Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Simultaneous Penalization and Subsidization for Stabilizing Grand Cooperation

Lindong Liu, Xiangtong Qi, Zhou Xu

informs
http://pubsonline.informs.org/journal/opre/

OPERATIONS RESEARCH
Vol. 66, No. 5, September–October 2018, pp. 1362–1375
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Simultaneous Penalization and Subsidization for Stabilizing Grand Cooperation

Lindong Liu,[a] Xiangtong Qi,[b] Zhou Xu[c]

[a] International Institute of Finance, School of Management, University of Science and Technology of China, 230026, P.R. China;
[b] Department of Industrial Engineering and Decision Analytics, The Hong Kong University of Science and Technology, Kowloon, Hong Kong;
[c] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong

Contact: ldliu@ustc.edu.cn, http://orcid.org/0000-0003-1015-3097 (LL); ieemqi@ust.hk, http://orcid.org/0000-0003-4573-0157 (XQ); lgtzx@polyu.edu.hk, http://orcid.org/0000-0003-1528-116X (ZX)

Received: June 3, 2015
Revised: September 30, 2016; July 14, 2017; November 30, 2017
Accepted: December 23, 2017
Published Online in Articles in Advance: July 27, 2018

Subject Classifications: cooperative game; grand coalition stability; simultaneous penalization and subsidization; parallel machine scheduling game
Area of Review: Optimization

https://doi.org/10.1287/opre.2018.1723

Abstract. In this paper we propose a new instrument, a simultaneous penalization and subsidization, for stabilizing the grand coalition and enabling cooperation among all players of an unbalanced cooperative game. The basic idea is to charge a penalty z from players who leave the grand coalition, and at the same time provide a subsidy ω to players who stay in the grand coalition. To formalize this idea, we establish a penalty-subsidy function $\omega(z)$ based on a linear programming model, which allows a decision maker to quantify the trade-off between the levels of penalty and subsidy. By studying function $\omega(z)$, we identify certain properties of the trade-off. To implement the new instrument, we design two algorithms to construct function $\omega(z)$ and its approximation. Both algorithms rely on solving the value of $\omega(z)$ for any given z, for which we propose two effective solution approaches. We apply the new instrument to a class of machine scheduling games, showing its wide applicability.

Funding: The work is partially supported by the Hong Kong Research Grants Council [Grant 16225316], the National Natural Science Foundation of China [Grants 71401149, 71701192, and 71731010], and the Hong Kong Polytechnic University [Grant 1-ZVDS].
Supplemental Material: The e-companion is available at https://doi.org/10.1287/opre.2018.1723.

Keywords: cooperative game • grand coalition stability • simultaneous penalization and subsidization • parallel machine scheduling game

1. Introduction

In many decision making problems that involve multiple players, minimizing total cost can be pursued by centralized optimization, which essentially requires all the players to form a grand coalition for cooperation. To ensure that the grand coalition is stable, the minimal total cost incurred must be entirely allocated to all the players so that no player or coalition of players can be better off by leaving the grand coalition. This is one of the central themes in cooperative game theory, which has wide applications. See, for example, facility location games (Goemans and Skutella 2000, Mallozzi 2011), inventory games (Anily and Haviv 2007, Chen 2009, Zhang 2009, Chen and Zhang 2016), and outsourcing games (Aydinliyim and Vairaktarakis 2010, Cai and Vairaktarakis 2012), to name a few. The set of such cost allocations is known as the *core* of a cooperative game (Shapley and Shubik 1969). If the core is not empty, the grand coalition's stability is guaranteed.

However, there are many situations where the core is empty, implying that the grand coalition is not stable; in the literature, these are known as *unbalanced cooperative games* (see Bondareva 1963, Shapley and Shubik 1969). In this paper, we study how a central authority, such as the government, can stabilize the grand coalition for an unbalanced cooperative game.

The central authority wishes to stabilize the grand coalition in two common situations. One is where the cooperation of the grand coalition helps the central authority minimize the total cost for all the players to complete their tasks. See, for example, facility location games (Goemans and Skutella 2000, Puerto et al. 2011), and bin packing games (Faigle and Kern 1993, Liu 2009). The other is where the total cost can be minimized only by partitioning the players into subcoalitions, but the grand coalition helps the central authority reduce certain negative social externalities, such as the number of machines used in machine scheduling games (Schulz and Uhan 2010, 2013), and the number of trucks used in travelling salesman games (Tamir 1989, Caprara and Letchford 2010, Kimms and Kozeletskyi 2016).

To stabilize the grand coalition, there are two known instruments that can be applied, i.e., *penalization* and *subsidization*. By penalization, the central authority can penalize players who leave the grand coalition. By subsidization, the central authority can subsidize players who stay in the grand coalition. However, applying either instrument alone has some drawbacks, as charging a penalty causes players to be dissatisfied, while providing a subsidy to the grand coalition injects external resources.

To ease these drawbacks of the two known aforementioned instruments, we propose and study a new instrument, i.e., *simultaneous penalization and subsidization*, which is based on a "stick-and-carrot" method that simultaneously charges penalties and provides subsidies. To illustrate application of this new instrument, let us consider its potential application to a water resource allocation problem in a region with a water shortage (e.g., see Fredericks et al. 1998, Van der Zaag et al. 2002, Sadegh and Kerachian 2011), where each water user owns a water supply source, but has to pay a shortage cost if the water consumption cannot be satisfied. The government, as a central authority, can pool all water resources together, and re-allocate them to different users to minimize the total shortage cost for the best social welfare of the entire region. However, it is possible that, no matter how the total cost is transferred, there is always a certain group of unsatisfied users who pay costs higher than they would have paid if they had stayed outside the grand coalition. In such a situation, the central government can implement a policy to charge a penalty to any group of users who are unwilling to cooperate. When the penalty is sufficiently large, no users will be better off by paying the penalty to leave the grand coalition, so that the grand coalition is enforced and stable with the best social welfare achieved. At the same time, though, a high penalty often causes such users to be dissatisfied. To avoid such dissatisfaction, the central government can lower the penalty, and simultaneously subsidize the grand coalition by injecting certain external resources, albeit at some additional cost. For instance, new water diversion projects can be constructed to bring in an external water supply from outside regions. When the external water supply is sufficient, the grand coalition for cooperation can then be stabilized, with all the water users satisfied.

The previous illustration indicates that the basic idea of our newly proposed instrument is to charge some penalty that may be insufficient to totally stabilize the grand coalition, but can nevertheless help to reduce the subsidy that needs to be provided. In other words, the penalty and subsidy become complementary. As a result, the subsidy can be reduced if the penalty increases, and vice versa, enabling the central authority to evaluate a whole spectrum of options. This is the motivation behind our study on the trade-off between penalty and subsidy.

Despite its practical relevance and theoretical interest, stabilizing the grand coalition of an unbalanced cooperative game has not been given adequate attention in the literature. Following the concept of core, which has been extensively studied (e.g., see Curiel 2013), two relaxed concepts, i.e., *the least core* and *the γ-core*, have been proposed for allocating costs to players in an unbalanced cooperative game to form a grand coalition. Under the concept of the least core (e.g., see Maschler et al. 1979), the cost allocated to each coalition must be no more than the minimum cost of the coalition plus the minimum value of z that allows the cost allocation to be stable. The minimum value of z is called *the least core value*. Under the concept of the γ-core (e.g., see Faigle and Kern 1993), the total cost allocated to all players is relaxed to be no smaller than γ times the cost of the grand coalition, where $0 < \gamma < 1$. Although not often explicitly mentioned, these concepts are relevant to the use of penalization and subsidization for stabilizing the grand coalition of an unbalanced cooperative game. For penalization, the central authority can charge a penalty to each coalition that wants to leave the grand coalition. The penalty can be set at the least core value. Alternatively, with a nonempty γ-core, the penalty for each coalition can be set at $(1/\gamma - 1)$ times its own cost. For subsidization, the central authority can provide a particular subsidy to all the players if they all choose to cooperate together. With a nonempty γ-core, the subsidy can be set at $(1-\gamma)$ times the cost of the grand coalition. Although there are other relaxed concepts, such as the concept of restricted coalition structures (Yi 1997, Demange 2004), that can be used to stabilize the grand coalition, we focus on those relevant to penalization and subsidization, using them to develop and study our new instrument.

The concept of γ-core has been extensively studied, but mainly for the design of cross-monotonic cost sharing methods, not for stabilizing the grand coalition (e.g., see Jain and Vazirani 2001, Könemann et al. 2005, Immorlica et al. 2008). Bachrach et al. (2009) were the first to formally propose the concept of *cost of stability*, i.e., the minimum external subsidy that can stabilize the grand coalition of an unbalanced cooperative game. In their work, although various bounds on the cost of stability were derived for several classes of unbalanced cooperative games, no general algorithms were provided to calculate the cost of stability. Following this work, Resnick et al. (2009) derived tight bounds on the cost of stability under various restrictions, and Meir et al. (2011) studied how to approximate the cost of stability for network flow games. Recently, Caprara and Letchford (2010) and Liu et al. (2016) developed various algorithms that can be applied to compute the cost of stability for unbalanced cooperative games.

Existing studies on the instrument of penalization are mainly based on the concept of the least core. For example, Faigle et al. (2001), and Kern and Paulusma (2003) studied how to compute the least core value for some special unbalanced cooperative games. Recently, Schulz and Uhan (2010, 2013) showed how to approximate the least core value for games with supermodular costs, particularly, for machine scheduling games.

As we have shown, most of the existing work on stabilizing the grand coalition uses the instrument of penalization or the instrument of subsidization. Although the newly proposed idea of simultaneously using both of these instruments is easy to understand, it is unclear how one can quantify the tradeoff between the levels of penalty and subsidy. This raises the following research question: What is the appropriate amount of penalty that a central authority needs to charge so that the required amount of external resources is affordable? As the first (to our knowledge) to address this question, our study makes the following contributions.

First, we introduce a penalty-subsidy function (PSF) to characterize the relationship between any given penalty and its corresponding minimum subsidy needed for stabilizing the grand coalition. We prove that the PSF is strictly decreasing, piecewise linear, and convex in the penalty, which reveals the diminishing effect of increasing the penalty to reduce the subsidy required to achieve the grand coalition's stability.

Second, we develop an algorithm to iteratively construct the exact PSF on its effective domain, with the number of iterations bounded by four times the number of breakpoints on the effective domain of the PSF. For a case where the PSF has an exponential number of breakpoints, we develop another algorithm to iteratively construct an ϵ-approximation of the PSF, with the number of iterations bounded by a polynomial function of the number of players, and with the cumulative error approaching zero as the parameter ϵ approaches zero.

Third, the two algorithms to construct the PSF and its approximation rely on solving the value of the PSF for any specific penalty, for which we derive its computational complexity and propose two effective solution approaches. The first approach follows a cutting plane method; the second is based on the theory of linear programming and its duality. Both approaches can be applied to a broad class of unbalanced cooperative games.

Fourth, we apply our new model, algorithms, and solution approaches to a class of parallel machine scheduling games. This not only demonstrates the wide applicability of our newly proposed instrument of simultaneous penalization and subsidization for stabilizing the grand coalition but also reveals some interesting properties of these games.

The paper is organized as follows. In Section 2 we introduce some preliminaries, and define the PSF to formulate the new instrument of simultaneous penalization and subsidization. In Section 3 we study its properties and present the construction algorithms for the PSF. In Section 4 we illustrate the two approaches for solving the value of the PSF for any given penalty. In Section 5 we demonstrate the applications of the proposed model, algorithms, and solution approaches. In Section 6 we conclude with a discussion of directions for future research. All proofs are provided in the electronic companion.

2. Formulation for Simultaneous Penalization and Subsidization

2.1. Preliminaries

A cooperative game with transferable utilities can be described by a pair (V, c), where $V = \{1, 2, \ldots, v\}$ denotes a set of v players with $v \geqslant 2$, and $c\colon 2^V \to \mathbb{R}$ denotes a characteristic function. A coalition is defined as a nonempty subset of players, and V is the grand coalition. Let $S = 2^V \backslash \{\varnothing\}$ denote the set of all coalitions. For each coalition $s \in S$, the characteristic function specifies a value $c(s)$ that indicates the minimum total cost for the members in s to accomplish their work when they cooperate. The game requires a cost allocation vector $\theta = [\theta_1, \theta_2, \ldots, \theta_v] \in \mathbb{R}^v$ with θ_k being the cost allocated to each player $k \in V$. By slightly abusing the notation for convenience, we use $\theta(s) = \sum_{k\in s} \theta_k$ to denote the total cost allocated to each coalition $s \in S$.

One of the most important concepts for a cooperative game (V, c) is the core, denoted by Core(V, c), which is defined as the set of cost allocation vectors $\theta \in \mathbb{R}^v$ that satisfy a budget balance constraint, i.e., $\theta(V) = c(V)$, as well as coalition stability constraints, i.e., $\theta(s) \leqslant c(s)$ for each $s \in S\backslash\{V\}$. In other words, we have

$$\mathrm{Core}(V,c) = \{\theta\colon \theta(V) = c(V),\ \theta(s) \leqslant c(s) \text{ for all } s \in S\backslash\{V\},\ \theta \in \mathbb{R}^v\}.$$

A cooperative game (V, c) is balanced if $\sum_{s\in S} \lambda_s c(s) \geqslant c(V)$ holds for every balanced collection of weights $(\lambda_s)_{s\in S}$ with $0 \leqslant \lambda_s \leqslant 1$ for $s \in S$ and $\sum_{s\in S: k\in s} \lambda_s = 1$ for $k \in V$ (Osborne and Rubinstein 1994). It is well known that Core(V, c) is not empty if and only if the cooperative game (V, c) is balanced (Bondareva 1963, Shapley and Shubik 1969). Thus, if (V, c) is balanced, there is no incentive for any coalition $s \in S\backslash\{V\}$ to deviate from the grand coalition V.

However, as mentioned earlier, many cooperative games are unbalanced. To stabilize the grand coalition for an unbalanced cooperative game (V, c), a central authority can use two instruments known in the literature. One is penalization, by which the central authority charges a penalty z to any coalition that wants to leave the grand coalition. Because a high penalty often causes high player dissatisfaction, the central authority needs to find the minimum penalty z^*, along with a cost allocation $\beta^* \in \mathbb{R}^v$, such that, for any coalition $s \in S\backslash\{V\}$, the assigned cost $\beta^*(s)$ is no larger than its own cost $c(s)$ plus the penalty z^*. This can be formulated as the following linear program (LP):

$$z^* = \min_{\beta, z}\{z\colon \beta(V) = c(V),\ \beta(s) \leqslant c(s) + z \text{ for all } s \in S\backslash\{V\},\ z \in \mathbb{R},\ \beta \in \mathbb{R}^v\}. \quad (1)$$

The minimum penalty z^* is the least core value, and the optimal solution, denoted by β^*, is called the least core cost allocation (Maschler et al. 1979). It can be seen that setting $z = c(V)$ and $\beta_k = c(V)/v$ for all $k \in V$ forms a feasible solution to LP (1), which implies that $z^* \leqslant c(V)$, and so z^* is bounded from above by $c(V)$.

The other common instrument is subsidization, by which the central authority provides a certain subsidy to all the players in V if they choose to cooperate as a grand coalition, so that the actual total cost shared among the players can be less than $c(V)$. The central authority is committed to finding the minimum subsidy ω^*, along with a cost allocation $\alpha^* \in \mathbb{R}^v$, that satisfies the coalition stability constraints. This can also be formulated as the following LP:

$$\omega^* = \min_{\alpha}\{c(V) - \alpha(V)\colon \alpha(s) \leqslant c(s) \text{ for all } s \in S,\ \alpha \in \mathbb{R}^v\}. \tag{2}$$

The optimal objective value ω^* is called the minimum subsidy, or the cost of stability as defined by Bachrach et al. (2009). The optimal solution, denoted by $\alpha^* \in \mathbb{R}^v$, is called the optimal cost allocation. Note that ω^* is non-negative, and it is positive if and only if game (V, c) is unbalanced (i.e., it has an empty core).

By definition it can be seen that the LP (2) for the instrument of subsidization is equivalent to the following optimal cost allocation problem introduced in Caprara and Letchford (2010),

$$\max_{\alpha}\{\alpha(V)\colon \alpha(s) \leqslant c(s) \text{ for all } s \in S,\ \alpha \in \mathbb{R}^v\}, \tag{3}$$

as well as being equivalent to the following γ-core problem (see Jain and Mahdian 2007),

$$\gamma^* = \max_{\alpha,\gamma}\{\gamma\colon \alpha(V) = \gamma c(V),\ \alpha(s) \leqslant c(s) \text{ for all } s \in S,\ \alpha \in \mathbb{R}^v,\ \gamma \in \mathbb{R}\}. \tag{4}$$

This is because every optimal cost allocation α^* of (2) is also optimal to (3) and to (4).

2.2. Penalty-Subsidy Function

To formulate the new instrument of simultaneous penalization and subsidization, we now state our definition of the penalty-subsidy function, z-penalized optimal cost allocation, and z-penalized minimum subsidy for a cooperative game.

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

$$\omega(z) = \min_{\beta}\{c(V) - \beta(V)\colon \beta(s) \leqslant c(s) + z \text{ for all } s \in S\backslash\{V\},\ \beta \in \mathbb{R}^v\}. \tag{5}$$

Its optimal solution, denoted by $\beta(\cdot, z)$, is called a *z-penalized optimal cost allocation*, and its optimal objective value $\omega(z)$ is called the *z-penalized minimum subsidy*. In addition, $\omega(z)$ as a function of z is referred to as the *penalty-subsidy function* (*PSF*).

From Definition 1 it can be seen that by capturing the trade-off between penalty and subsidy, the PSF $\omega(z)$ formulates the concept of simultaneous penalization and subsidization for stabilizing the grand coalition. For any penalty z that is not sufficient to prevent players from deviating from the grand coalition, the central authority needs to provide a subsidy of at least $\omega(z)$ to make the grand coalition cooperate. Under the joint effect of penalty z and subsidy $\omega(z)$, no player or coalition of players can be better off by deviating from the grand coalition, and hence the grand coalition is stabilized.

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 $z \in (0, z^*)$.*

Lemma 1 reveals the monotonicity of the new instrument, where z^*, as defined in (1), is the minimum penalty needed to stabilize the grand coalition by penalization alone, and ω^*, as defined in (2), is the minimum subsidy needed to stabilize the grand coalition by subsidization alone. It implies that the instrument of penalization on its own and the instrument of subsidization on its own are two extreme cases of the new instrument, and that a central authority can evaluate a spectrum of options provided by penalty-subsidy pairs $(z, \omega(z))$ for $z \in [0, z^*]$.

In this paper, we restrict the penalty value z in an effective domain $[0, z^*]$ of $\omega(z)$, so that the penalty and subsidy are non-negative. In fact, the main results derived in this paper, such as the algorithms in Section 3 to construct function $\omega(z)$, and the solution approaches in Section 4 to compute the value of $\omega(z)$ for any given z, can be directly applied to other cases where $z < 0$ or $z > z^*$. Although these cases are outside the scope of this paper, they may have some meaningful implications. For example, if $z > z^*$, then $\omega(z) < 0$, implying that the central authority is charging a high penalty so as to extract some profit from the grand coalition.

We now illustrate the instruments of penalization, subsidization, and simultaneous penalization and subsidization, by using the following game instance of single machine scheduling with weighted jobs (SMW) introduced by Schulz and Uhan (2010, 2013).

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.

For the grand coalition in Example 1, its optimal job processing sequence is $1 \to 2 \to 3 \to 4$ with a minimum total weighted completion time of 115. However,

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

the coalition stability constraints in (2) imply that the total cost that can be shared among the players cannot exceed $\sum_{k\in V} w_k t_k = 60 < 115$. Thus, the game is unbalanced. To stabilize the grand coalition by penalization, we can solve LP (1) to obtain the minimum penalty $z^* = 19.5$. To stabilize the grand coalition by subsidization, we can solve LP (2) to obtain the minimum subsidy $\omega^* = 55$.

To demonstrate our new instrument of simultaneous penalization and subsidization, we set the penalty z at some discrete values, and then compute the corresponding z-penalized minimum subsidy $\omega(z)$ and z-penalized optimal cost allocations $\beta(\cdot,z)$ by solving LP (5). The results, shown in Table 1, indicate that the PSF $\omega(z)$ and the rate of decrease are decreasing in z. Later, in Figure 2 of Section 3.2.1, we fully characterize the PSF $\omega(z)$ for z in the effective domain $[0, 19.5]$.

3. Analyses of Simultaneous Penalization and Subsidization

In this section, we first derive some structural properties of the PSF $\omega(z)$, which helps us to understand the trade-off between penalty and subsidy for an unbalanced cooperative game. Based on these properties, we then analyze how to construct the function $\omega(z)$ and its approximation on the effective domain $[0, z^*]$.

3.1. Structural Properties

We begin by exploring the properties of the new instrument from the players' perspective. For any penalty z, consider any z-penalized optimal cost allocation $\beta(\cdot,z)$ determined by LP (5). It can be seen that there must exist some coalitions $s\in S\backslash\{V\}$ of players who have to overpay for their deviation from the grand coalition with $c(s)\leqslant \beta(s,z)\leqslant c(s)+z$. In particular, for any coalitions s with $\beta(s,z)=c(s)+z$, their overpaid amount is the highest, which, to a certain extent, indicates that they are the most unsatisfied coalitions. We define such coalitions as *maximally unsatisfied coalitions*. In Example 1, for $z=5$, players in $\{1,4\}$ form a maximally unsatisfied coalition, since $\beta(1,5)+\beta(4,5)=25+13=38=w_1t_1+w_4(t_1+t_4)+z=c(\{1,4\})+z$.

Let $S^{\beta z}=\{s_1^{\beta z}, s_2^{\beta z},\ldots, s_{h(\beta,z)}^{\beta z}\}$ denote the collection of all maximally unsatisfied coalitions, where $h(\beta,z)=|S^{\beta z}|$. Taking the dual of LP (5), by strong duality we have:

$$\omega(z)=\max_{\rho}\Big\{c(V)+\sum_{s\in S\backslash\{V\}}-\rho_s[c(s)+z]:\ \sum_{s\in S\backslash\{V\}:k\in s}\rho_s=1,\ \forall k\in V,\ \rho_s\geqslant 0,\ \forall s\in S\backslash\{V\}\Big\}. \quad (6)$$

From this, we can establish Theorem 1, showing that the union of coalitions in $S^{\beta z}$ has a complete coverage of players.

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

$$s_1^{\beta z}\cup s_2^{\beta z}\cup\cdots\cup s_{h(\beta,z)}^{\beta z}=V. \quad (7)$$

By Theorem 1 we know that every player $k\in V$, regardless of its specific role in the game, must appear in at least one of the maximally unsatisfied coalitions, i.e., the coalitions of players who overpaid the most. This suggests a sense of fairness in the cost allocation to all players under $\beta(\cdot,z)$. In addition, by Theorem 1 we can develop bounds on the derivatives of points on the PSF curve, which will be shown later in this section.

Next, by examining the PSF $\omega(z)$, we explore some properties of the new instrument from the perspective of the central authority, so as to gain a greater understanding of the trade-off between penalty and subsidy. The results are presented in Theorems 2 and 3.

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

The properties of the PSF $\omega(z)$ shown in Theorem 2 have the following implications: First, the strictly decreasing property of $\omega(z)$ implies a strong complementarity between the penalty z and the corresponding minimum subsidy $\omega(z)$ desired; when z increases $\omega(z)$ strictly decreases. Second, the piecewise linearity of $\omega(z)$ implies that the derivative at point $(z,\omega(z))$ only changes a finite number of times when z increases from 0 to z^*. This allows us to fully characterize the PSF by evaluating $\omega(z)$ at only a finite number of values of z. Third, the convexity of $\omega(z)$ reveals a diminishing effect of increasing the penalty to reduce the minimum subsidy desired. When no penalty is charged, the central authority needs to provide the highest subsidy to stabilize the grand coalition. As the penalty increases, the minimum subsidy desired is reduced. However, as each additional unit of the penalty is charged, the reduction in the minimum subsidy desired decreases. Because penalization is at the cost of dissatisfaction of the players, Theorem 2 sheds light on how to make the best use of the penalty.

To further understand the trade-off between penalty and subsidy, we now study the derivatives of each linear segment of the PSF $\omega(z)$. Theorem 3 shows that the derivatives of $\omega(z)$ may have large variations, depending on the number of players v, which implies the challenges involved in developing efficient algorithms for the construction of $\omega(z)$ and the importance of using the properties of $\omega(z)$ in the construction.

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

For any given penalty z, consider the derivative $\omega'(z)$ of the PSF $\omega(z)$. From LP (6), we know that $\omega(z)$ is the point-wise maximum of a set of straight lines whose slopes are given in the set $\{\sum_{s\in S\backslash\{V\}}(-\rho_s)\colon \sum_{s\in S\backslash\{V\}:k\in s}\rho_s = 1, \forall\, k\in V;\ \rho_s \geqslant 0, \forall\, s\in S\backslash\{V\}\}$. Therefore, at any point $(z,\omega(z))$ on the PSF curve, the left and right derivatives, K_l^z and K_r^z, can be obtained by computing the respective minimum and maximum slopes of the corresponding straight lines that pass through point $(z,\omega(z))$. To formalize this, let Π^z denote the set of all optimal solutions ρ to LP (6), so that every value in $\{\sum_{s\in S\backslash\{V\}}(-\rho_s)\colon \rho\in\Pi^z\}$ corresponds to the slope of a straight line that passes through point $(z,\omega(z))$. Thus, we obtain that

$$K_l^z = \min\left\{\sum_{s\in S\backslash\{V\}}(-\rho_s)\colon \rho\in\Pi^z\right\} \quad \text{and} \quad K_r^z = \max\left\{\sum_{s\in S\backslash\{V\}}(-\rho_s)\colon \rho\in\Pi^z\right\}. \tag{8}$$

Moreover, it can be seen that if and only if $K_l^z \neq K_r^z$, point $(z,\omega(z))$ is a breakpoint on the PSF curve, i.e., a point that connects two adjacent linear segments of $\omega(z)$.

However, computing the left and right derivatives by (8) can be very difficult since it requires obtaining the set Π^z of all optimal solutions ρ to LP (6). To avoid such difficulty, in Section 3.2 we use a weak left derivative $K_{l'}^z$ and a weak right derivative $K_{r'}^z$ in the construction of $\omega(z)$. These are defined as follows:

Definition 2. We refer to $(K_{l'}^z, K_{r'}^z)$ as a pair of weak derivatives at point $(z,\omega(z))$ on the curve of the PSF $\omega(z)$ if and only if $K_l^z \leqslant K_{l'}^z \leqslant K_{r'}^z \leqslant K_r^z$. In addition, $K_{l'}^z$ is called a weak left derivative, and $K_{r'}^z$ is called a weak right derivative.

Definition 2 implies that if $(z,\omega(z))$ is not a breakpoint on the curve of $\omega(z)$, there exists a unique pair of weak derivatives $(K_{l'}^z, K_{r'}^z)$ that satisfies $K_{l'}^z = K_l^z = K_{r'}^z = K_r^z$. Thus, $(z,\omega(z))$ is a breakpoint on the curve of $\omega(z)$ if and only if there exists a pair of weak derivatives $(K_{l'}^z, K_{r'}^z)$ with $K_{l'}^z \neq K_{r'}^z$.

Compared with K_l^z and K_r^z, the computation of weak derivatives $K_{l'}^z$ and $K_{r'}^z$ is more tractable. For example, we can first compute a collection $S^{\beta z}$ of all the maximally unsatisfied coalitions under any $\beta(\cdot, z)$ optimal to LP (5). Define $\Pi^{\beta z} = \{\rho\colon \sum_{s\in S\backslash\{V\}:k\in s}\rho_s = 1$ for all $k\in V$, $\rho_s = 0$ for all $s\notin S^{\beta z}$, and $\rho_s \geqslant 0$ for all $s\in S^{\beta z}\}$. By the complementary slackness conditions, we obtain that every $\rho\in\Pi^{\beta z}$ is an optimal solution to LP (6). Thus, it can be verified that $K_{l'}^{\beta z}$ and $K_{r'}^{\beta z}$, as defined by

$$K_{l'}^{\beta z} = \min\left\{\sum_{s\in S\backslash\{V\}}-\rho_s\colon \rho\in\Pi^{\beta z}\right\} \quad \text{and} \quad K_{r'}^{\beta z} = \max\left\{\sum_{s\in S\backslash\{V\}}-\rho_s\colon \rho\in\Pi^{\beta z}\right\}, \tag{9}$$

are weak left and weak right derivatives, respectively, at point $(z,\omega(z))$ because $\Pi^{\beta z}$ is only a subset of the complete set Π^z of optimal solutions to LP (6).

There are also other approaches to obtaining weak derivatives, such as those based on the computation of the z-penalized minimum subsidy, which we explain in Section 4.

3.2. Construction of the PSF $\omega(z)$ and Its Approximation

3.2.1. Construction of the Exact PSF. According to Theorem 2, the PSF $\omega(z)$ is piecewise linear on its effective domain $[0,z^*]$. Thus, to construct $\omega(z)$ we only need to construct a set P^* of values from $[0,z^*]$ that cover all the breakpoints of $\omega(z)$, and then connect points $(z,\omega(z))$ for all $z\in P^*$.

Following the previous idea, we develop an Intersection Points Computation (IPC) algorithm to construct the PSF $\omega(z)$. It iteratively updates P^* by adding new values from $[0,z^*]$. The update of P^* is done along with an update of $\mathbb{P}$, which is a set of intervals that may contain new breakpoints of $\omega(z)$ not yet covered by P^*. Initially, the algorithm sets $P^* = \{0,z^*\}$, and $\mathbb{P} = \{[0,z^*]\}$, and then it iteratively updates P^* and $\mathbb{P}$ until $\mathbb{P}$ is empty.

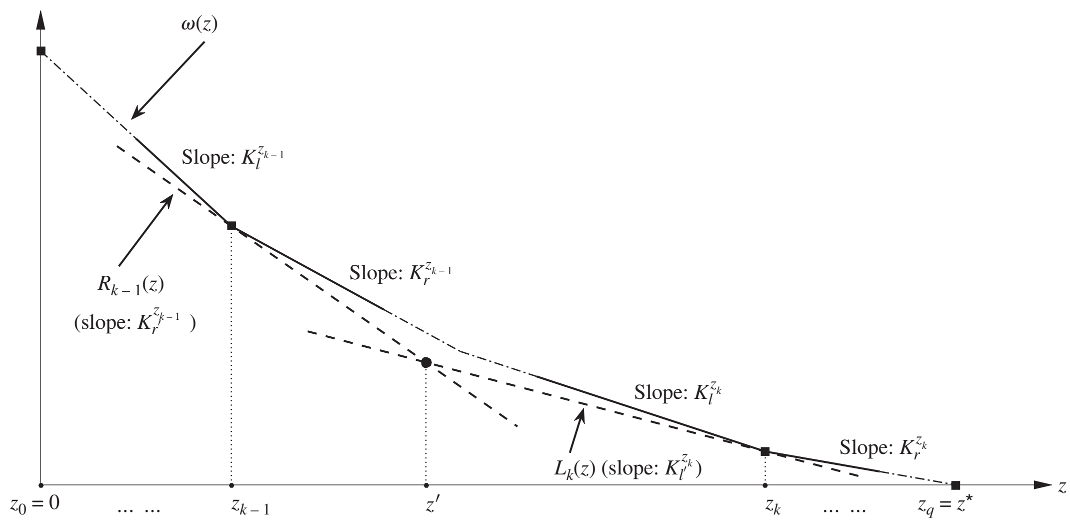
In each iteration, the IPC algorithm attempts to find for P^* a new value z' from an interval in $\mathbb{P}$ by computing an intersection point of two constructed linear functions. More specifically, as illustrated in Figure 1, it first relabels values in P^* by $z_0 < z_1 < \cdots < z_q$, where $z_0 = 0$, $z_q = z^*$ and $q = |P^*| - 1$, and then selects any interval from $\mathbb{P}$, denoted by $[z_{k-1}, z_k]$ with $1\leqslant k\leqslant q$. To examine whether $[z_{k-1}, z_k]$ contains any new breakpoint of $\omega(z)$, it constructs two linear functions, denoted by $R_{k-1}(z)$ 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. By the definition of left and right weak derivatives, and due to the convexity of $\omega(z)$, we have that

$$R_{k-1}(z)\leqslant\omega(z) \text{ and } L_k(z)\leqslant\omega(z), \quad \text{for each } z\in[0,z^*]. \tag{10}$$

With $R_{k-1}(z)$ and $L_k(z)$ the IPC algorithm then examines 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 $R_{k-1}(z)$ or $L_k(z)$ passes points $(z_{k-1},\omega(z_{k-1}))$ and $(z_k,\omega(z_k))$ of $\omega(z)$. Thus, by (10) and

Figure 1. Illustration of the Construction of the PSF $\omega(z)$ by the IPC Algorithm



the convexity of $\omega(z)$ we obtain that $\omega(z) = R_{k-1}(z)$ for $z \in [z_{k-1}, z_k]$ or $\omega(z) = L_k(z)$ for $z \in [z_{k-1}, z_k]$, thus implying that $\omega(z)$ has no breakpoint in (z_{k-1}, z_k). Therefore, for this case, no update of P^* is required, and the interval $[z_{k-1}, z_k]$ needs to be removed from $\mathbb{P}$.

Case 2. If neither $R_{k-1}(z)$ passes $(z_k, \omega(z_k))$, nor $L_k(z)$ passes $(z_{k-1}, \omega(z_{k-1}))$, then since $R_{k-1}(z)$ passes $(z_{k-1}, \omega(z_{k-1}))$ and $L_k(z)$ passes $(z_k, \omega(z_k))$, by (10) and the convexity of $\omega(z)$ we obtain that $R_{k-1}(z)$ and $L_k(z)$ must have a unique intersection point at $z = z'$ for some $z' \in (z_{k-1}, z_k)$ (see Figure 1). This implies that z' may be a breakpoint of $\omega(z)$. Therefore, for this case, we update P^* by adding the new value z', and update $\mathbb{P}$ by removing $[z_{k-1}, z_k]$ and adding two new intervals $[z_l, z']$ and $[z', z_r]$.

Finally, when $\mathbb{P}$ is empty, implying that P^* has covered all breakpoints of $\omega(z)$, the iteration stops. A linear piecewise function is obtained by connecting points $(z, \omega(z))$ for all $z \in P^*$.

We summarize the IPC algorithm in Algorithm 1, and establish Theorem 4 to show the effectiveness and efficiency of the algorithm, which indicates that the function obtained equals the PSF $\omega(z)$ for $z \in [0, z^*]$, and that the number of iterations is less than four times the number of breakpoints of $\omega(z)$.

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 $1 \leqslant k \leqslant q$.

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.

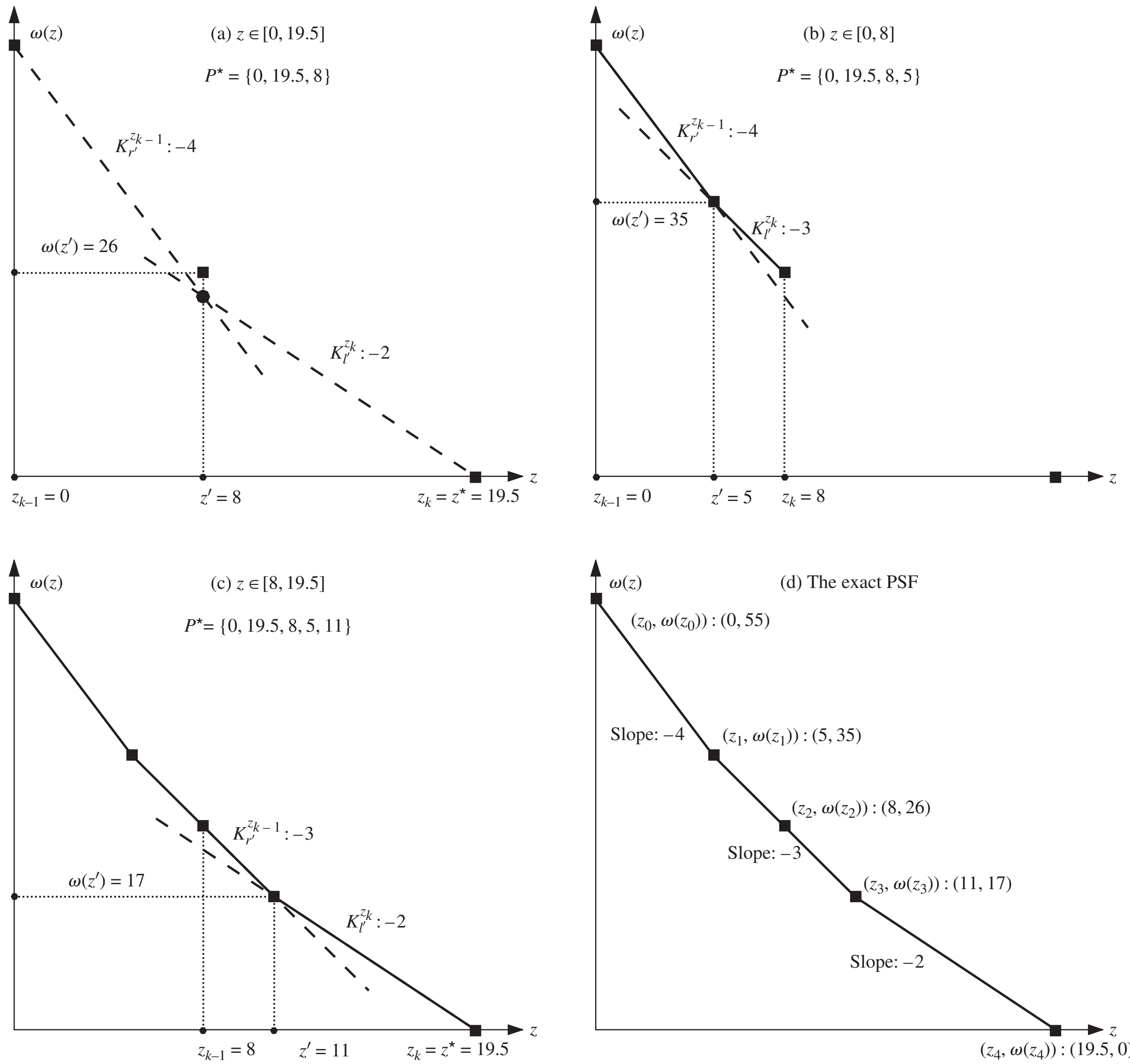
Step 3. Return a piecewise linear function by connecting points $(z, \omega(z))$ for all $z \in P^*$.

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

Figure 2 illustrates how the IPC algorithm iteratively constructs the PSF $\omega(z)$ for the instance in Example 1. The algorithm updates set P three times by adding new breakpoints z' equal to 8, 5, and 11, respectively, as shown in Figure 2(a)–(c). Accordingly, set $\mathbb{P}$ is updated from $\{[0, 19.5]\}$, to $\{[0, 8], [8, 19.5]\}$, to $\{[8, 19.5]\}$, and then to an empty set (when the algorithm stops). Figure 2(d) shows the curve of the final function obtained for $\omega(z)$, on which there are four breakpoints, i.e., $(0, 55)$, $(5, 35)$, $(11, 17)$, and $(19.5, 0)$.

In Figure 2(d), the PSF $\omega(z)$ is strictly decreasing, piecewise linear, and convex in z, as stated in Theorem 2. The slopes of its linear segments, from left to right, are -4, -3 and -2, respectively, which are all in the interval $[-4, -4/3]$, as stated in Theorem 3. Moreover, to illustrate the computation of weak derivatives,

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



consider the case of $z = 5$ as an example. By solving (5), we obtain a z-penalized optimal cost allocation $[25, 23, 19, 13]$ with the corresponding collection of maximally unsatisfied coalitions being $\{\{1\}, \{2\}, \{3\}, \{4\}, \{1, 4\}\}$. According to LP (9) and the definition of $\Pi^{\beta z}$ in Section 3.1, we can obtain a pair of weak derivatives at point $z = 5$, denoted by $(K_{l'}^5, K_{r'}^5)$ with $K_{l'}^5 = -4$ and $K_{r'}^5 = -3$, which, in fact, equal the actual derivatives (see Figure 2(d)). Since $K_{l'}^5 \neq K_{r'}^5$, we know that $(z_1, \omega(z_1)) = (5, 35)$ is a breakpoint of $\omega(z)$.

Remark 1. When the PSF $\omega(z)$ has a large number of breakpoints, it is time consuming for the IPC algorithm to construct function $\omega(z)$ exactly. In this situation, we can force the algorithm to stop after a number of iterations in step 2 (even when $\mathbb{P}$ is not empty), and then use values currently in P^*, denoted by $0 = z_0 < z_1 < \cdots < z_q = z^*$, to construct an upper bound function $\mathrm{UB}(z)$ and a lower bound function $\mathrm{LB}(z)$ of $\omega(z)$ for $z \in [0, z^*]$:

- To construct $\mathrm{UB}(z)$, we simply connect points $(z, \omega(z))$ for all $z \in P^*$ to obtain a piecewise linear function. By the convexity of $\omega(z)$ we obtain that for each $k \in \{1, 2, \ldots, q\}$, $\mathrm{UB}(z) \geqslant \omega(z)$ for $z \in [z_{k-1}, z_k]$, implying that $\mathrm{UB}(z) \geqslant \omega(z)$ for $z \in [0, z^*]$. Thus, $\mathrm{UB}(z)$ is an upper bound function of $\omega(z)$.
- To construct $\mathrm{LB}(z)$, we need to use the linear functions $R_{k-1}(z)$ and $L_k(z)$ defined in step 2.3 of the IPC algorithm for $1 \leqslant k \leqslant q$. By (10), we have that $R_{k-1}(z) \leqslant \omega(z)$ and $L_k(z) \leqslant \omega(z)$ for $z \in [0, z^*]$ and $1 \leqslant k \leqslant q$. Define $\mathrm{LB}(z) = \max\{R_0(z), L_1(z), R_1(z), L_2(z), \ldots, R_{q-1}(z), L_q(z)\}$. We obtain that $\mathrm{LB}(z) \leqslant \omega(z)$ for $z \in [0, z^*]$. Thus, $\mathrm{LB}(z)$ is a lower bound function of $\omega(z)$.

3.2.2. ϵ-Approximation of the PSF. Remark 1 shows that, by forcing it to stop after a number of iterations in step 2, the IPC algorithm can be modified to obtain upper and lower bound functions, UB(z) and LB(z), as approximations of the PSF $\omega(z)$. However, such approximations may significantly deviate from $\omega(z)$, especially if $\mathbb{P}$ still contains large intervals when the IPC algorithm is forced to stop.

Next, we present an efficient algorithm to construct an upper bound function as an ϵ-approximation of function $\omega(z)$. It converges to function $\omega(z)$ when the parameter ϵ approaches zero. Under the joint effect of any penalty-subsidy pair on the curve of this upper bound function, the grand coalition is stabilized.

Algorithm 2 (Approximation algorithm to construct an ϵ-approximation of the PSF)

Step 1. Divide $[0, z^*]$ into $\lceil 2v/\epsilon \rceil$ subintervals denoted by $[z_0, z_1)$, $[z_1, z_2), \ldots,$ $[z_{\lceil v/\epsilon\rceil-2}, z_{\lceil 2v/\epsilon\rceil-1})$, 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 $0 \leqslant k \leqslant \lceil 2v/\epsilon\rceil$, 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 $\{(z_0, \omega(z_0)), (z_1, \omega(z_1)), \ldots, (z_{\lceil 2v/\epsilon\rceil-1}, \omega(z_{\lceil 2v/\epsilon\rceil-1})), (z_{\lceil 2v/\epsilon\rceil}, \omega(z_{\lceil 2v/\epsilon\rceil}))\}$.

To construct an ϵ-approximation of function $\omega(z)$, we propose an approximation algorithm in Algorithm 2 that connects points $(z, \omega(z))$ for only $(\lceil 2v/\epsilon\rceil + 1)$ different values of z in $[0, z^*]$. Following an argument similar to that for UB(z) in Remark 1, by the convexity of $\omega(z)$ we can obtain that $U_\epsilon(z)$ returned by Algorithm 2 is an upper bound function of $\omega(z)$.

To show the effectiveness of Algorithm 2, we now evaluate the cumulative error E_c and the maximum error $E_{\max}$ between functions $U_\epsilon(z)$ and $\omega(z)$, where $E_c = \int_0^{z^*} |U_\epsilon(z) - \omega(z)|\,dz$, and $E_{\max} = \max\{|U_\epsilon(z) - \omega(z)|: z \in [0, z^*]\}$. As shown in Section 2.1, z^* is bounded by $c(V)$. Theorem 5 shows that when ϵ approaches zero, the cumulative error E_c and the maximum error $E_{\max}$ also approach zero, implying that $U_\epsilon(z)$ converges to function $\omega(z)$. It also shows that the relative cumulative error, $E_c/\int_0^{z^*}\omega(z)\,dz$, is bounded by ϵ. Thus, $U_\epsilon(z)$ is an ϵ-approximation of function $\omega(z)$.

Theorem 5. *For the two types of errors, we have that* $E_c \leqslant (\epsilon/2)(z^*)^2 \leqslant \epsilon \int_0^{z^*}\omega(z)\,dz$, *and* $E_{\max} \leqslant (\epsilon z^*)/2$, *for any given* $\epsilon > 0$.

4. Solution Approaches to Computing the z-Penalized Minimum Subsidy

In this section, we present two solution approaches to computing the value of $\omega(z)$ for any given z, which can be further used to compute weak derivatives of $\omega(z)$. With these, we apply the algorithms proposed in Section 3.2 to construct the PSF $\omega(z)$ and its approximation.

Our solution approaches are applicable to a broad class of cooperative games, i.e., integer minimization (IM) games. As introduced by Caprara and Letchford (2010), the class of IM games includes many well known unbalanced cooperative games, such as machine scheduling games, facility location games, travelling salesman games, etc.

Definition 3. A cooperative game (V, c) is an IM game if there exist (i) positive integers e and t; (ii) a left hand side (LHS) matrix $A \in \mathbb{Z}^{e\times t}$; (iii) a right hand side (RHS) matrix $B \in \mathbb{Z}^{e\times v}$; (iv) an RHS vector $D \in \mathbb{Z}^e$; (v) an objective function vector $c \in \mathbb{Z}^t$; and (vi) for each coalition $s \in S$, an incidence vector $y^s \in \{0,1\}^v$, with $y_k^s = 1$ if $k \in s$, and with $y_k^s = 0$ otherwise, for all $k \in V$, such that the coalition cost $c(s)$ equals the optimal objective value of the following integer linear program:

$$c(s) = \min_x \{cx: Ax \geqslant By^s + D,\ x \in \mathbb{Z}^t\}. \qquad (11)$$

As we show in Section EC.2 of the electronic companion, computing $\omega(z)$ for any z and for any IM game is NP-hard, even for some special cases where $c(s)$ for each $s \in S$ is polynomially solvable. Therefore, it is of interest for us to develop solution approaches that can efficiently compute the exact value of $\omega(z)$ for some special IM games, or produce bounds on the value of $\omega(z)$ for any IM game in general.

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

$$\pi(z) = \max_\beta \{\beta(V): \beta(s) \leqslant c(s) + z \text{ for all } s \in S\backslash\{V\},\ \beta \in \mathbb{R}^v\}. \qquad (12)$$

From (5) it can be seen that $\omega(z) = c(V) - \pi(z)$. Thus, instead of directly computing $\omega(z)$, we can compute the value of $\pi(z)$ first, and obtain $\omega(z)$ by $\omega(z) = c(V) - \pi(z)$. When $c(V)$ or $\pi(z)$ is hard to obtain, we can compute their bounds to obtain a bound on $\omega(z)$.

The value of $\pi(z)$ can be interpreted as the maximum total cost that can be shared stably by the grand coalition V, given that the penalty for deviation by any subcoalition is z. Thus, we refer to LP (12) as the z-penalized optimal cost allocation problem. Although important, this problem is rarely studied in the literature. Only its special case $\pi(0)$, where penalty z is zero, has recently been studied by Caprara and Letchford (2010) and Liu et al. (2016).

Following the previous discussion, we develop two solution approaches to compute the exact value of $\omega(z)$ for IM games by solving $\pi(z)$ of LP (12). The first approach, presented in Section 4.1, is a cutting plane method. The second approach, presented in Section 4.2, is based on the theory of linear programming and its duality. Both approaches can also be adopted to efficiently derive lower or upper bounds on the value of $\omega(z)$ for all IM games.

4.1. Cutting Plane Approach

For any IM game (V, c) and any z to obtain the value of $\pi(z)$, we can solve LP (12), which contains an exponential number of constraints. Thus, a natural way to solve it is to follow a cutting plane (CP) approach (see Bertsimas and Tsitsiklis 1997). As described in Algorithm 3, the CP approach starts with a restricted coalition set $S' \subseteq (S\backslash\{V\})$, and finds an optimal solution $\bar{\beta}(\cdot, z)$ to a relaxation of LP (12) with only constraints $\beta(s, z) \leqslant c(s)+z$ for $s \in S'$ included. It then checks whether $\bar{\beta}(\cdot, z)$ violates any constraints $\beta(s, z) \leqslant c(s)+z$ not included, with $s \in S\backslash\{V\}\backslash S'$. For this, it needs to find an optimal solution s^* to a separation problem $\delta = \min\{c(s) + z - \bar{\beta}(s, z) \colon \forall\, s \in S\backslash\{V\}\}$. If $\delta < 0$, then $\bar{\beta}(\cdot, z)$ violates the constraint $\beta(s^*, z) \leqslant c(s^*) + z$, and we add s^* to S'. We then solve the relaxation of LP (12) again, with constraints based on the new S'; otherwise, we know that $\bar{\beta}(\cdot, z)$ is also an optimal solution to LP (12), implying that $\omega(z) = c(V) - \bar{\beta}(V, z)$, and that a pair of weak derivatives $(K_{l'}^{\bar{\beta}z}, K_{r'}^{\bar{\beta}z})$ can be computed by solving (9) with $\Pi^{\beta z}$ replaced by $\Pi^{\bar{\beta}z}$.

Algorithm 3 (Cutting Plane (CP) approach to computing $\omega(z)$ for a given z)

Step 1. Let $S' \subseteq S\backslash\{V\}$ 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 $\max_\beta\{\beta(V, z) \colon \beta(s, z) \leqslant c(s) + z$, for all $s \in S', \beta \in \mathbb{R}^v\}$.

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

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 CP approach is how to efficiently solve the separation problem in step 3 to find a violated constraint $\beta(s^*, z) \leqslant c(s^*) + z$, and this depends on the specific game being studied. In fact, if we can separate the constraints $\beta(s, z) \leqslant c(s) + z$ for all $s \in S\backslash\{V\}$ in (pseudo-)polynomial time, then by the equivalence between optimization and separation we can solve LP (12) to obtain $\pi(z)$ in (pseudo-)polynomial time by the well known ellipsoid method, which follows a more complicated cutting plane approach (see Grötschel et al. 2012).

When the separation problem is hard to solve, we can compute a lower bound $\omega^l(z)$ for $\omega(z)$ by revising the CP approach as follows: In step 3, we solve the separation problem simply by using a heuristic method, which implies that when the CP approach stops, the obtained $\bar{\beta}(V, z)$ may be greater than $\pi(z)$, and thus the returned value $c(V) - \bar{\beta}(V, z)$, denoted by $\omega^l(z)$, is a lower bound of $\omega(z)$. Moreover, if the coalition problem $c(s)$ is also hard to solve, we can further replace $c(s)$ with its upper bound $c^u(s)$ in step 2 to compute $\bar{\beta}(\cdot, z)$. Using any lower bound $c^l(V)$ of $c(V)$, we can compute $\omega^l(z) = c^l(V) - \bar{\beta}(V, z)$, which is also a lower bound of $\omega(z)$. In this situation, since the exact value of $\omega(z)$ is not known, we cannot compute weak derivatives.

4.2. Linear Programming Approach

For any IM game (V, c) and any z to obtain the value of $\pi(z)$, we can follow another solution approach, which is inspired by the approach that Caprara and Letchford (2010) proposed to compute $\pi(z)$ with $z = 0$. We refer to it as the LP approach since it is based on the theory of linear programming and its duality.

Let Q^{xy} denote the overall set of feasible solutions to (11) of $c(s)$ for all $s \in S\backslash\{V\}$:

$$Q^{xy} = \{(x, y) \colon Ax \geqslant By + D,\ y = y^s \text{ for some } s \in S\backslash\{V\},\ x \in \mathbb{Z}^t,\ y \in \{0,1\}^v\}, \quad (13)$$

and extend Q^{xy} to $Q^{x\mu y}$ as follows by introducing a new decision variable μ, but fixing $\mu = 1$:

$$Q^{x\mu y} = \{(x, \mu, y) \colon Ax \geqslant By + D\mu,\ y = y^s \text{ for some } s \in S\backslash\{V\},\ \mu = 1,\ x \in \mathbb{Z}^t,\ y \in \{0,1\}^v\}. \quad (14)$$

Let cone $Q^{x\mu y}$ represent the conic hull of $Q^{x\mu y}$. By intersecting cone $Q^{x\mu y}$ with $\{(x, \mu, y) \in \mathbb{R}^{t+1+v} \colon y = \mathbf{1}\}$, and projecting the intersection onto (x, μ)-space, we define $C^{x\mu}$ as follows:

$$C^{x\mu} = \text{proj}_{x\mu}(\{(x, \mu, y) \in \mathbb{R}^{t+1+v} \colon y = \mathbf{1}\} \cap \text{cone } Q^{x\mu y}). \quad (15)$$

To this end, we can establish Lemma 2, showing that $\pi(z) = \min\{cx + z\mu \colon (x, \mu) \in C^{x\mu}\}$.

Lemma 2. *The optimal objective value* $\pi(z)$ *of LP* (12) *equals* $\min\{cx + z\mu \colon (x, \mu) \in C^{x\mu}\}$.

Although by Lemma 2 we can obtain $\omega(z) = c(V) - \pi(z) = c(V) - \min\{cx + z\mu \colon (x, \mu) \in C^{x\mu}\}$, it is not easy to solve $\min\{cx + z\mu \colon (x, \mu) \in C^{x\mu}\}$ directly, especially when explicit expressions defining the facets of the feasible region $C^{x\mu}$ (which is a convex polyhedron) are not known. Thus, we turn to finding a lower bound of $\pi(z)$ by relaxing Q^{xy} to some convex polyhedron P^{xy} that can be represented in the form $\{(x, y) \colon A'x \geqslant B'y + D'\}$. Note that one intuitive way to obtain such P^{xy} is to relax the integral constraints in Q^{xy}.

We then solve $\min\{cx + z\mu \colon A'x \geqslant B'\mathbf{1} + D'\mu\}$, and denote its optimal solution by $[x^*, \mu^*]$. According to Lemma 3, we know that $cx^* + z\mu^*$ provides a lower bound of $\pi(z)$, which equals $\pi(z)$ if P^{xy} equals the convex hull of Q^{xy}.

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

By Lemma 3 and our earlier argument, the value $c(V)-(cx^*+z\mu^*)$ provides an upper bound of $\omega(z)$. When $c(V)$ is computationally intractable, we can also apply the LP approach to obtain an upper bound $\omega^u(z)$ of $\omega(z)$ by replacing $c(V)$ with its upper bound $c^u(V)$. Under subsidy $\omega^u(z)$ and penalty z, the grand coalition of the IM game (V,c) can still be stabilized.

Furthermore, if polyhedron P^{xy} equals the convex hull of Q^{xy}, then by Lemma 3, the value $c(V)-(cx^*+z\mu^*)$ equals $\omega(z)$, and $[x^*,\mu^*]$ is also an optimal solution to $\min\{cx+z\mu\colon (x,\mu)\in C^{x\mu}\}$. Thus, in this case, it can be verified that setting $K^z_{l'}=-\mu^*$ and $K^z_{r'}=-\mu^*$ forms a pair of weak derivatives of $\omega(z)$ at point z, which can be used in the IPC algorithm of Section 3.1.

We summarize the LP approach in Algorithm 4, and show its effectiveness and efficiency by establishing Theorem 6, which is based on Lemma 3 and the previous discussion.

Algorithm 4 (Linear Programming approach to computing $\omega(z)$ for a given z)

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

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

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

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 $K^z_{r'}=-\mu^*$.

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

(i) *the LP appr/oach 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};*

(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 $\min\{cx+z\mu\colon A'x\geqslant B'\mathbf{1}+D'\mu\}$, which is optimal if P^{xy} equals the convex hull of Q^{xy}.*

5. Applications to Parallel Machine Scheduling Games

To demonstrate their wide applicability, we apply our newly proposed model, algorithms, and solution approaches for the instrument of simultaneous penalization and subsidization to a class of parallel machine scheduling games. The results also reveal some interesting properties of these games. Next, we present the results for a game arising from identical parallel machine scheduling of unweighted jobs, which we call the IPU game. Results for other games are provided in Section EC.3 of the electronic companion.

In an IPU game, each player k in $V=\{1,2,\ldots,v\}$ has a job k that needs to be processed on one of m identical machines in $M=\{1,2,\ldots,m\}$, where $m\in\mathbb{Z}^+$. Each job $k\in V$ has a processing time denoted by t_k. Each coalition $s\in S$, where $S=2^V\backslash\{\varnothing\}$, aims to schedule the jobs in s on the machines in M so that the total completion time of the jobs in s is minimized, i.e., to minimize $\sum_{k\in s}C_k$, where C_k is the completion time of job $k\in s$. Thus, the value of the characteristic function for s, denoted by $c_{\mathrm{IPU}}(s)$, equals the minimum value of $\sum_{k\in s}C_k$. Using the notation in the scheduling literature, $c_{\mathrm{IPU}}(s)$ corresponds to problem $P\|\sum C_k$, which can be solved by the shortest processing time first (SPT) rule (Pinedo 2015).

Consider any instance (V,c_{IPU}) of the IPU game. It can be formulated as the following IM game: Let $O=\{1,2,\ldots,v\}$. For $k\in V$ and $j\in O$, define x_{kj} to be a binary variable, where $x_{kj}=1$ if and only if job k is scheduled on a machine as the jth to last job, contributing jt_k to the total completion time. Thus, for each coalition $s\in S$, the total completion time to be minimized for $c_{\mathrm{IPU}}(s)$ equals $\sum_{k\in s}\sum_{j\in O}(jt_k)x_{kj}$, which can be written as $\sum_{k\in s}\sum_{j\in O}c_{kj}x_{kj}$ by setting each $c_{kj}=jt_k$. To this end, $c_{\mathrm{IPU}}(s)$ can be formulated as the following integer linear program:

$$\begin{aligned} c_{\mathrm{IPU}}(s)=\min&\sum_{k\in V}\sum_{j\in O}c_{kj}x_{kj}\\ \text{s.t.}\ &\sum_{j\in O}x_{kj}-y^s_k=0,\quad \forall k\in V,\\ &\sum_{k\in V}x_{kj}\leqslant m,\quad \forall j\in O,\\ &0\leqslant x_{kj}\leqslant 1,\ x_{kj}\in\mathbb{Z},\quad \forall k\in V,\forall j\in O. \end{aligned}\tag{16}$$

Constraints $\sum_{j\in O}x_{kj}-y^s_k=0$, $\forall k\in V$, are equivalent to $\sum_{j\in O}x_{kj}-1=0$, $\forall k\in s$, indicating that only jobs in s are included and are processed only once. Constraints $\sum_{k\in V}x_{kj}\leqslant m$ for all $j\in O$ indicate that at most m machines can be used, and no jobs are processed on the same machine at the same time. Each decision variable x_{kj} is binary. Thus, (V,c_{IPU}) is an IM game.

For game (V,c_{IPU}), as we will illustrate in Sections 5.1 and 5.2 that the CP approach and the LP approach can be efficiently applied to computing the z-penalized minimum subsidy for any penalty z. Based on this, we will show in Section 5.3 that the IPC algorithm can be applied to constructing the PSF $\omega(z)$ in polynomial time.

5.1. Computing the z-Penalized Minimum Subsidy for Game (V,c_{IPU}) by the CP Approach

For game (V,c_{IPU}), we can apply the CP approach in Section 4.1 to computing the z-penalized minimum subsidy, i.e., the value of $\omega(z)$, for any penalty z. For this, we need to solve the following separation problem for any given cost allocation $\beta\in\mathbb{R}^v$:

$$\delta_{\mathrm{IPU}}=\min_{s\in S\backslash\{V\}}\left\{c_{\mathrm{IPU}}(s)+z-\sum_{k\in s}\beta_k\right\}.\tag{17}$$

The separation problem is devoted to finding an optimal coalition s^* among all coalitions $s \in S\backslash\{V\}$ that minimizes the difference between z plus the minimum total completion time $c_{\rm IPU}(s)$ for jobs in s and the total cost $\beta(s)$ assigned to players in s. Thus, if $\delta_{\rm IPU} < 0$, then constraint $\beta(s^*) \leqslant c_{\rm IPU}(s^*) + z$ is violated.

As mentioned earlier, for each $s \in S\backslash\{V\}$, $c_{\rm IPU}(s)$ can be solved by the SPT rule. Thus, it is optimal to process jobs in s with the shortest processing time first, which is equivalent to processing jobs in s with the longest processing time last. This results in the largest m jobs in s each being processed on different machines as the last processing jobs, the second m largest jobs in s each being processed on different machines as the second to last processing jobs, and so on. In other words, for $u = 1, 2, \ldots, |s|$, it is optimal to process the uth largest job in s on machine $[(u-1) \bmod m + 1]$ as the $\lceil u/m \rceil$th to last processing job.

To this end, we can show how to solve the separation problem (17) by dynamic programming (DP). Without loss of generalities, let us assume that $t_1 \geqslant t_2 \geqslant \cdots \geqslant t_v$. For each (k, u) with $k \in \{1, 2, \ldots, v\}$ and $u \in \{0, 1, \ldots, v\}$, let $P(k, u)$ indicate the minimum objective value of a restricted problem of (17), subject to the additional constraints that coalition s is a subset of $\{1, 2, \ldots, k\}$, and that s contains exactly u players. From our previous discussion, we know that if an optimum s^* to $P(k, u)$ contains player k, it is optimal to process job k on machine $[(u-1) \bmod m + 1]$ as the $\lceil u/m \rceil$th to last processing job, contributing $\lceil u/m \rceil t_k$ to the total completion time. Thus, it can be verified that $P(k, u)$ satisfies the following recursion:

$$P(k,u) = \min \begin{cases} P(k-1,u), \text{ for the case when} \\ \quad s^* \text{ does not contain } k, \\ P(k-1,u-1) + \lceil u/m \rceil t_k - \beta_k, \text{ for} \\ \quad \text{the case when } s^* \text{ contains } k. \end{cases} \tag{18}$$

The initial conditions for the recursion are $P(1, 0) = z$ and $P(1, 1) = t_1 - \beta_1 + z$, and the boundary conditions are $P(k, u) = \infty$ if $u > k$, for all $k \in V$. It can be seen that the optimal objective value $\delta_{\rm IPU}$ of the separation problem (17) is given by

$$\delta_{\rm IPU} = \min\{P(v, u) \colon u \in \{1, 2, \ldots, v-1\}\}.$$

We can now establish Lemma 4 which, together with Algorithm 3, implies that, for any given penalty z, the CP approach efficiently returns the value and the weak derivatives of $\omega(z)$ for game $(V, c_{\rm IPU})$ in polynomial time if the ellipsoid method is adopted.

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

5.2. Computing the z-Penalized Minimum Subsidy for Game $(V, c_{\rm IPU})$ by the LP Approach

For game $(V, c_{\rm IPU})$, we can also apply the LP approach in Section 4.2 to computing the z-penalized minimum subsidy, i.e., the value of $\omega(z)$, for any penalty z. Following (13), we use $Q^{xy}_{\rm IPU}$ to denote the overall set of feasible solutions to $c_{\rm IPU}(s)$ for all $s \in S\backslash\{V\}$. Let y_k be a binary variable, where $y_k = 1$ if and only if k is in some coalition $s \in S\backslash\{V\}$. From (13) it can be seen that $(x, y) \in Q^{xy}_{\rm IPU}$ if and only if (x, y) satisfies (i) constraints in (16) with y^s_k replaced by y_k, (ii) constraints $1 \leqslant \sum_{k\in V} y_k \leqslant v - 1$ (to exclude empty and grand coalitions), and (iii) constraints $0 \leqslant y_k \leqslant 1$ and $y_k \in \mathbb{Z}$ for all $k \in V$.

Let $P^{xy}_{\rm IPU}$ indicate the polyhedron defined by the LP relaxation of $Q^{xy}_{\rm IPU}$, with the integral constraints, $y_k \in \mathbb{Z}$ and $x_{kj} \in \mathbb{Z}$ for $k \in V$ and $j \in O$, being relaxed. We can establish Lemma 5.

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

It can be seen that the polyhedron $P^{xy}_{\rm IPU}$ can be represented as $\{(x, y) \colon A'x \geqslant B'y + D'\}$, where the dimensions of matrices A', B', and D' are polynomially bounded. Thus, $\min\{cx + z\mu \colon A'x \geqslant B'\mathbf{1} + D'\mu\}$ is an LP model whose optimal solution $[x^*, \mu^*]$ can be solved in polynomial time. Hence, since $c_{\rm IPU}(V)$ can be computed in polynomial time by the SPT rule, by Lemma 5, Theorem 6, and Algorithm 4 we obtain that the LP approach runs in polynomial time for game $(V, c_{\rm IPU})$, and that for any penalty z, it returns the exact value of $\omega(z)$ equal to $c_{\rm IPU}(V) - (cx^* + z\mu^*)$, as well as a pair of weak derivatives $K^z_{l'} = -\mu^*$ and $K^z_{r'} = -\mu^*$.

5.3. Construction of the PSF $\omega(z)$ for Game $(V, c_{\rm IPU})$

For game $(V, c_{\rm IPU})$, since both the CP approach and the LP approach can be used to compute the exact value of $\omega(z)$ in polynomial time for any given z, we can follow Algorithm 2 to obtain an ϵ-approximation of the PSF $\omega(z)$ for $z \in [0, z^*]$ in polynomial time.

Moreover, as indicated by Theorem 7, for game $(V, c_{\rm IPU})$, the number of breakpoints of $\omega(z)$ is polynomially bounded; therefore, we can obtain the exact PSF $\omega(z)$ in polynomial time by the IPC algorithm.

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

6. Conclusion

In this paper, we proposed a novel instrument for enabling a central authority to stabilize the grand coalition in an unbalanced cooperative game, which is of theoretical value and practical importance. The novelty lies in linking two previously unconnected concepts, i.e., penalization and subsidization, and using them simultaneously, which provides more flexibility for the central authority. To formulate the trade-off between the levels of penalty and subsidy for this new instrument, we introduced a penalty-subsidy function $\omega(z)$, and characterized its structural properties. To provide an overall picture of the trade-off, we proposed two algorithms to construct the curve of $\omega(z)$. Both algorithms rely on solving the value of $\omega(z)$, i.e., the minimum subsidy needed to stabilize the grand coalition for any given penalty z, for which we developed solution approaches based on the cutting plane method and the theory of linear programming and its duality. Our algorithms and solution approaches for the new instrument can be applied to a broad class of unbalanced cooperative games. We demonstrated their applicability using several parallel machine scheduling games, which in turn revealed some interesting new properties of these games.

Our work has opened several directions for future study on this new instrument of simultaneous penalization and subsidization. First, when the penalty-subsidy function $\omega(z)$ has a large number of breakpoints, it would be interesting to further investigate how to obtain an even better ϵ-approximation of function $\omega(z)$, in terms of smaller approximation errors, shorter running time, and faster convergence speed. Second, as we have shown, it is challenging to find the value of $\omega(z)$ for any given penalty z. We have revealed that finding $\omega(z)$ is equivalent to optimizing a linear function over a specific convex polyhedron. The LP approach we proposed can find $\omega(z)$ if we can obtain all the inequalities of the convex polyhedron, which, however, can be exponentially many. To efficiently find $\omega(z)$ in such situations will require new solution approaches and techniques. Third, when applying the new instrument to various unbalanced cooperative games, such as machine scheduling games, facility location games, travelling salesman games, etc., a considerable number of new and interesting research questions arise for future study. For example, in these games, how many breakpoints does the function $\omega(z)$ have? Also, can the value of $\omega(z)$ be fully solved or approximated in polynomial time under given z? The results obtained in this work have laid a solid foundation for addressing such questions.

Acknowledgments

The authors thank two anonymous referees and the associate editor for their valuable comments. The proofs of Theorems 1–3 were suggested by the associate editor, and have simplified the proofs in the earlier version of this paper.

References

Anily S, Haviv M (2007) The cost allocation problem for the first order interaction joint replenishment model. *Oper. Res.* 55(2):292–302.

Aydinliyim T, Vairaktarakis GL (2010) Coordination of outsourced operations to minimize weighted flow time and capacity booking costs. *Manufacturing Service Oper. Management* 12(2):236–255.

Bachrach Y, Elkind E, Meir R, Pasechnik D, Zuckerman M, Rothe J, Rosenschein JS (2009) The cost of stability in coalitional games. Mavronicolas M, Papadopoulou VG, eds. *Algorithmic Game Theory* (Springer, Berlin), 122–134.

Bertsimas D, Tsitsiklis JN (1997) *Introduction to Linear Optimization*, Vol. 6 (Athena Scientific Belmont, Nashua, NH).

Bondareva ON (1963) Some applications of linear programming methods to the theory of cooperative games. *Problemy Kibernetiki* 10:119–139.

Cai X, Vairaktarakis GL (2012) Coordination of outsourced operations at a third-party facility subject to booking, overtime, and tardiness costs. *Oper. Res.* 60(6):1436–1450.

Caprara A, Letchford AN (2010) New techniques for cost sharing in combinatorial optimization games. *Math. Programming* 124(1–2):93–118.

Chen X (2009) Inventory centralization games with price-dependent demand and quantity discount. *Oper. Res.* 57(6):1394–1406.

Chen X, Zhang J (2016) Duality approaches to economic lot-sizing games. *Production Oper. Management* 25(7):1203–1215.

Curiel I (2013) *Cooperative Game Theory and Applications: Cooperative Games Arising from Combinatorial Optimization Problems*, Vol. 16 (Springer Science & Business Media, Berlin).

Demange G (2004) On group stability in hierarchies and networks. *J. Political Econom.* 112(4):754–778.

Faigle U, Kern W (1993) On some approximately balanced combinatorial cooperative games. *Zeitschrift für Oper. Res.* 38(2): 141–152.

Faigle U, Kern W, Kuipers J (2001) On the computation of the nucleolus of a cooperative game. *Internat. J. Game Theory* 30(1): 79–98.

Fredericks JW, Labadie JW, Altenhofen JM (1998) Decision support system for conjunctive stream-aquifer management. *J. Water Resources Planning Management* 124(2):69–78.

Goemans MX, Skutella M (2000) Cooperative facility location games. *J. Algorithms* 50(2):76–85.

Grötschel M, Lovász L, Schrijver A (2012) *Geometric Algorithms and Combinatorial Optimization*, Vol. 2 (Springer Science & Business Media, Berlin).

Immorlica N, Mahdian M, Mirrokni VS (2008) Limitations of cross-monotonic cost-sharing schemes. *ACM Trans. Algorithms* 4(2):602–611.

Jain K, Mahdian M (2007) Cost sharing. Nisan N, Roughgarden T, Tardos E, Vazirani V, eds. *Algorithmic Game Theory* (Cambridge University Press, New York), 385–410.

Jain K, Vazirani V (2001) Applications of approximation algorithms to cooperative games. *Proc. 33rd Annual ACM Sympos. Theory Comput*. (ACM, Hersonissos, Crete, Greece), 364–372.

Kern W, Paulusma D (2003) Matching games: The least core and the nucleolus. *Math. Oper. Res.* 28(2):294–308.

Kimms A, Kozeletskyi I (2016) Core-based cost allocation in the cooperative traveling salesman problem. *Eur. J. Oper. Res.* 248(3): 910–916.

Könemann J, Leonardi S, Schäfer G (2005) A group-strategyproof mechanism for Steiner forests. *Proc. 16th Annual ACM-SIAM Sympos. Discrete Algorithms* (SIAM, Vancouver), 612–619.

Liu L, Qi X, Xu Z (2016) Computing near-optimal stable cost allocations for cooperative games by Lagrangian relaxation. *INFORMS J. Comput.* 28(4):687–702.

Liu Z (2009) Complexity of core allocation for the bin packing game. *Oper. Res. Lett.* 37(4):225–229.

Mallozzi L (2011) Cooperative games in facility location situations with regional fixed costs. *Optim. Lett.* 5(1):173–181.

Maschler M, Peleg B, Shapley LS (1979) Geometric properties of the kernel, nucleolus, and related solution concepts. *Math. Oper. Res.* 4(4):303–338.

Meir R, Rosenschein JS, Malizia E (2011) Subsidies, stability, and restricted cooperation in coalitional games. *Proc. 22nd Internat. Joint Conf. Artificial Intelligence*, Vol. 11 (AAAI Press, Barcelona, Catalonia, Spain), 301–306.

Osborne MJ, Rubinstein A (1994) *A Course in Game Theory* (MIT Press, Cambridge, MA).

Pinedo M (2015) *Scheduling: Theory, Algorithms, and Systems* (Springer, Berlin).

Puerto J, Tamir A, Perea F (2011) A cooperative location game based on the 1-center location problem. *Eur. J. Oper. Res.* 214(2): 317–330.

Resnick E, Bachrach Y, Meir R, Rosenschein JS (2009) The cost of stability in network flow games. Královiç R, Niwiński D, eds. *Proc. 34th Internat. Sympos. Math. Foundations Comput. Sci.* (Springer, Berlin), 636–650.

Sadegh M, Kerachian R (2011) Water resources allocation using solution concepts of fuzzy cooperative games: Fuzzy least core and fuzzy weak least core. *Water Resources Management* 25(10): 2543–2573.

Schulz AS, Uhan NA (2010) Sharing supermodular costs. *Oper. Res.* 58(4):1051–1056.

Schulz AS, Uhan NA (2013) Approximating the least core value and least core of cooperative games with supermodular costs. *Discrete Optim.* 10(2):163–180.

Shapley LS, Shubik M (1969) On market games. *J. Econom. Theory* 1(1):9–25.

Tamir A (1989) On the core of a traveling salesman cost allocation game. *Oper. Res. Lett.* 8(1):31–34.

Van der Zaag P, Seyam IM, Savenije HHG (2002) Towards measurable criteria for the equitable sharing of international water resources. *Water Policy* 4(1):19–32.

Yi S-S (1997) Stable coalition structures with externalities. *Games Econom. Behav.* 20(2):201–237.

Zhang J (2009) Cost allocation for joint replenishment models. *Oper. Res.* 57(1):146–156.

Lindong Liu is associate professor in the School of Management at the University of Science and Technology of China. His research interests are game theory and optimization.

Xiangtong Qi is professor in the Department of Industrial Engineering and Decision Analytics at the Hong Kong University of Science and Technology. His research interests are scheduling, logistics, and supply chain management.

Zhou Xu is associate professor in the Department of Logistics and Maritime Studies at the Hong Kong Polytechnic University. His research interests are operations research and its applications in logistics, transportation, shipping, and supply chain management.