

Lagrangian Heuristic for Simultaneous Subsidization and Penalization: Implementations on Rooted Travelling Salesman Games

Lindong Liu^{a,*}, Yuqian Zhou^a, Zikang Li^a

^a *International Institute of Finance, School of Management,
University of Science and Technology of China, Hefei 230026, China*

Abstract

This work examines the problem of stabilizing the grand coalition of an unbalanced cooperative game under the concept of simultaneous subsidization and penalization (S&P). We design a generic framework for developing heuristic algorithms to evaluate the trade-off between subsidy and penalty in the S&P instrument. By incorporating some Lagrangian relaxation techniques, we develop an approach for computing feasible subsidy–penalty pairs under which the grand coalition is stabilized in unbalanced cooperative games. This approach is particularly applicable when the characteristic functions of a cooperative game involve intractable integer programmes. To illustrate the performance of the Lagrangian relaxation based approach, we investigate the rooted travelling salesman game, and the computational results obtained show that our new approach is both efficient and effective.

Keywords: Cooperative Game, Travelling Salesman Game, Cost Allocation

1. Introduction

Cooperative game theory investigates ways to enforce and sustain cooperation among a set of players; the key difficulty is in how to fairly allocate the costs (or profits) among different players by taking individual and group incentives into consideration (e.g., see [Jain and Mahdian 2007](#)).

Strictly, in terms of cost minimization, a cooperative game involving transferable utilities can be represented by a pair (V, c) , where $V = \{1, 2, \dots, v\}$ denotes a set of players and $c : 2^V \rightarrow \mathbb{R}$ is the characteristic function of the game. A subset of players, denoted by S , is called a coalition;

*Corresponding author

Email addresses: ldliu@ustc.edu.cn (Lindong Liu), zyq94819@mail.ustc.edu.cn (Yuqian Zhou), kk1996@mail.ustc.edu.cn (Zikang Li)

while V is the grand coalition, the set of all feasible coalitions is denoted by $\mathbb{S} = 2^{|V|} \setminus \emptyset$. The characteristic function of the game is to specify for every coalition $S \in \mathbb{S}$, a value $c(S)$, representing the minimum coalitional cost the members in S have to pay in order to complete their work with cooperation.

To sustain cooperation in the grand coalition, the game requires a cost allocation vector $\theta = [\theta_1, \dots, \theta_v]$, where θ_k is the cost assigned to player k such that no individual or group of players has the incentive to deviate. By slightly modifying the notation for the sake of convenience, we use $\theta(S) = \sum_{k \in S} \theta_k$ to denote the total cost assigned to the players in coalition S . Now, the grand coalition of the cooperative game (V, c) will be stable, or the cooperative game (V, c) will be balanced, if and only if the associating *core* is non-empty (Bondareva 1963), that is

$$\text{Core}(V, c) = \left\{ \theta \in \mathbb{R}^v : \theta(V) = c(V) \text{ and } \theta(S) \leq c(S), \forall S \in \mathbb{S} \setminus V \right\} \neq \emptyset, \quad (1)$$

where $\{\theta(V) = c(V)\}$ is the budget balance constraint and $\{\theta(S) \leq c(S), \forall S \in \mathbb{S} \setminus V\}$ are the coalitional stability constraints (Shapley and Shubik 1969).

Due to the joint restrictions of the budget balance and coalitional stability constraints, the core of the cooperative game is sometimes empty. In this case, to stabilize the grand coalitions in unbalanced cooperative games, researchers have developed several effective instruments, such as the instruments of subsidization, penalization, and simultaneous subsidization and penalization (S&P). The common idea of these instruments is to relax the concept of core (Curiel 2013).

In the instrument of subsidization, the central authority provides the grand coalition with a subsidy if they cooperate; the key problem is to find the minimum subsidy needed. This instrument is closely related to the concept of γ -core (Faigle and Kern 1993) and the *optimal cost allocation problem* (Caprara and Letchford 2010; Liu et al. 2016), where the central authority has to assign the players as much cost as possible such that the coalitional stability constraints in (1) still hold. Mathematically, the budget balance constraint in (1) can be relaxed from $\{\theta(V) = c(V)\}$ to $\{\theta(V) \leq c(V)\}$, then we have

$$\omega^* = \min_{\alpha} \left\{ c(V) - \alpha(V) : \alpha(S) \leq c(S) \text{ for all } S \in \mathbb{S}, \alpha \in \mathbb{R}^v \right\},$$

where ω^* is the *minimum subsidy* required for the grand coalition to cooperate and $\alpha(S) = \sum_{k \in S} \alpha_k$ for all $S \in \mathbb{S}$.

As for the instrument of penalization, the central authority penalizes each sub-coalition for deviating from the grand coalition; hence, the problem is to minimize the penalty amount and

reduce the players' dissatisfaction. This instrument is similar to the concept of *least core value* (Maschler et al. 1979; Schulz and Uhan 2010), where the cost assigned to each sub-coalition should not be more than the minimum coalitional cost plus the least core value, while the budget balance constraint in (1) still holds. Mathematically, the coalitional stability constraints in (1) can be relaxed from $\{\theta(S) \leq c(S)\}$ to $\{\theta(S) \leq c(S) + z\}$ for all $S \in \mathbb{S} \setminus V$, then we can formulate the penalization instrument as

$$z^* = \min_{\beta, z} \left\{ z : \beta(V) = c(V), \beta(S) \leq c(S) + z \text{ for all } S \in \mathbb{S} \setminus \{V\}, z \in \mathbb{R}, \beta \in \mathbb{R}^v \right\},$$

where z^* is the *minimum penalty* required for the grand coalition to cooperate and $\beta(S) = \sum_{k \in S} \beta_k$ for all $S \in \mathbb{S}$.

Recently, Liu et al. (2018) introduced the new instrument of simultaneous subsidization and penalization (S&P) by integrating the existing two instruments. In the S&P instrument, the central authority subsidizes the grand coalition if they cooperate, but at the same time penalizes each sub-coalition if they deviate. For any given penalty z , the goal is to minimize the subsidy required to stabilize the grand coalition. Mathematically, by relaxing both the budget balance and coalition stability constraints in (1), the S&P instrument solves

$$z(\omega) = \min_{\beta, z} \left\{ z \in \mathbb{R} : \beta(V) = c(V) - \omega \text{ and } \beta(S) \leq c(S) + z, \forall S \in \mathbb{S} \setminus \{V\} \right\}, \quad (2)$$

where $z(\omega)$ is the *subsidized minimum penalty*. In addition, as a function of subsidy ω , $z(\omega)$ is also a subsidy–penalty function (SPF)¹.

As an integration of subsidization and penalization, the S&P instrument includes the two traditional instruments as its special cases. To be specific, when subsidy is zero, the corresponding subsidized minimum penalty $z(0)$ will be equal to the least core value z^* , whereas when subsidy is ω^* , the resulting minimum penalty $z(\omega^*)$ will be equal to zero. Therefore, following the S&P idea and stabilizing the grand coalition in unbalanced cooperative games would be a good choice.

By broadly focusing on the integer minimization (IM) game, Liu et al. (2018) developed the cutting plane (CP) and linear programming (LP) approaches based on the CP method and LP duality theory, respectively, to compute $\omega(z)$ for a given z . The two approaches are effective under certain situations: the CP approach is effective when the corresponding separation prob-

¹Note that Liu et al. (2018) study the trade-off between subsidy and penalty in the format of penalty–subsidy function, i.e., PSF, $\omega(z)$. Changing the format from PSF to SPF does not affect our future studies, but provides us with a perspective opposite to the S&P instrument.

lem is computationally tractable, and the LP approach is effective when the associating convex polyhedron is well identified. However, computing the subsidized minimum penalty $z(\omega)$ for any given ω is in general hard due to the following reasons. First, LP (2) is a **linear programming problem involving an exponential number of constraints**. Second, the characteristic function $c(S)$ is intractable in some IM games, such as machine scheduling (Liu et al. 2018), facility location (Kolen 1983; Goemans and Skutella 2000; Puerto et al. 2011), and travelling salesman (TSP) (Tamir 1989; Potters et al. 1992; Kimms and Kozeletskyi 2016).

Following the concept of simultaneous S&P, we develop a new heuristic algorithm based on the Lagrangian relaxation to compute the subsidized minimum penalty, $z^*(\omega)$, for an IM game. We briefly summarize our major contributions to the literature as follows.

First, we design a new framework to develop heuristic algorithms for computing $z(\omega)$. The key idea of the framework is to replace $c(S)$ for all $S \in \mathbb{S}$ with its tractable bounds and reduce the computational difficulties for solving (2). Compared with the CP approach in Liu et al. (2018), this framework is efficient especially when $c(S)$ is computational intractable in some IM games.

Second, following the heuristic framework, we introduce the well-known Lagrangian bounds and develop an algorithm based on Lagrangian relaxation (LRB algorithm) for computing $z(\omega)$. Compared with the LP approach in Liu et al. (2018), the LRB algorithm is effective since, **for those integer linear programmings which do not fulfill the integrality property**, Lagrangian bounds are sharper than linear relaxation bounds. Furthermore, the LRB algorithm is able to deal with nonlinear situations while the LP approach is not.

Third, we implement our ideas in the rooted TSP games. The result indicates the effectiveness and efficiency of the LRB algorithm in computing $z(\omega)$, associating with a fair cost allocation. It also contributes to the TSP game’s literature, which previously mainly focused on core existence checking and the time complexity analysis of the γ -core.

This paper is organized follows. Section 2 formally introduces the concepts of simultaneous S&P and IM games, and provides a TSP game example for illustration. Section 3 develops the LRB algorithm for computing the subsidized minimum penalty for IM games. Section 4 implements the LRB algorithm on the TSP game in order to demonstrate its effectiveness and efficiency. Section 5 concludes the paper.

2. Preliminaries

There are some cooperative games in the field of operations research, such as facility location and machine scheduling games, whose characteristic functions are defined using intractable integer programmes. These games can be included in the class of IM games studied in [Caprara and Letchford \(2010\)](#). We once again define the IM game as follows for better illustration.

Definition 1. A cooperative TU game (V, c) is called an IM game if there exist

- positive integers $e, e',$ and t ;
 - left-hand side matrices $A \in \mathbb{R}^{e \times t}$ and $A' \in \mathbb{R}^{e' \times t}$;
 - right-hand side matrices $B \in \mathbb{R}^{e \times v}$ and $B' \in \mathbb{R}^{e' \times v}$;
 - non-negative right-hand side column vectors $D \in \mathbb{R}_+^e$ and $D' \in \mathbb{R}_+^{e'}$;
 - an objective function $f(x)$ in x ($f(x)$ can be non-linear in this paper);
 - an incidence vector $\gamma^S \in \{0, 1\}^{v \times 1}$ with $\gamma_i^S = 1$, where if $i \in S$, $\gamma_i^S = 0$, otherwise, $\forall i \in V$,
- such that the cost $c(S)$ equals the optimal value of the following integer programme for all $S \in \mathbb{S}$:

$$c(S) = \min_x \left\{ f(x) : Ax \geq B\gamma^S + D, A'x \geq B'\gamma^S + D', x \in \{0, 1\}^{t \times 1} \right\}, \quad (3)$$

where constraints in (3) are divided into two parts for the usage of Lagrangian relaxation later in [Section 3.2](#).

It is known that many IM games are unbalanced, i.e., the associating grand coalition is unstable. As described in Introduction, the S&P instrument is proposed by [Liu et al. \(2018\)](#) to stabilize the grand coalition, where an amount of subsidy, ω , is provided to the grand coalition and an amount of penalty, $z(\omega)$, is imposed to the deviating sub-coalitions. The key problem for the S&P instrument is computing the subsidized minimum penalty $z(\omega)$ for any given subsidy ω , where the resulting **linear programming** problem is

$$z(\omega) = \min_{\beta, z} \left\{ z \in \mathbb{R} : \beta(V) = c(V) - \omega \text{ and } \beta(S) \leq c(S) + z, \forall S \in \mathbb{S} \setminus \{V\} \right\}.$$

Below, we illustrate the S&P instrument with a small example of a rooted TSP game as follows.

Example 1. Consider a rooted TSP game where node 1 is the root, nodes $N_t = \{2, 3, 4, 5, 6\}$ is a set of players, and $c(S)$ is the minimum transportation cost of the Hamiltonian cycle consisting of $S \cup \{1\}$ for any coalition $S \subseteq N_t$. The network structure is given in [Figure 1](#), where the values on the arcs represent the corresponding transportation costs.

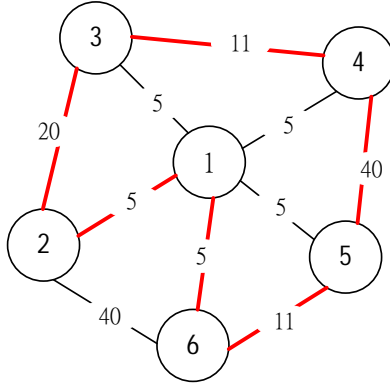


Figure 1: A rooted TSP game

In Example 1, the grand coalition, representing the social optimum, has a minimum cost of 92, with the optimal path being $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 1$. By checking LP (1), we can find that Example 1 is unbalanced, that is, the core is empty. Now, we stabilize the grand coalition by the S&P instrument at different subsidy levels, and then compute the corresponding subsidized minimum penalty levels $z(\omega)$ with LP (2). Some of the results are shown in Table 1.

Table 1: subsidized minimum penalty for the rooted TSP game under different subsidy levels

ω	0	6	12	18	24	30	36	42
$z(\omega)$	15	12	9.33	7.33	5.33	3.33	1.33	0
$\beta(2, \omega)$	25	22	19.33	17.33	15.33	13.33	11.33	10
$\beta(3, \omega)$	20	20	19.33	17.33	15.33	13.33	11.33	10
$\beta(4, \omega)$	11	11	11	11	11	11	11	10
$\beta(5, \omega)$	25	22	19.33	17.33	15.33	13.33	11.33	10
$\beta(6, \omega)$	11	11	11	11	11	11	11	10

From Table 1, we can see that, for example, if the subsidy provided by the central authority to the grand coalition is $\omega = 6$, then the minimum penalty needed to be imposed to the deviating sub-coalitions is $z(\omega) = 12$, and the resulting cost allocation vector is $\beta = [22, 20, 11, 22, 11]$. According to Liu et al. (2018), as well as the results shown in Table 1, the SPF $z(\omega)$ is decreasing, piecewise linear, and convex in ω . In addition, $z(0)$ is equal to the least core value, z^* , and ω^* , such that $z(\omega^*) = 0$, is the minimum subsidy required to stabilize the grand coalition.

We can solve LP (2) and derive all $z(\omega)$ and β in Table 1 efficiently since there are only five

players in the sample game. However, when there are numerous number of players, it would be hard to optimally solve LP (2) for the rooted TSP game due to its exponential number of constraints. Note that, for each constraint in LP (2), the computation of $c(S)$ is already hard in the TSP game (Woeginger 2003).

To apply the S&P instrument to the IM game, Liu et al. (2018) proposed two approaches (i.e., the CP and LP approaches) both of which can serve as approximation algorithms to solve LP (2), i.e., $z(\omega)$, under certain conditions. In the following section, we will develop an alternative approach to solve LP (2), which outperforms the CP and the LP approaches from the perspective of efficiency and effectiveness, respectively.

3. Design of Heuristic Algorithms

In this section, we first propose a generic framework to develop heuristic algorithms to solve LP (2), assuming that we can obtain some upper and lower bounds for $c(S)$. There are several specific methods, such as LP-based ones, which can be used to get the bounds for $c(S)$. For the sake of consistency and generality, the upper and lower bounds of $c(S)$ are both obtained by the Lagrangian relaxation method.

We then apply the well-known Lagrangian bounds to develop a practical approach for an approximate computation of the subsidized minimum penalty $z(\omega)$.

3.1. A Generic Framework for Heuristic Algorithms

For any $S \in \mathbb{S}$, we denote $c_l(S)$ and $c_u(S)$, respectively, as the respective lower and upper bounds of $c(S)$. By respectively replacing $c(S)$ and $c(V)$ with $c_l(S)$ and $c_u(V)$, we can derive a variant LP for (2), expressed as

$$\begin{aligned} z_r(\omega) = \min_{\beta, z} \quad & z \\ \text{s.t.} \quad & \beta(S) \leq c_l(S) + z, \quad \forall S \in \mathbb{S} \setminus \{V\}, \\ & \beta(V) = c_u(V) - \omega. \end{aligned} \tag{4}$$

Theorem 1. *The optimal value of LP (4) is an upper bound of the subsidized minimum penalty for an IM game (V, c) ; that is $z_r(\omega) \geq z(\omega)$ for any given subsidy ω .*

As a penalty function shown in LP (4), $z_r(\omega)$ is referred to as the restricted subsidy-penalty

function (R-SPF). Since $z_r(\omega)$ is an upper bound of $z(\omega)$, the joint effect of penalty $z_r(\omega)$ and subsidy ω is that no coalition will deviate from the grand coalition V .

Similar to Liu et al. (2018), we can derive some properties of the R-SPF and evaluate the trade-off between subsidy ω and penalty $z_r(\omega)$ for stabilizing the grand coalition of IM game (V, c) . The results are shown in Remark 1.

Remark 1. *Given any $c_u(V) \geq c(V)$ and $c_l(S) \leq c(S)$, $\forall S \in \mathbb{S} \setminus V$, for the R-SPF $z_r(\omega)$, we have,*

- (i) *the R-SPF $z_r(\omega)$ is strictly decreasing, piecewise linear, and convex in subsidy ω , and*
- (ii) *for each linear segment of $z_r(\omega)$, its derivative $z'_r(\omega)$ is in the range $[-\frac{v-1}{v}, -\frac{1}{v}] \subset [-1, 0]$, where v represents the cardinal number of coalition V , i.e., $|V|$.*

Compared with LP (2), we replace $c(S)$ with its tractable upper or lower bound. Thus, we might be able to solve LP (4) with some conventional techniques. Theorem 1 and Remark 1 can be applied to any situation with different types of $c(S)$ bounds. In the following subsection, we develop a solution by applying Lagrangian bounds.

3.2. The Lagrangian Subsidized Minimum Penalty

Following Theorem 1, we first need to obtain the Lagrangian lower bounds for $\{c(S) : \forall S \in \mathbb{S}\}$. We can carry out a typical Lagrangian relaxation procedure by relaxing constraints $\{A'x \geq B'\gamma^S + D'\}$ in LP (3) to the objective function with non-negative Lagrangian multipliers λ , and derive the Lagrangian characteristic function $c_\lambda(S)$ for an IM game (V, c) :

$$c_\lambda(S) = \min_x \left\{ f(x) - \lambda A'x + \lambda B'\gamma^S + \lambda D' : Ax \geq B\gamma^S + D, x \in \{0, 1\}^{t \times 1} \right\}, \forall S \in \mathbb{S}. \quad (5)$$

In ILP (5), λ is a non-negative row vector with dimension e' , that is $\lambda \in \mathbb{R}_+^{1 \times e'}$. In particular, for the grand coalition V , the ILP expression (5) is

$$c_\lambda(V) = \min_x \left\{ f(x) - \lambda A'x + \lambda B'\mathbf{1} + \lambda D' : Ax \geq B\mathbf{1} + D, x \in \{0, 1\}^{t \times 1} \right\}.$$

As with usual Lagrangian relaxation, constraints $\{A'x \geq B'\gamma^S + D'\}$ can be chosen such that $c_\lambda(S)$ is relatively easy to solve, for example in (pseudo-)polynomial for any $S \in \mathbb{S}$.

Since $c_\lambda(V)$ is a lower bound of $c(V)$ for any non-negative λ , to obtain the sharpest Lagrangian lower bound for $c(V)$, we need to solve the Lagrangian dual problem; that is we have to find the

best Lagrangian multipliers λ to maximize $c_\lambda(V)$:

$$d(V) = \max_{\lambda} \left\{ \min_x \left\{ f(x) - \lambda A'x + \lambda B'\mathbf{1} + \lambda D' : Ax \geq B\mathbf{1} + D, x \in \{0, 1\}^{t \times 1} \right\} : \lambda \geq \mathbf{0} \right\}.$$

According to the Lagrangian dual theory, $d(V)$ is a concave function, then the sub-gradient method (e.g., see Fisher 1981) can be applied to derive the optimal Lagrangian multipliers λ^* for $d(V)$. In the sub-gradient method, an initial Lagrangian multiplier is given as λ^0 and the sequence $\{\lambda^k\}$ is generated by $\lambda^{k+1} = \lambda^k + t_k(Ax^k - b)$, where x^k is an optimal solution to the Lagrangian problem $Z_T(\lambda^k)$, $Z_T(\lambda^k) = \min \{cx + \lambda^k(Ax - b) : Tx \geq e, x \in \{0, 1\}^{n \times 1}\}$ and t_k is a positive scalar step size. And the step size used most commonly is $t_k = u_k (Z^* - Z_T(\lambda^k)) / \|Ax^k - b\|^2$, where u_k is determined by setting $u_0 = 2$ and halving u_k whenever $Z_T(\lambda)$ does not change its value after some fixed number of iterations, $\|\cdot\|$ represents L_2 -norm of \cdot . And Z^* is an upper bound on $Z_T = \max_{\lambda} Z_T(\lambda)$, commonly the initial upper bound can be an arbitrary feasible solution for the original integer programming and can be updated when finding a new feasible solution during the iteration.

The following discussion applies to any λ , not necessarily λ^* , although usually λ^* leads to better results.

By substituting $c_l(S)$ with $c_\lambda(S)$ in LP (4), we can derive the associating Lagrangian subsidized minimum penalty problem as follows:

$$\begin{aligned} z_\lambda(\omega) &= \min_{\beta, z} z \\ \text{s.t. } &\beta(S) \leq c_\lambda(S) + z, \quad \forall S \in \mathbb{S} \setminus V, \\ &-\beta(V) \leq -c_u(V) + \omega. \end{aligned} \tag{6}$$

Note that by showing the second constraint in (6) as an inequality, we can simplify our future computations. This transformation is feasible and the explanation is shown below. Firstly, LP(4) is to find the minimum value, thus, we just need to limit $c_u(V) - \beta(V)$ which is equal to ω in (4), not to increase to realize the feasible transformation because $z_\lambda(\omega)$ is decreasing with respect to ω as in Remark 1. If $c_u(V) - \beta(V)$ can be larger than ω in a new feasible region, $z_\lambda(\omega)$ will be less than the original minimum one in LP(4), which violates the equivalence of transformation. In this way, $c_u(V) - \beta(V)$ need to be no larger than ω , that is, $c_u(V) - \beta(V) \leq \omega$. After reformulating, we can obtain the second constraint in (6).

From Theorem 1, the optimal $z_\lambda(\omega)$ to LP (6), or the Lagrangian subsidized minimum penalty, is an upper bound of the subsidized minimum penalty for the IM game (V, c) . In this case, given

any subsidy ω , the IM game (V, c) can be balanced by letting the penalty deviate from the grand coalition to become $z_\lambda(\omega)$.

Motivated by [Liu et al. \(2016\)](#), the remainder of this subsection shows how to compute $z_\lambda(\omega)$ with Lagrangian decomposition and column generation techniques.

First, for any non-negative Lagrangian multipliers λ , we can decompose the Lagrangian characteristic function c_λ into two sub-characteristic functions $c_{\lambda 1}$ and $c_{\lambda 2}$ such that $c_\lambda(S) = c_{\lambda 1}(S) + c_{\lambda 2}(S)$ under any $S \in \mathbb{S}$, where

$$c_{\lambda 1}(S) = \lambda B' \gamma^S,$$

and

$$c_{\lambda 2}(S) = \min_x \left\{ f(x) - \lambda A'x + \lambda D' : Ax \geq B\gamma^S + D, x \in \{0, 1\}^{t \times 1} \right\}.$$

We define IM subgames 1 and 2 as $(V, c_{\lambda 1})$ and $(V, c_{\lambda 2})$, respectively. Note that the decomposition mode $(V, c_{\lambda 1})$ is modular, that is $c_{\lambda 1}(s_1) + c_{\lambda 1}(s_2) = c_{\lambda 1}(s_1 \cup s_2)$ for any two disjoint coalitions $s_1, s_2 \in \mathbb{S}$. Now, we have an LP equivalent to (6), as shown in Lemma 1.

Lemma 1. *LP (6) has the same optimal objective function value as LP*

$$\begin{aligned} z_\lambda(\omega) = & -\max_{\beta, z} -z \\ \text{s.t. } & \beta(S) - z \leq c_{\lambda 2}(S), \quad \forall S \in \mathbb{S} \setminus V, \\ & -\beta(V) \leq -c_u(V) + \omega + c_{\lambda 1}(V). \end{aligned} \tag{7}$$

Next, we deal with the problems arising from the exponential number of constraints in (7). By applying the column generation technique, we can solve (7) via its dual problem:

$$\begin{aligned} \min_{\mu} \quad & \sum_{S \in \mathbb{S} \setminus V} c_{\lambda 2}(S) \mu_s + [\omega + c_{\lambda 1}(V) - c_u(V)] \mu_V, \\ \text{s.t. } \quad & \sum_{S \in \mathbb{S} \setminus V} \gamma_k^S \mu_s - \mu_V = 0, \quad \forall k \in V, \\ & -\sum_{S \in \mathbb{S} \setminus V} \mu_s = -1, \\ & \mu_s \geq 0, \quad \forall S \in \mathbb{S}, \end{aligned} \tag{8}$$

where $\{\mu_s : \forall S \in \mathbb{S}\} \in \mathbb{R}^{2^v - 1}$ are the dual variables. We summarize the procedures for solving the Lagrangian subsidized minimum penalty $z_\lambda(\omega)$ in Algorithm 1.

Algorithm 1 LRB Algorithm to Compute $z_\lambda(\omega)$

Step 1. For an IM game (V, c) , compute the Lagrangian bounds $c_\lambda(S)$ for $c(S)$, to obtain LP (6).

Step 2. Decompose $c_\lambda(S)$ into $c_{\lambda 1}(S)$ and $c_{\lambda 2}(S)$, and derive LP (7) and its dual LP (8).

Step 3. Start from the restricted master problem of (8), where the restricted coalition set is $\mathbb{S}' \subset \mathbb{S}$, and compute its optimal dual solution $\pi^* \in \mathbb{R}^{v+1}$. **The restricted coalition set should contain a polynomial number of elements, for example, $\mathbb{S}' = \{\{1\}, \{2\}, \dots, \{v\}\}$.**

Step 4. Find a coalition S^* with the minimum negative cost by solving the pricing problem,

$$\min_{S \in \mathbb{S} \setminus \mathbb{S}'} c_{\lambda 2}(S) - \sum_{k \in V} \gamma_k^s \pi_k^* + \pi_{v+1}^*. \quad (9)$$

Step 5. Following the duality theory, if there is a \bar{S} with negative value of (9), add the \bar{S} into \mathbb{S}' , and go back to step 3; otherwise (7) is optimally solved with its dual problem (8), go to step 6.

Step 6. From the updated restricted coalition family \mathbb{S}' and its corresponding characteristic function values $\{c_{\lambda 2}(S) : \forall S \in \mathbb{S}'\}$, the following LP gives the optimal Lagrangian subsidized minimum penalty:

$$\begin{aligned} z_\lambda(\omega) = \min_{\beta, z} \quad & z \\ \text{s.t.} \quad & \beta(S) - z \leq c_{\lambda 2}(S), \quad \forall S \in \mathbb{S}' \setminus V, \\ & -\beta(V) \leq -c_u(V) + \omega + c_{\lambda 1}(V). \end{aligned} \quad (10)$$

As we know, in column generation, when the pricing problem is hard to solve, one can take a compromised approach and find an \bar{S} with a negative **reduced cost**, which will suffice till the last iteration. However, this compromised approach is problem specific. In Section 4, we show how to efficiently deal with the pricing problem in a rooted TSP game. We refer to $z_\lambda(\omega)$, a function of ω , as the Lagrangian R-SPF (LR-SPF).

Lemma 2. *By connecting a set of points $\left\{ (\omega_0, z_\lambda(\omega_0)), (\omega_1, z_\lambda(\omega_1)), \dots, (\omega_l, z_\lambda(\omega_l)) \right\}$ in the interval $[\omega_0, \omega_l]$, we can derive the upper bound $U_\lambda(\omega)$ on the curve of LR-SPF $z_\lambda(\omega)$ as well as SPF $z(\omega)$.*

Lemma 2 implies that the upper bound of the SPF curve $z(\omega)$ can be obtained by simply linking some discrete subsidy–penalty pairs $(\omega, z_\lambda(\omega))$. The accuracy of $U_\lambda(\omega)$ depends on both the accuracy of $z_\lambda(\omega)$ and number of points sampled. From Remark 1, because the derivative of $z_\lambda(\omega)$ is within $[-\frac{v-1}{v}, -\frac{1}{v}]$, the error between curves $U_\lambda(\omega)$ and $z_\lambda(\omega)$ is somehow bounded, but the error between $z_\lambda(\omega)$ and $z(\omega)$ for given ω still remains unknown. Therefore, we need to find the lower bound of $z(\omega)$ in order to evaluate the effectiveness of the Lagrangian subsidized minimum penalty $z_\lambda(\omega)$.

3.3. The Lower Bound of $z(\omega)$

We propose a method to calculate the lower bound $z_k(\omega)$, also called the k -level lower bound, of $z(\omega)$ by merely considering a subset of the constraints in LP (2). The value of $z_k(\omega)$ is obtained from LP,

$$\begin{aligned}
z_k(\omega) = \min_{\beta, z} z \\
s.t. \quad & \beta(S) \leq c_u(S) + z, \text{ where } |S| \leq k \text{ or } |S| > v - k, \forall S \in \mathbb{S} \setminus V, \\
& \beta(S) \leq c_u(S) + z, \forall S \in \mathbb{S}'_\lambda, \\
& \beta(V) = c_\lambda(V) - \omega.
\end{aligned} \tag{11}$$

In LP (11), $|S|$ represents the number of players in coalition S , and the coalition set \mathbb{S}'_λ is a subset of the restricted coalition family \mathbb{S}' derived at the end of the LRB algorithm. For a given k , LP (11) includes coalitions with no more than k or no less than $v + 1 - k$ players, rendering the number of constraints in the order of $O(v^k)$. To further tighten the lower bound, we add constraints $\{\beta(S) \leq c_u(S) + z, \forall S \in \mathbb{S}'_\lambda\}$ to (11), where $\mathbb{S}'_\lambda = \{S \in \mathbb{S}' : \beta_\lambda(S) - z_\lambda(\omega) = c_{\lambda 2}(S)\}$, since constraints $\{\beta(S) \leq c(S) + z, \forall S \in \mathbb{S}'_\lambda\}$ are more likely to be tight when solving LP (2).

Lemma 3. *The optimal $z_k(\omega)$ derived from (11) is the lower bound of the subsidized minimum penalty $z(\omega)$.*

The lower bound given by LP (11) is sharper when k is larger, but when k is no smaller than $\frac{v}{2}$, the corresponding lower bound $z_k(\omega)$ would be equal to $z(\omega)$, with the computation taking a very long time. For time efficiency, we compute only the two-level lower bound $z_2(\omega)$ in the rooted TSP game.

4. Implementations on the Rooted TSP Game

In this section, we illustrate the LRB algorithm on the rooted TSP game, compute its Lagrangian subsidized minimum penalty, and investigate the quantitative relationship between subsidy and penalty in stabilizing its grand coalition.

4.1. Game Definition

The TSP game examined here is symmetric and rooted; that is the distances between two cities are the same in both directions, and there is a depot city. There is a complete undirected graph $G = (N_p \cup \{1\}, E)$ with $N_p \cup \{1\}$ representing the city sites, E being the set of edges linking these cities and city '1' being the depot city. Each edge $\{i, j\} \in E$ has a unit transportation cost of $c_{ji} = c_{ij} > 0$. For a given coalition $S \in N_p$, the goal is to find a Hamiltonian circuit on cities $S \cup \{1\}$ with minimum total transportation cost. The notations used in the rooted TSP game are listed in Table 2.

Table 2: Notations used in rooted TSP game

N	Set of city sites, $N = \{1, 2, \dots, n\}$ with city 1 being the depot.
N_t	Set of players, $N_t = \{2, 3, \dots, n\}$.
c_{ij}	Cost to travel from city i to city j , $\forall \{i, j\} \in N^2$.
\mathbb{S}	Feasible player coalition set, $\mathbb{S} = 2^{N_t} \setminus \emptyset$.
S	A feasible player coalition, $S \in \mathbb{S}$.
γ^S	Incidence vector $[\gamma_2^S, \gamma_3^S, \dots, \gamma_n^S]^T$, where $\gamma_j^S = 1$, if $j \in S$, and $\gamma_j^S = 0$, otherwise.

Definition 2. A rooted TSP game is defined as (N_t, c_t) , with the player set N_t and characteristic function $c_t(S)$ determined by integer linear program (ILP),

$$c_t(S) = \min_x \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}$$

$$s.t. \quad \sum_{i \in N} x_{ij} + \sum_{i \in N} x_{ji} = 2\gamma_j^{S \cup \{1\}}, \quad \forall j \in N, \quad (12)$$

$$\sum_{i \in \varphi} \sum_{j \in \varphi} x_{ij} \leq |\varphi| - 1, \quad \forall \varphi \subset 2^{S \cup \{1\}} \setminus \emptyset, \quad (13)$$

$$\sum_{i \in N} \sum_{j \in N} x_{ij} = |S| + 1, \quad (14)$$

$$\begin{aligned} x_{ij} &\leq \gamma_i^{S \cup \{1\}}, \quad \forall i \in N, j \in N, \\ x_{ij} &\leq \gamma_j^{S \cup \{1\}}, \quad \forall i \in N, j \in N, \end{aligned} \quad (15)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in N, j \in N. \quad (16)$$

In the above ILP, constraints (12) are known as degree equations, which require the degree of a city to be ‘2’ on the final trip if the city is in $S \cup \{1\}$; otherwise, the degree should be ‘0’.

Constraints (13) are known as subtour elimination constraints **which were developed by Dantzig et al. (1954)**, ensuring that feasible tours should cover all cities in $S \cup \{1\}$. Constraints (14) and (15) are redundant due to (12), but they will be helpful in computing the Lagrangian lower bound and Lagrangian subsidized minimum penalty for the rooted TSP game.

In view of Definition 1, the rooted TSP game (N_t, c_t) is an IM game (V, c) with $V = N_t$ and $c = c_t$. The specific expressions of matrices C , A , A' , B , B' , D , and D' can be obtained by transforming c_t to a matrix format.

4.2. Computation of the Lagrangian Subsidized Minimum Penalty

In $c_t(S)$, by relaxing constraints $\{\sum_{i \in N} x_{ij} + \sum_{i \in N} x_{ji} = 2\gamma_j^S : j \in N_t\}$ to the objective of Lagrangian multipliers σ , we can derive the Lagrangian characteristic function for the rooted TSP game as

$$\begin{aligned} c_\sigma(S) &= \min_x \sum_{i \in N} \sum_{j \in N} (c_{ij} + \sigma_i + \sigma_j) x_{ij} - 2 \sum_{j \in N_t} \sigma_j \gamma_j^S \quad (\text{Let } \sigma_1 = 0) \\ &\quad s.t. \quad \sum_{i \in N} x_{i1} + \sum_{i \in N} x_{1i} = 2, \\ &\quad \sum_{i \in \varphi} \sum_{j \in \varphi} x_{ij} \leq |\varphi| - 1, \quad \forall \varphi \subset 2^{S \cup \{1\}} \setminus \emptyset, \\ &\quad \sum_{i \in N} \sum_{j \in N} x_{ij} = |S| + 1, \\ &\quad x_{ij} \leq \gamma_i^{S \cup \{1\}}, \quad \forall i \in N, j \in N, \\ &\quad x_{ij} \leq \gamma_j^{S \cup \{1\}}, \quad \forall i \in N, j \in N, \\ &\quad x_{ij} \in \{0, 1\}, \quad \forall i \in N, j \in N. \end{aligned} \quad (17)$$

ILP (17) shows that the redundant constraints $\{\sum_{i \in N} \sum_{j \in N} x_{ij} = |S| + 1\}$ and $\{x_{ij} \leq \gamma_i^{S \cup \{1\}}, x_{ij} \leq \gamma_j^{S \cup \{1\}} : \forall i \in N, j \in N\}$ can improve the Lagrangian lower bound for $c(N_t)$. By dividing c_σ into two parts, we can define the rooted TSP sub-games 1 and 2 as $(N_t, c_{\sigma 1})$ and $(N_t, c_{\sigma 2})$, respectively.

Sub-game 1 is modular, where the expression of $c_{\sigma 1}(S)$ is

$$c_{\sigma 1}(S) = -2 \sum_{j \in N_t} \sigma_j \gamma_j^S,$$

and the specific expression of $c_{\sigma 2}(S)$ is

$$\begin{aligned} c_{\sigma 2}(S) = \min_x & \sum_{i \in N} \sum_{j \in N} (c_{ij} + \sigma_i + \sigma_j) x_{ij}, \\ \text{s.t.} & \sum_{i \in N} x_{i1} + \sum_{i \in N} x_{1i} = 2, \\ & \sum_{i \in \varphi} \sum_{j \in \varphi} x_{ij} \leq |\varphi| - 1, \quad \forall \varphi \subset 2^{S \cup \{1\}} \setminus \emptyset, \\ & \sum_{i \in N} \sum_{j \in N} x_{ij} = |S| + 1, \\ & x_{ij} \leq \gamma_i^{S \cup \{1\}}, \quad \forall i \in N, j \in N, \\ & x_{ij} \leq \gamma_j^{S \cup \{1\}}, \quad \forall i \in N, j \in N, \\ & x_{ij} \in \{0, 1\}, \quad \forall i \in N, j \in N. \end{aligned}$$

In the literature, $c_{\sigma 2}(N_t)$ is called a ‘1-tree problem’ (e.g. [Held and Karp 1971](#)). Similarly, we refer to $c_{\sigma 2}(S)$ as a ‘1-S tree problem’ with a subgraph $G = (N_t \cup \{1\}, E)$ consisting of a tree on cities S and two distinct edges at city 1 linked to S , and minimum transportation cost. To solve the 1-S tree problem polynomially, we need to first construct a minimum spanning tree on cities S and then add two edges linked to city 1 with the smallest transportation cost.

For ease of exposition, we rewrite $c_{\sigma 2}(S)$ in an equivalent expression as

$$\begin{aligned} c_{\sigma 2}(S) = \min_x & \sum_{i \in N} \sum_{j \in N} c'_{ij} x_{ij} \quad (c'_{ij} = c_{ij} + \sigma_i + \sigma_j) \\ \text{s.t.} & \quad x \text{ is a 1-S tree.} \end{aligned}$$

Now, we apply the LRB algorithm and compute the Lagrangian subsidized minimum penalty $z_\sigma(\omega)$ for the rooted TSP game. For this particular application of LRB algorithm, we omit all the detailed descriptions and analyse only its hardest part, the corresponding pricing problem.

Specifically, this pricing problem can find the coalition $S \in N_t$ with the minimum negative cost:

$$\begin{aligned} \min_{S \in N_t} \sum_{i \in N} \sum_{j \in N} c'_{ij} x_{ij} - \sum_{j \in N_t} \gamma_j^S \pi_j^* + \pi_{n+1}^* \\ \text{s.t. } x \text{ is a 1-}S \text{ tree,} \end{aligned} \quad (18)$$

where π^* gives the optimal dual solution to the corresponding master problem defined in LP (8). We denote the resulting Lagrangian subsidized minimum penalty for the rooted TSP game as $z_\sigma(\omega)$.

As stated in the description of Algorithm 1, it is hard to optimally solve ILP (18), but we can still apply the LRB algorithm before it is completed, because we only need to find column \bar{S} with a negative cost to continue the algorithm.

To further enhance computational efficiency, we adopt the Minimum Adjacency Test (see Duin and Volgenant 1987; Ljubić et al. 2006) and reduce the magnitude of the problem (18). This method is widely used to reduce the magnitude of prize-collecting Steiner tree problems. The key idea is to merge two cities into one if it does not affect the optimal value. With some modifications, we can obtain an efficient reduction technique for ILP (18), as shown in Lemma 4.

Lemma 4. *If the pricing problem (18) has two nodes $a, b \in N_t$ such that*

$$\min \{ \pi_a^*, \pi_b^* \} - c'_{ab} > 0 \text{ and } c'_{ab} = \min_{ag \in N_t^2} c'_{ag},$$

then a and b can be merged into a new city c with prize $\pi_c^ = \pi_a^* + \pi_b^* - c'_{ab}$ and transportation costs $c_{ic} = c_{ci} = \min \{ c_{ai}, c_{bi} \}$ for any $i \in N \setminus \{a, b\}$.*

Although this does not guarantee polynomial time complexity, Lemma 4 can greatly reduce the problem size when solving (18) and ensure the time efficiency of the LRB algorithm when implementing a rooted TSP game. The computational results are shown in Section 4.3.

4.3. Computational Results

In this subsection, we present our results of applying the LRB algorithm to the rooted TSP game. We conduct experiments using a Windows 7 PC with an Intel Core i7-2600 running at 3.4GHz and 16G RAM. The algorithms are implemented with Matlab 2015a.

For our preliminary experiments, we first developed 50 TSP instances, each with 20 cities, including one root. The distances between two cities are uniformly distributed random values

from 1 to 1000. Note that further requiring triangle inequality in distance settings will not change our future computational procedures, but uniformly distributed distances can be used in some practical situations, such as for cases with weighted transportation costs.

For each instance, when computing the Lagrangian subsidized minimum penalty $z_\sigma(0)$ using Algorithm 1, we found that 42 out of 50 instances are balanced; that is $z_\sigma(0) \leq 0$. This result is not surprising, because when the transportation distance is uniformly distributed, players tend to visit less number of arcs, implying the preference of a single Hamiltonian circle including all the players; that is the grand coalition is stable.

To avoid trivial cases and consider more practical instances, we further clustered the cities from several regions whose distances from other cities in different regions are larger than those in the same region. To be more specific, we divided the cities into groups and then multiplied the distances between two cities from different groups with random values ranging from 1 to 5. Intuitively, it is hard to balance regional TSP cases because players from different regions are less willing to accept larger regional travelling costs. The intuition is implied by our computational results where $z_\sigma(0) > 0$ for all the 50 regional TSP instances. Note that regional situations are not rare in TSP problems. For example, retailers always gather at the centre of different cities and request deliveries from a particular manufacturer. In this case, the resulting TSP problem will be regional.

We conduct the following experiments with rooted and regional TSP instances, where the players are partitioned into 4, 5, 6, 7 and 8 regions, respectively, when there are 20, 25, 30, 35 and 40 cities.

First, for some visible examples, we give four representative Lagrangian SPF curves for TSP games in Figure 2. For example, (a) represents that the respective lines passing the two squared points have the same slope, thus there is no breakpoint between them. (c) represents that the two lines passing the two squared points at the beginning and the end intersects a point under the third squared point. Then with x-coordinate value of this intersection point, the third squared point can be obtained by Algorithm 1 (the LRB algorithm). With the IPC algorithm proposed by Liu et al. (2018), we can generate the line passing the third squared point. Finally, this line intersects the two lines passing the two points at the beginning and the end at the second and fourth squared points, respectively. As mentioned in the example, the curves are generated with the IPC algorithm proposed by Liu et al. (2018), and each subsidy–penalty pair on the curves is

computed with Algorithm 1 (the LRB algorithm).

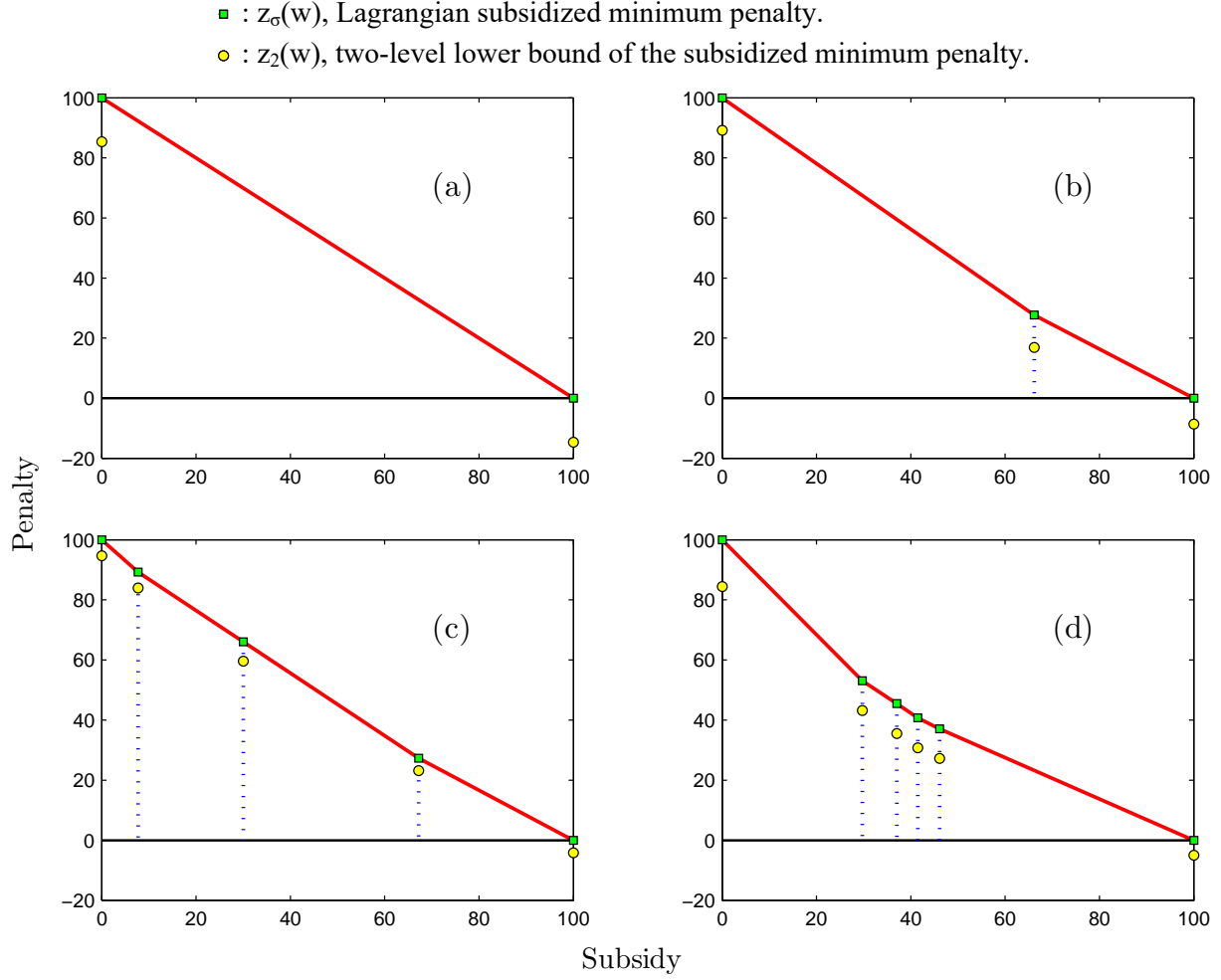


Figure 2: Four representative Lagrangian SPFs for TSP games with 20 cities

In each diagram of Figure 2, the curve represents the Lagrangian SPF $z_\sigma(\omega)$, and the dots below the curve represent the two-level lower bounds $z_2(\omega)$ at the corresponding subsidy levels. For better understanding, all the subsidies and penalties are normalized in percentages of the corresponding Lagrangian minimum subsidy ω_σ and Lagrangian minimum penalty $z_\sigma(0)$, respectively. Here, the Lagrangian minimum subsidy ω_σ is given by $\phi(N) - \alpha_\sigma(N_t)$, where $\phi(N)$ is the best social optimum (an upper bound) of the centralized TSP $c_t(N)$, and α_σ is the best cost allocation (a lower bound) of the optimal cost allocation problem α_t^* . Therefore, the Lagrangian minimum subsidy ω_σ is in fact the upper bound of the minimum subsidy ω_t^* for the TSP game.

From Figure 2, the gap between $z_\sigma(\omega)$ (upper bound of $z(\omega)$) and $z_2(\omega)$ (lower bound of $z(\omega)$) is small. To further show the effectiveness and efficiency of applying the LRB algorithm

to TSP games, we present the average percentage gaps of $z_\sigma(\omega)$ against $z_2(\omega)$ and the average computational time of $z_\sigma(\omega)$ under different game sizes in Table 3.

Table 3: Computational results of the LBR algorithms effectiveness and efficiency

Size	Average Gaps (%)			Time (in seconds)		
	Avg.	Max	Min	Avg.	Max	Min
$n = 20$	6.57	12.12	2.54	224	468	89
$n = 25$	6.32	11.43	3.06	264	879	83
$n = 30$	6.39	12.04	2.13	361	960	98
$n = 35$	6.21	10.78	2.78	440	1319	122
$n = 40$	6.18	11.41	1.87	564	1552	92

For each game size n in Table 3, we average the computational results over 50 randomized instances. We find the LRB algorithm effective for computing Lagrangian subsidized minimum penalties for TSP games. Compared to the two-level lower bounds $z_2(\omega)$, the average gap of $z_\sigma(\omega)$ is around 6.25% with different game sizes. We also observe an increasing tendency of performance of the LRB algorithm implemented on larger sizes of TSP games. Note that the gap of $z_\sigma(\omega)$ could be further narrowed by comparing it with sharper lower bounds $z_k(\omega)$ ($k > 2$), or finding tighter $\phi(N)$ for the cost of grand coalition in the TSP game.

We now check the overall trade-off between the subsidy ω and penalty $z_\sigma(\omega)$ in Figure 3 for TSP games. We again average the results over 50 randomized instances for each game size. For each game instance, we consider five different subsidy levels by setting ω at 0%, 20%, 40%, 60% and 80% of ω_σ respectively, and the penalty levels are normalized in the percentage ratios against $z_\sigma(0)$.

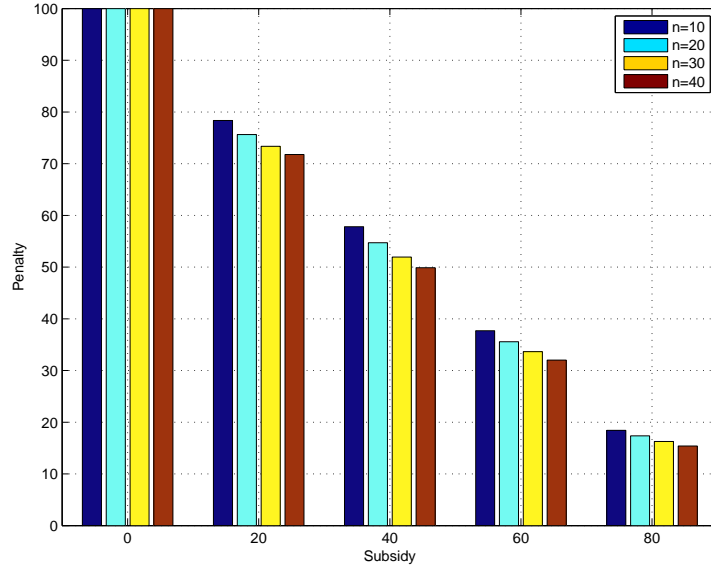


Figure 3: Subsidy efficiency results for different TSP game sizes

In addition to the convexity of the Lagrangian SPF as shown in Remark 1, the numerical results in Figure 3 imply stronger convexity as the game becomes bigger. When n is small (e.g. $n = 10$), the subsidy at different levels are almost equally efficient, but when n is large (e.g. $n = 40$), the additional subsidy is much more efficient at the earlier stages than at the later stages.

5. Conclusion

In this paper, we proposed a new LRB heuristic algorithm to compute feasible subsidy–penalty pairs and stabilize grand coalitions in unbalanced cooperative games. The LRB algorithm is an effective alternative to the existing approaches, such as the CP and LP approaches proposed in Liu et al. (2018), especially for IM games with computationally intractable characteristic functions.

We then applied the LRB algorithm to TSP games and computed the Lagrangian subsidized minimum penalty. By taking good care of the pricing problem, the resulting numerical results confirm that the LRB algorithm is both effective and efficient for TSP games.

The S&P instrument is a novel idea for stabilizing grand coalitions in cooperative game theory. However, to apply it to more practical problems, we should have more effective and efficient algorithms to solve feasible subsidy–penalty pairs. The LRB algorithm is one of such practical algorithms that can lead to interesting results when applied to specific cooperative games.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China [Grant 71701192] and the Fundamental Research Funds for the Central Universities [Grant WK2040160024].

Bondareva, O. N., 1963. Some applications of linear programming methods to the theory of cooperative games. *Problemy Kibernetiki* 10, 119–139.

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

Curiel, I., 2013. Cooperative game theory and applications: cooperative games arising from combinatorial optimization problems. Vol. 16. Springer Science & Business Media.

Dantzig, G., Fulkerson, R., Johnson, S., 1954. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America* 2 (4), 393–410.

Duin, C., Volgenant, A., 1987. Some generalizations of the steiner problem in graphs. *Networks* 17 (3), 353–364.

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

Fisher, M. L., 1981. The lagrangian relaxation method for solving integer programming problems. *Management science* 27 (1), 1–18.

Goemans, M. X., Skutella, M., 2000. Cooperative facility location games. *Journal of Algorithms* 50 (2), 76–85.

Held, M., Karp, R. M., 1971. The traveling-salesman problem and minimum spanning trees: Part ii. *Mathematical Programming* 1 (1), 6–25.

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.

Kimms, A., Kozeletskyi, I., 2016. Core-based cost allocation in the cooperative traveling salesman problem. *European Journal of Operational Research* 248 (3), 910–916.

- Kolen, A., 1983. Solving covering problems and the uncapacitated plant location problem on trees. *European Journal of Operational Research* 12 (3), 266–278.
- Liu, L., Qi, X., Xu, Z., 2016. Computing near-optimal stable cost allocations for cooperative games by Lagrangian relaxation. *INFORMS Journal on Computing* 28 (4), 687–702.
- Liu, L., Qi, X., Xu, Z., 2018. Simultaneous penalization and subsidization for stabilizing grand cooperation. *Operations Research* 66 (5), 1362–1735.
- Ljubić, I., Weiskircher, R., Pferschy, U., Klau, G. W., Mutzel, P., Fischetti, M., 2006. An algorithmic framework for the exact solution of the prize-collecting steiner tree problem. *Mathematical Programming* 105 (2-3), 427–449.
- Maschler, M., Peleg, B., Shapley, L. S., 1979. Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of Operations Research* 4 (4), 303–338.
- Potters, J. A., Curiel, I. J., Tijs, S. H., 1992. Traveling salesman games. *Mathematical Programming* 53 (1-3), 199–211.
- Puerto, J., Tamir, A., Perea, F., 2011. A cooperative location game based on the 1-center location problem. *European Journal of Operational Research* 214 (2), 317–330.
- Schulz, A. S., Uhan, N. A., 2010. Sharing supermodular costs. *Operations Research* 58 (4), 1051–1056.
- Shapley, L. S., Shubik, M., 1969. On market games. *Journal of Economic Theory* 1 (1), 9–25.
- Tamir, A., 1989. On the core of a traveling salesman cost allocation game. *Operations Research Letters* 8 (1), 31–34.
- Woeginger, G. J., 2003. Exact algorithms for np -hard problems: A survey. In: *Combinatorial optimization: Eureka, You Shrink!* Springer, Berlin, Heidelberg., pp. 185–207.

Appendix

Proof of Theorem 1.

The proof is straightforward. Denote an optimal solution for LP (4) as $[\beta_r(\cdot, \omega), z_r(\omega)]$. Now,

we consider vector $[\bar{\beta}_r(\cdot, \omega), z_r(\omega)]$, where $\bar{\beta}_r(k, \omega) = \beta_r(k, \omega) - (c_u(V) - c(V))/v$, for all $k \in V$. Then, for each $S \in \mathbb{S} \setminus V$, we have

$$\bar{\beta}_r(S, \omega) = \beta_r(S, \omega) - \frac{|S|[c_u(V) - c(V)]}{v} \leq \beta_r(S, \omega) \leq c_l(S) + z_r(\omega) \leq c(S) + z_r(\omega).$$

Furthermore, since $\bar{\beta}_r(V, \omega) = \beta_r(V, \omega) - \frac{v[c_u(V) - c(V)]}{v} = c(V) - \omega$, we argue that $[\bar{\beta}_r(\cdot, \omega), z_r(\omega)]$ is feasible to LP (2). Thus, $z_r(\omega)$ is an upper bound of $z(\omega)$. \square

Proof of Remark 1.

From the strong duality of LP (4), we have

$$\begin{aligned} z_r(\omega) &= \max_{\rho} \rho_V [c_u(V) - \omega] - \sum_{S \in \mathbb{S} \setminus \{V\}} \rho_S c_l(S) \\ \text{s.t.} \quad &\sum_{S \in \mathbb{S} \setminus \{V\}} \rho_S = 1, \\ &\sum_{S \in \mathbb{S} \setminus \{V\}: k \in S} \rho_S - \rho_V = 0, \forall k \in V, \\ &\rho_S \geq 0, \forall S \in \mathbb{S}. \end{aligned} \tag{19}$$

In the objective function of LP(19), $\rho_V [c_u(V) - \omega] - \sum_{S \in \mathbb{S} \setminus \{V\}} \rho_S c_l(S)$ can be seen as a line l_ρ which is related to ρ . “A finite set of straight lines” denotes l_ρ with a finite set of ρ . Consequently, $z_r(\omega)$, the maximum of the objective function, is the point-wise maximum of a finite set of straight lines, each with a slope of $-\rho_V$. Thus, $z_r(\omega)$ is a piece-wise linear and convex function ω . Moreover, since the slope $-\rho_V$ of each straight line constructing $z_r(\omega)$ is negative (ρ_V is clearly positive), the R-SPF $z_r(\omega)$ must be strictly decreasing in ω .

From the constraints in LP (19), we have

$$1 = \sum_{S \in \mathbb{S} \setminus \{V\}} \rho_S \leq \sum_{k \in V} \sum_{S \in \mathbb{S} \setminus \{V\}: k \in S} \rho_S \leq (v-1) \sum_{S \in \mathbb{S} \setminus \{V\}} \rho_S = v-1.$$

Note that $\sum_{k \in V} \sum_{S \in \mathbb{S} \setminus \{V\}: k \in S} \rho_S = v\rho_V$; therefore, we have $1 \leq v\rho_V \leq v-1$, that is $-(v-1)/v \leq -\rho_V \leq -1/v$. \square

Proof of Lemma 1.

Denote an optimal solution of (6) as $[\beta_\lambda(\cdot, \omega), z_\lambda(\omega)]$. Now, the vector $[\bar{\beta}_\lambda(\cdot, \omega), z_\lambda(\omega)]$ with $\{\bar{\beta}_\lambda(k, \omega) = \beta_\lambda(k, \omega) - c_{\lambda 1}(k) : \forall k \in V\}$ is clearly a feasible solution for (7) since first, for all $S \in \mathbb{S} \setminus V$,

$$\bar{\beta}_\lambda(S, \omega) - z_\lambda(\omega) = \beta_\lambda(S, \omega) - z_\lambda(\omega) - c_{\lambda 1}(S) \leq c_\lambda(S) - c_{\lambda 1}(S) = c_{\lambda 2}(S),$$

and second,

$$-\bar{\beta}_\lambda(V, \omega) = -\beta_\lambda(V, \omega) + c_{\lambda 1}(V) \leq -c_u(V) + \omega + c_{\lambda 1}(V),$$

where $\sum_{k \in S} \bar{\beta}_\lambda(k, \omega) = \bar{\beta}_\lambda(S, \omega)$ because of the modularity of IM sub-game 1 $(V, c_{\lambda 1})$.

Similarly, given the optimal solution $[\beta_{\lambda 2}(\cdot, \omega), z_\lambda(\omega)]$ of (7), we can show that vector $[\bar{\beta}_{\lambda 2}(\cdot, \omega), z_\lambda(\omega)]$ with $\{\bar{\beta}_{\lambda 2}(k, \omega) = \beta_{\lambda 2}(k, \omega) + c_{\lambda 1}(k) : \forall k \in V\}$ is feasible for (6). Hence, the optimal objective values $z_\lambda(\omega)$ of LPs (7) and (6) are the same. \square

Proof of Lemma 2.

Note that from Theorem 1, $z_\lambda(\omega)$ is an upper bound of $z(\omega)$ for any given ω . In addition, from Remark 1, the LR-SPF $z_\lambda(\omega)$ is convex in ω , implying that $U_\lambda(\omega)$ is an upper bound of LR-SPF $z_\lambda(\omega)$ and SPF $z(\omega)$. \square

Proof of Lemma 3.

The proof is straightforward. As for the proof of Theorem 1, we show that

$$z_l(\omega) = \min_{\beta, z} \{z \in \mathbb{R} : \beta(V) = c_\lambda(V) - \omega \text{ and } \beta(S) \leq c_u(S) + z, \forall S \in \mathbb{S} \setminus V\},$$

is a lower bound of $z(\omega)$. In addition, we argue that $z_k(\omega) \leq z_l(\omega)$, since the latter feasible region is a subset of the former one. Therefore, we have $z_k(\omega) \leq z_l(\omega) \leq z(\omega)$. \square