



# MECHANISM DESIGN IN THE WILD

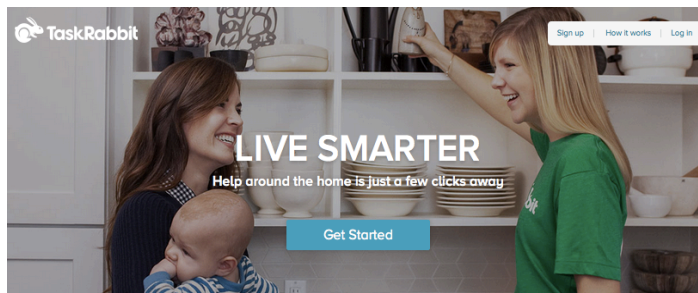
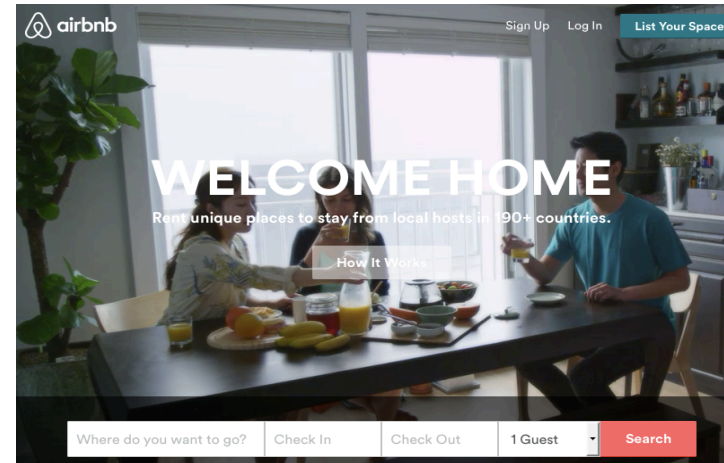
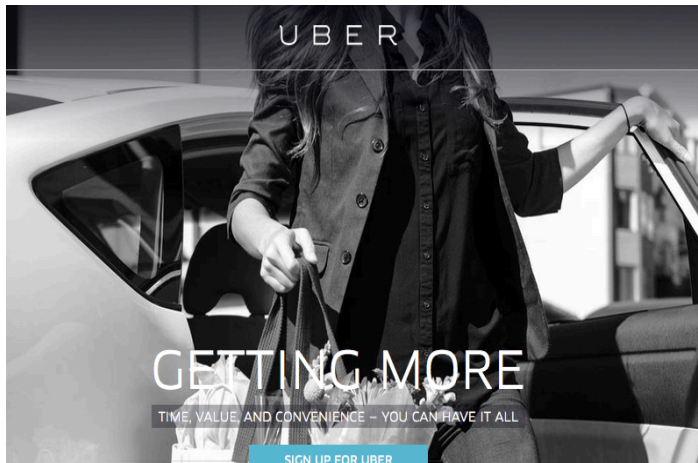
*(WHAT) CAN WE LEARN FROM THE EMERGENT  
DIGITAL SHARING ECONOMY?*

---

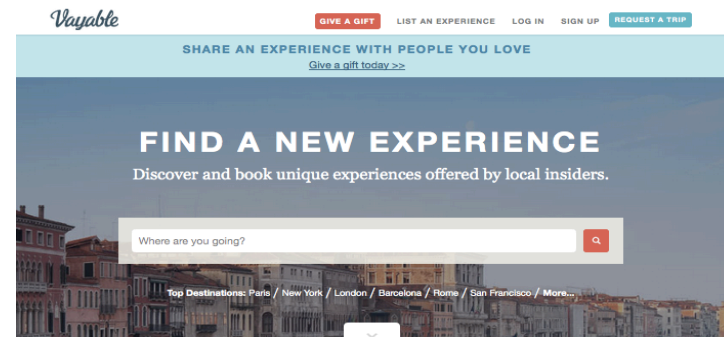
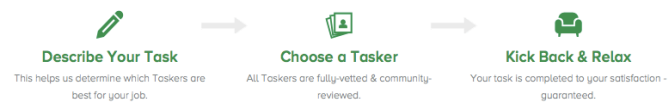
Michael Rovatsos  
University of Edinburgh

AMEC/TADA Workshop @ AAMAS 2015, Istanbul, 3<sup>rd</sup> May 2015

# The Sharing Economy



## How TaskRabbit Works



“A historian, guide, and friend all rolled into one. You won't find this in guidebook or a traditional tour service.”

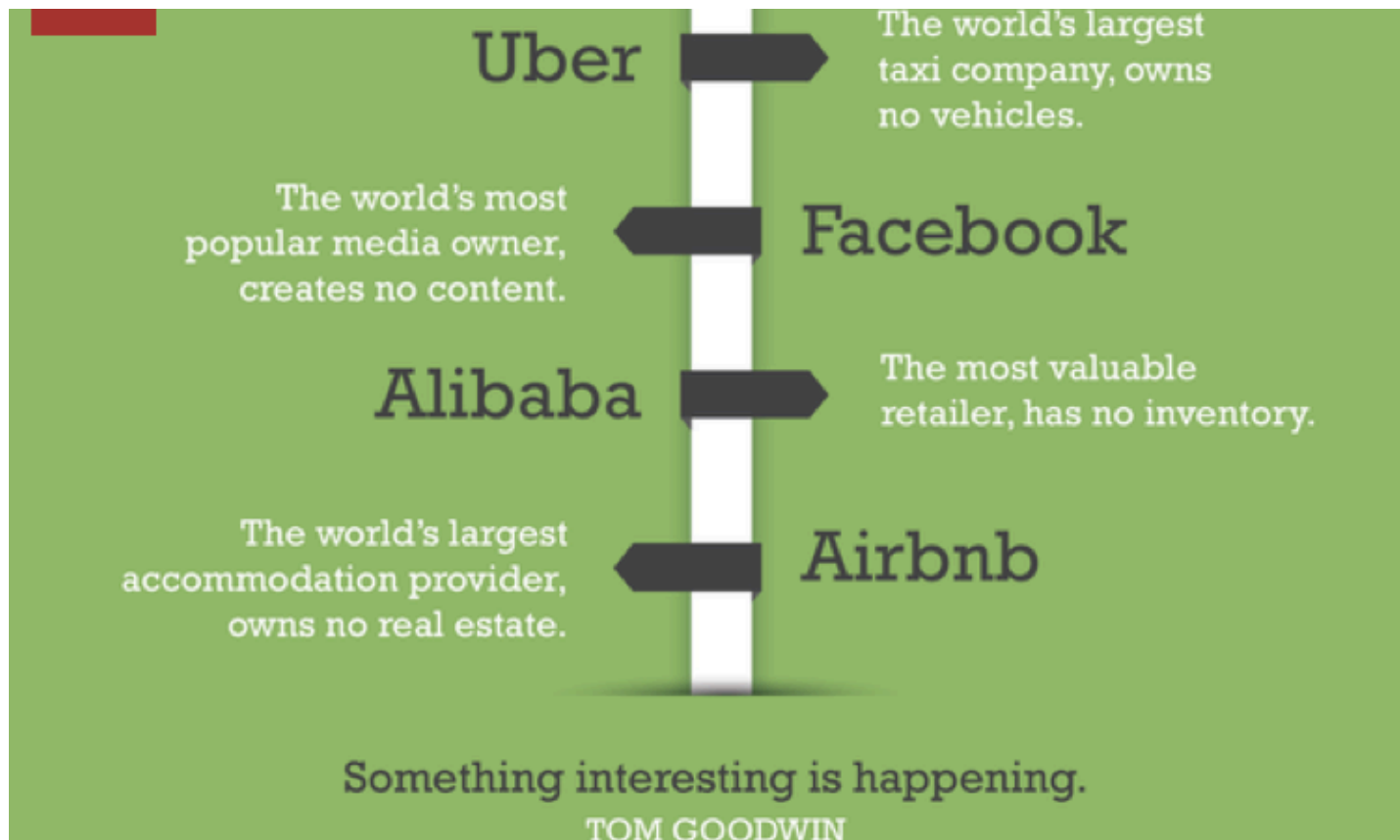
MATT C. (BUSINESS EXECUTIVE)



“Always the highlight of our trip. Put Vayable on your “Must do” list.”

CALLIE H. (CONSULTANT)

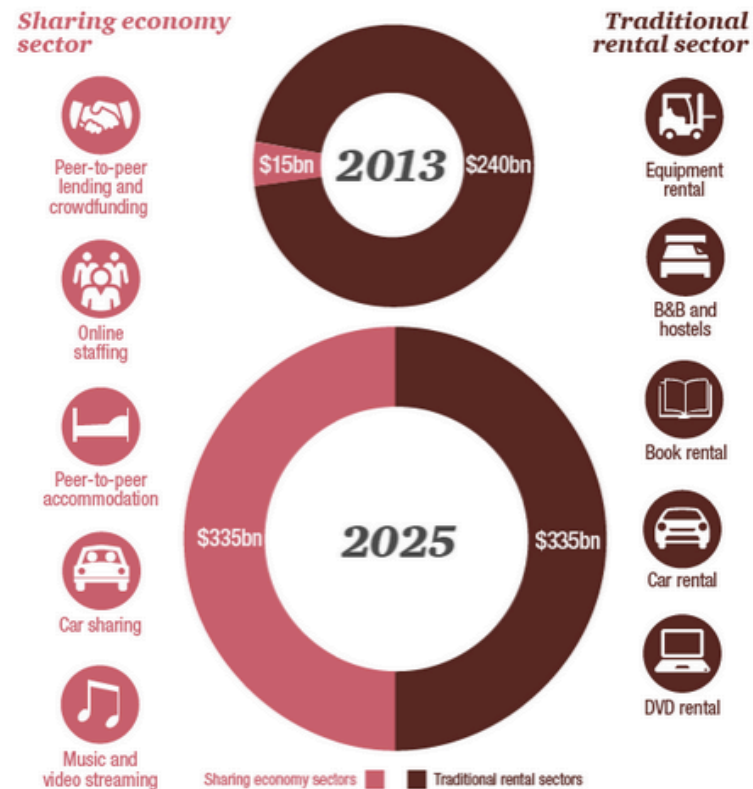
# The Sharing Economy



Source: @WetPaintMENA

# The Sharing Economy

## Sharing economy sector and traditional rental sector projected revenue growth



source: PriceWaterhouseCoopers

# The Sharing Economy

- IT-enabled distribution, sharing and reuse of excess capacity in goods and services
- Web platforms mostly manage search/matching, (de-)commitment, remuneration
- Coordination mechanisms used largely ignore mechanism design literature
- What can we learn from these emerging systems, and what can they learn from agents research?

# Example: Ridesharing

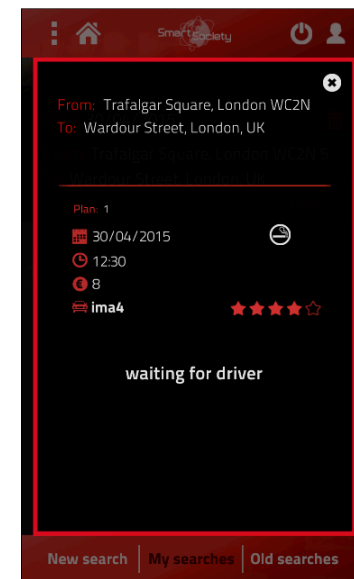
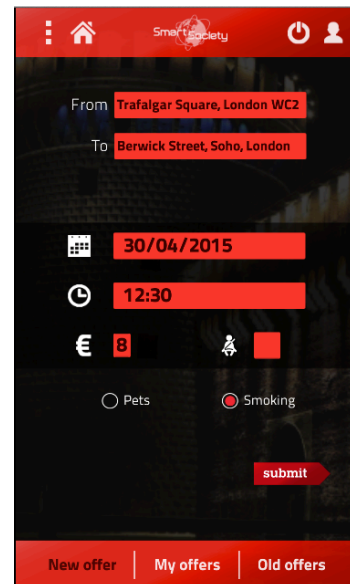
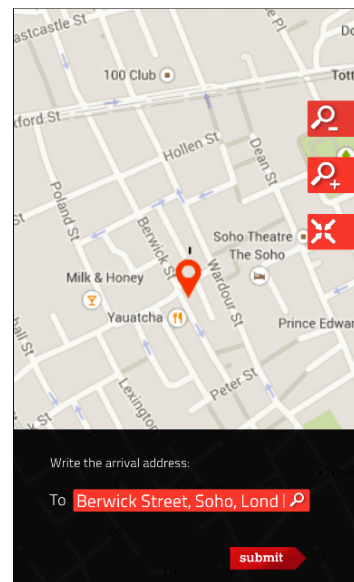
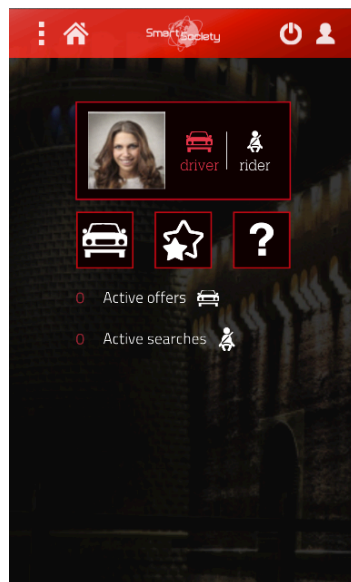
- Over the past two years we've built the web-based ridesharing system *SmartShare*
- Study of human behaviour *in situ* to test models of human collaboration
- Part of a €6.8M project on *hybrid and diversity-aware collective adaptive systems*
- Preliminary user study in Israel, upcoming larger trial in Italy + lab experiments



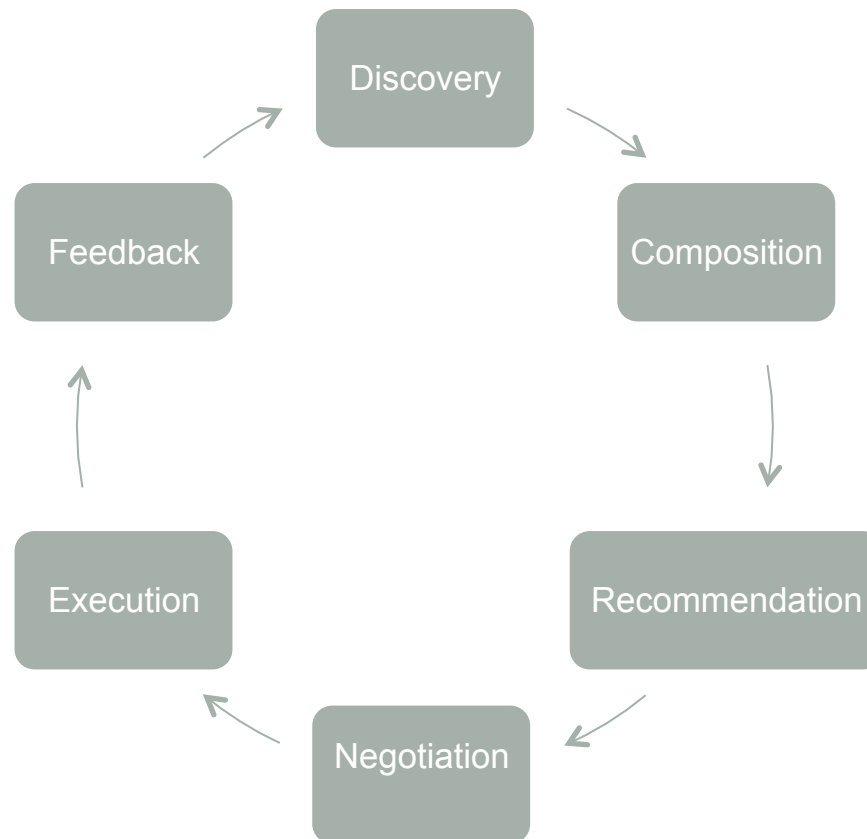
[www.smart-society-project.eu](http://www.smart-society-project.eu)

@SmartSocietyFP7

# SmartShare



# Sharing app orchestration cycle





# “Canonical” mechanism design

- Game-theoretic rationality assumptions
- Focus on social welfare maximisation
- Truthfulness and stability as core concerns
- Provable properties obviate agent reasoning

## A traditional resource allocation problem?

- Possible services not known a priori
- Part-route sharing creates vast solution space
- Sequential dependencies
- Complex, context-dependent preferences
- Optimality less important than availability
- Mechanism acceptability culture-sensitive

# Interesting problems

1. Unmanageable solution spaces
2. Plasticity of interaction models
3. Designing incentive schemes
4. The social “frame problem”

# 1. Unmanageable solution spaces

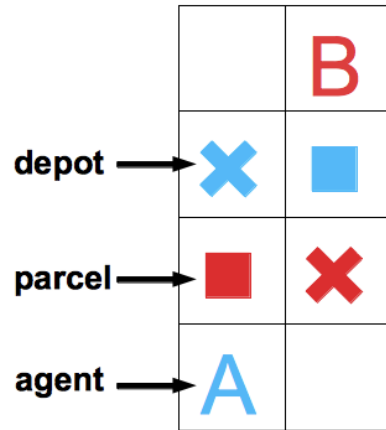
- At finer levels of granularity, there is a vast number of possible collective behaviours
- Synthesising these needs to take strategic properties and user preferences into account
- “Softer” solution concepts might provide some guarantees without excessive computation cost
- **Opportunity**: designing heuristic algorithms that generate “reasonable” solutions

# Calculating complex group tasks

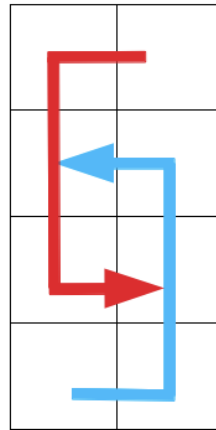
- In complex strategic domains, joint strategies cannot be enumerated *a priori*
- Amounts to a strategic multiagent planning problem
  - Like concurrent planning with additional constraints on plan cost to individuals
  - Problem definition depends on whether contracts can be enforced and utility can be transferred
- Hard to define meaningful solution concepts if goals are incompatible or agents untrustworthy

# Example

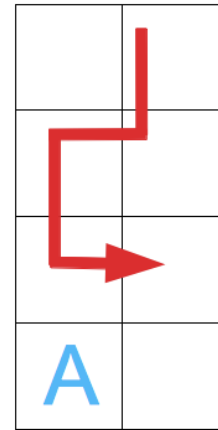
- Delivery domain



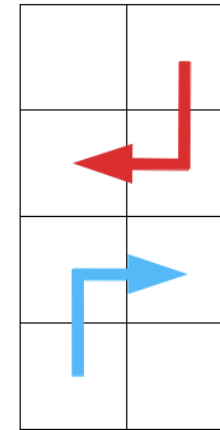
utility = reward - cost



“isolated”  
cost: 6/6  
(inefficient)



“selfish”  
cost: 0/8  
(irrational)



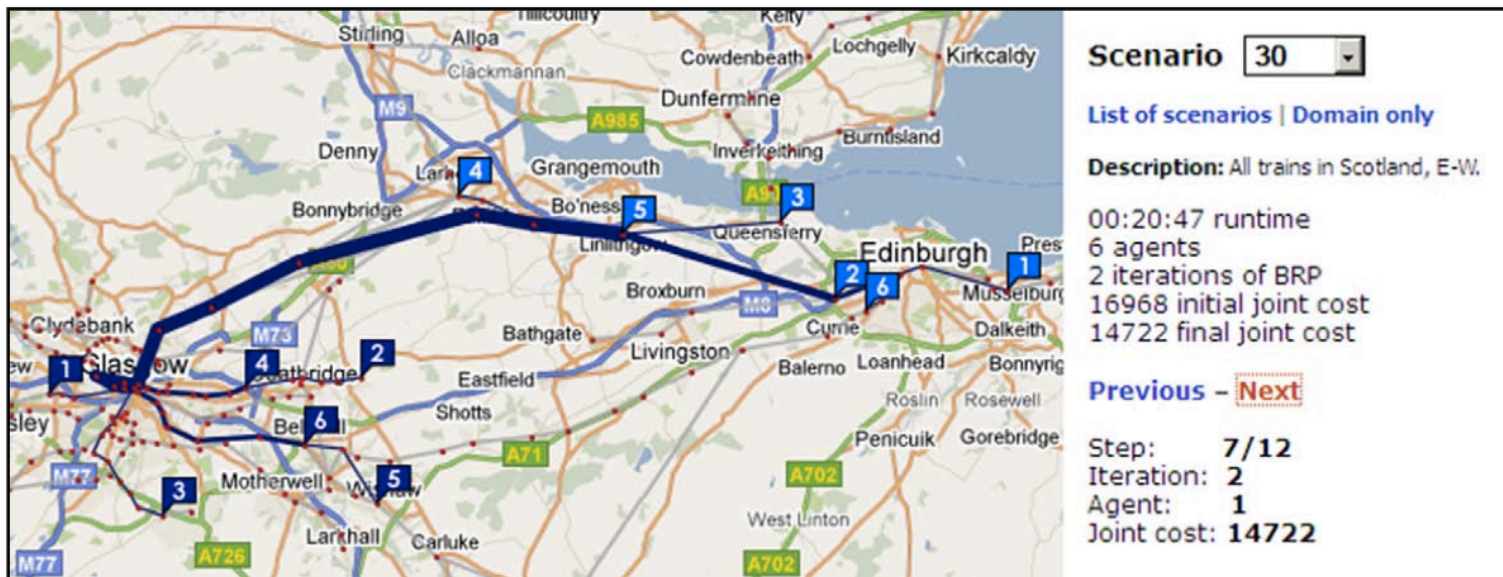
“cooperative”  
cost: 4/4  
(stable)

# Planning for Self-Interested Agents

- **Best-Response Planning (Jonsson & MR):**
  - Iterative method of optimising agents' individual plans without breaking others' plans
  - Computes equilibrium plans fast in congestion games, restricted to interactions regarding cost
- **Extended by “compress-and-expand” algorithm to produce initial concurrent plan**
  - Only for domains where agents can achieve their individual goals alone; where they can't, it's still useful for plan cost optimisation

# Empirical results

- We used BRP to calculate travel routes using real-world UK public transportation data and private cars (>200,000 connections)





## 2. Human-created interaction models

- Platforms let users design a broad range of interaction models
- Not possible to analyse them mathematically before deployment
- Many of them might fall into known classes of well-studied mechanism design problems
- **Opportunity:** automated mapping/verification of interaction protocol properties
- For limited case of ridesharing, we can take more traditional mechanism design approach

# Mechanism Design for Ridesharing

- Ridesharing calls for design of preference elicitation and allocation mechanisms on a huge scale
- Low churn rate, i.e. ensuring commuters are willing to use the service again, is a key concern
- Can be interpreted as a *stability* constraint on allocations computed by the mechanism
- Practical mechanisms can only support *incomplete reporting* of commuter preferences
- **Problem: How do we design mechanisms that form stable allocations with incomplete information?**

# Mechanism Design for Ridesharing

- Any ridesharing mechanism consists of three components:
  - A signaling protocol to support communication between commuters and providers
  - The message sets that the commuters and provider can communicate
  - An allocation mechanism that matches groups (*coalitions*) of commuters to vehicles
- We consider the *posted goods signaling protocol (PGP)*, motivated by real-world ridesharing websites
- Generalizes signaling semantics for posted price mechanisms, ensures incentive compatible reporting

# Posted Goods Signaling Protocol

- 1) Each commuter sends a request signal to the platform
- 2) The platform computes an allocation
- 3) The platform sends a signal to each commuter, consisting of offers
- 4) Each commuter sends a signal indicating whether they accept
- 5) At the time of transport, each commuter sends a *commit signal*, indicating they took/liked the service

# Stable Mechanisms

- To design Nash stable mechanisms we require:
  - Message sets for commuters to report *incomplete preferences*
  - Allocations of passengers to vehicles that yield stable coalitions, accounting for incomplete reporting
- Key observation: the structure of stable coalition formation mechanisms depends on passenger preferences, e.g.
  - *hedonic preferences*: utility depends on other passengers in the same vehicle
  - *topological preference*: utility depends on pick-up times, locations, and tradeoffs between them

# Key Results

- Mechanisms for hedonic preferences
  - To ensure Nash stability we need to allocate one commuter at a time
  - Previously allocated commuters need to admit new commuters into their coalition
  - Key design problem is the allocation, not the message sets

# Key results

- Mechanisms for topological preferences:
  - To ensure Nash stability all commuters can be allocated simultaneously
  - However, message sets need to be carefully designed
  - The message sets depend heavily on the topology of commuter preferences
  - Requires assumption that provider has side information about maximum size of the set of acceptable journeys:

$$\sup_{t, t' \in \Psi_{\mathcal{P}_i}} d_{\Psi_i}(t, t') \leq \alpha$$

## 3. Designing incentives

- Global goals of interaction platforms can be supported by creating additional rewards
- Monetary and “virtual” benefits (badges, scoreboards etc) can be used – gamification
- Feedback mechanisms affect collective behaviour, provide additional incentives
- **Opportunity**: largely overlooked problem, learning over parametrised mechanisms might be a solution



## 4. The “social” frame problem

- Very large numbers of users, possibly small sets of preference types/coarse preferences
- Parametrisation of search and solution mechanisms requires known parameters
- More customisability means less data – how can we balance adaptability with optimality?
- **Opportunity:** human-oriented methods, e.g. solution recommendation

# Group task recommendation

- We don't know whether a solution exists for a requested objective a priori (cannot just propose nearest “product”)
- Impossible to compute all possible solutions offline (and annotate them for retrieval), computation takes time
- We require agreement of all parties for a task to happen, i.e. solution must rank high on everyone's preferences
- Data obtained from negotiation/execution/feedback refers to whole teams (correlated views), not just individuals

# Conclusions

- Sharing economy presents mechanism design with novel, interesting problems
- Adaptive mechanisms and weaker stability/optimality guarantees possibly the answer
- Not covered, but extremely important: ethical issues (privacy, safety, fairness, transparency)
- Opportunities for closer interaction among different communities and across sectors