Stabilizing Grand Cooperation of Machine Scheduling Game via Setup Cost Pricing

Lindong Liu

School of Management; International Institute of Finance

University of Science and Technology of China

Co-authored with Zikang Li (PG Student, USTC)

NWPU-Online, July, 2020

Outline

Preliminaries

Cooperative Game

A **cooperative game** is defined by a pair (V, C):

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

The game requires:

- A **cost allocation** $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_v] \in \mathbb{R}^v$, where $\alpha_k =$ the cost allocated to each player $k \in V$.

Core

Define $\alpha(S) = \sum_{k \in S} \alpha_k$.

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

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

$$\mathrm{Core}(V, C) = \Big\{ \alpha : \ \alpha(V) = C(V), \\ \alpha(S) \leq C(S), \ \forall S \in \mathbb{S} \setminus \{V\}, \ \alpha \in \mathbb{R}^V \Big\}.$$

Core

Define $\alpha(S) = \sum_{k \in S} \alpha_k$.

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

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

$$\mathrm{Core}(V, C) = \left\{ \alpha : \ \alpha(V) = C(V), \right.$$
$$\left. \alpha(S) \le C(S), \ \forall S \in \mathbb{S} \setminus \{V\}, \ \alpha \in \mathbb{R}^v \right\}.$$

However, $\mathrm{Core}(V, c)$ can be empty.

Existing Instruments

$$\mathrm{Core}(V, C) = \left\{ \alpha : \ \alpha(V) = C(V), \ \alpha(S) \leq C(S), \ \forall S \in \mathbb{S} \setminus \{V\} \right\}$$

Existing Instruments

$$\mathrm{Core}(V, C) = \left\{ \alpha : \ \alpha(V) = C(V), \ \alpha(S) \leq C(S), \ \forall S \in \mathbb{S} \setminus \{V\} \right\}$$

- Subsidization: $\alpha(V) = C(V) - \theta$, $\epsilon-$**core**;

Existing Instruments

$$\mathrm{Core}(V, C) = \left\{ \alpha : \ \alpha(V) = C(V), \ \alpha(S) \leq C(S), \ \forall S \in \mathbb{S} \setminus \{V\} \right\}$$

- Subsidization: $\alpha(V) = C(V) - \theta$, $\epsilon-$**core**;
- Penalization: $\alpha(S) \leq C(S) + z$, **least core**;

Existing Instruments

$$\mathrm{Core}(V, C) = \left\{ \alpha : \ \alpha(V) = C(V), \ \alpha(S) \leq C(S), \ \forall S \in \mathbb{S} \setminus \{V\} \right\}$$

- Subsidization: $\alpha(V) = C(V) - \theta$, $\epsilon-$**core**;
- Penalization: $\alpha(S) \leq C(S) + z$, **least core**;
- Simul. S & P: $\alpha(V) = C(V) - \theta$ and $\alpha(S) \leq C(S) + z$, **PSF**;

Existing Instruments

$$\mathrm{Core}(V, C) = \left\{\alpha : \ \alpha(V) = C(V), \ \alpha(S) \leq C(S), \ \forall S \in \mathbb{S} \setminus \{V\}\right\}$$

- Subsidization: $\alpha(V) = C(V) - \theta$, $\epsilon-$**core**;
- Penalization: $\alpha(S) \leq C(S) + z$, **least core**;
- Simul. S & P: $\alpha(V) = C(V) - \theta$ and $\alpha(S) \leq C(S) + z$, **PSF**;
- Inv. Opt.: Changing c to d such that $\mathrm{Core}(V, D)$ is non-empty.

Existing Instruments

$$\mathrm{Core}(V, C) = \left\{ \alpha : \ \alpha(V) = C(V), \ \alpha(S) \le C(S), \ \forall S \in \mathbb{S} \setminus \{V\} \right\}$$

- Subsidization: $\alpha(V) = C(V) - \theta$, $\epsilon-$**core**;
- Penalization: $\alpha(S) \le C(S) + z$, **least core**;
- Simul. S & P: $\alpha(V) = C(V) - \theta$ and $\alpha(S) \le C(S) + z$, **PSF**;
- Inv. Opt.: Changing c to d such that $\mathrm{Core}(V, D)$ is non-empty.

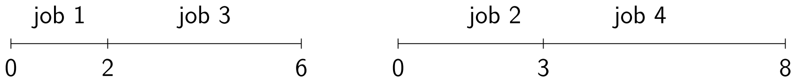
S.	Caprara and Letchford (2010, MP), Liu et al. (2016, IJOC)
P.	Faigle et al. (2001, IJGT), Schulz and Uhan (2010, OR)
P&S	Liu et al. (2018, OR)
Inv. Opt.	Liu et al. (2020, under review)

Illustrative Example

Example: Machine Scheduling Game (MSG)

Game of Parallel Machine Scheduling with Setup Cost:

- Grand coalition: $V = \{1, 2, 3, 4\}$;
- Processing times: $t_1 = 2,\ t_2 = 3,\ t_3 = 4,\ t_4 = 5$;
- Machine setup cost: $t_0 = 9.5$;
- $c(S)$ 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
$\{1,2,3,4\}$	38

Optimal Cost Allocation Problem

$$\begin{aligned} \max \ & \left(\alpha_1+\alpha_2+\alpha_3+\alpha_4\right) = 37.25 < 38 \\ s.t.\ & \alpha_1 \le 11.5, \ \cdots, \ \alpha_4 \le 14.5, \\ & \alpha_1+\alpha_2 \le 16.5, \ \cdots, \ \alpha_3+\alpha_4 \le 22.5, \\ & \cdots, \\ & \alpha_1+\alpha_2+\alpha_3+\alpha_4 \le 38. \end{aligned}$$

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

Models & Analyses

Problem Definition and Formulation

Definition

A cooperative TU game (V,c) is called an Identical Variable Parallel machine scheduling of Unweighted jobs(IVPU) if it satisfies the following formulations:

Problem Definition and Formulation

Definition

A cooperative TU game (V,c) is called an Identical Variable Parallel machine scheduling of Unweighted jobs(IVPU) if it satisfies the following formulations:

- Optimality: an initial feasible solution x^0 is optimal to $D(V)$;

Problem Definition and Formulation

Definition

A cooperative TU game (V,c) is called an Identical Variable Parallel machine scheduling of Unweighted jobs(IVPU) if it satisfies the following formulations:

- Optimality: an initial feasible solution x^0 is optimal to $D(V)$;
- Consistency: the social optimum is consistent $C(V) = D(V)$;

Problem Definition and Formulation

Definition

A cooperative TU game (V,c) is called an Identical Variable Parallel machine scheduling of Unweighted jobs(IVPU) if it satisfies the following formulations:

- Optimality: an initial feasible solution x^0 is optimal to $D(V)$;
- Consistency: the social optimum is consistent $C(V) = D(V)$;
- Balancedness: the updated IM game (V, D) is balanced.

Problem Definition and Formulation

Definition

A cooperative TU game (V,c) is called an Identical Variable Parallel machine scheduling of Unweighted jobs(IVPU) if it satisfies the following formulations:

- Optimality: an initial feasible solution x^0 is optimal to $D(V)$;
- Consistency: the social optimum is consistent $C(V) = D(V)$;
- Balancedness: the updated IM game (V, D) is balanced.

$$\min_{\delta} \left\{ f(\delta) : \ d \in \mathbb{O}, \ D(V) = C(V), \ dx^0 = D(V), \ \delta = d - c \right\},$$

where $f(\delta) = \omega \times |\delta|$ (L_1 norm); $\mathbb{O}$ is the Balanced Cost Vector Set.

Problem Definition and Formulation

Definition

A cooperative TU game (V,c) is called an Identical Variable Parallel machine scheduling of Unweighted jobs (IVPU) if it satisfies the following formulations:

Problem Definition and Formulation

Definition

A cooperative TU game (V,c) is called an Identical Variable Parallel machine scheduling of Unweighted jobs (IVPU) if it satisfies the following formulations:

$$
\begin{aligned}
c(s,P) = &\min \sum_{k\in V}\sum_{j\in O} c_{kj}x_{kj} + P\sum_{k\in s} x_{k1} \\
s.t.\quad & \sum_{j\in O} x_{kj} - y_k^s = 0, \forall k \in V, \\
& \sum_{k\in V} x_{kj} \le m, \forall j \in O, \\
& x_{kj} \in \{0,1\}, \forall k \in V, \forall j \in O, \\
& y_k^s = 1, k \in s \,;\, y_k^s = 0, k \notin s.
\end{aligned}
$$

Properties

Algorithms & Computations

… Algorithm

Computational Results

Extension & Generalization

Machine Scheduling Game with Weighted Jobs

Pricing in General IM Games

Conclusions

- ⋆ Cooperative Game Theory:
  - New Instrument for Stabilization via Cost Adjustment.
- ⋆ Inverse Problem:
  - Constrained Inverse Optimization Problem.
- ⋆ Models, Solution Methods and Applications:
  - Several equivalent LP formulations;
  - Feasibility analyses & How to handle infeasibility;
  - Implementations on WMG and UFL games.

The End

Thank you!