中国科学技术大学
University of Science and Technology of China

Simultaneous Penalization and Subsidization for Stabilizing Grand Cooperation

Reporter： Chengcheng Yu
Date: 2023.10.25

Title: Simultaneous Penalization and Subsidization for Stabilizing Grand Cooperation

- Published on ***Operations Research***

  - Received: ***June 3, 2015***
  - Revised: ***September 30, 2016; July 14, 2017; November 30, 2017***
  - Accepted: ***December 23, 2017***
  - Area of Review: ***Optimization***
  - Keywords: cooperative game; grand coalition stability; simultaneous penalization and subsidization; parallel machine scheduling game

  - Authors: Lindong Liu, Xiangtong Qi, Zhou Xu

Paper Information

- **PhD thesis:** To Stabilize Grand Coalitions in Unbalanced Cooperative Games
  - Computing Near-Optimal Stable Cost Allocations for Cooperative Games by Lagrangian Relaxation
  - Simultaneous Penalization and Subsidization for Stabilizing Grand Cooperation
  - Stabilizing Grand Cooperation via Cost Adjustment: An Inverse Optimization Approach

- The most important reference: *Caprara, A., & Letchford, A. N. (2010). New techniques for cost sharing in combinatorial optimization games. Mathematical programming, 124, 93-118.*

1. Background

University of Science and Technology of China

■ Motivation:

✓ Two known instruments:

- **Penalization and subsidization**
- **Penalization**: Charging a penalty causes players to be dissatisfied
- **Subsidization**: Providing a subsidy to the grand coalition is at the cost of injecting external resources.

✓ Stick-and-carrot:

- Charges penalties and provides subsidies simultaneously

Research Problem

✓ **Simultaneous Penalization and Subsidization for Stabilizing Grand Cooperation**

- Penalty and subsidy become complementary
- Trade-off between penalty and subsidy

中国科学技术大学
University of Science and Technology of China

Preliminaries:

- Core:

$$\text{Core}(V,c)=\{\theta{:}\,\theta(V)=c(V),\theta(s)\le c(s)\ \textbf{for all}\ s\in S\setminus\{V\},\theta\in\mathbb{R}^v\}.$$

- Two instruments to stabilize the grand coalition:
- Penalization:

$$z^*=\min_{\beta,z}\{z{:}\,\beta(V)=c(V),\beta(s)\le c(s)+z\ \textbf{for all}\ s\in S\setminus\{V\},z\in\mathbb{R},\beta\in\mathbb{R}^v\}.$$

- Subsidization:

$$\omega^*=\min_{\alpha}\{c(V)-\alpha(V){:}\,\alpha(s)\le c(s)\ \textbf{for all}\ s\in S,\alpha\in\mathbb{R}^v\},$$

which is equivalent to the OCAP in Caprara and Letchford (2010):

$$\max_{\alpha}\{\alpha(V){:}\,\alpha(s)\le c(s)\ \textbf{for all}\ s\in S,\alpha\in\mathbb{R}^v\}$$

学管
院理

中国科学技术大学
University of Science and Technology of China

■ Penalty-subsidy Function:

➢ **Penalty-subsidy function (PSF):**

Definition 1. *In a cooperative game* (V, c), *for any penalty* $z \in \mathbb{R}$, *consider the following LP:*

$$\omega(z) = \min_{\beta}\{c(V) - \beta(V) \colon \beta(s) \leq c(s) + z \ \textbf{for all}\ s \in S \setminus \{V\}, \beta \in \mathbb{R}^n\}.$$

➢ For any penalty z that is not sufficient, the central authority provides a subsidy ω(z) to make the grand coalition cooperate.

➢ Monotonicity and two extreme cases:

Lemma 1. *The penalty-subsidy function* $\omega(z)$ *is strictly decreasing in* z *for* $z \in [0, z^*]$. *In addition,* $\omega(0) = \omega^*, \omega(z^*) = 0, and 0 < \omega(z) < \omega^*$ *for any* $z \in (0, z^*)$.

学院 管理

Penalty-subsidy Function:

Example 1:

EXAMPLE 1. Consider an SMW game with $V=\{1,2,3,4\}$ of four players. Each player $k\in V$ has a job with weight w_k and processing time t_k, where $w_1=4$, $w_2=3$, $w_3=2$, $w_4=1$, $t_1=5$, $t_2=6$, $t_3=7$, and $t_4=8$. Each coalition $s\in S$ aims to minimize the total weighted completion time by processing all their jobs on a single machine.

Table 1 z-penalized minimum subsidies and z-penalized optimal cost allocations for Example 1.

z	0	5	10	15	19.5
$\omega(z)$	55	35	20	9	0
$\beta(1,z)$	20.00	25.00	29.29	31.62	34.70
$\beta(2,z)$	18.00	23.00	28.00	31.45	34.12
$\beta(3,z)$	14.00	19.00	24.00	27.38	28.80
$\beta(4,z)$	8.00	13.00	13.71	15.55	17.38

Structural Properties:

- **Taking the dual of LP (5), by strong duality we have:**

$$\omega(z) = \boldsymbol{max}_{\rho}\{c(V) + \sum_{s\in S\setminus\{V\}} -\rho_s[c(s)+z] : \sum_{s\in S\setminus\{V\},k\in s} \rho_s = 1, \forall k \in V, \rho_s \geq 0, \forall s \in S \setminus \{V\}\}.$$

- ***Maximally unsatisfied coalitions***: $\beta(s,z) = c(s) + z.$
- Let $S^{\beta z} = \left\{s_1^{\beta z}, s_2^{\beta z}, \dots, s_{h(\beta,z)}^{\beta z}\right\}$ denote the collection of all maximally unsatisfied coalitions, where $\mathrm{h}(\beta, \mathrm{z}) = |\mathrm{S}^{\beta \mathrm{z}}|$.

Theorem 1. *Consider any penalty z, and any z -penalized optimal cost allocation β(· , z). The union of all maximally unsatisfied coalitions in $S^{\beta z}$ equals the grand coalition V , i.e.,*

$$s_1^{\beta z} \cup s_2^{\beta z} \cup \cdots \cup s_{h(\beta,z)}^{\beta z} = V.$$

Structural Properties:

Theorem 2. *ω(z) is strictly decreasing, piecewise linear, and convex in penalty z for* $z \in [0, z^*]$.

➢ **Implications**:

- Strong complementarity;
- Fully characterize the PSF at only a finite number;
- A diminishing effect.

Theorem 3. *For each linear segment of ω(z), its derivative* $\omega'(z)$ *is in the range* $[-v, -\frac{v}{v-1}]$.

Implications: The derivatives of ω(z) may have large variations, depending on the number of players v.

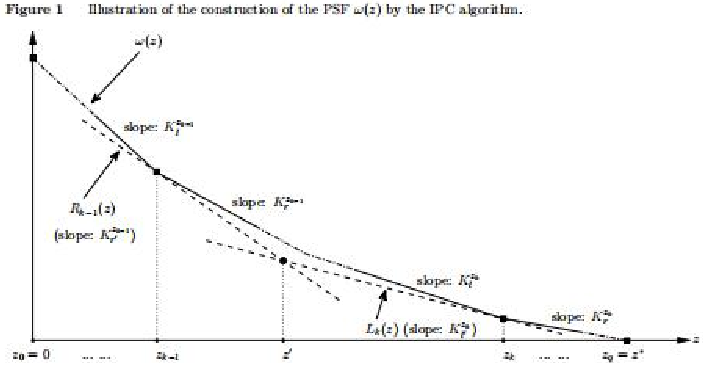
Structural Properties:

- Let $\prod^{z}$ denote the set of all optimal solutions ρ to LP (6).

$$K_l^z = \min\left\{\sum\nolimits_{s\in S\setminus\{V\}} -\rho_s : \rho \in \Pi^z\right\} \textbf{ and } K_r^z = \max\left\{\sum\nolimits_{s\in S\setminus\{V\}} -\rho_s : \rho \in \Pi^z\right\}.$$

If, and only if $K_l^z \neq K_r^z$, point $\left(z, \omega(z)\right)$ is a breakpoint on the PSF curve.

Figure 1 Illustration of the construction of the PSF $\omega(z)$ by the IPC algorithm.



- **Weak derivatives** : $K_l^z \leq K_{l'}^z \leq K_{r'}^z \leq K_r^z$

3. Analyses

■ Construction of the PSF *ω(z)* :

➢ Construction of the Exact PSF:

- To construct a set P^* of values from $[0, z^*]$ that cover all the breakpoints of *ω(z)* , and then connect points $(z, \omega(z))$ for all $z \in P^*$.

Algorithm 1 Intersection Points Computation (IPC) Algorithm to Construct the PSF

Step 1. Initially, set $P^* = \{0, z^*\}$ and $\mathbb{P} = \{[0, z^*]\}$.

Step 2. If $\mathbb{P}$ is not empty, update P^* and $\mathbb{P}$ by the following steps:

Step 2.1. Relabel values in P^* by $z_0 < z_1 < \cdots < z_q$, where $z_0 = 0$, $z_q = z^*$ and $q = |P^*| - 1$.

Step 2.2. Select any interval from $\mathbb{P}$, denoted by $[z_{k-1}, z_k]$ with $1 \le k \le q$.

Step 2.3. Construct two linear functions $R_{k-1}(z)$ and $L_k(z)$ so that $R_{k-1}(z)$ passes $(z_{k-1}, \omega(z_{k-1}))$ with a slope equal to a right weak derivative $K_{r'}^{z_{k-1}}$ of $\omega(z)$ at z_{k-1}, and that $L_k(z)$ passes $(z_k, \omega(z_k))$ with a slope equal to a left weak derivative $K_{l'}^{z_k}$ of $\omega(z)$ at z_k.

Step 2.4. Consider the following two cases:

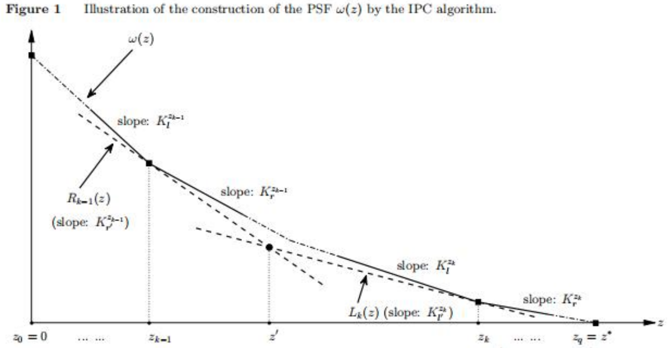
Case 1: If $R_{k-1}(z)$ passes $(z_k, \omega(z_k))$ or $L_k(z)$ passes $(z_{k-1}, \omega(z_{k-1}))$, then update $\mathbb{P}$ by removing $[z_{k-1}, z_k]$.

Case 2: Otherwise, $R_{k-1}(z)$ and $L_k(z)$ must have a unique intersection point at $z = z'$ for some $z' \in (z_{k-1}, z_k)$. Update P^* by adding z', and update $\mathbb{P}$ by removing $[z_{k-1}, z_k]$ and adding $[z_l, z']$ and $[z', z_r]$.

Step 2.5. Go to step 2.

Step 3. Return a piecewise linear function by connecting points $(z, \omega(z))$ for all $z \in P^*$.

Figure 1 Illustration of the construction of the PSF $\omega(z)$ by the IPC algorithm.



University of Science and Technology of China

■ Construction of the PSF *ω(z)* :

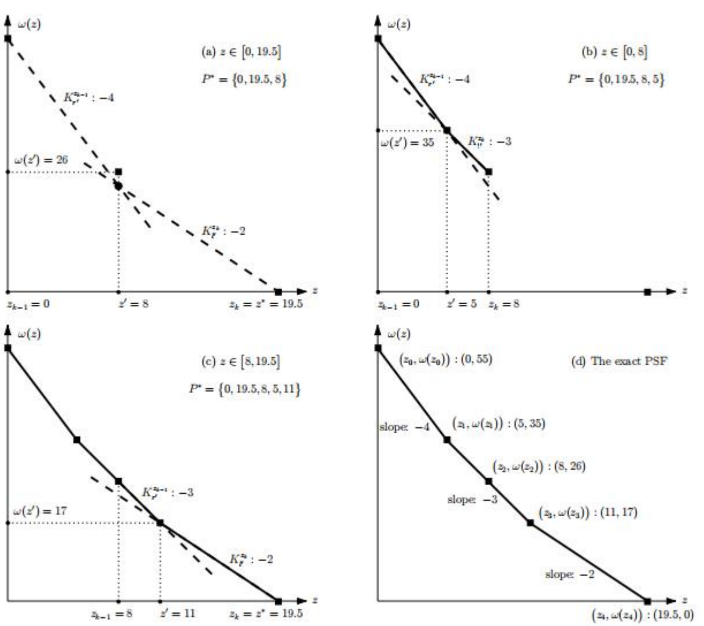
➢ Construction of the Exact PSF:

Theorem 4. *(i) The function returned by the IPC algorithm equals the PSF ω(z) for* $z \in [0, z^*]$. *(ii) If function ω(z) has* $\hat{q} \geq 2$ *linear segments (or equivalently,* $\hat{q}+\mathbf{1}$ *breakpoints), then the IPC algorithm will terminate after at most* $4\hat{q}-\mathbf{1}$ *iterations.*

➢ Instance in Example 1:

Three times: {[0,19.5]} **to**{[0,8], [8,19.5]}, **to**{[8,19.5]}, **and then to an empty set**

Figure 2 Applying the IPC algorithm to constructing the PSF $\omega(z)$ for Example 1.



- Construction of the PSF $\omega(z)$:
  - **ϵ−Approximation of the PSF:**
  - To construct an upper bound function.

Algorithm 2 Approximation Algorithm to Construct an ϵ-Approximation of the PSF

Step 1. Divide $[0, z^*]$ into $\lceil 2v/\epsilon \rceil$ sub-intervals denoted by $[z_0, z_1)$, $[z_1, z_2)$, ..., $[z_{\lceil v/\epsilon \rceil -2}, z_{\lceil 2v/\epsilon \rceil -1})$, and $[z_{\lceil 2v/\epsilon \rceil -1}, z_{\lceil 2v/\epsilon \rceil}]$, such that each segment has the same length of $(z^*/\lceil 2v/\epsilon \rceil)$, where $z_0 = 0$ and $z_{\lceil 2v/\epsilon \rceil} = z^*$.

Step 2. For each $0 \le k \le \lceil 2v/\epsilon \rceil$, compute the z-penalized minimum subsidy $\omega(z)$ for $z = z_k$.

Step 3. Obtain an upper bound $U_\epsilon(z)$ for the PSF $\omega(z)$ by connecting points in $\{(z_0, \omega(z_0)), (z_1, \omega(z_1)), \ldots, (z_{\lceil 2v/\epsilon \rceil -1}, \omega(z_{\lceil 2v/\epsilon \rceil -1})), (z_{\lceil 2v/\epsilon \rceil}, \omega(z_{\lceil 2v/\epsilon \rceil}))\}$.

Theorem 5. $E_c \le (\epsilon/2)(z^*)^2 \le \epsilon \int_0^{z^*} \omega(z)\mathrm{d}z$ and $E_{\max} \le (\epsilon z^*)/2$, for any given $\epsilon > 0$.

4. Solution Approaches

■ Computing the value of *ω(z)* :

➢ **Integer Minimization (IM) games:**

$$c(s) = \min_x \{cx: Ax \geq By^s + D, x \in \mathbb{Z}^t\}.$$

➢ Let $\pi(z)$ denote the optimal objective value of the following LP:

$$\pi(z) = \max_\beta \{\beta(V): \beta(s) \leq c(s) + z \textbf{ for all } s \in S \setminus \{V\}, \beta \in \mathbb{R}^v\},$$

where $\omega(z) = c(V) - \pi(z)$.

➢ Remarks:

- *When either c(V) or π(z) is hard to obtain, we can compute their bounds to obtain a bound on ω(z).*
- *Only its special case π(0), has recently been studied by Caprara and Letchford (2010) and Liu et al. (2016).*

Cutting Plane Approach:

- **LP (12) contains an exponential number of constraints.**

Algorithm 3 Cutting Plane (CP) Approach to Computing $\omega(z)$ for a Given z

Step 1. Let $S' \subseteq S \setminus \{V\}$ indicate a restricted coalition set, which includes some initial coalitions, e.g., $\{1\}, \{2\}, \ldots,$ and $\{v\}$.

Step 2. Find an optimal solution $\bar{\beta}(\cdot, z)$ to a relaxed LP of (12) defined as $\max_\beta \{\beta(V,z) : \beta(s,z) \leq c(s) + z, \text{ for all } s \in S', \beta \in \mathbb{R}^v\}$.

Step 3. Find an optimal solution s^* to the separation problem $\delta = \min\{c(s) + z - \bar{\beta}(s,z) : \forall s \in S \setminus \{V\}\}$.

Step 4. If $\delta < 0$, then add s^* to S', and go to step 2; otherwise, return (i) the z-penalized minimum subsidy $\omega(z) = c(V) - \bar{\beta}(V,z)$; and (ii) a pair of weak derivatives $(K_{l'}^{\bar{\beta}z}, K_{r'}^{\bar{\beta}z})$ computed by solving (9) with $\Pi^{\beta z}$ replaced by $\Pi^{\bar{\beta}z}$.

- The critical part of the above CP approach is how to efficiently solve the separation problem in step 3 to find a violated constraint $\beta(s^*, z) \leq c(s^*) + z$, and this depends on the specific game being studied.

Linear Programming Approach:

➢ **LP approach: based on the theory of linear programming and duality.**

- Let Q^{xy} denote the overall set of feasible solutions to ILP of $c(s)$ for all $s \in S \setminus \{V\}$:

$$Q^{xy} = \{(x, y): Ax \geq By + D, y = y^s \textbf{ for some } s \in S \setminus \{V\}, x \in \mathbb{Z}^t, y \in \{0,1\}^v\}$$

Lemma 3. *If $P^{xy} = \{(x, y): A'x \geq B'y + D'\}$ is a relaxation of Q^{xy}, then $\min\{cx + z\mu: A'x \geq B'1 + D'\mu\} \leq \pi(z)$, which holds with equality if P^{xy} equals the convex hull of Q^{xy}.*

THEOREM 6. *Consider any $P^{xy} = \{(x,y): A'x \geq B'y + D'\}$ that is a relaxation of Q^{xy}, where the dimensions of A', B', and D' are polynomially bounded. Then, we have that:*

(i) the LP approach runs in polynomial time with an upper bound of $\omega(z)$ returned for any given penalty z, which equals $\omega(z)$ if P^{xy} equals the convex hull of Q^{xy}; and that

(ii) there exists a polynomial time algorithm that can produce a z-penalized feasible cost allocation $\beta(\cdot, z)$ with a total shared value of $\min\{cx + z\mu : A'x \geq B'1 + D'\mu\}$, which is optimal if P^{xy} equals the convex hull of Q^{xy}.

Linear Programming Approach:

- **LP approach: based on the theory of linear programming and duality.**

Algorithm 4 Linear Programming (LP) Approach to Computing $\omega(z)$ for a Given z

---

Step 1. Denote the overall set of solutions to programs $c(s)$ for all $s \in S \setminus \{V\}$ by $Q^{xy} = \{(x, y) :$
$Ax \geq By + D, y = y^s$ for some $s \in S \setminus \{V\}, x \in \mathbb{Z}^t, y \in \{0,1\}^v\}$.

Step 2. Relax Q^{xy} to some convex polyhedron $P^{xy} = \{(x, y) : A'x \geq B'y + D'\}$.

Step 3. Find an optimal solution $[x^*, \mu^*]$ to min $\{cx + z\mu : A'x \geq B'\mathbf{1} + D'\mu\}$.

Step 4. Return the value of $c(V) - (cx^* + z\mu^*)$ as an approximation of $\omega(z)$, and return a pair of
$K^z_{l'} = -\mu^*$ and $K^z_{r'} = -\mu^*$.

---

5. Applications to Parallel Machine Scheduling Games

■ Identical parallel machine scheduling of unweighted jobs (**IPU**)

➢ **IPU game: Minimize the total completion time**

$$c_{\mathrm{IPU}}(s) = \min \sum_{k \in s} \sum_{j \in O} c_{kj} x_{kj}$$

$$\textbf{s.t.} \sum_{j \in O} x_{kj} - y_k^s = 0, \forall k \in V,$$

$$\sum_{k \in V} x_{kj} \leq m, \forall j \in O,$$

$$0 \leq x_{kj} \leq 1, x_{kj} \in \mathbb{Z}, \forall k \in V, \forall j \in O.$$

➢ Can be solved by the shortest processing time first (SPT) rule.

CP Approach:

- Separation problem for any given cost allocation $\beta \in \mathbb{R}^v$:

$$\delta_{\mathrm{IPU}} = \min_{s \in S \setminus \{V\}} \{c_{\mathrm{IPU}}(s) + z - \sum_{k \in s} \beta_k \}.$$

- **Solve the separation problem (17) by dynamic programming.**

$$P(k,u) = \min \begin{cases} P(k-1,u), & \text{for the case when } s^* \text{ does not contain } k, \\ P(k-1,u-1) + \lceil u/m \rceil t_k - \beta_k, & \text{for the case when } s^* \text{ contains } k. \end{cases}$$

- **Lemma 4.** *For game (V, c_{IPU}), the separation problem (17) can be solved in $O(v^2)$ time.*

5. Applications to Parallel Machine Scheduling Games

中国科学技术大学 University of Science and Technology of China

LP Approach:

- Let $P^{xy}_{\boldsymbol{IPU}}$ indicate the polyhedron defined by the LP relaxation of $\boldsymbol{Q}^{xy}_{\boldsymbol{IPU}}$, with the integral constraints being relaxed.

Lemma 5. *$P^{xy}_{\boldsymbol{IPU}}$ equals the convex hull of $\boldsymbol{Q}^{xy}_{\boldsymbol{IPU}}$.*

Theorem 7. *For game $(V, c_{\boldsymbol{IPU}})$, the PSF ω(z) has $O(v^4)$ breakpoints, and it can be exactly constructed in polynomial time by the IPC algorithm.*

管理学院

中国科学技术大学
University of Science and Technology of China

Thank You!

Reporter： Chengcheng Yu