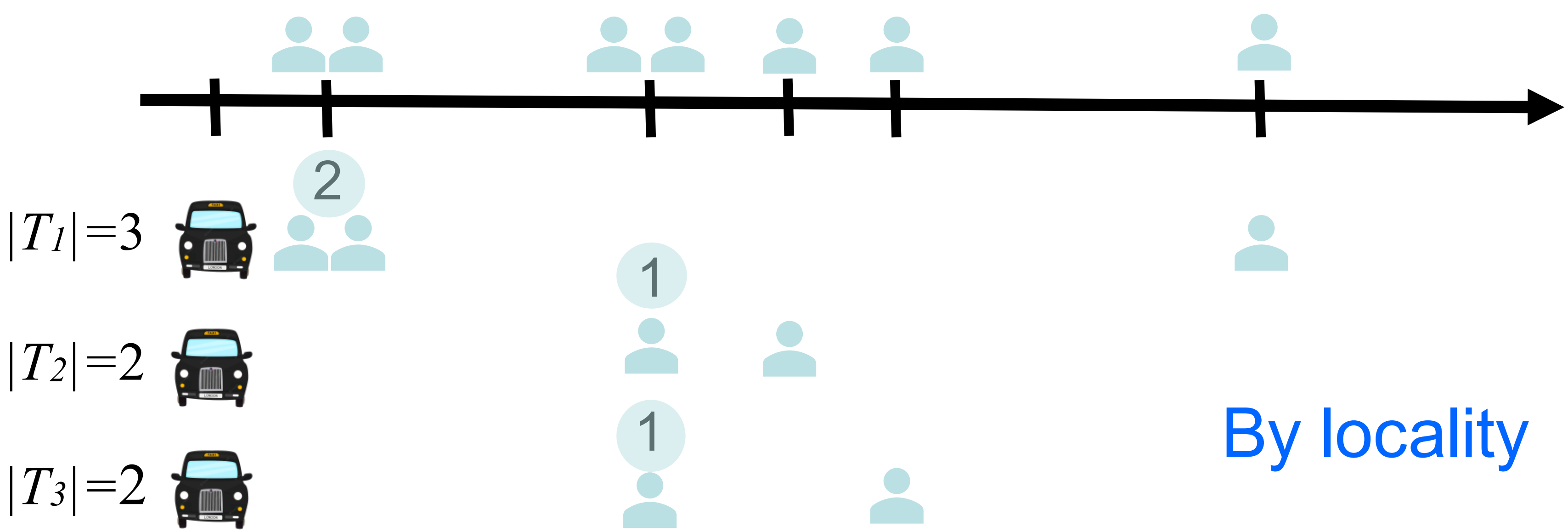
Algorithm for envy-freeness:

If the following 3 parameters are known

⇨ Envy-free allocation can be computed efficiently.

For each taxi i

1. the number $|T_i|$ of agents
2. the first drop-off point $\min_{a \in T_i} x_a$
3. the number of agents who first drop off



Envy-free:

- Envy-free allocations may not exist.
- Three structural properties:
  monotonicity, split property, and locality
- **Efficient algorithms for envy-freeness when**
  (1) the number of taxis is a constant,
  (2) the capacity of each taxi is at most four, or
  (3) the number of destination types is small.
- NP-hardness for two relaxations of envy-freeness

Hardness for Envy-freeness:

Th 1. *It is NP-complete to decide whether there exists an allocation with split property.*

Let *S*={$S_1, S_2, \ldots, S_l$} be a partition of agent set *A*,

an allocation is *envy-free in S*

if for each $S \in$ *S*, the agents in *S* do not envy each other

Th 2. *It is NP-complete to decide whether there exists an envy-free allocation in a given S.*

Conclusion and future work

Introduce our model as a generalization of the airport problem

Th. Backward greedy efficiently computes an allocation which is **Nash stable**, **strongly swap-stable**, and **socially optimal**.

Th. An **envy-free** allocation can be computed efficiently when

(1) the number of taxis is a constant,
(2) the capacity of each taxi is at most four, or
(3) the number of destination types is small.

Th. It is NP-complete to decide whether there exists an allocation with split property.

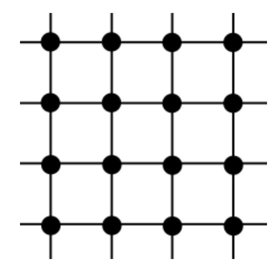
Th. It is NP-complete to decide whether there exists an envy-free allocation in a given S.

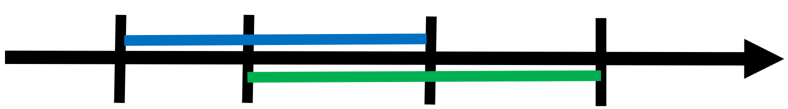
Conclusion and future work

Open problem

- Complexity of computing an envy-free allocation.

Extensions of our model

- Heterogeneous facilities 
  e.g., costs
  some taxis are more comfortable than others
- More general metric space beyond line structure 
- Agents may have different starting points.



- People arrive online.

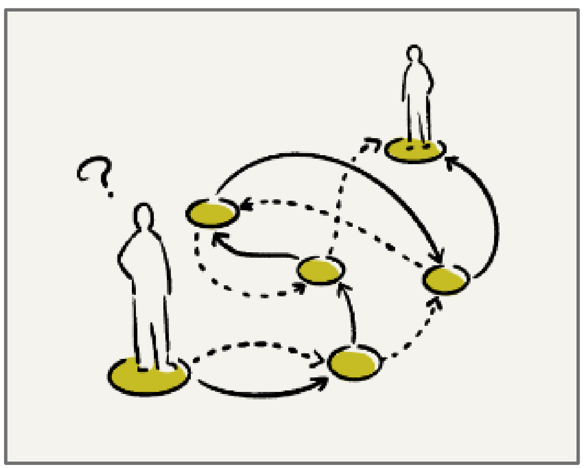
Support &

Advanced Mathematical Science for Mobility Society

Joint project of Kyoto Univ. and Toyota Motor Corp.

京都大学 KYOTO UNIVERSITY　　TOYOTA

Period: 2020～2023 ＋ 2023～2026



Team: Algorithms for future mobility society

- Mechanism design
- Network optimization
- Online algorithm

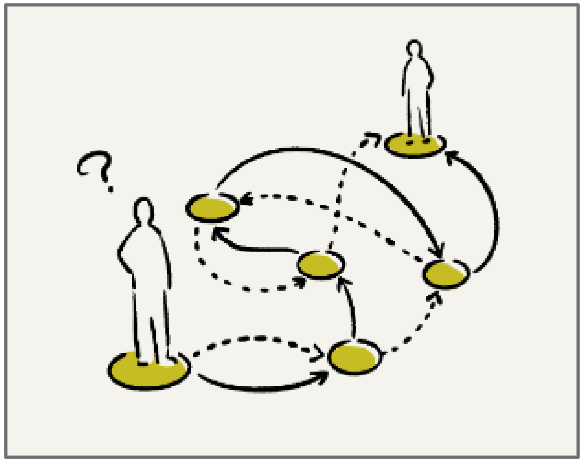
Support &

Advanced Mathematical Science for Mobility Society

Joint project of Kyoto Univ. and Toyota Motor Corp.

京都大学 KYOTO UNIVERSITY　　TOYOTA

Period: 2020～2023　＋　2023～2026



Team: Algorithms for future mobility society

- Mechanism design
- Network optimization
- Online algorithm

#【Paper】

[1] Y. Kobayashi, R. Mahara, An approximation algorithm for Steiner tree problem with neighbor-induced cost, JORSJ, 2023.

[2] K. Makino, H. Ono, Y. Kawase, A. Igarashi, Y. Amano, Fair ride allocation on a line, SAGT 2022.

[3] T. Ishii, A. Kawamura, Y. Kobayashi, K. Makino, Trade-offs among degree, diameter, and number of paths, Discrete Applied Mathematics, 2023.

[4] Y. Amano, K. Makino, A 3/4 differential approximation algorithm for traveling salesman, TAMC 2022.

[5] H. Goko, A. Igarashi, Y. Kawase, K. Makino, H. Sumita, A. Tamura, Y. Yokoi, M. Yokoo, Fair and truthful mechanism with limited subsidy, AAMAS 2022.

[6] H. Goko, K. Makino, S. Miyazaki, Y. Yokoi, Maximally satisfying lower quotas in the hospitals/residents problem, STACS 2022.

[7] R. Mahara, On the existence of EFX for general valuations, ESA2021.

[8] K. Makino, T. Ishii, J. Kawahara, H. Ono, Reallocation problems with minimum completion time, COCOON 2022.

[9] Y. Yokoi, An approximation algorithm for maximum stable matching with ties and constraints, ISAAC2021.

[10] H. Goko, A. Kawamura, Y. Kawase, K. Makino, H. Sumita, Online scheduling on identical machines with a metrical state space, STACS 2022.

[11] Y. Amano, Locally defined independence systems on graphs, Discrete Applied Mathematics, 2023.

[12] K. Makino, S. Miyazaki, Y. Yokoi, Incomplete list setting of the hospitals/residents problem with maximally satisfying lower quotas, SAGT 2022.

[13] Y. Kobayashi, R. Mahara, Proportional allocation of indivisible goods up to the Least valued good on average. ISAAC 2022