Fault Tolerant Mechanism Design for General Task Allocation

Dengji Zhao[1] Sarvapali Ramchurn[1] Nick Jennings[2]

[1]University of Southampton, UK
[2]Imperial College London, UK

Task Allocation with Execution Uncertainty

Car repairing problem

Which garage should I visit to successfully fix my car with a minimal cost?

Garage	*Cost/Suggestion*	*Rating (Prob. of Success)*
	\$100	★★★☆☆ (0.5)
AUBURN AUTO DOCTORS	\$150	★★★½☆ (0.8)
EASTLEIGH AUTOWORKS	\$200	★★★★★ (0.7)

The Key Literature

Work	*Domain*					
	R	*T*	*V*	*W*	*C*	Others
[Porter 08]	1	multi	comb.	multi	fixed	

- R. Porter et al. *Fault tolerant mechanism design*. AIJ 2008.
  - single task requester with multiple tasks (combinatorial valuation), multiple workers with fixed and additive costs

The Key Literature

Work	*Domain*					
	R	*T*	*V*	*W*	*C*	Others
[Porter 08]	1	multi	comb.	multi	fixed	
[Ramchurn 09]	multi	multi	comb.	multi	fixed	trust

- S. D. Ramchurn et al. *Trust-based mechanisms for robust and efficient task allocation in the presence of execution uncertainty*. JAIR 2009.
  - multiple task requesters/workers (combinatorial valuation/fixed-cost), multiple tasks (exchange), trust-based execution uncertainty

The Key Literature

Work	*Domain*					
	R	*T*	*V*	*W*	*C*	Others
[Porter 08]	1	multi	comb.	multi	fixed	
[Ramchurn 09]	multi	multi	comb.	multi	fixed	trust
[Stein 11]	1	1	time	multi	fixed	redundancy

- S. Stein et al. *Algorithms and mechanisms for procuring services with uncertain durations using redundancy*. AIJ 2011.
  - single task with completion deadline, multiple workers (uncertain execution time, fixed costs), allow allocation redundancy

The Key Literature

Work	*Domain*					
	R	*T*	*V*	*W*	*C*	Others
[Porter 08]	1	multi	comb.	multi	fixed	
[Ramchurn 09]	multi	multi	comb.	multi	fixed	trust
[Stein 11]	1	1	time	multi	fixed	redundancy
[Conitzer 14]	1	1	time	multi	p-fix	pree.-redu.

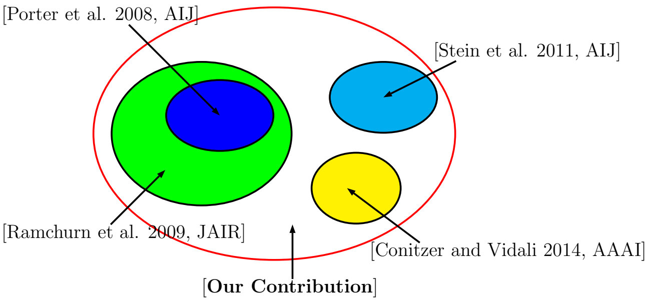
- V. Conitzer et al. *Mechanism design for scheduling with uncertain execution time*. AAAI 2014.
  - single task with completion deadline, uncertain execution time, allocation redundancy with preemption (partially fixed costs)

The Key Literature

Work	*Domain*					
	R	*T*	*V*	*W*	*C*	Others
[Porter 08]	1	multi	comb.	multi	fixed	
[Ramchurn 09]	multi	multi	comb.	multi	fixed	trust
[Stein 11]	1	1	time	multi	fixed	redundancy
[Conitzer 14]	1	1	time	multi	p-fix	pree.-redu.

- Their common features:
  - efficient (*optimal in its range* [Nisan and Ronen 2007]) task allocation
  - pay according to the execution outcome (verification-based)

Our Contribution (Overview)



- generalise the task allocation domain
- characterise the applicability of the Vickrey-type mechanisms with verification (fault tolerant mechanisms)

Outline

A Single Task Allocation Problem

without Execution Uncertainty:

task (valuation)	costs
	$\vdots$
	$\vee\vert$
	c_2
	$\vee\vert$
τ_0 (v_0)	c_1

- Vickrey outcome (efficient, truthful and individually rational):
    - allocate τ_0 to worker 1
    - pays worker 1 c_2
    - worker 1's utility is $c_2 - c_1$

A Single Task Allocation Problem

task	costs
	$\vdots$
	$\vee\vert$
	c_2
	$\vee\vert$
τ_0 (v_0)	c_1

- Vickrey outcome:
  - allocate τ_0 to worker 1
  - pay worker 1 c_2
  - worker 1's utility is $c_2 - c_1$

task	costs (fixed)	prob.
	$\vdots$	$\vdots$
	$\vee\vert$	$\wedge\vert$
	c_2	p_2
	$\vee\vert$	$\wedge\vert$
τ_0 (v_0)	c_1	p_1

- Vickrey outcome:
  - allocate τ_0 to worker 1
  - pay worker 1
    $v_0 \times p_1 - (v_0 \times p_2 - c_2) =$
    $v_0 \times (p_1 - p_2) + c_2$
  - worker 1's utility is
    $v_0 \times (p_1 - p_2) + c_2 - c_1$

Fault Tolerant Mechanism Design

task	costs (fixed)	prob.
	$\vdots$	$\vdots$
	$\vee\vert$	$\wedge\vert$
	c_2	p_2
	$\vee\vert$	$\wedge\vert$
τ_0 (v_0)	c_1	p_1

- Vickrey outcome:
  - allocate τ_0 to worker 1
  - pay worker 1
    $v_0 \times p_1 - (v_0 \times p_2 - c_2) =$
    $v_0 \times (p_1 - p_2) + c_2$

Vickrey with verification (*fault tolerant [Porter et al. 2008]*):

- allocate τ_0 to worker 1,
- pay worker 1 $v_0 \times (1 - p_2) + c_2$ if τ_0 is completed,
- charge him $v_0 \times p_2 - c_2$, otherwise.
- worker 1's expected payment is
  $p_1 \times (v_0 \times (1 - p_2) + c_2) - (1 - p_1) \times (v_0 \times p_2 - c_2) =$
  $v_0 \times (p_1 - p_2) + c_2$

The General Model

- n agents $N = \{1, ..., n\}$.
- task allocation space T, each $\tau \in T$ is defined by $\tau = (\tau_i)_{i \in N}$, where τ_i is a set of tasks assigned to agent i.
- $p_i^\tau \in [0, 1]$ is the probability that i will successfully complete her tasks τ_i. $p_i = (p_i^\tau)_{\tau \in T}$ is i's probability of success (PoS) profile.
- i's valuation is defined by $v_i : T \times [0, 1]^N \to \mathbb{R}$. v_i considers variable-costs, externalities, and task interdependences.
- define agent i's type by $\theta_i = (v_i, p_i)$.

The Post-Execution Verification (PEV) Mechanism

Given the agents' true type profile θ, their reports $\hat{\theta}$, and efficient allocation π^*, **PEV-based payment** x^{PEV} for each agent i is defined as:

$$x_i^{PEV}(\hat{\theta}) = \begin{cases} h_i(\hat{\theta}_{-i}) - V^1_{-i}(\hat{\theta}, \pi^*) & \text{if } i \text{ succeeded,} \\ h_i(\hat{\theta}_{-i}) - V^0_{-i}(\hat{\theta}, \pi^*) & \text{if } i \text{ failed.} \end{cases}$$

where

- $h_i(\hat{\theta}_{-i}) = \sum_{j \in N \setminus \{i\}} \hat{v}_j(\pi^*(\hat{\theta}_{-i}), (0, \hat{p}_{-i}^{\pi^*(\hat{\theta}_{-i})}))$
- $V^1_{-i}(\hat{\theta}, \pi^*) = \sum_{j \in N \setminus \{i\}} \hat{v}_j(\pi^*(\hat{\theta}), (1, p_{-i}^{\pi^*(\hat{\theta})}))$,
  $V^0_{-i}(\hat{\theta}, \pi^*) = \sum_{j \in N \setminus \{i\}} \hat{v}_j(\pi^*(\hat{\theta}), (0, p_{-i}^{\pi^*(\hat{\theta})}))$.

Conditions for PEV to be Ex-Post Truthful and IR

Theorem

Mechanism (π^*, x^{PEV}) is ex-post truthful iff for all $i \in N$, v_i is multilinear in probability of success (i.e., risk-neutral).

- ex-post truthfulness requires that reporting truthfully maximises an agent's expected utility, if everyone else also reports truthfully.

Theorem

Mechanism (π^*, x^{PEV}) is individually rational iff for all $i \in N$, for all $\tau \in T$, if $\tau_i = \emptyset$, then $v_i(\tau, p^\tau) \geq 0$ for any $p^\tau \in [0, 1]^N$.

Extension to Trust-Based Environments

Trust-based environments:

- each agent's probability of success p_i is not completely predictable by the agent alone but aggregated from all agents' private opinions (known as trust [Ramchurn et al. 2009, JAIR]).
- $p_i^\tau = f_i^\tau(p_{1,i}^\tau, ..., p_{n,i}^\tau)$ where $p_{i,j}^\tau$ is the probability that i believes j will complete j's tasks in τ.

Theorem

Mechanism (π^*, x^{PEV}) is ex-post truthful in the trust-based environments iff for all $i \in N$,

- *v_i is multilinear in probability of success and*
- *aggregation f_i is multilinear.*

In Summary

Post-Execution Verification Mechanism is “almost" universally applicable.

Garage	c_i	p_i
	\$100	0.5
AUBURN AUTO DOCTORS	\$150	0.8
EASTLEIGH AUTOWORKS	\$200	0.7

- The social welfare of visiting each garage are:
  - $300 \times 0.5 - 100 = 50$
  - $300 \times 0.8 - 150 = 90$
  - $300 \times 0.7 - 200 = 10$
- The payments are:
  - pay the garage $300 - 50$ (if succeeded)
  - charge the garage 50 (otherwise)

In Summary

Post-Execution Verification Mechanism is “almost" universally applicable.

Garage	c_i	p_i
	\$100	0.5
AUBURN AUTO DOCTORS	\$150	0.8
EASTLEIGH AUTOWORKS	\$200	0.7

- The social welfare of visiting each garage are:
  - $300 \times 0.5 - 100 = 50$
  - $300 \times 0.8 - 150 = 90$
  - $300 \times 0.7 - 200 = 10$
- The payments are:
  - pay the garage $300 - 50$ (if succeeded)
  - charge the garage 50 (otherwise)

Question

Possible/how to design truthful mechanisms for non-risk-neutral settings?