Submitted to *INFORMS Journal on Computing*
manuscript JOC-2015-04-OA-064.R1

Computing Near-Optimal Stable Cost Allocations for Cooperative Games by Lagrangian Relaxation

Lindong Liu, Xiangtong Qi
Department of Industrial Engineering and Logistics Management
The Hong Kong University of Science and Technology
{ldliu@ust.hk, ieemqi@ust.hk}

Zhou Xu
Department of Logistics and Maritime Studies
Faculty of Business
The Hong Kong Polytechnic University
lgtzx@polyu.edu.hk

For a cost-sharing cooperative game with an empty core, we study the problem of calculating a near-optimal cost allocation that satisfies coalitional stability constraints and maximizes the total cost allocated to all players. One application of such a problem is finding the minimum level of subsidy required to stabilize the grand coalition. To obtain solutions, we propose a new generic framework based on Lagrangian relaxation, which has several advantages over existing work that exclusively relies on linear programming (LP) relaxation techniques. Our approach can generate better cost allocations than LP-based algorithms, and is also applicable to a broader range of problems. To illustrate the efficiency and performance of the Lagrangian relaxation framework, we investigate two different facility location games. The results demonstrate that our new approach can find better cost allocations than the LP-based algorithm, or provide alternative optimal cost allocations for cases that the LP-based algorithm can also solve to optimality.

Key words: game theory, cooperative game, cost allocation, Lagrangian relaxation, facility location game

1. Introduction

Cooperative game theory addresses situations involving collaboration between multiple independent decision makers. It has applications in a variety of areas, such as economics, finance, operations research and telecommunications, to name just a few. In the context of cost reduction, a cooperative game (with transferable utility) can be roughly stated as follows: There are n players, each of whom needs to complete a specific task at minimum cost using a certain resource she owns. Some, or all of the players, may form a coalition by pooling their resources together to jointly work on all their tasks, with the goal being to reduce their total cost. The set of all players is called the grand coalition. The major concern is how to share the total cost of the grand coalition in a “fair” way among all players such that no player has any incentive to quit.

While there are different approaches to defining the “fairness” of a cost allocation, one fundamental concept is coalitional stability, which requires that the cost allocated to each coalition (the sum of the cost allocated to each player in the coalition) is no more than the minimum cost incurred by the coalition if its members do not join the grand coalition. In addition, it is desirable to have a budget balance constraint requiring that the total cost allocated to all players is equal to the minimum cost of the grand coalition. The core of a cooperative game is defined as the set of cost allocations satisfying (1) coalitional stability and (2) the budget balance constraint. The core is not empty if there exists at least one such allocation. When this is the case, the grand coalition is stable. Various conditions and methods have been developed to test the non-emptiness of the core of different cooperative games.

Unfortunately, it is well known that many cooperative games have an empty core. For such games, alternative concepts have been proposed that can be used to motivate a solution. The basic idea is to relax one of the two conditions specified in the definition of core. For example, the concept of *least core* (e.g., Maschler et al. 1979, Kern and Paulusma 2003, Schulz and Uhan 2010, 2013) is defined by relaxing the requirement of coalitional stability. Under the least core concept, the cost allocated to each coalition is limited to no more than a value z plus the minimum cost of the coalition, where z is a parameter to be minimized.

In an alternative concept known as γ-*core* (e.g., Jain and Mahdian 2007), the budget balance constraint is replaced with a γ-budget balance constraint requiring that the total cost allocated to all players is no less than γ times the minimum cost of the grand coalition, where $0 < \gamma \leq 1$. The γ-core is mathematically equivalent to another concept known as the ϵ-*approximate core* (e.g., Faigle et al. 1998, Bläser and Ram 2008) which enforces the budget balance constraint and relaxes coalitional stability constraints such that the total cost allocated to each coalition is no more than $(1+\epsilon)$ times the minimum cost of the coalition. Generally speaking, the main focus of studying the γ-core or ϵ-approximate core has centered on finding a constant bound on γ or ϵ for specific games.

In this paper, we study the idea of γ-core, but with a different focus. Instead of looking for a constant bound on γ, we study the optimal cost allocation problem (OCAP) introduced by Caprara and Letchford (2010), to design an algorithm to exactly calculate the best value of γ for any given instance of the game. Specifically, OCAP tries to maximize the total cost allocated to all players subject to coalitional stability. As pointed out by Caprara and Letchford (2010), this can be viewed equivalently as calculating the “cost” of stabilizing the social optimum under the grand coalition where a third party, representing the social welfare, is willing to subsidize the stability of the grand coalition. Here, the third party may be a government agency, and the players a group of private companies. Alternatively, the third party may be the headquarters of a large corporation, and the players different branches. In such cases, the objective of OCAP is equivalent to minimizing the gap between the total cost allocated to all players and the cost incurred by the grand coalition, where the gap is to be subsidized by the third party.

Roughly speaking, there are at least two difficulties in finding solutions to OCAP. First, as we will show later, the common linear programming (LP) formulation requires an exponential number of constraints in the number of players, n, i.e., 2^n constraints. Second, for a given cost allocation, just to verify that one of the LP constraints is satisfied often involves solving an optimization problem that is itself NP-hard. Hence it is always hard to solve OCAP directly from its LP formulation.

Solving OCAP is a natural solution concept for cooperative games with an empty core. However, it had not been fully studied until recently, when Caprara and Letchford (2010) developed an LP-based framework, referred to herein as the LPB algorithm, to handle games in which the minimum cost of each coalition has an integer linear programming (ILP) formulation. Many games with this property originate from operations research applications. The basic idea is to generate a stable cost allocation that achieves an LP relaxation lower bound on the grand coalition cost. Theoretically speaking, the LPB algorithm can solve OCAP optimally, but doing so requires that all the “assignable” constraints, as defined in Appendix A, are first identified and added to the ILP formulation. However, sometimes it is hard to identify all assignable constraints (e.g., rooted travelling salesman game and vehicle routing game in Caprara and Letchford 2010). Even in a case where all assignable constraints can be identified, such as the unrooted travelling salesman game studied in Caprara and Letchford (2010), the LPB algorithm may still not be applicable if there is no polynomial time separation algorithm over the exponential number of assignable constraints.

In this paper, we propose a new framework to tackle OCAP that is based on Lagrangian relaxation rather than LP relaxation. Our approach, referred to as the LRB algorithm, also tries to find a cost allocation that achieves a lower bound on the grand coalition cost, but has the following advantages compared to the LPB algorithm:

First, it is a generic framework that can be applied to a broader class of cooperative games than those addressed by Caprara and Letchford (2010). Unlike the LPB algorithm, which is restricted to problems with only linear objectives, the new approach is also applicable to problems with non-linear objectives.

Second, the LRB algorithm can find better solutions than the LPB algorithm under the same formulation of the grand coalition problem, due to the well-known fact that the Lagrangian relaxation bound is no worse than the bound provided by the relaxed LP solution. To a certain extent, this avoids the requirement of identifying all “assignable” constraints in the LPB algorithm. In addition, even for certain cases where it is easy to find all assignable constraints to guarantee the optimality of the LPB algorithm, the LRB algorithm is still valuable because it can offer alternative optimal cost allocations to the players and hence, more choices for evaluation.

Third, in solving the OCAP our algorithm takes a decomposition approach and creates two sub-games, both of which are relatively easier to solve than the original game. One sub-game has a simple optimal solution which can be represented in closed form. The other sub-game has some beneficial properties that the original game lacks. In many cases, the minimum cost incurred by each coalition in the sub-games is relatively easier to calculate than in the original game. In some cases, the optimal cost allocations of the sub-games are polynomially solvable, though that of the original game is not.

Finally, the LRB algorithm rests on a large amount of research developed over decades to efficiently solve the Lagrangian dual. In applying it to the OCAP, we can take advantage of various techniques that have been developed to speed up convergence and produce sharper bounds. Such results can be incorporated in the LRB algorithm in its first step.

2. Literature Review

The research on cooperative games has been extensive ever since the seminal work of Shapley (1952). Some of the most prominent examples related to operations research applications include assignment games (Shapley and Shubik 1971, Martínez-de Albéniz et al. 2013), bin packing games (Faigle and Kern 1993, Liu 2009), linear production games (Owen 1975), minimum spanning tree games (Granot and Huberman 1981), travelling salesman games (Tamir 1989, Potters et al. 1992), vehicle routing games (Göthe-Lundgren et al. 1996, Engevall et al. 2004), inventory games (Hartman et al. 2000, Chen 2009, Chen and Zhang 2009, He et al. 2012, Zhang 2009), production outsourcing games (Aydinliyim and Vairaktarakis 2010, Cai and Vairaktarakis 2012), and some packing and covering games on graphs (Deng et al. 1999), to name just a few. The major interest in studying these games is usually the existence of the core.

Since this paper uses facility location games for illustration purposes, we now discuss the work on cooperative facility location games. An early important result is given by Kolen (1983), who showed that for uncapacitated facility location games, the maximum shared cost among players is equal to the classic LP relaxation cost for the grand coalition optimization problem. Later, Goemans and Skutella (2000) extended this result, and proved non-emptiness of the core for several special facility location games where facility locations are on a line, a cycle, and a tree. Others have also studied variants of the problem. For example, Puerto et al. (2011, 2012) introduced the minimum radius location game and the minimum diameter location game, respectively. Xu and Yang (2009), Mallozzi (2011) and Li et al. (2013) studied facility location games with various cost components, such as service installation costs, regional fixed costs, and concave facility location costs.

As previously mentioned, the least core is one type of relaxation that can be used to deal with games with empty cores. Faigle et al. (2000) showed that computing the least core allocation for the minimum spanning tree game is NP-hard. Kern and Paulusma (2003) studied the nucleolus based on a polynomial description of the least core for the cardinality matching game. Schulz and Uhan (2010) showed that computing the least core value for a single-machine scheduling game with supermodular cost is weakly NP-hard, and provided a framework in Schulz and Uhan (2013) for a 3-approximation algorithm for computing the least core value. With respect to the γ-core and ϵ-approximate core, there has been more research using the latter concept. For example, Faigle et al. (1998) developed an LP-based algorithm that generates a $\frac{1}{3}$-approximate core for the Euclidean TSP game. Bläser and Ram (2008) provided a polynomial time algorithm that finds cost allocations lying in a $(\log_2(n-1)-1)$-approximate core for the asymmetric TSP game. We refer the reader to Jain and Mahdian (2007) for a more comprehensive review on the above concepts.

Only a few papers directly address the OCAP. Bachrach et al. (2009) raised the problem of stabilizing the grand coalition of a cooperative game at a value that minimizes the subsidy, and derived appropriate upper and lower bounds for the minimum value. Following Bachrach et al. (2009), a similar study on restricted cooperation in coalitional games was undertaken by Meir et al. (2011). The only algorithmic work on solving OCAP is by Caprara and Letchford (2010), who proposed a comprehensive framework for optimally solving a large class of problems based on LP relaxation and duality theory. In their approach, it is usually necessary to re-formulate the optimization problems by introducing constraints with special structures. The details are presented in the next section.

3. Preliminaries

A cooperative game with transferable utilities is described by a pair (V, c), where $V = \{1, 2, ..., v\}$ denotes a set of players, and $c : S \to \mathbb{R}$ denotes the characteristic function, with $S = 2^V \setminus \emptyset$ indicating the set of non-empty coalitions of players. The characteristic function assigns to every coalition $s \in S$ a value $c(s)$, representing the minimum total cost that the members in s need to pay when they cooperate. The problem of cost allocation studied here is to share the grand coalition cost $c(V)$ among the players in V in such a way that for any smaller coalition s of players, there is no incentive for them to break away from the grand coalition and form their own coalition.

A (coalitional) stable cost allocation for a game (V, c) is a vector $\alpha \in \mathbb{R}^v$ which satisfies coalitional stability: $\sum_{k\in s} \alpha(k) \le c(s)$, $\forall s \in S$. An ideal cost allocation additionally satisfies the budget balance constraint: $\sum_{k\in V} \alpha(k) = c(V)$. The core of game (V, c) is defined as:

$$Core(V, c) = \{\alpha \in \mathbb{R}^v : \sum_{k\in s} \alpha(k) \le c(s), \forall s \in S, \text{ and } \sum_{k\in V} \alpha(k) = c(V)\}.$$

It is known that not every game (V,c) has a non-empty core. To address games with an empty core, our goal is to find a stable cost allocation that covers the grand coalition cost $c(V)$ as much as possible, which motivates the Optimal Cost Allocation Problem ($OCAP$), defined as:

$$\begin{aligned} &\max_{\alpha} \sum_{k\in V} \alpha(k) \\ s.t. &\sum_{k\in s} \alpha(k) \le c(s),\ \forall s\in S. \end{aligned} \tag{1}$$

Solving (1) directly is often intractable, because it consists of an exponential number of constraints, and the values of characteristic functions $c(s)$ can even be NP-hard to compute. The focus of this paper is to compute good stable cost allocations for a class of games called Operations Research (OR) games whose core may be empty and whose characteristic functions are defined by an integer program (IP). The OR game is a generalization of the Integer Minimization (IM) game investigated in Caprara and Letchford (2010). Unlike the IM game, which can only use integer linear programming (ILP) to define characteristic functions, the OR game allows using non-linear integer programming to define characteristic functions. For easy comparison, our formulation follows Caprara and Letchford (2010) as much as possible.

Definition 1. A cooperative game (V,c) is called an Operations Research game or OR game if there exist

- positive integers r, r' and t,
- left hand side matrices $A\in\mathbb{R}^{r\times t}$ and $A'\in\mathbb{R}^{r'\times t}$,
- right hand side matrices $B\in\mathbb{R}^{r\times v}$ and $B'\in\mathbb{R}^{r'\times v}$,
- non-negative right hand side column vectors $D\in\mathbb{R}^{r}$ and $D'\in\mathbb{R}^{r'}$,
- an objective function $f(x)$ which can be either linear or nonlinear in x, and
- an incidence column vector $\gamma^s\in\{0,1\}^v$ with $\gamma^s_k=1$ if $k\in s$ and $\gamma^s_k=0$ otherwise, $\forall k\in V$,

such that for all $s\in S$, the characteristic function $c(s)$ is given by the following IP:

$$c(s)=\min_x \left\{f(x): Ax\ge B\gamma^s+D, A'x\ge B'\gamma^s+D', x\in\{0,1\}^{t\times 1}\right\}. \tag{2}$$

Note that in (2), the constraints are partitioned into two parts to facilitate the use of Lagrangian relaxation later. Following Caprara and Letchford (2010), it is easy to show that every OR game with non-negative D and D' is sub-additive, i.e., $c(s_1\cup s_2)\le c(s_1)+c(s_2)$, for all $s_1,s_2\in S$ with $s_1\cap s_2=\emptyset$. This defines a proper game where it makes sense for the players to cooperate, and such games are the focus of our analyses in this paper.

For an IM game, a special case of the OR game, its characteristic function $c(s)$ is given by ILP

$$c(s)=\min_x \left\{Cx: Ax\ge B\gamma^s+D, A'x\ge B'\gamma^s+D', x\in\{0,1\}^{t\times 1}\right\}, \tag{3}$$

where C is a row vector of dimension t. We use $c_{LP}(V)$ to denote the LP lower bound of $c(V)$ defined in (3), where $x \in \{0,1\}^{t\times 1}$ is relaxed to $\mathbf{0} \le x \le \mathbf{1}$.

Caprara and Letchford (2010) explained how OCAP for IM games can be solved by using column generation, row generation, or both. To make the paper self-contained, we summarize a few highlights of these methods in Appendix A. Roughly speaking, the column generation approach has a straightforward formulation; however, the associated pricing problem is usually difficult to handle because it needs to do optimization over a very large solution space. The row generation approach, which is more promising, needs to re-formulate ILP (3) by identifying a set of so-called assignable constraints $\{Ex \ge F\gamma\}$. Then a cost allocation can be obtained by solving the LP relaxation $c_{LP}^{EF}(V) = \min\{Cx : Ex \ge F\gamma\}$ with only assignable constraints, and the total cost allocated is equal to $c_{LP}^{EF}(V)$, a lower bound of $c(V)$. Note that $c_{LP}^{EF}(V)$ might be different from $c_{LP}(V)$, the LP lower bound of $c(V)$ under the original formulation (3).

For IM games, the quality of the LPB cost allocation greatly depends on the assignable constraints that have been identified. Theoretically speaking, the LPB algorithm can find the optimal stable cost allocation for an IM game if all assignable constraints can be identified and added. However, for different IM games, the ways of identifying assignable constraints are often different, as no general approaches are known. For some assignable constraints, no polynomial separation algorithms are known, raising another difficulty in computing the LPB cost allocations. Despite these difficulties, the LPB algorithm can be used as an effective heuristic to find good stable cost allocations when only a subset of assignable constraints are added.

4. Lagrangian Relaxation Based Cost Allocation Algorithm

In this section, we present our LRB cost allocation algorithm. In general, for an OR game (V, c), when the objective function $f(x)$ in (2) is linear in x, both LPB and LRB algorithms are possible alternatives in computing good stable cost allocations, but only the LRB algorithm can be used if $f(x)$ is nonlinear. We first explain the framework of the LRB algorithm and show its effectiveness, and then give more details of the algorithm implementation.

4.1. Lagrangian Relaxation and Game Decomposition

In a Lagrangian relaxation procedure, by relaxing constraints $\{A'x \ge B'\gamma^s + D'\}$ in (2) and bringing them into the objective function with non-negative Lagrangian multiplier λ, we can derive the resulting Lagrangian characteristic function $c_{LR}(\,\cdot\,;\lambda)$ for an OR game as

$$c_{LR}(s;\lambda) = \min_x \left\{ f(x) - \lambda A'x + \lambda B'\gamma^s + \lambda D' : Ax \ge B\gamma^s + D, x \in \{0,1\}^{t\times 1} \right\}, \forall s \in S, \tag{4}$$

where λ is a non-negative row vector of dimension r', i.e., $\lambda \in \mathbb{R}_+^{1\times r'}$. In particular, for the grand coalition V, its Lagrangian characteristic function is

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

As is typical with Lagrangian relaxation, constraints $\{A'x \geq B'\gamma^s + D'\}$ can be carefully chosen such that $c_{LR}(s;\lambda)$ is relatively easy to solve, e.g., in polynomial or pseudo-polynomial time for any $s \in S$. It is known that $c_{LR}(V;\lambda)$ is a lower bound of $c(V)$ for any non-negative λ. To achieve the sharpest lower bound, the Lagrangian dual problem $d_{LR}(V)$ finds the best Lagrangian multiplier λ that maximizes $c_{LR}(V;\lambda)$, i.e.,

$$d_{LR}(V) = \max_{\lambda} \big\{ \min_{x} \big\{ f(x) - \lambda A'x + \lambda B'\mathbf{1} + \lambda D' : Ax \geq B\mathbf{1} + D, x \in \{0,1\}^{t\times 1} \big\} : \lambda \geq \mathbf{0} \big\}. \tag{5}$$

By the subgradient method (e.g., see Ahuja et al. 1993), we can compute the optimal Lagrangian multiplier λ^* for $d_{LR}(V)$.

Under any non-negative Lagrangian multiplier λ, we can decompose the Lagrangian characteristic function $c_{LR}(\,\cdot\,;\lambda)$ into two sub-characteristic functions $c_{LR1}(\,\cdot\,;\lambda)$ and $c_{LR2}(\,\cdot\,;\lambda)$, such that $c_{LR}(s;\lambda) = c_{LR1}(s;\lambda) + c_{LR2}(s;\lambda)$, for any $s \in S$, where

$$c_{LR1}(s;\lambda) = \lambda B'\gamma^s, \text{ and} \tag{6}$$

$$c_{LR2}(s;\lambda) = \min_{x} \big\{ f(x) - \lambda A'x + \lambda D' : Ax \geq B\gamma^s + D, x \in \{0,1\}^{t\times 1} \big\}. \tag{7}$$

We then define sub-game 1, denoted by $\big(V, c_{LR1}(\,\cdot\,;\lambda)\big)$, with its characteristic function being $c_{LR1}(s;\lambda)$, and sub-game 2 $\big(V, c_{LR2}(\,\cdot\,;\lambda)\big)$ similarly. In some specific implementations, we can further decompose $c_{LR2}(\,\cdot\,;\lambda)$ into more sub-characteristic functions in order to make the LRB algorithm more efficient. One example is given in Section 5.2.1.

Theorem 1. *Given any non-negative Lagrangian multiplier λ, if α^{λ}_{LR1} and α^{λ}_{LR2} are stable cost allocations for sub-games $\big(V, c_{LR1}(\,\cdot\,;\lambda)\big)$ and $\big(V, c_{LR2}(\,\cdot\,;\lambda)\big)$, respectively, then $\alpha^{\lambda}_{LR} = \alpha^{\lambda}_{LR1} + \alpha^{\lambda}_{LR2}$ is a stable cost allocation for OR game (V,c).*

Proof. For any $s \in S$, the stability of α^{λ}_{LR1} and α^{λ}_{LR2} implies that

$$\sum_{k \in s} \big[\alpha^{\lambda}_{LR1}(k) + \alpha^{\lambda}_{LR1}(k)\big] \leq c_{LR1}(s;\lambda) + c_{LR2}(s;\lambda) = c_{LR}(s;\lambda) \leq c(s).$$

Therefore, we have $\sum_{k\in s} \alpha^{\lambda}_{LR}(k) = \sum_{k\in s} \big[\alpha^{\lambda}_{LR1}(k) + \alpha^{\lambda}_{LR1}(k)\big] \leq c(s)$. This completes the proof. □

By Theorem 1, we can design an LRB algorithm to obtain a good stable solution to OCAP by finding a good stable cost allocation for each of the sub-games. In fact, we are able to find an optimal stable cost allocation for each sub-game under a specific λ. Before introducing the details of handling the two sub-games, we first have the following result regarding the effectiveness of the LRB algorithm compared to the LPB algorithm. To be specific, for an IM game, different ILP formulations of the characteristic function might lead to different LPB and LRB cost allocations. Theorem 2 gives a sufficient condition for the LRB cost allocation algorithm to be better than or as good as the LPB one.

Theorem 2. *For an IM game, under the same ILP formulation for the characteristic function $c(s)$, the LRB cost allocation value $\sum_{k\in V}\alpha_{LR}^{\lambda}(k)$ is no less than the LPB cost allocation value $\sum_{k\in V}\alpha_{LP}(k)$, when the following two conditions hold:* (1) *the Lagrangian multiplier is optimal for $d_{LR}(V)$, i.e., $\lambda=\lambda^*$, and* (2) *$\alpha_{LR1}^{\lambda^*}$ and $\alpha_{LR2}^{\lambda^*}$ lie in the core of the sub-games $\big(V, c_{LR1}(\,\cdot\,;\lambda^*)\big)$ and $\big(V, c_{LR2}(\,\cdot\,;\lambda^*)\big)$, respectively.*

Proof. It is well known that, for the same formulation, the lower bound obtained by Lagrangian relaxation is no worse than that obtained by LP relaxation (e.g., see Ahuja et al. 1993), i.e., $c_{LR}(V;\lambda^*)\geq c_{LP}(V)$. From condition (2), we have

$$\sum_{k\in V}\alpha_{LR}^{\lambda^*}(k)=\sum_{k\in V}\left[\alpha_{LR1}^{\lambda^*}(k)+\alpha_{LR2}^{\lambda^*}(k)\right]=c_{LR1}(V;\lambda^*)+c_{LR2}(V;\lambda^*)=c_{LR}(V;\lambda^*).$$

In addition, the LPB cost allocation value $\sum_{k\in V}\alpha_{LP}(k)$ is clearly no larger than the LP lower bound $c_{LP}(V)$. Therefore, we have $\sum_{k\in V}\alpha_{LR}^{\lambda^*}(k)=c_{LR}(V;\lambda^*)\geq c_{LP}(V)\geq\sum_{k\in V}\alpha_{LP}(k)$. □

We have the following remarks about Theorem 2:

Remark 1. Theorem 2 implies the competitiveness of the LRB algorithm against the LPB algorithm. Both LPB and LRB cost allocations can be improved by adding more constraints to the conventional ILP formulation of characteristic function $c(s)$. However, as shown in Theorem 2, when applied to the same ILP formulation of $c(s)$, the LRB cost allocation value is no smaller than the LPB one if both conditions (1) and (2) hold. Even if the LPB algorithm can be proven optimal for some cases, our LRB algorithm can still be valuable in that it can provide alternative cost allocations with different features. An example of this is given in Section 5.1.3.

Remark 2. The conditions given in Theorem 2 to ensure that the LRB cost allocation value $\sum_{k\in V}\alpha_{LR}^{\lambda}$ is no less than the LPB cost allocation value $\sum_{k\in V}\alpha_{LP}^{\lambda}$ are sufficient, but not necessary. In fact, even without condition (1), for a non-optimal Lagrangian multiplier $\bar{\lambda}$, as long as $c_{LR}(V;\bar{\lambda})\geq c_{LP}(V)$, the result still holds.

Remark 3. Condition (2) requires that both sub-games have a non-empty core. As will be shown in the following Lemma 1, sub-game 1 $\big(V, c_{LR1}(\,\cdot\,;\lambda)\big)$ always has a non-empty core. However, sub-game 2 $\big(V, c_{LR2}(\,\cdot\,;\lambda)\big)$ may have an empty core. Therefore, the effectiveness of the LRB algorithm depends on sub-game 2, specifically on the value of $\sum_{k\in V}\alpha_{LR2}^{\lambda}(k)$, the cost that can be allocated for game $\big(V, c_{LR2}(\,\cdot\,;\lambda)\big)$. Our numerical results show that, even in cases where sub-game 2 does not have a non-empty core, the LRB algorithm can still be competitive. Examples are given in Section 5.2.2.

Remark 4. In the situation where $c(V)$ is relaxed in such a way that $d_{LR}(V)$ has the integrality property, i.e., $d_{LR}(V)$ is not increased by removing the integrality restriction on x from the constraints of the Lagrangian problem (see, Geoffrion 1974), the Lagrangian lower bound

$c_{LR}(V;\lambda^*)$ is equal to the LP lower bound $c_{LP}(V)$. In addition, if all the constraints in $c_{LP}(V)$ are assignable, then our LRB algorithm cannot obtain a solution better than the LPB algorithm, since $\sum_{k\in V}\alpha_{LP}(k)=c_{LP}(V)=c_{LR}(V;\lambda^*)\geq\sum_{k\in V}\alpha_{LR}^{\lambda^*}(k)$.

The above Remark 3 is also related to the convergence of the LRB algorithm when sub-game 2 may have an empty core. In such a case, we first need a tight Lagrangian lower bound $c_{LR2}(V;\lambda)$ which can be achieved by a large number of iterations in solving the Lagrangian dual problem; however, there is no guarantee that a higher bound $c_{LR2}(V;\lambda)$ will also lead to a higher cost allocation value $\sum_{k\in V}\alpha_{LR2}^{\lambda}(k)$. In other words, the final optimal Lagrangian multiplier λ^* may not necessarily correspond to the best cost allocation. Computational examples are shown in Section 5.2.2. As such, we propose the following algorithm:

Algorithm 1. The LRB cost allocation algorithm for an OR game (V,c).

Step 1: For problem (2), design a Lagrangian relaxation as shown in (4). Compute the optimal solution λ^* for the Lagrangian dual problem $d_{LR}(V)$ defined by (5) with the subgradient method, where we save a set of Lagrangian multipliers $\Lambda=\{\lambda_1,\lambda_2,\dots,\lambda_p\}$ at some intermediate iterations.

Step 2: For each $\lambda\in\Lambda$, decompose the Lagrangian characteristic function $c_{LR}(\,\cdot\,;\lambda)$ into two sub-characteristic functions $c_{LR1}(\,\cdot\,;\lambda)$ and $c_{LR2}(\,\cdot\,;\lambda)$ according to (6) and (7).

Step 3: Compute the optimal stable cost allocations α_{LR1}^{λ} and α_{LR2}^{λ} for sub-games $\big(V,c_{LR1}(\,\cdot\,;\lambda)\big)$ and $\big(V,c_{LR2}(\,\cdot\,;\lambda)\big)$, respectively. Details are given by Lemmas 1 and 2 in the next two subsections.

Step 4: Let $\alpha_{LR}^{\lambda}=\alpha_{LR1}^{\lambda}+\alpha_{LR2}^{\lambda}$ be a stable cost allocation for game (V,c). Find the best stable cost allocation with the maximal shared cost among α_{LR}^{λ} for $\lambda\in\Lambda$.

Note that if sub-game 2 has a non-empty core for the optimal Lagrangian multiplier λ^*, we only need to use the final Lagrangian multiplier λ^* in Λ to calculate a cost allocation. Otherwise, we can consider using more Lagrangian multipliers, due to the above discussion prior to Algorithm 1. While there is no theoretical study on choosing the λ values, our experience shows that it is useful to select five or six values from different iterations.

4.2. Solving Sub-Game 1

We now show how to calculate a cost allocation in the core of sub-game 1 $\big(V,c_{LR1}(\,\cdot\,;\lambda)\big)$. In this game, any player $k\in V$ will induce a cost $(\lambda B')_k$, which represents the value of the kth element in vector $\lambda B'$. Accordingly, we have

Lemma 1. *For sub-game 1 $\big(V,c_{LR1}(\,\cdot\,;\lambda)\big)$, a vector α_{LR1}^{λ} defined by equalities $\{\alpha_{LR1}^{\lambda}(k)=(\lambda B')_k:\forall k\in V\}$ lies in its core.*

Proof. The proof is straightforward: for any coalition $s\in S$, by the definition of game $\big(V,c_{LR1}(\,\cdot\,;\lambda)\big)$, the total induced cost will be $c_{LR1}(s;\lambda)=\sum_{k\in s}(\lambda B')_k$ which is equal to the total cost allocated to s under cost allocation $\alpha_{LR1}^{\lambda}(s)$. □

Lemma 1 shows that we can directly compute a core cost allocation for game $\big(V,c_{LR1}(\,\cdot\,;\lambda)\big)$.

4.3. Solving Sub-Game 2

Different from sub-game 1, sub-game 2 $\big(V, c_{LR2}(\,\cdot\,;\lambda)\big)$ may have an empty core. So we aim to calculate the optimal stable cost allocation, which lies in the core if it is non-empty.

We first provide a generic Column Generation Based (CGB) algorithm (see Barnhart et al. 1998 for an introduction on column generation) to compute the optimal stable cost allocation for game $\big(V, c_{LR2}(\,\cdot\,;\lambda)\big)$ in general cases. Note that the CGB algorithm here is more computationally tractable than the one described in Caprara and Letchford (2010). Specifically, we propose treating each coalition as a column in the master problem defined by the following LP (9), but Caprara and Letchford (2010) suggests re-formulating the master problem by treating each feasible solution of the optimization problem as a column (see LP (36) in Appendix A for details). The re-formulation is necessary there to avoid the strong NP-hardness of evaluating $c(s)$ for each coalition s, but it makes the search space much larger in column generation. We can directly treat each coalition as a column, since compared with $c(s)$, $c_{LR2}(s;\lambda)$ is much easier to solve after Lagrangian relaxation.

Recall that an optimal stable cost allocation α^{λ}_{LR2} for sub-game 2 is the optimal solution to the corresponding OCAP,

$$\begin{aligned} &\max_{\alpha^{\lambda}_{LR2}} \sum_{k\in V} \alpha^{\lambda}_{LR2}(k) \\ s.t.\quad &\sum_{k\in s} \alpha^{\lambda}_{LR2}(k) \le c_{LR2}(s;\lambda),\ \forall s\in S. \end{aligned} \tag{8}$$

Under the assumption that the Lagrangian relaxations will be easy to solve, $c_{LR2}(s;\lambda)$ would be computationally tractable for all $s\in S$.

To solve the OCAP in (8), consider its dual problem as follows:

$$\begin{aligned} &\min_{\beta} \sum_{s\in S} c_{LR2}(s;\lambda)\beta_s \\ s.t.\quad &\sum_{s\in S} \gamma^s_k \beta_s = 1,\quad \forall k\in V, \\ &\beta_s \ge 0,\quad \forall s\in S, \end{aligned} \tag{9}$$

where $\{\beta_s : \forall s\in S\} \in \mathbb{R}^{(2^v-1)\times 1}$ are decision variables. Based on the strong duality we know that the OCAP in (8) is equivalent to the LP in (9). By following the standard column generation framework, we propose the following Algorithm 2 that can solve (9) to optimality and generate an optimal cost allocation for sub-game 2.

Algorithm 2. A CGB algorithm to compute the optimal stable cost allocation α^{λ}_{LR2} for game $\big(V, c_{LR2}(\,\cdot\,;\lambda)\big)$ includes the following four steps:

Step 1. Start from a restricted master problem of (9) where the restricted coalition set $S'\subset S$ contains a polynomial number of elements, and compute its optimal dual solution π^*.

Step 2. Find an optimal coalition s^* to the pricing problem,

$$\min_{s\in S\setminus S'} \big\{c_{LR2}(s;\lambda) - \sum_{k\in V}\gamma_k^s\pi_k^*\big\}. \tag{10}$$

Step 3. If there exists an s^* with a negative value of (10), then add s^* into S', and go back to step 1; otherwise, the dual problem (9) is solved to optimality, so go to step 4.

Step 4. According to the updated restricted coalition family S' and its corresponding characteristic function values $\{c_{LR2}(s;\lambda):\forall s\in S'\}$, the following LP gives the optimal stable cost allocation α_{LR2}^{λ} for game $\big(V, c_{LR2}(\,\cdot\,;\lambda)\big)$:

$$\begin{aligned}&\max_{\alpha_{LR2}^{\lambda}}\sum_{k\in V}\alpha_{LR2}^{\lambda}(k)\\ s.t.\ &\sum_{k\in s}\alpha_{LR2}^{\lambda}(k)\le c_{LR2}(s,\lambda),\ \ \forall s\in S'.\end{aligned}$$

In the *CGB* algorithm, the most difficult and essential part is solving the pricing problem (10), which is specific to each game. We will give an example of this in Section 5.2.

To conclude, we present the following lemma on the CGB algorithm:

Lemma 2. *Vector α_{LR2}^{λ} computed by the CGB algorithm is an optimal stable cost allocation for sub-game 2 $\big(V, c_{LR2}(\,\cdot\,;\lambda)\big)$.*

The CGB algorithm above provides a general approach to computing an optimal stable cost allocation for sub-game 2. However, for some special cases, it is possible to obtain an optimal or core cost allocation for sub-game 2 by using certain simpler or faster approaches. For example, we consider the following two special cases:

First, if the LP relaxation of $c_{LR2}(V;\lambda)$ contains the complete assignable constraint set, one can apply the LPB method proposed by Caprara and Letchford (2010) to compute an optimal cost allocation for $\big(V, c_{LR2}(\,\cdot\,;\lambda)\big)$.

Second, when sub-game 2 turns out to be submodular, we can solve a core cost allocation by a greedy algorithm, i.e., by sorting the players and computing the marginal cost vector (see, Edmonds 1970 and Shapley 1971).

Definition 2. Denote a and b as two players in the grand coalition V. A characteristic function $c_{LR2}(\,\cdot\,;\lambda)$ is submodular if $c_{LR2}(s\cup\{a\};\lambda)+c_{LR2}(s\cup\{b\};\lambda)\ge c_{LR2}(s;\lambda)+c_{LR2}(s\cup\{a,b\};\lambda)$ holds for any coalition $s\in V\setminus\{a,b\}$.

We emphasize these two cases because sub-game 2 may include all assignable constraints or have the submodularity property even though the original game (V,c) does not. This is actually a potential advantage of the LRB algorithm, and one such example is given in Section 5.1. However, it often requires sophisticated analysis to detect the completeness of assignable constraints or to prove the submodularity of sub-game 2.

5. Implementations for Facility Location Games

We will illustrate the LRB cost allocation algorithm on two different facility location games, namely, the Uncapacitated Facility Location (UFL) game and the Non-Linear single source Capacitated Facility Location (NLCFL) game. The UFL game represents the case where both LPB and LRB algorithms can be used to compute optimal cost allocations; in addition, the resulting UFL subgame 2 is submodular, and its core cost allocation can be obtained in polynomial time. The NLCFL game has a different coalition definition and nonlinear cost function, showing the full power of our LRB algorithm.

5.1. The UFL Game

In a UFL game, there is a bipartite network defined by $G=(M,N,E)$, with M being the set of potential sites where facilities can be opened, N being the set of customer points that must be served, and E being the set of edges which link the facility sites and customer points. Each potential site $i\in M$ has a fixed opening cost f_i, and each edge $(i,j)\in E$ has a transportation cost c_{ij}. In a UFL game, the customers share the facility opening and transportation costs, i.e., the players in the game are customers. We list the notation used in the UFL game in Table 1.

Table 1 Notation used in the UFL game

M	The set of potential facility sites, $M=\{1,2,...,m\}$.
N	The set of customer points as well as game players, $N=\{1,2,...,n\}$.
c_{ij}	Transportation cost from facility i to customer j, $\forall i\in M, j\in N$.
f_i	Fixed opening cost of facility i, $\forall i\in M$.
s	Player coalition, $s\subseteq N$.
γ^s	Incidence vector $\left[\gamma^s_1,\gamma^s_2,...,\gamma^s_n\right]^T$, where $\gamma^s_j=1$ if player j is in coalition s and $\gamma^s_j=0$ otherwise.
v_i	Decision variable, where $v_i=1$ if facility i will be opened and $v_i=0$ otherwise, $\forall i\in M$.
u_{ij}	Decision variable, where $u_{ij}=1$ if customer j will be served by facility i and $u_{ij}=0$ otherwise, $\forall i\in M$ and $j\in N$.

Definition 3. A UFL game (N, c_{UFL}) is defined with the players being the customers in N and the characteristic function $c_{UFL}(s)$ determined by the following ILP,

$$c_{UFL}(s)=\min_{v,u}\sum_{i\in M} f_i v_i+\sum_{i\in M}\sum_{j\in N} c_{ij}u_{ij} \tag{11}$$

$$s.t.\ \sum_{i\in M} u_{ij}\geq \gamma^s_j,\ \forall j\in N, \tag{12}$$

$$u_{ij}-v_i\leq 0,\ \forall i\in M, j\in N, \tag{13}$$

$$v_i, u_{ij}\in\{0,1\},\ \forall i\in M, j\in N. \tag{14}$$

In the above ILP, the objective function (11) is to minimize the total facility opening and transportation cost for a coalition s; constraints (12) require that every customer in coalition s must be served, and constraints (13) ensure that only an opened facility can serve customers.

ILP (11)-(14) is the conventional formulation for an uncapacitated facility location problem. In view of Definition 1, we see that the UFL game (N, c_{UFL}) is an OR game (V, c) with $V = N$ and $c = c_{UFL}$. Specifically, decision variables x in c are now $[v; u]$ in c_{UFL}, and the specific expressions of matrices C, A, A', B, B', D, D' can be obtained by writing c_{UFL} using matrices. In particular, both D and D' are now $\mathbf{0}$, so the game (N, c_{UFL}) is sub-additive. This is also true for the NLCFL game which we are going to study in Section 5.2.

5.1.1. LPB Cost Allocation for the UFL Game Kolen (1983) and Goemans and Skutella (2000) proved that, for a UFL game, the maximum stable cost allocation value coincides with the LP lower bound of $c_{UFL}(N)$. To perform some in-depth analysis, we give more details on using the LPB algorithm to calculate the optimal stable cost allocation.

In $c_{UFL}(s)$, constraints (12) and (13) are already assignable. By adding assignable constraints $\{u_{ij} \geq 0 : i \in M, j \in N\}$ to relax the binary constraints (14), we can obtain an LP relaxation for the grand coalition optimization problem $c_{UFL}(N)$ as follows:

$$c_{LP\_UFL}(N) = \min_{v,u} \sum_{i \in M} f_i v_i + \sum_{i \in M} \sum_{j \in N} c_{ij} u_{ij}$$

$$s.t. \sum_{i \in M} u_{ij} \geq \gamma_j^N, \ \forall j \in N, \tag{15}$$

$$v_i - u_{ij} \geq 0, \ \forall i \in M, j \in N, \tag{16}$$

$$u_{ij} \geq 0, \ \forall i \in M, j \in N. \tag{17}$$

We sequentially label constraints (15), (16) and (17) from 1 to n, $n+1$ to $n+mn$ and $n+mn+1$ to $n+2mn$, respectively. For $c_{LP\_UFL}(N)$, we consider its dual LP. Let μ_k be the dual variable corresponding to the k-th constraint of $c_{LP\_UFL}(N)$, and μ^* an optimal solution to the dual LP. According to the Row Generation Approach in Appendix A, we have the following lemma:

Lemma 3. *For a UFL game, the LPB cost allocation α_{LP_UFL} given by*

$$\alpha_{LP\_UFL}(j) = \mu_j^*, \ \forall j \in \{1, 2, \ldots, n\},$$

is optimal, with total shared cost $c_{LP\_UFL}(N)$.

We note that a simpler way to obtain one optimal stable cost allocation to the UFL game is to solve $c_{LP\_UFL}(N)$ directly and obtain the optimal dual variables by calculating the shadow prices of the constraints. However, solving the dual LP facilitates finding alternative optimal solutions, because in the event that the dual LP has multiple optimal solutions, not all of them correspond to a shadow price of the primal.

5.1.2. LRB Cost Allocation for the UFL Game We next demonstrate how to apply the LRB algorithm to obtain optimal cost allocations for the UFL game. We will prove that the sub-game 2 of the UFL game is submodular. We will also show by a computational study that the optimal cost allocation obtained by the LRB algorithm for this game can be different from those obtained by the LPB algorithm, thus giving more choices for evaluation and comparison.

In $c_{UFL}(s)$, we add a set of new constraints

$$\big\{u_{ij} \le \gamma_j^s : \ \forall i \in M, j \in N \ \big\}, \tag{18}$$

and then bring constraints $\{\sum_{i\in M} u_{ij} \ge \gamma_j^s : j \in N\}$ into the objective function with non-negative Lagrangian multiplier σ to derive the UFL Lagrangian characteristic function,

$$\begin{aligned} c_{LR\_UFL}(s;\sigma) = \min_{v,u} & \sum_{i\in M} f_i v_i + \sum_{i\in M}\sum_{j\in N}\big(c_{ij}-\sigma_j\big)u_{ij} + \sum_{j\in N}\sigma_j\gamma_j^s \\ s.t.\ & u_{ij} - v_i \le 0,\ \forall i \in M, j \in N, \\ & u_{ij} \le \gamma_j^s,\ \forall i \in M, j \in N, \\ & v_i, u_{ij} \in \{0,1\},\ \forall i \in M, j \in N. \end{aligned}$$

The augmentation of constraints (18) is to strengthen the Lagrangian lower bound of $c_{UFL}(s)$, which may accordingly lead to a better LRB cost allocation. It prohibits setting $u_{ij'} = 1$ for any player j' not in coalition s, even though the coefficient $c_{ij'} - \sigma_{j'} < 0$ when computing $c_{LR\_UFL}(s;\sigma)$. It is easy to see that the augmentation of (18) is simply equivalent to replacing term $\sum_{i\in M}\sum_{j\in N}\big(c_{ij} - \sigma_j\big)u_{ij}$ by $\sum_{i\in M}\sum_{j\in s}\big(c_{ij}-\sigma_j\big)u_{ij}$ in the objective function of $c_{LR\_UFL}(s;\sigma)$.

Under Algorithm 1, the generic LRB cost allocation algorithm for any $s \in S$ and non-negative Lagrangian multiplier σ, we can decompose $c_{LR\_UFL}(s;\sigma)$ into $c_{LR1\_UFL}(\ \cdot\ ;\sigma)$ and $c_{LR2\_UFL}(\ \cdot\ ;\sigma)$ such that $c_{LR\_UFL}(s;\sigma) = c_{LR1\_UFL}(s;\sigma) + c_{LR2\_UFL}(s;\sigma)$, and define UFL sub-game 1 $\big(N, c_{LR1\_UFL}(\ \cdot\ ;\sigma)\big)$ and UFL sub-game 2 $\big(N, c_{LR2\_UFL}(\ \cdot\ ;\sigma)\big)$.

For the UFL sub-game 1, its characteristic function is

$$c_{LR1\_UFL}(s;\sigma) = \sum_{j\in N}\sigma_j\gamma_j^s. \tag{19}$$

According to Lemma 1, the optimal stable cost allocation $\alpha^{\sigma}_{LR1\_UFL}$ which lies in the core of game $\big(N, c_{LR1\_UFL}(\ \cdot\ ;\sigma)\big)$ is given by $\alpha^{\sigma}_{LR1\_UFL}(j) = \sigma_j$, for all $j \in N$.

For the UFL sub-game 2, its characteristic function is

$$\begin{aligned} c_{LR2\_UFL}(s;\sigma) = \min_{v,u} & \sum_{i\in M} f_i v_i + \sum_{i\in M}\sum_{j\in N}\big(c_{ij}-\sigma_j\big)u_{ij} \\ s.t.\ & u_{ij} - v_i \le 0,\ \forall i \in M, j \in N, \\ & u_{ij} \le \gamma_j^s,\ \forall i \in M, j \in N, \\ & v_i, u_{ij}, \in \{0,1\},\ \forall i \in M, j \in N. \end{aligned} \tag{20}$$

To solve $c_{LR2\_UFL}(s:\sigma)$, we can decompose it by facilities and derive a closed-form optimal objective function value given by $c_{LR2\_UFL}(s;\sigma) = \sum_{i=1}^{m} \min\big\{0, f_i + \sum_{j\in s}\min\{0, c_{ij}-\sigma_j\}\big\}$.

Lemma 4. *UFL sub-game 2 $\big(N, c_{LR2\_UFL}(\,\cdot\,;\sigma)\big)$ is submodular.*

Proof. Denote a and b as two players in N. To show the submodularity, we need to prove that, for any coalition $s \in N\setminus\{a,b\}$,

$$c_{LR2\_UFL}(s\cup\{a\};\sigma) - c_{LR\_UFL2}(s;\sigma) \geq c_{LR2\_UFL}(s\cup\{a,b\};\sigma) - c_{LR2\_UFL}(s\cup\{b\};\sigma). \tag{21}$$

For each $i\in M$, let $\Delta_i(s;\sigma) = \min\{0, f_i + \sum_{j\in s}\min\{0, c_{ij}-\sigma_j\}\}$. To show (21), it is sufficient to show

$$\Delta_i(s;\sigma) + \Delta_i(s\cup\{a,b\};\sigma) \leq \Delta_i(s\cup\{a\};\sigma) + \Delta_i(s\cup\{b\};\sigma),\ \forall s\in N\setminus\{a,b\}. \tag{22}$$

Let $\rho(x) = \min\{0,x\}$, and define $x_{\hat{s}} = f_i + \sum_{j\in\hat{s}}\min\{0, c_{ij}-\sigma_j\}$ for each $\hat{s}\in\{s, s\cup\{a\}, s\cup\{b\}, s\cup\{a,b\}\}$. It can be seen that $x_s + x_{s\cup\{a,b\}} = x_{s\cup\{a\}} + x_{s\cup\{b\}}$, and $x_{s\cup\{a,b\}} \leq \min\{x_{s\cup\{a\}}, x_{s\cup\{b\}}\} \leq \max\{x_{s\cup\{a\}}, x_{s\cup\{b\}}\} \leq x_s$. Thus, since $\rho(x)$ is a concave function of x, we have

$$\rho(x_s) + \rho(x_{s\cup\{a,b\}}) \leq \rho(x_{s\cup\{a\}}) + \rho(x_{s\cup\{b\}}),$$

from which we can obtain (22) directly, and complete the proof of Lemma 4. □

Due to the submodularity of UFL sub-game 2, one can easily compute its core cost allocation, denoted as $\alpha^{\sigma}_{LR2\_UFL}$, by the greedy algorithm mentioned in Section 4.3. Under the optimal Lagrangian multiplier σ^*, we can derive the optimal UFL LRB cost allocation given by $\alpha^{\sigma^*}_{LR\_UFL} = \alpha^{\sigma^*}_{LR1\_UFL} + \alpha^{\sigma^*}_{LR2\_UFL}$. Since both UFL sub-games 1 and 2 have non-empty cores, by Theorem 2 the optimal LRB cost allocation value achieves the Lagrangian lower bound $c_{LR\_UFL}(N;\sigma^*)$, which is no less than the LP lower bound $c_{LP\_UFL}(N)$.

The following theorem shows the optimality of the UFL LRB cost allocation, and reveals the equivalence of the LRB and LPB cost allocations.

Theorem 3. *For a UFL game, the LRB cost allocation $\alpha^{\sigma^*}_{LR\_UFL} = \sigma^* + \alpha^{\sigma^*}_{LR2\_UFL}$ is optimal. In addition, both the LRB cost allocation set and the LPB cost allocation set consist of all the optimal UFL cost allocations.*

Proof. For the UFL game, we first show that the LRB cost allocation is optimal. As stated earlier, the optimal LRB cost allocation value achieves the Lagrangian lower bound $c_{LR\_UFL}(N;\sigma^*)$, which is not less than the LP lower bound $c_{LP\_UFL}(N)$. It is known that the LP lower bound equals the maximum total shared cost for the UFL game (Kolen 1983, Goemans and Skutella 2000). Thus, the LRB cost allocation must be an optimal UFL cost allocation, and $c_{LR\_UFL}(N;\sigma^*) = c_{LP\_UFL}(N)$.

We next prove that both the LRB cost allocation set and the LPB cost allocation set consist of all the optimal UFL cost allocations. It is known that the LPB cost allocation set consists of all the optimal UFL cost allocations (Goemans and Skutella 2000). This implies that each LRB cost allocation must belong to the LPB cost allocation set. Therefore, it remains to be shown that each LPB cost allocation belongs to the LRB cost allocation set.

Consider each LPB cost allocation $\alpha_{LP\_UFL}(j) = \mu_j^*$ for $j \in N$, where μ^* together with some δ^* form an optimal solution to the following dual problem of $c_{LP\_UFL}(N)$:

$$\begin{aligned} &\max_{\mu,\delta} \sum_{j\in N} \mu_j \\ s.t.\ &\sum_{j\in N} \delta_{ij} = f_i,\ \forall i \in M, \\ &\mu_j - \delta_{ij} \le c_{ij},\ \forall i \in M, j \in N, \\ &\mu_j \ge 0, \delta_{ij} \ge 0,\ \forall i \in M, j \in N. \end{aligned}$$

For each $i \in M$, it can be seen that $f_i = \sum_{j\in N} \delta_{ij}^*$, and that $\delta_{ij}^* \ge \max\{0, \mu_j^* - c_{ij}\}$ for $j \in N$, which imply that $f_i \ge \sum_{j\in N} \max\{0, \mu_j^* - c_{ij}\} = -\sum_{j\in N} \min\{0, c_{ij} - \mu_j^*\}$. Thus,

$$\min\{0, f_i + \sum_{j\in N} \min\{0, c_{ij} - \mu_j^*\}\} = 0, \text{ for each } i \in M. \tag{23}$$

Since $c_{LR2\_UFL}(N;\sigma) = \sum_{i=1}^m \min\{0, f_i + \sum_{j\in N} \min\{0, c_{ij} - \sigma_j\}\}$ for any non-negative σ, by (23) we have $c_{LR2\_UFL}(N;\mu^*) = 0$. This, together with $c_{LR1\_UFL}(N;\mu^*) = \sum_{j\in N} \mu_j^*$, implies that $c_{LR\_UFL}(N;\mu^*) = \sum_{j\in N} \mu_j^* = c_{LP\_UFL}(N) = c_{LR\_UFL}(N;\sigma^*)$. Hence, μ^* is an optimal Lagrangian multiplier. The resulting LRB cost allocation is then given by $\alpha_{LR\_UFL}^{\mu^*}(j) = \mu_j^* + \alpha_{LR2\_UFL}^{\mu^*}(j)$ for $j \in N$. Note that for each $s \in S$, since $c_{LR2\_UFL}(s;\mu^*) \le 0$ and $c_{LR2\_UFL}(s;\mu^*) \ge c_{LR2\_UFL}(N;\mu^*) = 0$, we have $c_{LR2\_UFL}(s;\mu^*) = 0$, which leads to $\alpha_{LR2\_UFL}^{\mu^*}(j) = 0$ for $j \in N$. Therefore, we obtain that $\alpha_{LR\_UFL}^{\mu^*} = \mu^*$, implying that each LPB cost allocation μ^* belongs to the LRB cost allocation set. This completes the proof of Theorem 3. □

5.1.3. Alternative Optimal Stable Cost Allocations For a UFL game, it is known that every optimal cost allocation corresponds to an LPB solution, which must be a convex combination of all basic optimal solutions to the dual of $c_{LP\_UFL}(N)$. However, by applying only common LP solvers to the dual of c_{LP_UFL}, it is hard to obtain all the basic optimal solutions.

By Theorem 3, we know that the LRB algorithm provides an alternative optimal cost allocation for the UFL game. It is possible that the LRB solution obtained can be excluded from those LPB solutions produced by the common LP solvers. We illustrate this by the following example.

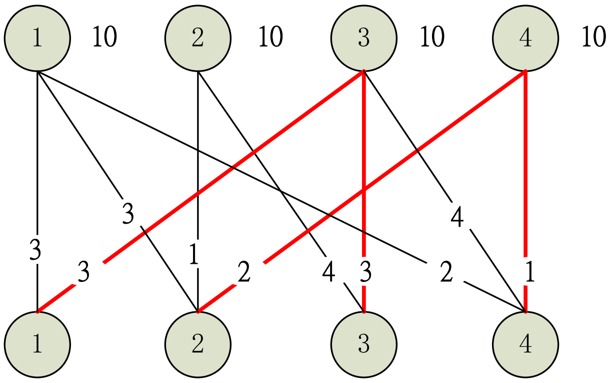

Figure 1 An example of a UFL game

In the UFL game shown in Figure 1, there are four facilities and four customers (players). Each facility has a fixed opening cost 10. The numbers on the links are the transportation costs from facilities to customers. An optimal decision for the grand coalition is to open facilities 3 and 4, and the links in bold are the optimal paths. Therefore, the grand coalition cost is 10+10+3+3+2+1=29.

For this example, we use two LP solvers, the "Simplex" and "Interior Point" methods, in MATLAB Release 2011a to compute the LPB allocations, respectively. Table 2 shows the cost assigned to each player under different approaches.

Table 2 Optimal stable UFL cost allocations under different approaches

Method	Player 1	Player 2	Player 3	Player 4	Total Shared Cost
LPB with Simplex	5.00	6.50	8.50	6.50	26.5
LPB with Interior Point	6.58	6.50	8.50	4.92	26.5
LRB	6.87	6.50	8.50	4.63	26.5

The example reveals that the LRB algorithm can generate optimal stable cost allocations that are different from those generated by common LP solvers. The LRB solution is beyond the range of convex combination of the two LPB solutions. This demonstrates the value of the LRB algorithm in providing alternative cost allocations.

To investigate the capability of the LRB algorithm under a general setting, we tested 30 uncapacitated facility location benchmark instances developed by Beresnev et al. (2006), all with $m = n = 100$. We conducted all computational experiments on a Windows 7 PC with an Intel Core i7-2600 running at 3.4GHz and 16G RAM. All algorithms were implemented in MATLAB Release 2011a. Among the 30 instances there are 22 for which the LRB solution is beyond the range of convex combination of the two LPB solutions. Again, this shows the value of the LRB algorithm in terms of computationally finding alternative optimal stable cost allocations even in cases where the LPB cost allocations are shown to be optimal.

5.2. The NLCFL Game

In a Non-Linear single source Capacitated Facility Location (NLCFL) game, there is a bipartite network $G=(M,N,E)$ defined similarly to the UFL game. Each potential facility site $i\in M$ now has a capacity Q_i, and each customer point $j\in N$ has a demand q_j. Every customer can only be served by a single facility. In addition to the opening cost, each facility i also has an operational cost that is increasing with the number of customers it is serving. To model the economy-of-scale effect, we use a quadratic function $\theta_i[h_i n_i - l_i\big(n_i\big)^2]$ to measure the operational cost, where n_i is the number of customers served by facility i, and θ_i, h_i and l_i are appropriate parameters ensuring the cost is concave and increasing for $n_i\in[0,n]$.

Unlike the UFL game in which players are only the customers in N, the player set of the NLCFL game includes both facility players in M and customer players in N. Similar settings of the players can also be found in the bin packing games (e.g., see Faigle and Kern 1993, Liu 2009). Our LRB algorithm can also handle the case where only customer players are involved. However, by including facility players, together with a non-linear cost function, we can demonstrate the broad range of applications to which our LRB algorithm can be applied.

We need the following additional notation in defining the NLCFL game:

Table 3 New notation used in the NLCFL game

Notation	Meaning
Q_i	The capacity of facility i, $\forall i\in M$, $Q_i\in\mathbb{Z}^+$.
q_j	The demand of customer j, $\forall j\in N$, $q_j\in\mathbb{Z}^+$.
s	A player coalition, $s=s_f\cup s_c$.
s_f	Facility player set in coalition s, $s_f\subseteq M$.
s_c	Customer player set in coalition s, $s_c\subseteq N$.
γ^s	Incidence vector $\big[\gamma_1^{s_f},\gamma_2^{s_f},...,\gamma_m^{s_f},\gamma_1^{s_c},\gamma_2^{s_c},...,\gamma_n^{s_c}\big]^T$, where $\gamma_i^{s_f}=1$, if $i\in s_f$ and $\gamma_i^{s_f}=0$, otherwise; $\gamma_j^{s_c}=1$ if $j\in s_c$ and $\gamma_j^{s_c}=0$ otherwise, $\forall j\in N, s_f\subseteq M, s_c\subseteq N$.

Definition 4. An NLCFL game $\big(M\cup N, c_{NLCFL}\big)$ is defined with players in $M\cup N$, where M is the facility player set, N is the customer player set, and the characteristic function $c_{NLCFL}(s)$ is determined by NLP

$$c_{NLCFL}(s_f\cup s_c)=\min_{v,u}\sum_{i\in M}f_iv_i+\sum_{i\in M}\sum_{j\in N}c_{ij}u_{ij}+\sum_{i\in M}\theta_i\big[\sum_{j\in N}h_iu_{ij}-l_i\big(\sum_{j\in N}u_{ij}\big)^2\big] \tag{24}$$

$$s.t.\ \sum_{i\in M}u_{ij}\geq\gamma_j^{s_c},\ \forall j\in N, \tag{25}$$

$$\sum_{j\in N}q_ju_{ij}-Q_iv_i\leq 0,\ \forall i\in M, \tag{26}$$

$$v_i \leq \gamma_i^{s_f},\ \forall i \in M, \tag{27}$$

$$v_i, u_{ij}, \in \{0,1\},\ \forall i \in M, j \in N. \tag{28}$$

Compared with $c_{UFL}(s)$, $c_{NLCFL}(s_f \cup s_c)$ has a few new constraints. Constraints (26) represent the capacity restrictions of the facilities; constraints (27) ensure that only if a facility player is in the coalition can the corresponding facility be used to serve customers.

Since the objective function of $c_{NLCFL}(s_f \cup s_c)$ has a non-linear term to measure the facility operational cost, the LPB algorithm is no longer applicable to computing cost allocations. Next we will illustrate the implementation of the LRB algorithm on the NLCFL game.

5.2.1. LRB Cost Allocation for the NLCFL Game In the NLCFL characteristic function $c_{NLCFL}(s_f \cup s_c)$, by adding a new set of constraints

$$u_{ij} \leq \gamma_i^{s_f},\ u_{ij} \leq \gamma_j^{s_c},\ \forall i \in M, j \in N, \tag{29}$$

and then bringing constraints $\{\sum_{i\in M} u_{ij} \geq \gamma_j^{s_c} : \forall j \in N\}$ into the objective function with non-negative Lagrangian multiplier σ, we can derive the NLCFL Lagrangian characteristic function,

$$\begin{aligned} c_{LR\_NLCFL}(s;\sigma) = \sum_{i\in M} f_i v_i + \sum_{i\in M}\sum_{j\in N} (c_{ij} - \sigma_j + \theta_i h_i) u_{ij} - \sum_{i\in M} \theta_i l_i \Big(\sum_{j\in N} u_{ij}\Big)^2 + \sum_{j\in N} \sigma_j \gamma_j^{s_c} \\ s.t. \sum_{j\in N} u_{ij} q_j - Q_i v_i \leq 0,\ \forall i \in M, \\ u_{ij} \leq \gamma_i^{s_f},\ \forall i \in M, j \in N, \\ v_i \leq \gamma_i^{s_f},\ \forall i \in M, \\ u_{ij} \leq \gamma_j^{s_c},\ \forall i \in M, j \in N, \\ v_i, u_{ij} \in \{0,1\},\ \forall i \in M, j \in N. \end{aligned}$$

Similar to constraints (18) for c_{UFL}, the augmentation of constraints (29) is to strengthen the Lagrangian lower bound for $c_{NLCFL}(s_f \cup s_c)$.

Following the LRB algorithm, we can decompose $c_{LR\_NLCFL}(\,\cdot\,;\sigma)$ into $c_{LR1\_NLCFL}(\,\cdot\,;\sigma)$ and $c_{LR2\_NLCFL}(\,\cdot\,;\sigma)$ such that $c_{LR\_NLCFL}(s_f \cup s_c;\sigma) = c_{LR1\_NLCFL}(s_f \cup s_c;\sigma) + c_{LR2\_NLCFL}(s_f \cup s_c;\sigma)$, for all $s_f \subseteq M$ and $s_c \subseteq N$, and define NLCFL sub-game 1 $\big(M \cup N; c_{LR1\_NLCFL}(\,\cdot\,;\sigma)\big)$ and NLCFL sub-game 2 $\big(M \cup N; c_{LR2\_NLCFL}(\,\cdot\,;\sigma)\big)$, respectively.

For NLCFL sub-game 1, the characteristic function is

$$c_{LR1\_NLCFL}(s_f \cup s_c, \sigma) = \sum_{j\in N} \sigma_j \gamma_j^{s_c}. \tag{30}$$

According to Lemma 1, we can derive the core cost allocation for game $\big(M \cup N, c_{LR1\_NLCFL}(\,\cdot\,;\sigma)\big)$, where each customer player $j \in N$ is assigned a cost exactly equal to the Lagrangian dual price of serving her, and no facility players are assigned any cost since they do not need to be served. The cost allocation is given by $\alpha^{\sigma}_{LR1\_NLCFL}(j) = \sigma_j$ for all $j \in N$, and $\alpha^{\sigma}_{LR1\_NLCFL}(i) = 0$ for all $i \in M$.

For NLCFL sub-game 2, the characteristic function is

$$\begin{aligned} c_{LR2\_NLCFL}(s_f \cup s_c;\sigma) = \min_{v,u} & \sum_{i\in M} f_i v_i + \sum_{i\in M}\sum_{j\in N}(c_{ij} - \sigma_j + \theta_i h_i)u_{ij} - \sum_{i\in M}\theta_i l_i \Big(\sum_{j\in N} u_{ij}\Big)^2 \\ s.t. & \sum_{j\in N} u_{ij}q_j - Q_i v_i \le 0,\ \forall i \in M, \\ & u_{ij} \le \gamma_i^{s_f},\ \forall i \in M, j \in N, \\ & v_i \le \gamma_i^{s_f},\ \forall i \in M, \\ & u_{ij} \le \gamma_j^{s_c},\ \forall i \in M, j \in N, \\ & v_i, u_{ij} \in \{0,1\},\ \forall i \in M, j \in N. \end{aligned}$$

To solve $c_{LR2\_NLCFL}(s_f \cup s_c;\sigma)$, we can decompose it by facilities, i.e., $c_{LR2\_NLCFL}(s_f \cup s_c;\sigma) = \sum_{i\in s_f} \psi^i(s_c;\sigma)$, where for each $i \in s_f$,

$$\begin{aligned} \psi^i(s_c;\sigma) = \min_{v_i,u_{ij}} & f_i v_i + \sum_{j\in s_c}(c_{ij} - \sigma_j + \theta_i h_i)u_{ij} - \theta_i l_i \Big(\sum_{j\in s_c} u_{ij}\Big)^2 \\ s.t. & \sum_{j\in s_c} q_j u_{ij} - Q_i v_i \le 0, \\ & v_i, u_{ij} \in \big\{0,1\big\},\ \forall j \in s_c. \end{aligned} \tag{31}$$

It can be seen that each problem $\psi^i(s_c;\sigma)$ corresponds to a variant of the knapsack problem with an objective of minimizing a non-linear total value function, where Q_i is the knapsack capacity, and s_c is a set of items with each item $j \in s_c$ having a weight q_j and a value $(c_{ij} - \sigma_j + \theta_i h_i)$. In addition to the total value of the items packed into the knapsack, one can obtain an extra value $-\theta_i l_i\Big(\sum_{j\in s_c} u_{ij}\Big)^2$, which is quadratic in the number of the items packed. When all weights q_j are integers, we can solve $\psi^i(s_c;\sigma)$ by a dynamic program in pseudo-polynomial time $O(Q_i n^2)$. **To be specific, we define $F_i^{s_c}(j,k,q)$ as the minimum value of $\sum_{j'\in s_c, j'\le j}(c_{ij'} - \sigma_{j'} + \theta_i h_i)u_{ij'}$ such that $\sum_{j'\in s_c,j'\le j} u_{ij'} = k$ and $\sum_{j'\in s_c,j'\le j} q_{j'}u_{ij'} \le Q_i$.** Liu: To be specific, we define $F_i^{s_c}(j,k,q)$ as the minimum value of $\sum_{j'\in s_c,j'\le j}(c_{ij'} - \sigma_{j'} + \theta_i h_i)u_{ij'}$ such that $\sum_{j'\in s_c,j'\le j} u_{ij'} = k$ and $\sum_{j'\in s_c,j'\le j} q_{j'}u_{ij'} \le q$. In other words, $F_i^{s_c}(j,k,q)$ represents the minimum item value packed by including exactly k items from set $\{1,2,\cdots,j\}$ within capacity q. The dynamic programming recursion is as follows: The authors should specify what q represents mathematically in the states of the DP.

$$F_i^{s_c}(j,k,q) = \begin{cases} F_i^{s_c}(j,k,q), & \text{if } j \notin s_c, \\ \min\big\{F_i^{s_c}(j-1,k,q), F_i^{s_c}(j-1,k-1,q-q_j)\big\}, & \text{if } j \in s_c, \end{cases}$$

with initial conditions $F_i^{s_c}(0,0,q)=0$ for $q\geq 0$, and boundary conditions $F_i^{s_c}(j,k,q)=+\infty$ for $q<0$. Then $\psi_i(s_c;\sigma)$ can be found by $\psi_i(s_c;\sigma)=\min_{k\leq|s_c|}\big\{0, f_i+F_i^{s_c}(n,k,Q_i)-\theta_i l_i k^2\big\}$, and we have $c_{LR2\_NLCFL}(s_f\cup s_c;\sigma)=\sum_{i\in s_f}\psi_i(s_c;\sigma)$.

Now we are ready to compute the optimal stable cost allocation $\alpha^{\sigma}_{LR2\_NLCFL}$ for NLCFL sub-game 2 by the CGB algorithm, where we need to solve a pricing problem. In this particular case, the pricing problem is to find a coalition (or column) $s=s_f\cup s_c$ with the smallest reduced cost, where the reduced cost for each $s=s_f\cup s_c$ is given by

$$\begin{aligned}\min_{v,u}\sum_{i\in s_f}f_iv_i+\sum_{i\in s_f}\sum_{j\in s_c}\big(c_{ij}-\sigma_j+\theta_ih_i\big)u_{ij}-\sum_{i\in s_f}\theta_il_i\Big(\sum_{j\in s_c}u_{ij}\Big)^2-\sum_{k\in M\cup N}\gamma_k^s\pi_k^*\\ s.t.\ \sum_{j\in s_c}q_ju_{ij}\leq Q_iv_i,\ \forall i\in s_f,\\ v_i,u_{ij}\in\big\{0,1\big\},\ \forall i\in s_f, j\in s_c,\end{aligned}\tag{32}$$

with π^* being the optimal dual of the corresponding master problem for NLCFL sub-game 2.

For each given $s=s_f\cup s_c$, one can obtain the optimal objective value of (32) directly, as it equals $\sum_{i\in s_f}\psi^i(s_c;\sigma)-\sum_{k\in M\cup N}\gamma_k^s\pi_k^*$, where each $\psi^i(s_c;\sigma)$, as shown earlier, can be computed by dynamic programming. However, due to the exponential number of coalitions, it is computationally intractable to find a column s with the most negative reduced cost by enumeration. We therefore attempt to first identify a column $\bar{s}$ with a negative reduced cost by considering the following two cases:

Case 1: There exists at least one k such that $\pi_k^*>0$. In this case, a coalition $\bar{s}$ can be defined by including k with $\pi_k^*>0$ for $k\in M\cup N$. The reduced cost for $\bar{s}$ is negative because it is at most $-\sum_{k\in M\cup N}\max\{0,\pi_k^*\}$ by setting all u and v to be zero.

Case 2: For all $k\in M\cup N$, $\pi_k^*\leq 0$. This case is more complicated. To efficiently find a coalition $\bar{s}$ with a negative reduced cost, we can consider the following ILP that treats membership in s_f and s_c as decisions by binary variables v_i and γ_j, respectively. The ILP essentially covers all coalitions.

$$\begin{aligned}\min_{v;u;\gamma}R(v,u,\gamma)=\min_{v,u}\sum_{i\in M}f_i'v_i+\sum_{i\in M}\sum_{j\in N}c_{ij}'u_{ij}-\sum_{i\in M}l_i'\big(\sum_{j\in N}u_{ij}\big)^2+\sum_{j\in N}\gamma_j\pi_j'\\ s.t.\ \sum_{j\in N}q_ju_{ij}\leq Q_iv_i,\ \forall i\in M,\\ u_{ij}\leq\gamma_j,\ \forall i\in M, j\in N,\\ v_i,u_{ij},\gamma_j\in\big\{0,1\big\},\ \forall i\in M, j\in N,\end{aligned}\tag{33}$$

where $f_i'=f_i-\pi_i^*$, $c_{ij}'=c_{ij}-\sigma_j+\theta_ih_i$, $l_i'=\theta_il_i$ and $\pi_j'=-\pi_j^*$. Although (34) is hard to solve to optimality, our purpose here is just to find a coalition $\bar{s}$, together with its corresponding optimal packing solution $\bar{u}$, which leads to a negative reduced cost. The following conditions are developed to find $\bar{s}$ efficiently.

The authors should clarify what this sentence means.

Case 2: For all $k \in M \cup N$, $\pi^*_k \leq 0$. This case is more complicated. To efficiently find a coalition $\bar{s}$ with a negative reduced cost, we can consider the following ILP where binary variables v_i and γ_j indicate whether or not players i in s_f and j in s_c, respectively, with $f'_i = f_i - \pi^*_i$, $c'_{ij} = c_{ij} - \sigma_j + \theta_i h_i$, $l'_i = \theta_i l_i$ and $\pi'_j = -\pi^*_j$.

$$\begin{aligned}\min_{v;u;\gamma} R(v,u,\gamma) = \min_{v,u} \sum_{i\in M} f'_i v_i + \sum_{i\in M}\sum_{j\in N} c'_{ij}u_{ij} - \sum_{i\in M} l'_i \Big(\sum_{j\in N} u_{ij}\Big)^2 + \sum_{j\in N}\gamma_j \pi'_j \\ s.t. \ \sum_{j\in N} q_j u_{ij} \leq Q_i v_i, \ \forall i \in M, \\ u_{ij} \leq \gamma_j, \ \forall i \in M, j \in N, \\ v_i, u_{ij}, \gamma_j \in \{0,1\}, \ \forall i \in M, j \in N,\end{aligned} \tag{34}$$

Therefore, it can be seen that feasible solutions to ILP of negative objective values are one-to-one correspondence with coalitions $\bar{s} = \bar{s}_f \cup \bar{s}_c$ of negative reduced costs. Moreover, such a coalition $\bar{s}$ can be obtained efficiently by exploiting the properties below:

Lemma 5. *For* (34), *without changing the optimal objective value, one can sequentially fix some variables to zero by the following steps:*
(1) *For each $(i,j) \in M \times N$, if $c'_{ij} - l'_i\big[n_i^2 - (n_i-1)^2\big] > 0$, then $u_{ij} = 0$, where n_i is the number of elements in set $\{c'_{ij} < \infty : \forall j \in N\}$. After that, set $c'_{ij} = \infty$.*
(2) *For each $j \in N$, if $\pi'_j + \sum_{i\in M} \min\{c'_{ij} - l'_i\big[n_i^2 - (n_i-1)^2\big], 0\} \geq 0$, then $\gamma_j = 0$, $u_{ij} = 0$, $\forall i \in M$.*
(3) *For each $i \in M$, solve a non-linear knapsack problem similar to* (31), *where Q_i is the knapsack capacity, and N is the item set, with each item $j \in N$ having a weight q_j and a value c'_{ij}. Let ω^i be the optimal objective function value of this knapsack problem. If $\omega^i + f'_i \geq 0$, then set $v_i = 0$ and $u_{ij} = 0$, for all $j \in N$.*

Proof. First, if there exists a pair of indices (i,j) such that $u_{ij} = 1$ and $c'_{ij} - l'_i\big[n_i^2 - (n_i-1)^2\big] > 0$ in a feasible solution of ILP (34), one can directly set $u_{ij} = 0$, and derive another feasible solution under which the objective function value is reduced by at least $c'_{ij} - l'_i\big[n_i^2 - (n_i-1)^2\big]$.

Second, if there exists a customer j such that $\gamma_j = 1$ and $\pi'_j + \sum_{i\in M} \min\{c'_{ij} - l'_i\big[n_i^2 - (n_i-1)^2\big], 0\} \geq 0$ in a feasible solution of ILP (34), then setting $\gamma_j = 0$ and $u_{ij} = 0$ for all $i \in M$ results in another feasible solution where the objective function value is reduced by at least $\pi'_j + \sum_{i\in M} \min\{c'_{ij} - l'_i\big[n_i^2 - (n_i-1)^2\big], 0\}$.

Third, if there exists a facility i such that $v_i = 1$ and $\omega^i + f'_i \geq 0$ in a feasible solution of ILP (34), then the resulting solution by setting $v_i = 0$ and $u_{ij} = 0$ for all $j \in N$ is also feasible and the objective function value is reduced by at least $\omega^i + f'_i$. □

Suffice it to say that making the above changes does not increase the value of $\min_{v;u;\gamma} R(v,u,\gamma)$. Though not theoretically ensuring polynomial time complexity, the steps in Lemma 5 can indeed greatly reduce the problem size when solving (34).

After deriving the optimal stable cost allocations $\alpha^{\sigma}_{LR1\_NLCFL}$ and $\alpha^{\sigma}_{LR2\_NLCFL}$ for NLCFL subgames 1 and 2, respectively, we can compute a stable cost allocation $\alpha^{\sigma}_{LR\_NLCFL} = \alpha^{\sigma}_{LR1\_NLCFL} + \alpha^{\sigma}_{LR2\_NLCFL}$ for an NLCFL game according to Theorem 1. Furthermore, by Theorem 2, the corresponding LRB cost allocation value $\sum_{k\in M\cup N} \alpha^{\sigma^*}_{LR\_NLCFL}(k)$ is equal to the Lagrangian relaxation lower bound $c_{LR\_NLCFL}(M\cup N;\sigma^*)$, if σ^* is the optimal Lagrangian multiplier and $\big(M\cup N; c_{LR2\_NLCFL}(\,\cdot\,;\sigma^*)\big)$ has a non-empty core.

5.2.2. Computational Results for the NLCFL Game To conduct the computational experiments, we use 20 single source facility location benchmark instances developed by Beresnev et al. (2006). Each instance has a bipartite network $G=(M,N,E)$ with $m=n=100$ and $f_i=100$ for all $i\in M$. For each instance, there are three capacity levels 10, 20 and 30. In addition, we use $h_1=h_2=...=h_m=n^2$, $l_1=l_2=...=l_m=1$, and $\theta_1=\theta_2=...=\theta_m=\theta$ to measure the operational cost, where θ indicates the relative weight of the operational cost. When solving the Lagrangian dual problem by the subgradient method, we set the number of maximal iterations to be 2500.

To show the effectiveness of the LRB cost allocation, ideally we need to compare the total shared cost against the grand coalition cost $c_{NLCFL}(M\cup N)$. However, the grand coalition cost is only available in the benchmark data set for instances with $\theta=\mathbf{0}$. Therefore, for a general comparison, we need to compromise by replacing the centralized optimum with a heuristic solution, called the Best Found Centralized Solution (BFCS), which is defined as the better of the following two feasible solutions. The first feasible solution is simply the optimal solution to the original benchmark instances of the NLCFL problems with $\theta=\mathbf{0}$, which is available in Bachrach et al. (2009). The second feasible solution is derived from the optimal solution of $c_{LR2\_NLCFL}(M\cup N;\sigma)$. Note that this optimal solution might be infeasible for the centralized problem $c_{NLCFL}(M\cup N)$, since some customers may not be served. If so, to derive a feasible solution out of the given infeasible solution, we can proceed as follows. For each unserved customer, we choose an opened facility with enough remaining capacity and the smallest transportation cost to serve this customer; if there is no such facility, we open a new feasible facility with the minimum transportation cost to serve this customer.

Table 4 shows the performance and computational efficiency of the LRB cost allocation algorithm implemented on the 20 instances under situations where the facility capacities are identically equal to 10, 20 and 30, respectively.

Table 4 Performance of LRB cost allocation algorithm for the NLCFL game

Capacity	θ	LRCA / BFCS (%)			LRCA / LRB(%)			Total time(s)		
		Average	Max	Min	Average	Max	Min	Average	Max	Min
$Q = 10$	0	98.79	99.12	98.33	100	100	100	–	–	–
	0.01	99.64	99.70	99.55	100	100	100	5683	6838	4987
	0.1	99.87	99.89	99.78	100	100	100	5690	6834	4980
	0.5	99.90	99.92	99.87	100	100	100	5742	6814	5036
	1	99.91	99.95	99.89	100	100	100	5822	6983	4764
$Q = 20$	0	98.32	99.30	97.66	100	100	99.95	–	–	–
	0.01	99.61	99.76	99.48	100	100	100	9925	10478	9485
	0.1	99.83	99.85	99.82	100	100	100	9835	10458	9322
	0.5	99.85	99.88	99.84	100	100	99.99	9825	10487	9315
	1	99.89	99.92	99.87	100	100	100	9973	11154	9812
$Q = 30$	0	95.25	96.95	93.93	100	100	100	–	–	–
	0.01	99.02	99.15	98.82	100	100	99.99	11686	12831	10410
	0.1	99.72	99.77	99.63	99.99	100	99.95	11755	12816	10421
	0.5	99.81	99.87	99.78	100	100	100	11485	13064	10277
	1	99.88	99.92	99.86	100	100	100	12621	14371	11955

In the table, "LRCA" represents the best found LRB cost allocation value $\sum_{k\in M\cup N}\alpha^{\sigma}_{LR\_NLCFL}(k)$ under different σ, and "LRB" represents the best Lagrangian lower bound $c_{LR\_NLCFL}(M\cup N;\sigma^*)$ obtained using the subgradient method. For each capacity, we list the computational results under different values of θ. From column "LRCA/BFCS" it can be seen that for all the examined instances, our LRB cost allocation algorithm can produce stable cost allocations that share at least 93.93% of BFCS. When θ increases so that the facility operational cost gains more weight, our LRB cost allocations can share more than 99% of BFCS. These findings demonstrate the high quality of the LRB cost allocations. **Moreover, although NLCFL sub-game 2 is not submodular in general, column "LRCA / LRB" suggests that almost every NLCFL sub-game 2 has a non-empty core. This indicates the superiority of the LRB algorithm over the LPB algorithm, even in cases where condition (2) of Theorem 2 does not hold.** {The authors should clarify this statement: superiority in what sense? Liu: Moreover, although NLCFL sub-game 2 is not submodular in general, column "LRCA / LRB" suggests that almost every NLCFL sub-game 2 has a non-empty core. In particular, when $\theta = 0$, the resulting LRB cost allocation value is very likely to equal the corresponding Lagrangian lower bound, and thus surpass the LP lower bound as well as the LPB cost allocation value. This indicates that even in cases where condition (2) of Theorem 2 does not hold, the LRB algorithm still has a great chance to beat the LPB algorithm in terms of total allocated cost. } As for time efficiency, we can see that the computational time tends to increase with Q and θ. Among all instances, the longest computation time is around four hours.

For instances with $\theta = \mathbf{0}$, where the NLCFL game has no nonlinear term in its characteristic function, we compare the LPB and LRB cost allocations, by showing in Table 5 the percentage ratios of the cost allocation values against the grand coalition costs given by the benchmarks. Here the LPB and LRB cost allocations are computed based on the same ILP formulation for the characteristic function $c_{NLCFL}(s_f \cup s_c)$ augmented by constraints (29).

Table 5 LPB vs. LRB cost allocations for the NLCFL game with $\theta = \mathbf{0}$ (in %)

Capacity	Average					LRCA − LPCA	
	LPCA	LRB	LRCA	LRB′	LRCA′	Max	Min
10	97.15	98.79	98.79	98.79	98.79	2.38	1.00
20	97.20	98.32	98.31	98.29	98.25	1.51	0.88
30	94.70	95.25	95.25	95.21	95.21	0.75	0.38
40	94.11	94.25	94.25	94.25	94.25	0.28	0.07
50	93.87	93.88	93.88	93.88	93.88	0.04	-0.02

To study the impact of constraints (29), we compare LRB and LRCA with a new Lagrangian lower bound LRB′ and a new LRB cost allocation value LRCA′, where LRB′ and LRCA′ are obtained from a revised characteristic function of $c_{LR_NLCFL}(s;\sigma)$ with constraints (29) being relaxed. Moreover, since the LPB cost allocation value and the LP lower bound are equal, they are both presented in column "LPCA" of Table 5. From the table, we have the following observations:

First, the Lagrangian lower bound is tighter than the LP lower bound on average, as shown in the first two columns under "Average". This implies the potential advantage of the LRB cost allocation over the LPB cost allocation. In addition, as indicated by the columns under "LRCA − LPCA", the LRB cost allocation is indeed superior to the LPB on average, especially for cases with a lower capacity.

Second, as shown in columns "LRB, LRCA, LRB′, LRCA′", adding constraints (29) to $c_{NLCFL}(s_f \cup s_c)$ can indeed improve the Lagrangian lower bounds, as well as the LRB cost allocation values. In addition, by comparing columns "LPCA" and "LRCA′", we find that even with no additional constraints, the resulting LRB cost allocation still beats the LPB one on average. This further implies the competitiveness of our LRB algorithm.

We next investigate the convergence of the LRB algorithm on the NLCFL game. This was not an issue for the UFL game. As long as the subgradient method for the Lagrangian dual problem converges to σ^* in a UFL game, Theorem 2 ensures the optimality of the stable cost allocation corresponding to σ^* because UFL sub-game 2 has a non-empty core. However, the NLCFL sub-game 2 may have an empty core, implying a possible gap between the Lagrangian lower bound and the total cost that can be allocated. Although we may expect a general trend where a tighter Lagrangian lower bound leads to a better cost allocation, there is no guarantee of the strict increase of cost allocation value when the Lagrangian lower bound increases.

To examine this, we apply Algorithm 1 to the NLCFL game on instances with $\theta = \mathbf{0}$, and compare LRB cost allocations that are obtained by using different Λ sets of Lagrangian multipliers. Table 6 reports the number of instances whose LRB cost allocations are improved, declined, and unchanged, respectively, when different pairs of Λ sets are used for comparison, where each σ^i represents the best Lagrangian multiplier found within i iterations of the subgradient method in Step 1 of Algorithm 1. The results show that it is possible that the cost allocation may become worse as the Lagrangian bound improves, though the chance of getting worse is very small in later iterations. For example, out of the one hundred instances, there are seven instances whose LRB cost allocations decline in quality when Λ is chosen to be $\{\sigma^{2500}\}$ instead of $\{\sigma^{500}, \sigma^{800}, \sigma^{1000}, \sigma^{1500}, \sigma^{2000}\}$. This finding confirms the need for using multiple Lagrangian multipliers in Algorithm 1.

{Its not clear from the table which set of multipliers is the baseline for improved and declined: it appears from your description in the text that the second set is the baseline?}

Liu: To examine this, we apply Algorithm 1 to the NLCFL game on instances with $\theta = \mathbf{0}$, and compare LRB cost allocations that are obtained by using different Λ sets of Lagrangian multipliers. Table 6 reports the number of instances whose LRB cost allocations are improved, declined, and unchanged, respectively, when new Lagrangian multiplier set Λ_1 is applied. Each σ^i represents the best Lagrangian multiplier found within i iterations of the subgradient method in Step 1 of Algorithm 1. The results show that it is possible that the cost allocation may become worse as the Lagrangian bound improves, though the chance of getting worse is very small in later iterations. For example, out of the one hundred instances, there are seven instances whose LRB cost allocations decline in quality when Λ is chosen to be $\{\sigma^{2500}\}$ instead of $\{\sigma^{500}, \sigma^{800}, \sigma^{1000}, \sigma^{1500}, \sigma^{2000}\}$. This finding confirms the need for using multiple Lagrangian multipliers in Algorithm 1.

Table 6 Comparisons of LRB cost allocations derived from different Λ sets of Lagrangian multipliers

Pairs of Λ sets		# of Improved	# of Declined	# of Unchanged
Λ_1	Λ_2 (as baseline)			
$\{\sigma^{800}\}$	$\{\sigma^{500}\}$	95	4	1
$\{\sigma^{1000}\}$	$\{\sigma^{500}, \sigma^{800}\}$	51	6	43
$\{\sigma^{1500}\}$	$\{\sigma^{500}, \sigma^{800}, \sigma^{1000}\}$	24	6	70
$\{\sigma^{2000}\}$	$\{\sigma^{500}, \sigma^{800}, \sigma^{1000}, \sigma^{1500}\}$	1	6	93
$\{\sigma^{2500}\}$	$\{\sigma^{500}, \sigma^{800}, \sigma^{1000}, \sigma^{1500}, \sigma^{2000}\}$	0	7	93

In summary, from the computational experiments we can conclude that the LRB algorithm is both effective and efficient in solving the OCAP for the NLCFL game.

6. Conclusion

The focus of this paper is on cooperative games whose core may be empty. We propose a generic framework to calculate a good stable cost allocation that satisfies coalitional stability and recovers the grand coalition cost as much as possible. In the literature, such a problem is usually treated by linear programming relaxation and duality techniques. We take the different approach of investigating Lagrangian relaxation techniques. Besides the competitiveness of the Lagrangian bound over the linear programming bound, our algorithm is not restricted to solving problems with assignable constraints and linear objective functions.

We demonstrate our new algorithm on two different facility location games, each representing a typical type of cooperative game that may be encountered. The computational experiments show that our algorithm can produce near-optimal cost allocations for all these games, outperforming the existing linear programming based algorithm. In fact, we have also successfully implemented our algorithm on other typical games, such as TSP games, e.g., see Liu and Qi (2015).

Acknowledgement

The authors appreciate the constructive and detailed comments by two anonymous reviewers. The work is partially supported by Hong Kong RGC GRF grant No. 618311.

References

Ahuja, Ravindra K, Thomas L Magnanti, James B Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ.

Aydinliyim, Tolga, George L Vairaktarakis. 2010. Coordination of outsourced operations to minimize weighted flow time and capacity booking costs. *Manufacturing Service Oper. Management* **12** 236–255.

Bachrach, Yoram, Edith Elkind, Reshef Meir, Dmitrii Pasechnik, Michael Zuckerman, Jörg Rothe, Jeffrey S Rosenschein. 2009. The cost of stability in coalitional games. *Mavronicolas, M., Papadopoulou, V.G.(eds.) SAGT 2009. LNCS, vol. 5814, pp. 122–134. Springer, Heidelberg.*

Barnhart, Cynthia, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, Pamela H Vance. 1998. Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* **46** 316–329.

Beresnev, Vladimir, Yuri Kochetov, Mihail Pashchenko, Alexander Kononov, Eugeni Goncharov, Alexander Plyasunov, Nina Kochetova, Ekaterina Alekseeva, Ivanenko Dmitry. 2006. Benchmark. `http://www.math.nsc.ru/AP/benchmarks/english.html`.

Bläser, Markus, L Shankar Ram. 2008. Approximately fair cost allocation in metric traveling salesman games. *Theory Comput. Syst.* **43** 19–37.

Cai, Xiaoqiang, George L Vairaktarakis. 2012. Coordination of outsourced operations at a third-party facility subject to booking, overtime, and tardiness costs. *Oper. Res.* **60** 1436–1450.

Caprara, Alberto, Adam N Letchford. 2010. New techniques for cost sharing in combinatorial optimization games. *Math. Programming* **124** 93–118.

Chen, Xin. 2009. Inventory centralization games with price-dependent demand and quantity discount. *Oper. Res.* **57** 1394–1406.

Chen, Xin, Jiawei Zhang. 2009. A stochastic programming duality approach to inventory centralization games. *Oper. Res.* **57** 840–851.

Deng, Xiaotie, Toshihide Ibaraki, Hiroshi Nagamochi. 1999. Algorithmic aspects of the core of combinatorial optimization games. *Math. Oper. Res.* **24** 751–766.

Edmonds, Jack. 1970. Submodular functions, matroids, and certain polyhedra. *In Combinatorial Structures and Their Applications (Calgary International Conference on Combinatorial Structures and Their Applications)* 69–87.

Engevall, Stefan, Maud Göthe-Lundgren, Peter Värbrand. 2004. The heterogeneous vehicle-routing game. *Transportation Sci.* **38** 71–85.

Faigle, Ulrich, Sándor P Fekete, Winfried Hochstättler, Walter Kern. 1998. On approximately fair cost allocation in Euclidean TSP games. *OR Spectrum* **20** 29–37.

Faigle, Ulrich, Walter Kern. 1993. On some approximately balanced combinatorial cooperative games. *Zeitschrift für Oper. Res.* **38** 141–152.

Faigle, Ulrich, Walter Kern, Daniël Paulusma. 2000. Note on the computational complexity of least core concepts for min-cost spanning tree games. *Math. Methods Oper. Res.* **52** 23–38.

Geoffrion, A.M. 1974. Lagrangian relaxation and its uses in integer programming. *Math. Programming Stud.* **2** 82–114.

Goemans, Michel X., Martin Skutella. 2000. Cooperative facility location games. *J. Algorithms* **50** 76–85.

Göthe-Lundgren, Maud, Kurt Jörnsten, Peter Värbrand. 1996. On the nucleolus of the basic vehicle routing game. *Math. Programming* **72** 83–100.

Granot, Daniel, Gur Huberman. 1981. Minimum cost spanning tree games. *Math. Programming* **21** 1–18.

Hartman, Bruce C, Moshe Dror, Moshe Shaked. 2000. Cores of inventory centralization games. *Games Econom. Behav.* **31** 26–49.

He, Simai, Jiawei Zhang, Shuzhong Zhang. 2012. Polymatroid optimization, submodularity, and joint replenishment games. *Oper. Res.* **60** 128–137.

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

Kern, Walter, Daniël Paulusma. 2003. Matching games: The least core and the nucleolus. *Math. Oper. Res.* **28** 294–308.

Kolen, Antoon. 1983. Solving covering problems and the uncapacitated plant location problem on trees. *Eur. J. Oper. Res.* **12** 266–278.

Li, Gaidi, Yu Li, Jia Shu, Dachuan Xu. 2013. A cross-monotonic cost-sharing scheme for the concave facility location game. *J. Global Optim.* **56** 1325–1334.

Liu, Lindong, Xiangtong Qi. 2015. To stabilize cooperative games by simultaneously subsidizing and penalizing. Working paper.

Liu, Zhixin. 2009. Complexity of core allocation for the bin packing game. *Oper. Res. Lett.* **37** 225–229.

Mallozzi, Lina. 2011. Cooperative games in facility location situations with regional fixed costs. *Optim. Lett.* **5** 173–181.

Martínez-de Albéniz, F Javier, Carles Rafels, Neus Ybern. 2013. A procedure to compute the nucleolus of the assignment game. *Oper. Res. Lett.* **41** 675–678.

Maschler, Michael, Bezalel Peleg, Lloyd S Shapley. 1979. Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of Operations Research* **4** 303–338.

Meir, Reshef, Jeffrey S. Rosenschein, Enrico Malizia. 2011. Subsidies, stability, and restricted cooperation in coalitional games. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume, 1, 301–306*. IJCAI'11, AAAI Press.

Owen, Guillermo. 1975. On the core of linear production games. *Math. Programming* **9** 358–370.

Potters, Jos AM, Imma J Curiel, Stef H Tijs. 1992. Traveling salesman games. *Math. Programming* **53** 199–211.

Puerto, Justo, Arie Tamir, Federico Perea. 2011. A cooperative location game based on the 1-center location problem. *Eur. J. Oper. Res.* **214** 317–330.

Puerto, Justo, Arie Tamir, Federico Perea. 2012. Cooperative location games based on the minimum diameter spanning steiner subgraph problem. *Discrete Appl. Math.* **160** 970–979.

Schulz, Andreas S, Nelson A Uhan. 2010. Sharing supermodular costs. *Oper. Res.* **58** 1051–1056.

Schulz, Andreas S, Nelson A Uhan. 2013. Approximating the least core value and least core of cooperative games with supermodular costs. *Discrete Optim.* **10** 163 – 180.

Shapley, Lloyd S. 1952. A value for n-person games. Tech. rep., DTIC Document.

Shapley, Lloyd S. 1971. Cores of convex games. *International journal of game theory* **1** 11–26.

Shapley, Lloyd S, Martin Shubik. 1971. The assignment game I: The core. *Internat. J. Game Theory* **1** 111–130.

Tamir, Arie. 1989. On the core of a traveling salesman cost allocation game. *Oper. Res. Lett.* **8** 31–34.

Xu, Dachuan, Ruichun Yang. 2009. A cost-sharing method for an economic lot-sizing game. *Oper. Res. Lett.* **37** 107–110.

Zhang, Jiawei. 2009. Cost allocation for joint replenishment models. *Oper. Res.* **57** 146–156.

Appendix A: Column and Row Generation Approaches for IM Games

We summarize the column generation approach and the row generation approach (LPB algorithm) discussed by Caprara and Letchford (2010) in this section.

For the column generation approach, consider the definition of OCAP by LP (1). Its dual problem is:

$$\min_{\beta}\Big\{\sum_{s\in S}c(s)\beta_s:\sum_{s\ni k}\beta_s=1,\forall k\in V,\beta_s\geq 0,s\in S\Big\}. \tag{35}$$

Denote by $Q^{x\gamma}$ the overall set of feasible solutions of ILP (3), i.e.,

$$Q^{x\gamma}=\big\{x\in\big\{0,1\big\}^{t\times 1},\gamma\in\big\{0,1\big\}^{v\times 1}:Ax\geq B\gamma+D,A'x\geq B'\gamma+D',\gamma=\gamma^s,\forall s\in S\big\}.$$

Then LP (35) can be re-formulated, for the purpose of doing column generation, by enumerating all values in $Q^{x\gamma}$. Specifically, for each $(\bar{x},\bar{\gamma})\in Q^{x\gamma}$, define variable $\beta_{\bar{x},\bar{\gamma}}$ with cost $C\bar{x}$. We will have a master LP

$$\min_{\beta}\Big\{\sum_{(\bar{x},\bar{\gamma})\in Q^{x\gamma}}(C\bar{x})\beta_{(\bar{x},\bar{\gamma})}:\sum_{(\bar{x},\bar{\gamma})\in Q^{x\gamma}}\bar{\gamma}_k\beta_{(\bar{x},\bar{\gamma})}=1,\forall k\in V,\beta_{\bar{x},\bar{\gamma}}\geq 0,(\bar{x},\bar{\gamma})\in Q^{x\gamma}\Big\}. \tag{36}$$

Though the formulation is straightforward, the above column generation is difficult to solve because the pricing problem amounts to optimizing over $Q^{x\gamma}$, which is usually NP-hard in the strong sense.

As to the row generation approach, it needs to identify a set of so-called assignable constraints, analogous to the cutting-plane method for solving IP where tight valid constraints are added to sharpen the LP bound. We let $P_I^x=conv\big\{x\in\big\{0,1\big\}^{t\times 1}:Ax\geq B\mathbf{1}+D,A'x\geq B'\mathbf{1}+D'\big\}$ and $P_I^{x\gamma}=conv\ Q^{x\gamma}$, where function $conv\{\cdot\}$ represents the convex hull of a vector. Note that P_I^x is the convex hull of the integer solutions to (3).

Definition 5. A valid inequality $ax\geq b$ for P_I^x is said to be *assignable* if there exists a valid inequality $ax\geq b'\gamma$ for $P_I^{x\gamma}$ such that $\sum_{k\in V}b'_k=b$.

Theorem 4. *For an IM game (V,c), if there exists an LP $\min_x\{Cx:Ex\geq F\gamma\}$ that gives a lower bound to ILP (3), where all constraints $Ex\geq F\gamma$ are assignable, then vector α_{LP}^{EF} given by*

$$\alpha_{LP}^{EF}(k)=\sum_{l=1}^{m_E}f_{lk}\mu_l^*,\ \forall k\in V,$$

is a stable cost allocation for the IM game, where μ^* is the LP dual variable value for an optimal solution to $\min_x\{Cx:Ex\geq F\mathbf{1}\}$, and m_E is the number of rows of matrix E. In addition, the total shared cost $\sum_{k\in V}\alpha_{LP}^{EF}(k)=\min_x\{Cx:Ex\geq F\mathbf{1}\}$.

In fact, Theorem 4 stands true if μ^* is relaxed to be an optimal solution to the dual of $\min_x\{Cx:Ex\geq F\mathbf{1}\}$. The proof is straightforward based on the results in Caprara and Letchford (2010). We will apply the extended results in our analysis.

Note that $c_{LP}^{EF}(V)=\min_x\{Cx:Ex\geq F\mathbf{1}\}$ gives an LP lower bound of the grand coalition cost $c(V)$. According to Theorem 4, the quality of the LPB cost allocation α_{LP}^{EF} greatly depends on the tightness of constraints set $\{Ex\geq F\mathbf{1}\}$, i.e., the tighter the constraints set is, the better the resulting LPB cost allocation α_{LP}^{EF} is. In addition, if given a basic optimal solution of $\min_x\{Cx:Ex\geq F\mathbf{1}\}$, then the resulting μ^* can be regarded as the shadow prices of constraints $Ex\geq F\mathbf{1}$, and therefore this leads to some LPB cost allocations with strong business insights. Such examples can be seen in the UFL LPB cost allocations.

Four IM games are investigated in Caprara and Letchford (2010), namely, the Uncapacitated Facility Location game, the Rooted and Unrooted Travelling Salesman games and the Vehicle Routing game. For each game, they give a tight constraint set such that the total shared cost $c_{LP}^{EF}(V)$ is no smaller than $c_{LP}(V)$, the LP lower bound of $c(V)$ from the original ILP formulation.