Summary

Dis·*count*

August 25, 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 $V = \{1, 2, \ldots, v\}$ denotes a set of v players and $c : 2^V \to R$ indicates the characteristic function of the game. A coalition is defined as a non-empty subset of players; while V is refered 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 $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_v] \in R^v$ 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 contraints, one is the budget balance contraint $\alpha(V) = c(V)$ and the other is the coalitional stability constrains $\alpha(s) \le c(s)$. In other words, core can be expressed as

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

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 instrment 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.

$$c(s, m(s, P)) = \min \sum_{k \in V} \sum_{j \in O} c_{kj} x_{kj} + Pm$$

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

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

$$\sum_{k \in s} x_{k1} = m(s),$$

$$x_{kj} \in \{0, 1\}, \forall k \in V, \forall j \in O,$$

$$y_k^s = 1, k \in s, y_k^s = 0, k \notin s.$$

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 $N = \{1, 2, \ldots, n\}$ 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 $t_1 < t_2 < \cdots < t_n$.

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

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

By obtaining its dual:

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

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 lemma1, 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 $S_1 = S_2 + \cdots + S_n = \sum_{i=2}^{n} S_i$.

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 don'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 constrains 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.

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 inequalitie, which will save us the trouble of solving $c(s)$ and speed up the solution.
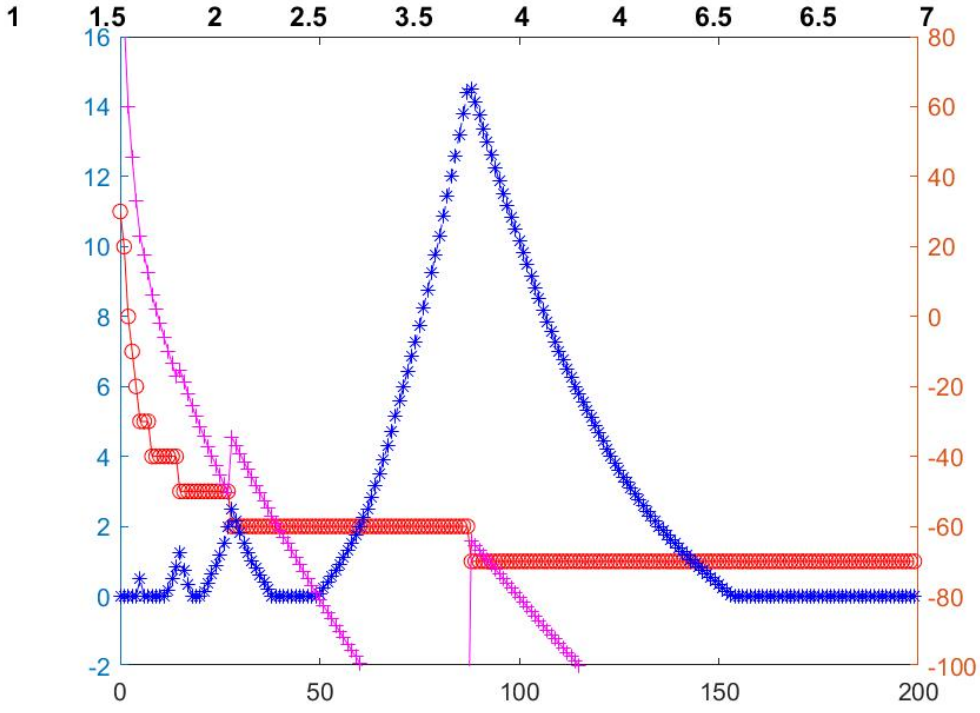
Figure 1: The number of machines and subsidy on setup cost.

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 $x \in Z^{t \times 1}$ and $\tilde{\alpha} \in Z^{1 \times t}$;

- left hand side matrix $A \in R^{e \times t}$;

- right hand side matrix $B \in R^{e \times v}$;

- a right hand side vector $D \in Z^e$;

- an objective function vector $c \in Z^{1 \times t}$;

- 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:

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

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

4

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.

$$c_0(s, m) - c_0(s, m + 1) < c_0(s, m - 1) - c_0(s, m)$$
$$\Leftrightarrow \quad P_i < P_{i-1}, i = 3, 4, \cdots, n$$

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:

$$c_0(s_1, m(s_1) - 1) - c_0(s_1, m(s_1)) \geq c_0(s_1, m(s_1)) - c_0(s_1, m(s_1) + 1) \tag{1}$$

and

$$c_0(s_1, m(s_1) - 1) - c_0(s_1, m(s_1)) \leq c_0(s_2, m(s_1) - 1) - c_0(s_2, m(s_1)) \tag{2}$$

, 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 (1) and (2).

Proof 1

For convenience of expression, we set the setup cost as $S_1, S_2, \ldots, S_n$ 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 have the equality

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

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.

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

Add these n equations together, and we can get

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

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

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

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

Proof * Remaining