

Summary

Dis·count

August 30, 2019

Introduction

At first, we will introduce some preliminary knowledge about cooperative game theory as follows. A pair (V, c) is usually used to represent a cooperative game with transferable utilities, among which $V = \{1, 2, \dots, v\}$ denotes a set of v players and $c : 2^V \rightarrow \mathbb{R}$ indicates the characteristic function of the game. A coalition is defined as a non-empty subset of players; while V is referred to as a coalition, $S = 2^V \setminus \{\emptyset\}$ denotes the set of all feasible coalitions. The characteristic function of the game, $c(s)$, represents the minimum coalitional cost the players in s have to pay in order to cooperate together. A cost allocation vector $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_v] \in \mathbb{R}^v$ is required by the game to maintain cooperation in the grand coalition, and α_k is the cost assigned to players $k \in V$ to assure no individual or group of players has the incentive to deviate. For the convenience of abbreviation, we use $\alpha(s) = \sum_{k \in s} \alpha_k$ to denote the total cost assigned to the players in coalition s . One of the most important concepts in cooperative game theory is core, which is a cost allocation satisfying two kinds of constraints, one is the budget balance constraint $\alpha(V) = c(V)$ and the other is the coalitional stability constraints $\alpha(s) \leq c(s)$. In other words, core can be expressed as

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

When the cost allocation exists, core is called non-empty. And if and only if the core is non-empty, the grand coalition of the associating cooperative game (V, c) will be stable or balanced.

However, cooperative games can be unbalanced in many cases owing to the joint restrictions of the above-mentioned two kinds of constraints. To stabilize the grand coalitions in unbalanced cooperative games, researchers have already developed several effective instruments, such as subsidization, penalization and simultaneous subsidization and penalization. The similarity of these instruments is that there exists an outside party who will take measures to stabilize the grand coalition. But the penalization will always arouse the discontent of players in coalitions, it's promising to find a new instrument to replace the penalization, we call it taxation. The significant idea of this instrument is that we can erase the role of the third party by collecting taxation as subsidy of the corresponding coalitions. That being said, the players of the games can stabilize themselves without the third party.

To make the project concrete, we will apply the cooperative theory on the machine schedule problem to

show our ideas.

Then we will illustrate the taxation instrument with a simple example of an IPU (Identical Parallel machine scheduling of Unweighted jobs) game as follows.

Example

There is a IPU game which contains four players, whose processing times on the identical parallel machine are $t_1 = 2, t_2 = 3, t_3 = 4, t_4 = 5$ respectively. Each machine setup cost is $t_0 = 9.5$, and $c(s)$ is the minimum total completion time of jobs in coalition s plus the machine setup cost.

In this example, the grand coalition has a minimum cost of 38, and the optimum number of machine is 2. It's easy to check that this example is unbalanced and calculate the subsidy which equals $c(V) - \alpha(V) = 0.75$.

When the setup cost increases from 9.5 to 13, that is, the taxation increases from 0 to 3.5, the number of machine will decrease from 2 to 1, accordingly.

Although this case is easily understood, it demonstrate the concept of taxation we have strong interests in. Then we will define the corresponding model to further elaborate on this concept.

Definition 1.

A cooperative TU game (V, c) is called an IPU game if it satisfies the following formulations.

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

Among these formulations, P and m denote the price and the number of using machines, respectively. Now we've already extended the number of players to a more complex case, that is, the number of players is set to n . Let $N = \{1, 2, \dots, n\}$ be a set of n players. The number of machines is m and the setup cost is t_0 . For convenience of expression, we set the processing times $t_i, i \in N$ satisfy $t_1 < t_2 < \dots < t_n$.

Then we need to define a function, that is, subsidy-price function (SPF) denoted by $\omega(P)$:

$$\omega(P) = \min_{\alpha} \{c(V, m(V, P)) - \alpha(V) : \alpha(s) \leq c(s, m(s, P)) \forall s \in S, \alpha \in \mathbb{R}^v\}.$$

By dividing the SPF function into two parts like $c(s) = c_0(s) + m_s P, s \in S$, as a common skill, we can

obtain its dual:

$$\omega^*(P) = \max_P \{c_0(V) + m(V)P + \sum_{s \in S \setminus \{V\}} -\rho_s [c_0(s) + m(s)P] : \sum_{s \in S \setminus \{V\} : k \in s} \rho_s = 1, \forall k \in V, \rho_s \geq 0, \forall s \in S \setminus V\} \quad (1)$$

We can establish Theorem 1 below, showing the property of the function.

Theorem 1. $\omega(P)$ is piecewise linear, and convex in price P when $m(V)$ is a fixed integer.

In this situation, the interval size of setup cost can be calculated when t_i is known. And we can obtain that under what circumstances the machine number will change. We set the different intervals where the number of machines remain unchanged as I_i respectively. The extreme points of these intervals are recorded as P_i respectively, which represents the price when the number of using machine changes from i to $i - 1$. Specially, P_1 denotes the extreme point of the whole interval, which is the value of price or setup cost when using one machine and subsidy equals zero.

Note that the costs of all the exponential coalitions can be easily calculated by the SPT rules, we have Lemma 1.

Lemma 1. The value at the extreme points of the sub-intervals I_i can be calculated with processing times t_i by comparing the costs of the grand coalitions where all the players use two adjacent numbers of machines.

According to the above lemma 1, we can obtain all the extreme points of the sub-intervals, i.e., the number of using machines decreases by one when the price or setup cost equals P_i .

Theorem 2. According to the foregoing description, we have the equation $S_1 = S_2 + \dots + S_n = \sum_{i=2}^n S_i$.

With Theorem 2, we obtain all the breakpoints during the interval of the price where the machine number changes and the subsidy is 0. Then we'll focus on the specific property of subsidy.

Theorem 3. The subsidy is always zero when m is larger than $\frac{n}{2}$. In other words, when the numbers of machine is larger than half of players, the grand coalition doesn't need any subsidy from the externality.

Theorem 4. When the number of machines used is 1, the range of slopes of the line segments in the interval is $\left(-1, -\frac{1}{n-1}\right]$, and the number of breakpoints is $O(v^2)$.

Until here, we described the main property of the whole figure. And a diagram of the number of machines and subsidy on setup cost, with its essential features being showed below.

The processing times of all players with the arrangement from small to large are listed on the top of this figure. The red and blue lines stand for the machine number and subsidy, respectively. The horizontal ordinate represents the price or setup cost. The left ordinate represents the value of machine number while the right represents the value of subsidy.

As the coalitional stability constraints showed above, the exponential inequality constraints are so tricky that we must figure out a method to simplify these constraints to obtain the optimal results. Fortunately for us, we find the Theorem 5.

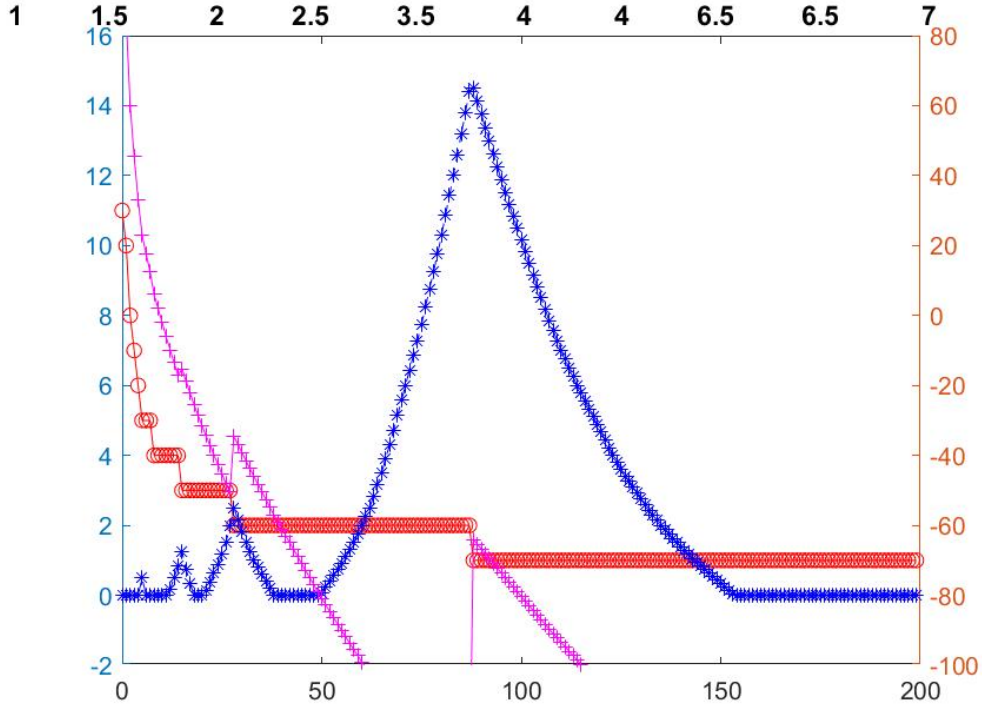


Figure 1: The number of machines and subsidy on setup cost.

Theorem 5. *When changing the price P , the effective inequalities correspond to the coalitions using only one machine.*

By establishing the Theorem 5, we can design the specific algorithm to eliminate the redundant inequalities, which will save us the trouble of solving $c(s)$ and speed up the solution.

Theorem 6. *The IPU game can be solved in polynomial time. (Remaining to be proved)*

Now that we've already solved the IPU game, we will try to extend the core concept to a more general case. So we need to define a new game as follow.

Definition 2

A cooperative TU game (V, c) is a PIM(Price Integer Minimization) game if there exist:

- positive integers e, v, t, P and m ;
- integer vectors $x \in \mathbb{Z}^{t \times 1}$ and $\tilde{\alpha} \in \mathbb{Z}^{1 \times t}$;
- left hand side matrix $A \in \mathbb{R}^{e \times t}$;
- right hand side matrix $B \in \mathbb{R}^{e \times v}$;
- a right hand side vector $D \in \mathbb{Z}^e$;

- an objective function vector $c \in \mathbb{Z}^{1 \times t}$;
- an incidence vector $y^s \in \{0, 1\}^v$ with $y_k^s = 1$ if $k \in s$ and $y_k^s = 0$ otherwise, $\forall k \in V$,

such that the characteristic function $c(s)$ equals the optimal objective value of the following integer linear program:

$$c(s, m(s, P)) = \min_x \{cx + Pm : Ax \geq By^s + D, \tilde{a}x \leq m, x \in \mathbb{Z}^{t \times 1}\}$$

Note that the object function is not easy to analyze due to the joint effect of cx and Pm , we need to divide the function into two parts for a further discussion.

With this idea, we divide $c(s, m(s, P))$ as $c_0(s, m(s)) + Pm$. Via analysis and study, we find some properties about $c_0(s, m(s))$ which we call primary function later.

The primary function must be supermodular, then we can get the positive price or setup cost when the number of facilities changes. This corresponds to the following two lemmas.

Lemma 2.

$$c_0(s, m) - c_0(s, m + 1) > 0 \Leftrightarrow P_i > 0$$

Lemma 3.

$$\begin{aligned} c_0(s, m) - c_0(s, m + 1) &< c_0(s, m - 1) - c_0(s, m) \\ \Leftrightarrow P_i &< P_{i-1}, i = 3, 4, \dots, n \end{aligned}$$

Once the primary function must be supermodular, we can specify this property in the following two aspects. One is the concavity about the number of facilities, the other is the nature about the number of players ($s_1 \subset s_2$). They correspond to the following two formulas:

$$c_0(s_1, m(s_1) - 1) - c_0(s_1, m(s_1)) \geq c_0(s_1, m(s_1)) - c_0(s_1, m(s_1) + 1) \quad (2)$$

and

$$c_0(s_1, m(s_1) - 1) - c_0(s_1, m(s_1)) \leq c_0(s_2, m(s_1) - 1) - c_0(s_2, m(s_1)) \quad (3)$$

, respectively.

Then we can develop the Theorem 7.

Theorem 7. When coalitions s_1, s_2 satisfy $s_1 \subset s_2$, the corresponding number of using facilities m_{s_1}, m_{s_2} have $m_{s_1} \leq m_{s_2}$, if the primary function satisfies (2) and (3).

Proof 1 (Lemma 1). In fact, according to the SPT rule, a fixed number of using machines for the given grand coalition corresponds to the only deterministic sort order working on the machines, i.e. the deterministic cost. Therefore, by comparing the cost of the number of adjacent machines $(k, k + 1)$ used by the grand coalition, the corresponding setup cost or the price P_i can be obtained. \square

Proof 2 (Theorem 2). For convenience of expression, we set the setup cost as S_1, S_2, \dots, S_n at interval points while the number of machine changes. And S_i denotes the setup cost when the machine number changes from i to $i - 1$, especially, S_1 denotes the least setup cost when machine number is 1 and the corresponding subsidy is 0. We just need to prove the following equality

$$S_1 = S_2 + \dots + S_n = \sum_{i=2}^n S_i.$$

Notice that

$$(n-1) \sum_{s \in S \setminus \{V\}} \rho_s \geq \sum_{k \in V} \sum_{s \in S \setminus \{V\} : k \in s} \rho_s = n.$$

The left side of the inequality means for every ρ_s can appear at most $(n-1)$ times, so we should know that if and only if for every $\rho_s > 0$ appears $n-1$ times the quality holds. That is to say, the coalitions which contains $n-1$ players are all maximally unsatisfied coalitions. Then we have $\binom{n}{n-1}$ equalities.

$$\begin{cases} \alpha_1 + \alpha_2 + \dots + \alpha_{n-1} &= x_1 \\ \alpha_1 + \alpha_3 + \dots + \alpha_n &= x_2 \\ \vdots &\vdots \\ \alpha_2 + \alpha_3 + \dots + \alpha_n &= x_n. \end{cases}$$

Add these n equations together, and we can get

$$(n-1)(\alpha_1 + \alpha_2 + \dots + \alpha_n) = \sum_{i=1}^n x_i$$

As we know, x_1, x_2, \dots, x_n can be expressed as follows:

$$\begin{cases} x_1 = S_0 + (n-1)t_1 + (n-2)t_2 + \dots + t_{n-1} \\ x_2 = S_0 + (n-1)t_1 + (n-2)t_3 + \dots + t_{n-1} \\ \vdots \\ x_n = S_0 + (n-1)t_2 + (n-2)t_3 + \dots + t_n \end{cases}$$

According to SPT rule, we can obtain the equality $c(V) = \alpha_1 + \alpha_2 + \dots + \alpha_n = S_0 + nt_1 + (n-1)t_2 + \dots + t_n$. By replacing x_1, x_2, \dots, x_n together with the expression of $c(V)$, we can get a equality only with $S_0, x_1, x_2, \dots, x_n$.

Finally, we can obtain $S_0 = \sum_{k=1}^n (n-k)t_k$. \square

Proof 3 (Theorem 1). By deriving from $\omega^*(P)$ (1), it can be seen that the SPF $\omega(P)$ is in fact the point-wise maximum of a set of straight lines, $c_0(V) + m(V)P + \sum_{s \in S \setminus \{V\}} -\rho_s[c_0(s) + m(s)P]$, each with a slope of $m(V) + \sum_{s \in S \setminus \{V\}} \rho_s m(s)$, for ρ in $\{\rho : \sum_{s \in S \setminus \{V\} : k \in s} \rho_s = 1 \text{ for all } k \in V, \text{ and } \rho_s \geq 0 \text{ for all } s \in S \setminus \{V\}\}$. Meanwhile, it's easy to see that $m(s)$ does not increase as P increases. Thus, we can claim that the SPF $\omega(P)$

is a convex function in P when $m(V)$ is a fixed integer.

Notice that the existence of $m(V)$ does not affect the property of the numbers of the breakpoints. To show that the SPF $\omega^*(P)$ is a piecewise linear function with a finite number of breakpoints, consider the feasible region of LP (1) for $\omega^*(P)$ without the part of $m(V)$ that is written as $\dot{\omega}(P)$, which is a convex polyhedron, denoted by \hat{R} . It can be seen that \hat{R} has a finite number of extreme points, and is independent of P . For any given $P \in [0, P^*]$, by LP (1) we know that there must exist an extreme point ρ of \hat{R} such that $\dot{\omega}(P)$ equals $c_0(V) + m(V)P + \sum_{s \in S \setminus \{V\}} -\rho_s [c_0(s) + m(s)P]$ and that the derivative of $\dot{\omega}(P)$ at P equals $\sum_{s \in S \setminus \{V\}} \rho_s m(s)$. Thus, the derivative of $\dot{\omega}(P)$ for $P \in [0, P^*]$ can have only a finite number of possible values. Moreover, due to the convexity of $\dot{\omega}(P)$, the derivative of $\dot{\omega}(P)$ is non-decreasing in P . Hence, the derivative of $\dot{\omega}(P)$ can change for only a finite number of times when P increases from 0 to P^* . Therefore, $\omega^*(P)$ is a piecewise linear function with a finite number of breakpoints. \square

Proof 4 (Theorem 3). From (1), we know that the slope is $m(V) - \sum_{s \in S \setminus \{V\}} \rho_s \cdot m(s)$ when setup cost is S_i . At the left side of S_i , the number of using machines is $m(V) = i$ and the slope equals $i - \sum_{s \in S \setminus \{V\}} \rho_s \cdot m(s)$, which is positive, while at the right side of S_i , the number of using machines is $m(V) = i - 1$ and the slope equals $\sum_{s \in S \setminus \{V\}} \rho_s \cdot m(s) - (i - 1)$, which is negative. So the sum of absolute values of slope on the left and right sides at S_i is 1. When $m(V)$ is 1, the expression of slope is $1 - \sum_{s \in S \setminus \{V\}} \rho_s \cdot m(s)$. Meanwhile, $m(s) \leq m(V)$, so $m(s) = 1$, then the slope is $1 - \sum_{s \in S \setminus \{V\}} \rho_s$. Because we know that the sum of absolute values of slope at both sides of S_2 is 1, so the absolute value of the right side of slope less than 1.

From Theorem 1, we have:

$$(n-1) \sum_{s \in S \setminus \{V\}} \rho_s \geq \sum_{k \in V} \sum_{s \in S \setminus \{V\}: k \in s} \rho_s = n$$

$\sum_{s \in S \setminus \{V\}} \rho_s$ minimum is $\frac{n}{n-1}$ the corresponding maximum slope is $1 - \sum_{s \in S \setminus \{V\}} \rho_s = -\frac{1}{n-1}$. \square

Proof 5 (Theorem 4.).

Proof 6 (Theorem 5.).

Proof 7 (Theorem 6.). Remaining

Proof 8 (Lemma 2.). $c_0(s)$