

Stabilizing Grand Cooperation in Unbalanced Cooperative Games

Xiangtong Qi

Department of Industrial Engineering and Decision Analytics,
The Hong Kong University of Science and Technology

Joint works with **Lindong Liu** (USTC), **Zhou Xu** (HKPU)

MISTA, Ningbo, Dec. 2019

INTRODUCTION

Cooperation is everywhere, from cells in your body, to people in a city, and partners in **a scheduling problem**.

Cooperation is everywhere, from cells in your body, to people in a city, and partners in **a scheduling problem**.

- centralized decision making, to minimize the total cost (for social optimum);
- enforced by an external party, to minimize negative externalities (e.g. the number of machines used, etc).

Is every player beneficial from the grand cooperation?

Is the grand cooperation stable?

Is every player beneficial from the grand cooperation?

Is the grand cooperation stable?

Not Always.

If the grand cooperation is not stable, how to stabilize it?

PRELIMINARIES

Cooperative Game

A **cooperative game** is defined by a pair (N, π) :

- A set $N = \{1, 2, \dots, n\}$ of players, **grand coalition**;
- A **characteristic function** $\pi(S)$ = the minimum total cost achieved by the cooperation of members in coalition $S \in \mathbb{S} = 2^N \setminus \{\emptyset\}$.

The game requires:

- A **cost allocation** $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n] \in \mathbb{R}^n$, where α_j = the cost allocated to each player $j \in N$.

Integer Minimization Games

$\pi(S)$: need to solve an optimization problem, not given.

Integer Minimization (IM) Games:

For each coalition $S \in \mathbb{S}$, an incidence vector $y^S \in \{0, 1\}^n$, with $y_j^S = 1$ if $j \in S$, and with $y_j^S = 0$ otherwise, for all $j \in N$, such that

$$\pi(S) = \min_x \{cx : Ax \geq By^S + E, x \in \mathbb{Z}^q\}.$$

Denote $\alpha(S) = \sum_{j \in S} \alpha_j$.

A cost allocation $\alpha \in \mathbb{R}^n$ is in the **core** if it satisfies:

- **Budget Balance** Constraint: $\alpha(N) = \pi(N)$;
- **Coalition Stability** Constraints: $\alpha(S) \leq \pi(S)$ for each $S \in \mathbb{S}$.

$$\text{Core}(N, \pi) = \left\{ \alpha : \begin{aligned} &\alpha(N) = \pi(N), \\ &\alpha(S) \leq \pi(S), \quad \forall S \in \mathbb{S} \setminus \{N\}, \quad \alpha \in \mathbb{R}^n \end{aligned} \right\}.$$

Denote $\alpha(S) = \sum_{j \in S} \alpha_j$.

A cost allocation $\alpha \in \mathbb{R}^n$ is in the **core** if it satisfies:

- **Budget Balance** Constraint: $\alpha(N) = \pi(N)$;
- **Coalition Stability** Constraints: $\alpha(S) \leq \pi(S)$ for each $S \in \mathbb{S}$.

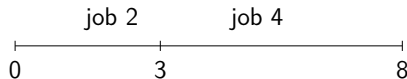
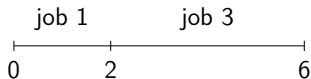
$$\text{Core}(N, \pi) = \left\{ \alpha : \begin{aligned} &\alpha(N) = \pi(N), \\ &\alpha(S) \leq \pi(S), \quad \forall S \in \mathbb{S} \setminus \{N\}, \quad \alpha \in \mathbb{R}^n \end{aligned} \right\}.$$

However, $\text{Core}(N, \pi)$ can be empty.

Example: Machine Scheduling Game (MSG)

Game of Single Machine Scheduling with Setup Cost:

- Grand coalition: $N = \{1, 2, 3, 4\}$;
- Processing times: $t_1 = 2$, $t_2 = 3$, $t_3 = 4$, $t_4 = 5$;
- Machine setup cost: $t_0 = 9.5$;
- $\pi(S)$ for $S \in \mathbb{S}$: minimizes the total completion time of jobs in S plus the machine setup cost;
- $\pi(N) = \pi(\{1, 3\}) + \pi(\{2, 4\}) = 38$ (SPT Rule).



Example: Empty Core

Coalitions	Cost
$\{1\}$	11.5
$\{2\}$	12.5
$\{3\}$	13.5
$\{4\}$	14.5
$\{1, 2\}$	16.5
$\{1, 3\}$	17.5
$\{1, 4\}$	18.5
$\{2, 3\}$	19.5
$\{2, 4\}$	20.5
$\{3, 4\}$	22.5
$\{1, 2, 3\}$	25.5
$\{1, 2, 4\}$	26.5
$\{1, 3, 4\}$	28.5
$\{2, 3, 4\}$	31.5
$\{1, 2, 3, 4\}$	38

Example: Empty Core

Coalitions	Cost
{1}	11.5
{2}	12.5
{3}	13.5
{4}	14.5
{1, 2}	16.5
{1, 3}	17.5
{1, 4}	18.5
{2, 3}	19.5
{2, 4}	20.5
{3, 4}	22.5
{1, 2, 3}	25.5
{1, 2, 4}	26.5
{1, 3, 4}	28.5
{2, 3, 4}	31.5
<u>{1, 2, 3, 4}</u>	<u>38</u>

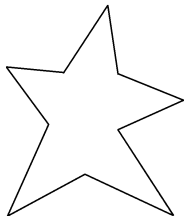
Optimal Cost Allocation Problem

$$\begin{aligned} \max \quad & (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) = 37.25 < 38 \\ \text{s.t.} \quad & \alpha_1 \leq 11.5, \dots, \alpha_4 \leq 14.5, \\ & \alpha_1 + \alpha_2 \leq 16.5, \dots, \alpha_3 + \alpha_4 \leq 22.5, \\ & \dots, \\ & \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \leq 38. \end{aligned}$$

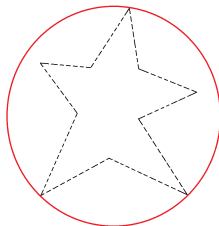
$$\alpha^* = [6; 8.75; 10.75; 11.75]$$

I INSTRUMENTS

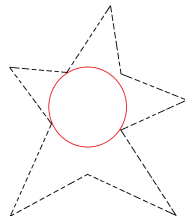
Instruments to Stabilize Grand Coalitions



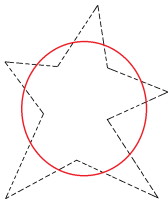
Unbalanced Game



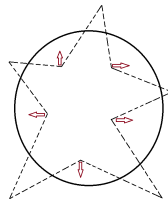
Subsidization



Penalization



Simultaneous Subsidization & Penalization



Cost Adjustment

Our Studies

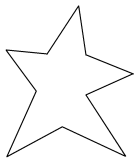
S. Caprara and Letchford (2010, MP), [Liu et al. \(2016, IJOC\)](#)

P. Schulz and Uhan (2010, OR; 2013, DO)

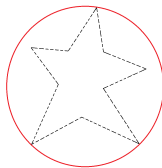
P.&S. [Liu et al. \(2018, OR\)](#)

C.A. [Liu et al. \(2019, under review\)](#)

1. INSTRUMENT OF SUBSIDIZATION



Unbalanced Game



Subsidization

Instrument 1: LRB Cost Allocation, 2016, IJOC

Instrument of Subsidization (ϵ -core)

$$\text{Core}(N, \pi) = \left\{ \alpha : \alpha(N) = \pi(N), \alpha(S) \leq \pi(S), \forall S \in \mathbb{S} \setminus \{N\}, \alpha \in \mathbb{R}^n \right\}.$$

Relax Budget Balanced constraint: $\alpha(N) = \pi(N)$

- **Minimum Subsidy** to stabilize the grand coalition:

$$\omega^* = \min \{ \pi(N) - \alpha(N) : \alpha(S) \leq \pi(S), \forall S \in \mathbb{S} \}.$$

Instrument 1: LRB Cost Allocation, 2016, IJOC

Instrument of Subsidization (ϵ -core)

$$\text{Core}(N, \pi) = \left\{ \alpha : \alpha(N) = \pi(N), \alpha(S) \leq \pi(S), \forall S \in \mathbb{S} \setminus \{N\}, \alpha \in \mathbb{R}^n \right\}.$$

Relax Budget Balanced constraint: $\alpha(N) = \pi(N)$

- **Minimum Subsidy** to stabilize the grand coalition:

$$\omega^* = \min \{ \pi(N) - \alpha(N) : \alpha(S) \leq \pi(S), \forall S \in \mathbb{S} \}.$$

Minimum Subsidy needed for MSG Example:

$$38 - 37.25 = 0.75$$

Instrument 1: LRB Cost Allocation, 2016, IJOC

Motivation: Advantages of Lagrangian Relaxation over Linear Relaxation

Optimal Cost Allocation Problem (OCAP):

$$\max_{\alpha} \{ \alpha(N) : \alpha(S) \leq \pi(S), \forall S \in \mathbb{S} \}.$$

All existing methods are based on Linear Relaxation

Lagrangian Relaxation v.s. Linear Relaxation

- Applicable in non-linear case;
- Generally more effective;
- New insights via decomposition;
- Adequate studies on LR for ILP.

Instrument 1: LRB Cost Allocation, 2016, IJOC

Research Question

Compute a cost allocation based on **Lagrangian Relaxation** technique such that

- the cost assigned to each coalition **does not exceed the total cost incurred** if they deviate from grand coalition;
- the total assigned cost is **as large as possible**.

Instrument 1: LRB Cost Allocation, 2016, IJOC

Lagrangian Relaxation and Decomposition

$$\pi(S) = \min_x \{f(x) : Ax \geq B\gamma^S + E, A'x \geq B'\gamma^S + E', x \in \{0,1\}^q\},$$

$$\pi_{LR}^\lambda(S) = \min_x \{f(x) - \lambda(A'x - B'\gamma^S - E') : Ax \geq B\gamma^S + E, x \in \{0,1\}^q\}.$$

- Lagrangian sub-characteristic functions 1 and 2:

$$\pi_{LR1}^\lambda(S) = \lambda B'\gamma^S + \lambda E', \quad \forall S \in \mathbb{S}.$$

$$\pi_{LR2}^\lambda(S) = \min_x \{f(x) - \lambda A'x : Ax \geq B\gamma^S + E, x \in \{0,1\}^q\}, \quad \forall S \in \mathbb{S}.$$

Theorem

Given any Lagrangian multiplier λ , if α_{LR1}^λ and α_{LR2}^λ are feasible cost allocations for sub-games (N, π_{LR1}^λ) and (N, π_{LR2}^λ) , respectively,

then $\alpha_{LR}^\lambda = \alpha_{LR1}^\lambda + \alpha_{LR2}^\lambda$ is a feasible cost allocation for game (N, π) .

Instrument 1: LRB Cost Allocation, 2016, IJOC

Computing Optimal Cost Allocations for Sub-games

- For **sub-game 1** (N, π_{LR1}^λ) , a vector α_{LR1}^λ with

$$\{(\alpha_{LR1}^\lambda)_j = (\lambda B')_j + \frac{1}{n} \lambda E' : \forall j \in N\}$$

is a feasible cost allocation and lies in the core.

- For **sub-game 2** (N, π_{LR2}^λ) , its optimal cost allocation α_{LR2}^λ can be computed by
 - Primal-Dual-Type algorithm, if this game is sub-modular;
 - Column Generation Based algorithm, if this game is not.

Instrument 1: LRB Cost Allocation, 2016, IJOC

Properties of LRB Cost Allocation

In general, the lower bound obtained by **Lagrangian relaxation** technique is **as sharp as** the lower bound obtained by **linear programming relaxation**.

Theorem

For game (N, π) , under the same ILP formulation for $\pi(S)$, the LRB cost allocation value is no less than the LPB cost allocation value, i.e., $\alpha_{LR}^\lambda(N) \geq \alpha_{LP}(N)$, when

- (1) **Given Lagrangian multiplier is optimal, i.e., $\lambda = \lambda^*$;**
- (2) **$\text{Core}(N, \pi_{LR2}^\lambda)$ is non-empty.**

Instrument 1: LRB Cost Allocation, 2016, IJOC

Applications of the LRB Cost Allocation Method

Applications on Four Types of Facility Location Games

	Sub-modular	Not Sub-modular	Applicable Method(s)
Linear	Type one: UFL	Type two: CFL	LPB, LRB
Non-linear	Type three: NLUFL	Type four: NLCFL	LRB

Instrument 1: LRB Cost Allocation, 2016, IJOC

Uncapacitated Facility Location Game

Theorem

LRB cost allocation and LPB cost allocation are both optimal.

The optimal cost allocations computed by different methods for the example

Method	Player 1	Player 2	Player 3	Player 4	Total
LPB with Simplex	5.00	6.50	8.50	6.50	26.5
LPB with Interior Point	6.58	6.50	8.50	4.92	26.5
LRB	6.87	6.50	8.50	4.63	26.5

- LRB cost allocation **shares the same amount of cost** as LPB cost allocation, in addition, LRB algorithm can sometimes generate cost allocations which are **not easy to obtain by conventional LPB methods**.

Instrument 1: LRB Cost Allocation, 2016, IJOC

Single Source Capacitated Facility Location Game

Q	Average (%)		LRCA - LPCA (%)		Total time (s)		
	LPCA	LRCA	Max	Min	Avg.	Max	Min
10	97.15	98.79	2.38	1.00	19	21	18
20	97.20	98.31	1.51	0.88	24	27	23
30	94.70	95.25	0.75	0.38	15	28	21
40	94.11	94.25	0.28	0.07	24	24	23
50	93.87	93.88	0.04	-0.02	31	35	27

- **Effectiveness** of LPB and LRB cost allocation methods applied in CFL.
- **Sharpness** of LRB cost allocation compared with LPB cost allocation.
- **Convergence** of LPB and LRB cost allocation as capacity increases.
- **Time efficiency** of LRB cost allocation method.

Instrument 1: LRB Cost Allocation, 2016, IJOC

Non-linear UFL Game

θ	LRCA / BFSC (%)			Total Time (s)		
	Avg.	Max	Min	Avg.	Max	Min
UFL	74.37	78.27	69.52	-	-	-
0.01	78.31	82.27	74.65	379	394	343
0.10	87.75	91.13	84.18	415	484	382
0.50	95.83	96.49	95.02	446	519	381
1.00	97.82	98.41	97.37	478	550	383

- **Effectiveness** of NLUFL LRB cost allocations.
- **Time efficiency** of LRB cost allocation method applied in NLUFL.

Instrument 1: LRB Cost Allocation, 2016, IJOC

Non-linear CFL Game

Q	θ	LRB / BFSC (%)			Total time(s)		
		Avg	Max	Min	Avg	Max	Min
10	0.01	99.64	99.70	99.55	5683	6838	4987
	0.10	99.87	99.89	99.78	5690	6834	4980
	0.50	99.90	99.92	99.87	5742	6814	5036
20	0.01	99.61	99.76	99.48	9925	10478	9485
	0.10	99.83	99.85	99.82	9835	10458	9322
	0.50	99.85	99.88	99.84	9825	10487	9315
30	0.01	99.02	99.15	98.82	11686	12831	10410
	0.10	99.73	99.77	99.67	11755	12816	10421
	0.50	99.81	99.87	99.78	11485	13064	10277

- Effectiveness of LRB cost allocations.
- Time efficiency of LRB cost allocation method applied in NLCFL.

★ Cooperative Game Theory:

- New Cost Allocation Method via Lagrangian Relaxation;
- Generic framework applicable to linear, non-linear cases;
- Effective method which in general can generate better cost allocations than LPB method.

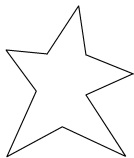
★ Models, Solution Methods:

- Lagrangian relaxation, Lagrangian decomposition, Subgames analyses;

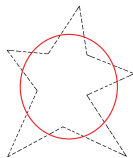
★ Applications:

- Implementations on four types of facility location games.

2. INSTRUMENT OF SIMULT. P&S



Unbalanced Game



Penalization & Subsidization

Instrument 2: Simultaneous P&S, 2018, OR

Instrument of Penalization (Least Core)

$$\text{Core}(N, \pi) = \left\{ \alpha : \alpha(N) = \pi(N), \alpha(S) \leq \pi(S), \forall S \in \mathbb{S} \setminus \{N\}, \alpha \in \mathbb{R}^n \right\}.$$

Relax Coalition Stability constraints: $\alpha(S) \leq \pi(S)$

- **Minimum Penalty** to stabilize the grand coalition:

$$z^* = \min \left\{ z : \alpha(N) = \pi(N), \alpha(S) \leq \pi(S) + z, \forall S \in \mathbb{S} \setminus \{N\} \right\}.$$

Instrument 2: Simultaneous P&S, 2018, OR

Instrument of Penalization (Least Core)

$$\text{Core}(N, \pi) = \left\{ \alpha : \alpha(N) = \pi(N), \alpha(S) \leq \pi(S), \forall S \in \mathbb{S} \setminus \{N\}, \alpha \in \mathbb{R}^n \right\}.$$

Relax Coalition Stability constraints: $\alpha(S) \leq \pi(S)$

- **Minimum Penalty** to stabilize the grand coalition:

$$z^* = \min \left\{ z : \alpha(N) = \pi(N), \alpha(S) \leq \pi(S) + z, \forall S \in \mathbb{S} \setminus \{N\} \right\}.$$

Minimum Penalty needed for MSG Example:

$$z^* = 0.5$$

Instrument 2: Simultaneous P&S, 2018, OR

Penalty-Subsidy Pair

$$\text{Core}(N, \pi) = \left\{ \alpha : \alpha(N) = \pi(N), \alpha(S) \leq \pi(S), \forall S \in \mathbb{S} \setminus \{N\}, \alpha \in \mathbb{R}^n \right\}.$$

Relax Coalition Stability and Budget Balance constraints

- **Penalty-Subsidy Pair** to stabilize the grand coalition:

$$\omega(z) = \min_{\alpha} \left\{ \pi(N) - \alpha(N) : \alpha(S) \leq \pi(S) + z, \forall S \in \mathbb{S} \setminus \{N\} \right\}$$

Instrument 2: Simultaneous P&S, 2018, OR

Penalty-Subsidy Pair

$$\text{Core}(N, \pi) = \left\{ \alpha : \alpha(N) = \pi(N), \alpha(S) \leq \pi(S), \forall S \in \mathbb{S} \setminus \{N\}, \alpha \in \mathbb{R}^n \right\}.$$

Relax Coalition Stability and Budget Balance constraints

- **Penalty-Subsidy Pair** to stabilize the grand coalition:

$$\omega(z) = \min_{\alpha} \left\{ \pi(N) - \alpha(N) : \alpha(S) \leq \pi(S) + z, \forall S \in \mathbb{S} \setminus \{N\} \right\}$$

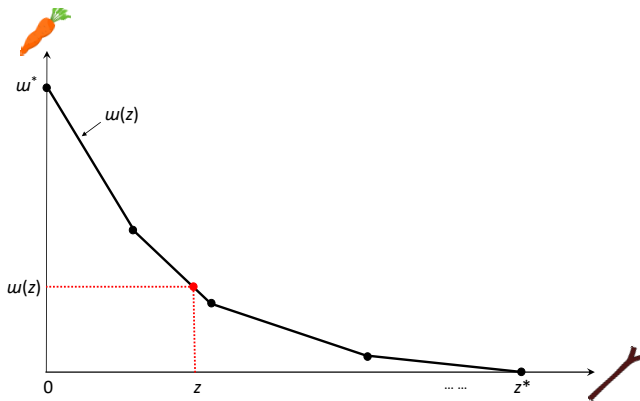
Penalty-Subsidy Pair for MSG Example:

E.g., combination of penalty 0.1667 and subsidy 0.5

Instrument 2: Simultaneous P&S, 2018, OR

Research Questions

How to use penalty and subsidy simultaneously to stabilize the grand cooperation? What's the trade-off?



Instrument 2: Simultaneous P&S, 2018, OR

Penalty Subsidy Function $\omega(z)$ – Properties

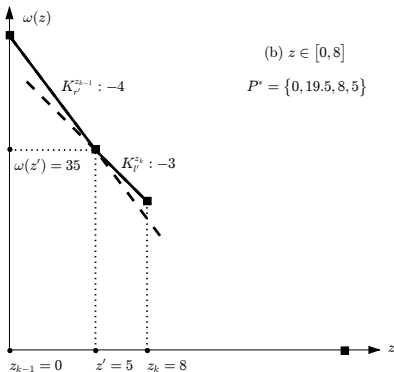
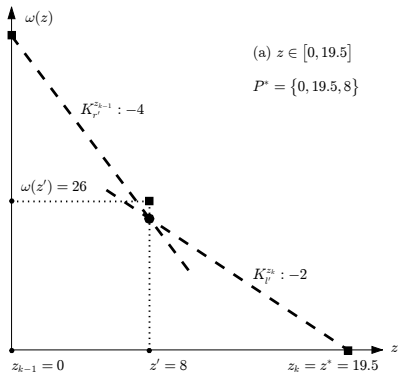
Theorem

- $\omega(z)$ is decreasing, piecewise linear, and convex in $z \in [0, z^*]$.
- For each segment of $\omega(z)$, the slope $\omega'(z) \in \left[-n, -\frac{n}{n-1}\right]$.

- Decreasing of $\omega(z)$: trade-off between penalty and subsidy;
- Convexity of $\omega(z)$: diminishing effect on increasing the penalty to reduce the minimum subsidy desired.

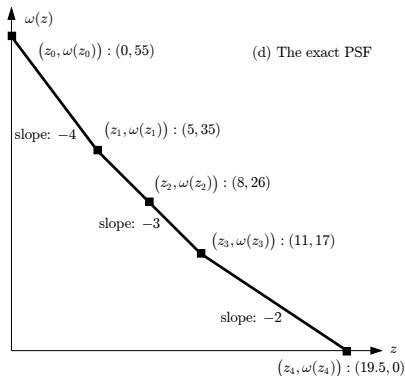
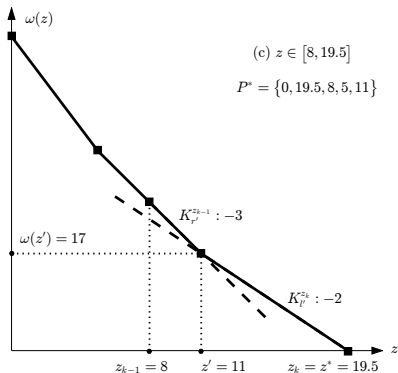
Instrument 2: Simultaneous P&S, 2018, OR

Intersection Points Computation Algorithm: Illustrations on Scheduling Game



Instrument 2: Simultaneous P&S, 2018, OR

Intersection Points Computation Algorithm: Illustrations on Scheduling Game

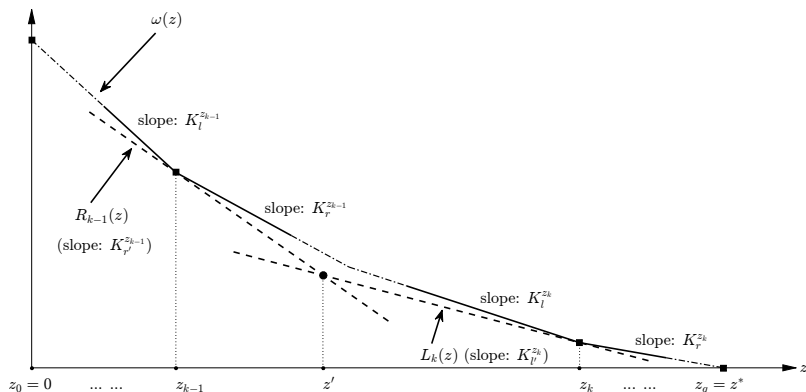


Instrument 2: Simultaneous P&S, 2018, OR

Efficiency of the IPC Algorithm

Theorem

If function $\omega(z)$ has $\hat{q} \geq 2$ linear segments, then the IPC algorithm will terminate after at most $4\hat{q} - 1$ iterations.



Instrument 2: Simultaneous P&S, 2018, OR

Heuristic to Construct $\omega(z)$: ϵ -Approximation

Step 1. Evenly divide $[0, z^*]$ into $\lceil 2v/\epsilon \rceil$ sub-intervals;

Step 2. For each sub-interval, compute $(z_i, \omega(z_i))$;

Step 3. Obtain upper bound $U_\epsilon(z)$ for $\omega(z)$ by linking these points.

Theorem

$$E_c \leq (\epsilon/2)(z^*)^2 \leq \epsilon \int_0^{z^*} \omega(z) dz \text{ and } E_{\max} \leq (\epsilon z^*)/2, (\epsilon > 0).$$

Cumulative error: $E_c = \int_0^{z^*} |U_\epsilon(z) - \omega(z)| dz,$

Maximum error: $E_{\max} = \max_{z \in [0, z^*]} \{|U_\epsilon(z) - \omega(z)|\}.$

Instrument 2: Simultaneous P&S, 2018, OR

How to compute $\omega(z)$ for given penalty z ?

$$\omega(z) = \min_{\alpha} \left\{ \pi(N) - \alpha(N) : \right. \\ \left. \alpha(S) \leq \pi(S) + z \text{ for all } S \in \mathbb{S} \setminus \{N\}, \alpha \in \mathbb{R}^n \right\}.$$

Define:

$$\tau(z) = \max_{\alpha} \left\{ \alpha(N) : \right. \\ \left. \alpha(S) \leq \pi(S) + z \text{ for all } S \in \mathbb{S} \setminus \{N\}, \alpha \in \mathbb{R}^n \right\}.$$

We obtain that $\omega(z) = c(N) - \tau(z)$.

Instrument 2: Simultaneous P&S, 2018, OR

Cutting-Plane (CP) Approach to Computing $\omega(z)$ for Given z

Step 1. $\mathbb{S}' \subseteq \mathbb{S} \setminus \{N\}$ indicates a **restricted coalition set**.

Step 2. Find an optimal $\bar{\alpha}(\cdot, z)$ to a **relaxed LP** of $\tau(z)$:

$$\max_{\alpha \in \mathbb{R}^n} \{ \alpha(N, z) : \alpha(S, z) \leq \pi(S) + z, \text{ for all } S \in \mathbb{S}' \}.$$

Step 3. Find an optimal S^* to the **separation problem**:

$$\delta = \min \{ \pi(S) + z - \bar{\alpha}(S, z) : \forall S \in \mathbb{S} \setminus \{N\} \}.$$

Step 4. If $\delta < 0$, then add S^* to \mathbb{S}' , and go to step 2; otherwise, return $\omega(z) = \pi(N) - \bar{\alpha}(N, z)$.

Generate a **lower bound** when serving as a heuristic

Instrument 2: Simultaneous P&S, 2018, OR

Linear Programming (LP) Approach to Computing $\omega(z)$ for a Given z

Inspired by Caprara & Letchford (2010) for $\tau(0)$, define:

$$\begin{aligned} Q^{x\mu y} &= \{(x, \mu, y) : Ax \geq By + D\mu, y = y^s \text{ for some } s \in S \setminus \{N\}, \\ &\quad \mu = 1, x \in \mathbb{Z}^t, y \in \{0, 1\}^n\}, \\ C^{x\mu} &= \text{proj}_{x\mu}(\{(x, \mu, y) \in \mathbb{R}^{t+1+n} : y = \mathbf{1}\} \cap \text{cone } Q^{x\mu y}). \end{aligned}$$

Theorem

$$\tau(z) = \min \{cx + z\mu : (x, \mu) \in C^{x\mu}\}.$$

Generate **an upper bound** when serving as a heuristic

Instrument 2: Simultaneous P&S, 2018, OR

Applications

Parallel Machine Scheduling Games:

- Players $N = \{1, \dots, n\}$; Machines $M = \{1, \dots, m\}$.
- Each coalition S minimizes the total (weighted) completion time of jobs of players in S .

Results on computing $\omega(z)$ for given z :

	Machines	Jobs	CP Approach	LP Approach
IPU	Identical	Unweighted	P-time	P-time
UPU	Unrelated	Unweighted	–	P-time
IPW	Identical	Weighted	Pseudo P-time (fixed m)	–
UPW	Unrelated	Weighted	Lower Bound	Upper Bound

Instrument 2: Simultaneous P&S, 2018, OR

Conclusions

★ Cooperative Game Theory:

- New Instrument for Stabilization via Simultaneous P&S.

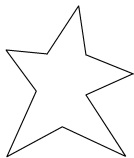
★ Models, Solution Methods:

- Characterize properties of the penalty-subsidy function $\omega(z)$, revealing the trade-off;
- Develop two algorithms to construct function $\omega(z)$;
- Develop two solution approaches to computing the values of $\omega(z)$ for any given z .

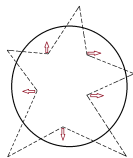
★ Applications:

- Implementations on several machine scheduling games.

3. INSTRUMENT OF COST ADJUSTMENT



Unbalanced Game



Cost Adjustment

Instrument 3: Cost Adjustment, 2019, Under Review

Ex-post Actions

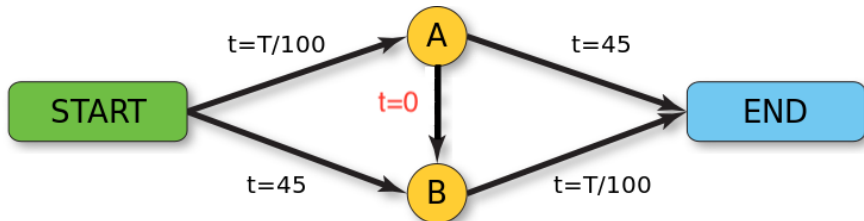
Ex-post actions on $\text{Core}(N, \pi)$

$$\text{Core}(N, \pi) = \left\{ \alpha : \alpha(N) = \pi(N), \alpha(S) \leq \pi(S), \forall S \in \mathbb{S} \setminus \{N\} \right\}$$

- Subsidization: $\alpha(N) = \pi(N) - \theta$, **ϵ -core**;
- Penalization: $\alpha(S) \leq \pi(S) + z$, **least core**;
- Simult. S & P: $\alpha(N) = \pi(N) - \theta$ and $\alpha(S) \leq \pi(S) + z$, **PSF**.

Instrument 3: Cost Adjustment, 2019, Under Review

Illustrative Example: Braess's Paradox



4000 cars in total

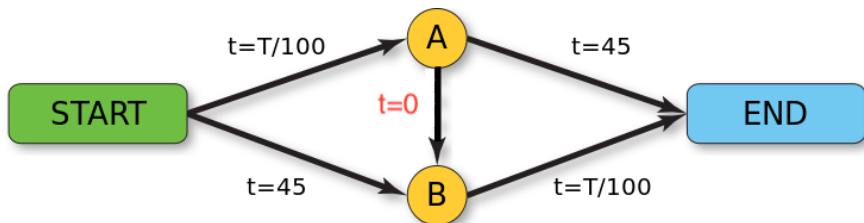
$$t = 0$$

$$START \rightarrow A \rightarrow B \rightarrow END$$

$$4000/100 + 4000/100 = 80min$$

Instrument 3: Cost Adjustment, 2019, Under Review

Illustrative Example: Braess's Paradox



4000 cars in total

$t = \infty$

$t = 0$

$START \rightarrow A \rightarrow B \rightarrow END$

$$4000/100 + 4000/100 = 80min$$

$START \rightarrow A \rightarrow END$

$START \rightarrow B \rightarrow END$

$$A = B = 2000$$

$$2000/100 + 45 = 65min$$

Instrument 3: Cost Adjustment, 2019, Under Review

Ex-ante Action

$$\text{Core}(N, \pi) = \left\{ \alpha : \alpha(N) = \pi(N; c), \alpha(S) \leq \pi(S; c), \forall S \in \mathbb{S} \right\}.$$

$$\pi(S; c) = \min_x \{ cx : Ax \geq By^S + E, x \in \mathbb{Z}^q \}.$$

Ex-ante Action

Cost Adjustment: $c \rightarrow d$ and $\pi(S; c) \rightarrow \pi(S; d)$

Instrument 3: Cost Adjustment, 2019, Under Review

Stabilization via Cost Adjustment

Definition

Grand Coalition Stabilization Problem (GCSP) via Cost Adjustment (CA):

$c \rightarrow d$, such that **BCC**

- **Balancedness**: the updated IM game $(N, \pi(\cdot; d))$ is balanced;
- **Cooperation Scheme**: an initial x^0 is optimal to $\pi(N; d)$;
- **Cost Sharing**: total cost $\pi(N; d)$ to share is within $[l, u]$.

Ex-post actions on $\text{Core}(N, \pi)$ V.S. Ex-ante action on $\pi(\cdot, c)$

Constrained Inverse Optimization Problem (CIOP)

$$\min \left\{ f(\delta) : \text{Core} | (N, \pi(\cdot; d)) | \geq 1, \, dx^0 = \pi(N; d), \, l \leq \pi(N; d) \leq u, \, d \in \mathbb{R}^q \right\}.$$

- Only Optimality: **Inverse Optimal Solution** Problem;
- Only Consistency: **Inverse Optimal Value** Problem;
- Only Balancedness: **Optimal Cost Allocation** Problem.

Instrument 3: Cost Adjustment, 2019, Under Review

\mathcal{NP} -hardness and Feasibility

Theorem

Solving the CIOP is in general \mathcal{NP} -hard.

Theorem

Feasibility – Sufficient and Necessary Conditions

Theorem

Feasibility – Necessary Conditions

Theorem

Feasibility – Sufficient Conditions

Instrument 3: Cost Adjustment, 2019, Under Review

Reformulation 1: Column Generation Method

Lemma

The CIOP is equivalent to the following LP.

$$\begin{aligned} \min \quad & \omega \times (\tau + \eta)^T \\ \text{s.t.} \quad & \alpha \mathbf{1} = dx^0, \\ & \alpha y \leq dx, \quad \forall (x, y) \in Q_{xy}, \\ & l \leq dx^0 \leq u, \\ & d - c = \tau - \eta, \text{ and } d \in \mathbb{R}^q, \tau \in \mathbb{R}_+^q, \eta \in \mathbb{R}_+^q. \end{aligned}$$

Instrument 3: Cost Adjustment, 2019, Under Review

Reformulation 1: Column Generation Method

- Step 1.** Let \hat{Q}_{xy} be a subset of Q_{xy} ;
- Step 2.** Find an optimal solution $[\hat{\tau}; \hat{\eta}; \hat{d}; \hat{\alpha}]$ to a relaxed LP of (1), where Q_{xy} is replaced by \hat{Q}_{xy} ;
- Step 3.** Find an optimal solution $[x'; y']$ to separation problem $\epsilon = \min \{ \hat{d}x - \hat{\alpha}y : \forall (x, y) \in Q_{xy} \}$;
- Step 4.** If $\epsilon < 0$, then add $[x'; y']$ to \hat{Q}_{xy} , go to step 2; otherwise, return (i) the updated cost coefficients \hat{d} ; and (ii) the total minimum perturbation $\omega \times (\tau + \eta)^T$.

Generate **a lower bound** when serving as a heuristic

Lemma

The CIOP is equivalent to the following LP.

$$\min_{\tau, \eta, d, \rho} \left\{ \omega \times (\tau + \eta)^T : \mathbb{B}^T \rho \geq dx^0; \mathbb{A}^T \rho = d^T; l \leq dx^0 \leq u; \right. \\ \left. \tau - \eta = d - c; \text{ and } d \in \mathbb{R}^q, \tau, \eta, \rho \in \mathbb{R}_+^q \right\},$$

where $C_x = \{x : \mathbb{A}x \geq \mathbb{B}\}$ and ρ is the associating dual variable.

Instrument 3: Cost Adjustment, 2019, Under Review

Reformulation 2: Cone Optimization Method

- Step 1.** Derive an expression of C_x , denoted as $\{x : Ax \geq B\}$, with finite number of constraints for IM game $(N, \pi(\cdot; c))$;
- Step 2.** Find an optimal solution $[\tau'; \eta'; d'; \rho']$ to the CIOP;
- Step 3.** Return (i) the optimal cost coefficients d' ; and (ii) the total minimum adjustment cost $\omega \times (\tau' + \eta')^T$.

Generate **an upper bound** when serving as a heuristic

Instrument 3: Cost Adjustment, 2019, Under Review

Computational Results: weighted matching and uncapacitated facility location

(M , E)	U	DV%	Number of Adjusted Arcs		
		Avg.	Avg.	Max.	Min.
(30, 435)	16	0.018	1.87	4	1
(40, 780)	21	0.005	1.57	3	1
(50, 1225)	33	0.003	1.64	4	1
(60, 1770)	26	0.002	1.96	6	1

(M , N)	U	DV%	Number of Adjusted Arcs			F
		Avg.	Avg.	Max.	Min.	
(20, 20)	75	0.226	5.32	17	1	0
(40, 40)	99	0.168	18.66	35	2	0
(60, 60)	100	0.133	35.32	81	3	0
(80, 80)	100	0.129	63.31	100	19	0

- ★ **Cooperative Game Theory:**
 - **New Instrument** for Stabilization via Cost Adjustment.
- ★ **Inverse Optimization:**
 - **Constrained** Inverse Optimization Problem.
- ★ **Models, Solution Methods and Applications:**
 - Several Equivalent LP Formulations;
 - Time complexity & Feasibility analyses;
 - Implementations on WM and UFL games.

The End

Thank You!