



## INFORMS Journal on Computing

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

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

Lindong Liu, Xiangtong Qi, Zhou Xu

To cite this article:

Lindong Liu, Xiangtong Qi, Zhou Xu (2016) Computing Near-Optimal Stable Cost Allocations for Cooperative Games by Lagrangian Relaxation. INFORMS Journal on Computing 28(4):687-702. <https://doi.org/10.1287/ijoc.2016.0707>

Full terms and conditions of use: <http://pubsonline.informs.org/page/terms-and-conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact [permissions@informs.org](mailto:permissions@informs.org).

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2016, INFORMS

Please scroll down for article—it is on subsequent pages



INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

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

Lindong Liu

The School of Management, University of Science and Technology of China, Hefei China, 230026; and  
Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong,  
[ldliu@ustc.edu.cn](mailto:ldliu@ustc.edu.cn)

Xiangtong Qi

Department of Industrial Engineering and Logistics Management, The Hong Kong University of Science and Technology,  
Kowloon, Hong Kong SAR, [ieemqi@ust.hk](mailto:ieemqi@ust.hk)

Zhou Xu

Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong SAR,  
[lgtxx@polyu.edu.hk](mailto:lgtxx@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.

*Keywords:* game theory; cooperative game; cost allocation; Lagrangian relaxation; facility location game

*History:* Accepted by Karen Aardal, Area Editor for Design and Analysis of Algorithms; received April 2015; revised November 2015, March 2016; accepted March 2016. Published online September 20, 2016.

## 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 (OR), 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 nonemptiness 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  $\gamma$ -core (e.g., Jain and Mahdian 2007), the budget balance constraint is replaced with a  $\gamma$ -budget balance constraint requiring

that the total cost allocated to all players is no less than  $\gamma$  times the minimum cost of the grand coalition, where  $0 < \gamma \leq 1$ . The  $\gamma$ -core is mathematically equivalent to another concept known as the  $\epsilon$ -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  $\gamma$ -core or  $\epsilon$ -approximate core has centered on finding a constant bound on  $\gamma$  or  $\epsilon$  for specific games.

In this paper, we study the idea of  $\gamma$ -core, but with a different focus. Instead of looking for a constant bound on  $\gamma$ , 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  $\gamma$  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 OR 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 the online supplement (available as supplemental material at <http://dx.doi.org/10.1287/ijoc.2016.0707>), are first identified and added to the ILP formulation. However, sometimes it is hard to identify all assignable constraints (e.g., rooted traveling salesman problem (TSP) 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 traveling 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 Lagrangian relaxation based (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 nonlinear 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 subgames, which are relatively easier to solve than the original game. One subgame has a simple optimal solution which can be represented in closed form. The other subgame has some beneficial properties that the original game lacks. In many cases, the minimum cost incurred by each coalition in the subgames is relatively easier to calculate than in the original game. In some cases, the optimal cost allocations of the subgames 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 (1953). Some of the most prominent examples related to OR 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), traveling 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 (UFL) 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 nonemptiness 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 three-approximation algorithm for computing the least core value. With respect to the

$\gamma$ -core and  $\epsilon$ -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. 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 prior 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 reformulate 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  $V = \{1, 2, \dots, v\}$  denotes a set of players, and  $c: S \rightarrow \mathbb{R}$  denotes the characteristic function, with  $S = 2^V \setminus \emptyset$  indicating the set of nonempty 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:  $\sum_{k \in s} \alpha(k) \leq c(s)$ ,  $\forall s \in S$ . 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:

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

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

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

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 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 ILP to define characteristic functions, the OR game allows using nonlinear 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 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  $B' \in \mathbb{R}^{r' \times v},$
- nonnegative 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  $\gamma_k^s = 1$  if  $k \in s$  and  $\gamma_k^s = 0,$  otherwise,  $\forall k \in V,$  such that for all  $s \in S,$  the characteristic function  $c(s)$  is given by the following IP:

$$\begin{aligned} c(s) = \min_x \{ & f(x): Ax \geq B\gamma^s + D, \\ & A'x \geq B'\gamma^s + D', x \in \{0, 1\}^{t \times 1} \}. \end{aligned} \quad (2)$$

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 nonnegative  $D$  and  $D'$  is subadditive, i.e.,  $c(s_1 \cup s_2) \leq c(s_1) + c(s_2)$  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:

$$\begin{aligned} c(s) = \min_x \{ & Cx: Ax \geq B\gamma^s + D, \\ & A'x \geq B'\gamma^s + D', x \in \{0, 1\}^{t \times 1} \}, \end{aligned} \quad (3)$$

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  $\mathbf{0} \leq x \leq \mathbf{1}.$

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 the online supplement. 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 reformulate ILP (3) by identifying a set of so-called assignable constraints  $\{Ex \geq F\gamma\}.$  Then, a cost allocation can be obtained by solving the LP relaxation  $c_{LP}^{EF}(V) = \min\{Cx: Ex \geq F\gamma\}$  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. LRB 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  $x,$  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  $\{A'x \geq B'\gamma^s + D'\}$  in (2) and bringing them into the objective function with nonnegative Lagrangian multiplier  $\lambda,$  we can derive the resulting Lagrangian characteristic function  $c_{LR}(\cdot; \lambda)$  for an OR game as

$$\begin{aligned} c_{LR}(s; \lambda) = \min_x \{ & f(x) - \lambda A'x + \lambda B'\gamma^s + \lambda D': \\ & Ax \geq B\gamma^s + D, x \in \{0, 1\}^{t \times 1} \}, \quad \forall s \in S, \end{aligned} \quad (4)$$

where  $\lambda$  is a nonnegative 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

$$c_{LR}(V; \lambda) = \min_x \{f(x) - \lambda A'x + \lambda B'1 + \lambda D'\} \\ Ax \geq B1 + D, x \in \{0, 1\}^{t \times 1}.$$

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 pseudopolynomial time for any  $s \in S$ . It is known that  $c_{LR}(V; \lambda)$  is a lower bound of  $c(V)$  for any nonnegative  $\lambda$ . To achieve the sharpest lower bound, the Lagrangian dual problem  $d_{LR}(V)$  finds the best Lagrangian multiplier  $\lambda$  that maximizes  $c_{LR}(V; \lambda)$ , i.e.,

$$d_{LR}(V) = \max_{\lambda} \left\{ \min_x \{f(x) - \lambda A'x + \lambda B'1 + \lambda D'\} : \right. \\ \left. Ax \geq B1 + D, x \in \{0, 1\}^{t \times 1} \right\}. \quad (5)$$

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

Under any nonnegative Lagrangian multiplier  $\lambda$ , we can decompose the Lagrangian characteristic function  $c_{LR}(\cdot; \lambda)$  into two subcharacteristic 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

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

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

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

**THEOREM 1.** *Given any nonnegative Lagrangian multiplier  $\lambda$ , if  $\alpha_{LR1}^\lambda$  and  $\alpha_{LR2}^\lambda$  are stable cost allocations for subgames  $(V, c_{LR1}(\cdot; \lambda))$  and  $(V, c_{LR2}(\cdot; \lambda))$ , respectively, then  $\alpha_{LR}^\lambda = \alpha_{LR1}^\lambda + \alpha_{LR2}^\lambda$  is a stable cost allocation for OR game  $(V, c)$ .*

**PROOF.** For any  $s \in S$ , the stability of  $\alpha_{LR1}^\lambda$  and  $\alpha_{LR2}^\lambda$  implies that

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

Therefore we have  $\sum_{k \in s} \alpha_{LR}^\lambda(k) = \sum_{k \in s} [\alpha_{LR1}^\lambda(k) + \alpha_{LR2}^\lambda(k)] \leq c(s)$ . This completes the proof.  $\square$

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 subgames. In fact, we are able to find an optimal stable cost allocation for each subgame under a specific  $\lambda$ . Before introducing the details of handling the two subgames, 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 subgames  $(V, c_{LR1}(\cdot; \lambda^*))$  and  $(V, c_{LR2}(\cdot; \lambda^*))$ , 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

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

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  $\sum_{k \in V} \alpha_{LR}^{\lambda^*}(k) = c_{LR}(V; \lambda^*) \geq c_{LP}(V) \geq \sum_{k \in V} \alpha_{LP}(k)$ .  $\square$

We have the following remarks about Theorem 2.

**REMARK 1.** Theorem 2 implies the competitiveness of the LRB algorithm against the LPB algorithm. 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 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  $\sum_{k \in V} \alpha_{LR}^\lambda(k)$  is no less than the LPB cost allocation value  $\sum_{k \in V} \alpha_{LP}(k)$  are sufficient, but not necessary. In fact, even without condition (1), for a nonoptimal 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 subgames have a nonempty core. As will be shown in the following Lemma 1, subgame 1  $(V, c_{LR1}(\cdot; \lambda))$  always has a nonempty core. However, subgame 2  $(V, c_{LR2}(\cdot; \lambda))$  may have an empty core. Therefore the effectiveness of the LRB algorithm depends on subgame 2, specifically on the value of  $\sum_{k \in V} \alpha_{LR2}^\lambda(k)$ , the cost that can be allocated for game  $(V, c_{LR2}(\cdot; \lambda))$ . Our numerical results show that, even in cases where subgame 2 does not have a nonempty 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  $\sum_{k \in V} \alpha_{LP}(k) = c_{LP}(V) = c_{LR}(V; \lambda^*) \geq \sum_{k \in V} \alpha_{LR}^{\lambda^*}(k)$ .

Remark 3 is also related to the convergence of the LRB algorithm when subgame 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 tighter bound  $c_{LR2}(V; \lambda)$  will also lead to a larger cost allocation value  $\sum_{k \in V} \alpha_{LR2}^\lambda(k)$ . In other words, the final optimal Lagrangian multiplier  $\lambda^*$  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  $\lambda^*$  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 subcharacteristic functions  $c_{LR1}(\cdot; \lambda)$  and  $c_{LR2}(\cdot; \lambda)$  according to (6) and (7).

Step 3. Compute the optimal stable cost allocations  $\alpha_{LR1}^\lambda$  and  $\alpha_{LR2}^\lambda$  for subgames  $(V, c_{LR1}(\cdot; \lambda))$  and  $(V, c_{LR2}(\cdot; \lambda))$ , 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  $\alpha_{LR}^\lambda$  for  $\lambda \in \Lambda$ .

Note that if subgame 2 has a nonempty core for the optimal Lagrangian multiplier  $\lambda^*$ , we only need to use the final Lagrangian multiplier  $\lambda^*$  in  $\Lambda$  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  $\lambda$  values, our experience shows that it is useful to select five or six values from different iterations.

#### 4.2. Solving Subgame 1

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

LEMMA 1. For subgame 1  $(V, c_{LR1}(\cdot; \lambda))$ , a vector  $\alpha_{LR1}^\lambda$  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  $(V, c_{LR1}(\cdot; \lambda))$ , 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  $\alpha_{LR1}^\lambda(s)$ .  $\square$

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

#### 4.3. Solving Subgame 2

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

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  $(V, c_{LR2}(\cdot; \lambda))$  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 reformulating the master problem by treating each feasible solution of the optimization problem as a column (see LP (A.2) in the online supplement for details). The reformulation 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  $\alpha_{LR2}^\lambda$  for subgame 2 is the optimal solution to the corresponding OCAP,

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

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:

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

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 that Algorithm 2 can solve (9) to optimality and generate an optimal cost allocation for subgame 2.

ALGORITHM 2. A CGB algorithm to compute the optimal stable cost allocation  $\alpha_{LR2}^\lambda$  for game  $(V, c_{LR2}(\cdot; \lambda))$  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  $\pi^*$ .

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

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

*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, therefore 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  $\alpha_{LR2}^\lambda$  for game  $(V, c_{LR2}(\cdot; \lambda))$ :

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

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  $\alpha_{LR2}^\lambda$  computed by the CGB algorithm is an optimal stable cost allocation for subgame 2  $(V, c_{LR2}(\cdot; \lambda))$ .

The CGB algorithm above provides a general approach to computing an optimal stable cost allocation for subgame 2. However, for some special cases, it is possible to obtain an optimal or core cost allocation for subgame 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  $(V, c_{LR2}(\cdot; \lambda))$ .

Second, when subgame 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, 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  $c_{LR2}(s \cup \{a\}; \lambda) + c_{LR2}(s \cup \{b\}; \lambda) \geq c_{LR2}(s; \lambda) + c_{LR2}(s \cup \{a, b\}; \lambda)$  holds for any coalition  $s \in V \setminus \{a, b\}$ .

We emphasize these two cases because subgame 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 subgame 2.

## 5. Implementations for Facility Location Games

We will illustrate the LRB cost allocation algorithm on two different facility location games, namely, the UFL game and the nonlinear single source capacitated facility location (NLCFL) game. The UFL game represents the case where 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.



**Table 1** Notation Used in the UFL Game

$M$	The set of potential facility sites, $M = \{1, 2, \dots, m\}$ .
$N$	The set of customer points as well as game players, $N = \{1, 2, \dots, n\}$ .
$c_{ij}$	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, $s \subseteq N$ .
$\gamma^s$	Incidence vector $[\gamma_1^s, \gamma_2^s, \dots, \gamma_n^s]^T$ , where $\gamma_j^s = 1$ if player $j$ is in coalition $s$ and $\gamma_j^s = 0$ 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 $j \in N$ .

### 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.

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

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

$$\text{s.t.} \quad \sum_{i \in M} u_{ij} \geq \gamma_j^s, \quad \forall j \in N, \quad (12)$$

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

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

In the previous 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 a UFL problem. In view of Definition 1, we see that the UFL game  $(N, c_{\text{UFL}})$  is an OR game  $(V, c)$  with  $V = N$  and  $c = c_{\text{UFL}}$ . Specifically, decision variables  $x$  in  $c$  are now  $[v; u]$  in  $c_{\text{UFL}}$ , and the specific expressions of matrices  $C, A, A', B, B', D, D'$  can be obtained by writing  $c_{\text{UFL}}$  using matrices. In particular,  $D$  and  $D'$  are now  $\mathbf{0}$ , therefore the game  $(N, c_{\text{UFL}})$  is subadditive. This is also true for the NLCFL game, which we 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  $c_{\text{UFL}}(N)$ . To perform some in-depth analysis, we give more details on using the LPB algorithm to calculate the optimal stable cost allocation.

In  $c_{\text{UFL}}(s)$ , 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  $c_{\text{UFL}}(N)$  as follows:

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

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

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

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

We sequentially label constraints (15)–(17) from 1 to  $n, n+1$  to  $n+mn$  and  $n+mn+1$  to  $n+2mn$ , respectively. For  $c_{\text{LP\_UFL}}(N)$ , we consider its dual LP. Let  $\mu_k$  be the dual variable corresponding to the  $k$ -th constraint of  $c_{\text{LP\_UFL}}(N)$ , and  $\mu^*$  an optimal solution to the dual LP. According to the row-generation approach in the online supplement, we have the following lemma.

**LEMMA 3.** For a UFL game, the LPB cost allocation  $\alpha_{\text{LP\_UFL}}$  given by

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

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

We note that a simpler way to obtain one optimal stable cost allocation to the UFL game is to solve  $c_{\text{LP\_UFL}}(N)$  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 subgame 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  $c_{\text{UFL}}(s)$ , we add a set of new constraints

$$\{u_{ij} \leq \gamma_j^s: \forall i \in M, j \in N\}, \quad (18)$$

and then bring constraints  $\{\sum_{i \in M} u_{ij} \geq \gamma_j^s; j \in N\}$  into the objective function with nonnegative Lagrangian multiplier  $\sigma$  to derive the UFL Lagrangian characteristic function,

$$c_{LR\_UFL}(s; \sigma) = \min_{v, u} \left\{ \sum_{i \in M} f_i v_i + \sum_{i \in M} \sum_{j \in N} (c_{ij} - \sigma_j) u_{ij} + \sum_{j \in N} \sigma_j \gamma_j^s \right\}$$

s.t.  $u_{ij} - v_i \leq 0, \quad \forall i \in M, j \in N,$   
 $u_{ij} \leq \gamma_j^s, \quad \forall i \in M, j \in N,$   
 $v_i, u_{ij} \in \{0, 1\}, \quad \forall i \in M, j \in N.$

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  $\sum_{i \in M} \sum_{j \in N} (c_{ij} - \sigma_j) u_{ij}$  by  $\sum_{i \in M} \sum_{j \in s} (c_{ij} - \sigma_j) u_{ij}$  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 nonnegative Lagrangian multiplier  $\sigma$ , 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 subgame 1  $(N, c_{LR1\_UFL}(\cdot; \sigma))$  and UFL subgame 2  $(N, c_{LR2\_UFL}(\cdot; \sigma))$ .

For the UFL subgame 1, its characteristic function is

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

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

For the UFL subgame 2, its characteristic function is as follows:

$$c_{LR2\_UFL}(s; \sigma) = \min_{v, u} \sum_{i \in M} f_i v_i + \sum_{i \in M} \sum_{j \in N} (c_{ij} - \sigma_j) u_{ij}$$

s.t.  $u_{ij} - v_i \leq 0, \quad \forall i \in M, j \in N,$  (20)  
 $u_{ij} \leq \gamma_j^s, \quad \forall i \in M, j \in N,$   
 $v_i, u_{ij} \in \{0, 1\}, \quad \forall i \in M, j \in N.$

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

LEMMA 4. UFL subgame 2  $(N, c_{LR2\_UFL}(\cdot; \sigma))$  is submodular.

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

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

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

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

Let  $\rho(x) = \min\{0, x\}$  and  $x_{\hat{s}} = f_i + \sum_{j \in \hat{s}} \min\{0, c_{ij} - \sigma_j\}$  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.  $\square$

Due to the submodularity of UFL subgame 2, one can easily compute its core cost allocation, denoted as  $\alpha_{LR2\_UFL}^\sigma$ , by the greedy algorithm mentioned in Section 4.3. Under the optimal Lagrangian multiplier  $\sigma^*$ , we can derive the optimal UFL LRB cost allocation given by  $\alpha_{LR\_UFL}^{\sigma^*} = \alpha_{LR1\_UFL}^{\sigma^*} + \alpha_{LR2\_UFL}^{\sigma^*}$ . Since UFL subgames 1 and 2 have nonempty 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  $\alpha_{LR\_UFL}^{\sigma^*} = \sigma^* + \alpha_{LR2\_UFL}^{\sigma^*}$  is optimal. In addition, 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 and LPB cost allocation sets 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 show 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  $\mu^*$  together with some  $\delta^*$  form an optimal solution to the following dual problem of  $c_{LP\_UFL}(N)$ :

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

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

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

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 nonnegative  $\sigma$ , 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  $\mu^*$  is an optimal Lagrangian multiplier. The resulting LRB cost allocation is then given by  $\alpha_{LR\_UFL}(j) = \mu_j^* + \alpha_{LR2\_UFL}(j)$  for  $j \in N$ . Note that for each  $s \in S$ , since  $c_{LR2\_UFL}(s; \mu^*) \leq 0$  and  $c_{LR2\_UFL}(s; \mu^*) \geq c_{LR2\_UFL}(N; \mu^*) = 0$ , we have  $c_{LR2\_UFL}(s; \mu^*) = 0$ , which leads to  $\alpha_{LR2\_UFL}(j) = 0$  for  $j \in N$ . Therefore we obtain that  $\alpha_{LR\_UFL} = \mu^*$ , implying that each LPB cost allocation  $\mu^*$  belongs to the LRB cost allocation set. This completes the proof of Theorem 3.  $\square$

**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.

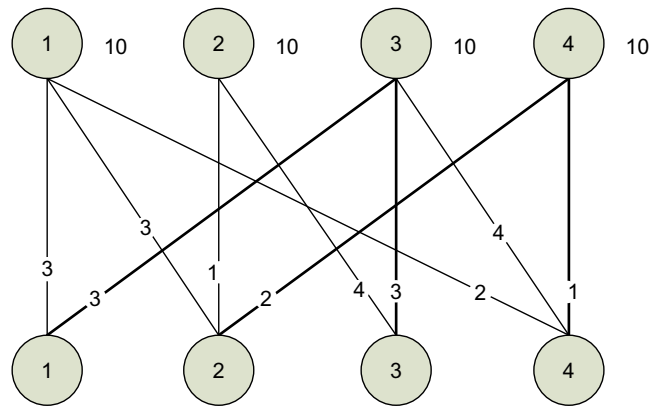


Figure 1 (Color online) An Example of a UFL Game

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.

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 UFL 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 3.4 GHz and 16 GB 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

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

**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 $s$ , $s_c \subseteq N$ .
$\gamma^s$	Incidence vector $[\gamma_1^{s_f}, \gamma_2^{s_f}, \dots, \gamma_m^{s_f}, \gamma_1^{s_c}, \gamma_2^{s_c}, \dots, \gamma_n^{s_c}]^T$ , 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, $\forall j \in N$ , $s_f \subseteq M$ , $s_c \subseteq N$ .

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 an 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  $\theta_i[h_i n_i - l_i(n_i)^2]$  to measure the operational cost, where  $n_i$  is the number of customers served by facility  $i$ , and  $\theta_i$ ,  $h_i$ , 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 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 nonlinear cost function, we can demonstrate the broad range of applications to which our LRB algorithm can be applied.

We need the additional notation shown in Table 3 in defining the NLCFL game.

**DEFINITION 4.** An NLCFL game  $(M \cup N, c_{\text{NLCFL}})$  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  $c_{\text{NLCFL}}(s)$  is determined by NLP:

$$c_{\text{NLCFL}}(s_f \cup s_c) = \min_{v, u} \left\{ \sum_{i \in M} f_i v_i + \sum_{i \in M} \sum_{j \in N} c_{ij} u_{ij} + \sum_{i \in M} \theta_i \left[ \sum_{j \in N} h_i u_{ij} - l_i \left( \sum_{j \in N} u_{ij} \right)^2 \right] \right\} \quad (24)$$

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

$$\sum_{j \in N} q_j u_{ij} - Q_i v_i \leq 0, \quad \forall i \in M, \quad (26)$$

$$v_i \leq \gamma_i^{s_f}, \quad \forall i \in M, \quad (27)$$

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

Compared with  $c_{\text{UFL}}(s)$ ,  $c_{\text{NLCFL}}(s_f \cup s_c)$  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  $c_{\text{NLCFL}}(s_f \cup s_c)$  has a nonlinear 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  $c_{\text{NLCFL}}(s_f \cup s_c)$ , by adding a new set of constraints

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

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

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

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

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

For NLCFL subgame 1, the characteristic function is

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

According to Lemma 1, we can derive the core cost allocation for game  $(M \cup N, c_{LR1\_NLCFL}(\cdot; \sigma))$ , 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 because they do not need to be served. The cost allocation is given by  $\alpha_{LR1\_NLCFL}^\sigma(j) = \sigma_j$  for all  $j \in N$ , and  $\alpha_{LR1\_NLCFL}^\sigma(i) = 0$  for all  $i \in M$ .

For NLCFL subgame 2, the characteristic function is

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

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  $i \in s_f$ ,

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

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 nonlinear 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  $-\theta_i l_i (\sum_{j \in s_c} u_{ij})^2$ , 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 pseudopolynomial time  $O(Q_i n^2)$ . To be specific, we define  $F_i^{s_c}(j, k, q)$  as the minimum value of  $\sum_{j' \in s_c, j' \leq j} (c_{ij'} - \sigma_{j'} + \theta_i h_i) u_{ij'}$  such that  $\sum_{j' \in s_c, j' \leq j} u_{ij'} = k$  and  $\sum_{j' \in s_c, j' \leq j} q_{j'} u_{ij'} \leq q$ . In other words,  $F_i^{s_c}(j, k, q)$  represents the minimum item value packed by including

exactly  $k$  items from set  $\{1, 2, \dots, j\}$  within capacity  $q$ . The dynamic programming recursion is as follows:

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

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  $\psi_i(s_c; \sigma) = \min_{k \leq |s_c|} \{0, f_i + F_i^{s_c}(n, k, Q_i) - \theta_i l_i k^2\}$ , 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  $\alpha_{LR2\_NLCFL}^\sigma$  for NLCFL subgame 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

$$\begin{aligned} \min_{v, u} \quad &\sum_{i \in s_f} \left\{ f_i v_i + \sum_{i \in s_f} \sum_{j \in s_c} (c_{ij} - \sigma_j + \theta_i h_i) u_{ij} \right. \\ &\quad \left. - \sum_{i \in s_f} \theta_i l_i \left( \sum_{j \in s_c} u_{ij} \right)^2 - \sum_{k \in M \cup N} \gamma_k^s \pi_k^* \right\} \\ \text{s.t.} \quad &\sum_{j \in s_c} q_j u_{ij} \leq Q_i v_i, \quad \forall i \in s_f, \\ &v_i, u_{ij} \in \{0, 1\}, \quad \forall i \in s_f, j \in s_c, \end{aligned} \quad (32)$$

with  $\pi^*$  being the optimal dual of the corresponding master problem for NLCFL subgame 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  $\bar{s} = \bar{s}_f \cup \bar{s}_c$  with a negative reduced cost, we can consider the following ILP where binary variables  $v_i$  and  $\gamma_j$  indicate whether  $\bar{s}$  includes facility players  $i$  and customer

player  $j$  with  $f'_i = f_i - \pi_i^*$ ,  $c'_{ij} = c_{ij} - \sigma_j + \theta_i h_i$ ,  $l'_i = \theta_i l_i$ , and  $\pi'_j = -\pi_j^*$ .

$$\begin{aligned} \min_{v, u, \gamma} R(v, u, \gamma) = \min_{v, u} & \left\{ \sum_{i \in M} f'_i v_i + \sum_{i \in M} \sum_{j \in N} c'_{ij} u_{ij} \right. \\ & \left. - \sum_{i \in M} l'_i (\sum_{j \in N} u_{ij})^2 + \sum_{j \in N} \gamma_j \pi'_j \right\} \\ \text{s.t.} & \sum_{j \in N} q_j u_{ij} \leq Q_i v_i, \quad \forall i \in M, \\ & u_{ij} \leq \gamma_j, \quad \forall i \in M, j \in N, \\ & v_i, u_{ij}, \gamma_j \in \{0, 1\}, \\ & \quad \quad \quad \forall i \in M, j \in N. \end{aligned} \quad (33)$$

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 following properties.

LEMMA 5. For (33), 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  $c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2] > 0$ , 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  $\pi'_j + \sum_{i \in M} \min\{c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2], 0\} \geq 0$ , then  $\gamma_j = 0$ ,  $u_{ij} = 0$ ,  $\forall i \in M$ .

(3) For each  $i \in M$ , solve a nonlinear 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  $\omega^i$  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  $c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2] > 0$  in a feasible solution of ILP (33), one can directly set  $u_{ij} = 0$ , and derive another feasible solution under which the objective function value is reduced by at least  $c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2]$ .

Second, if there exists a customer  $j$  such that  $\gamma_j = 1$  and  $\pi'_j + \sum_{i \in M} \min\{c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2], 0\} \geq 0$  in a feasible solution of ILP (33), 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  $\pi'_j + \sum_{i \in M} \min\{c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2], 0\}$ .

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 (33), 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  $\omega^i + f'_i$ .  $\square$

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 (33).

After deriving the optimal stable cost allocations  $\alpha_{\text{LR1\_NLCFL}}^\sigma$  and  $\alpha_{\text{LR2\_NLCFL}}^\sigma$  for NLCFL subgames 1 and 2, respectively, we can compute a stable cost allocation  $\alpha_{\text{LR\_NLCFL}}^\sigma = \alpha_{\text{LR1\_NLCFL}}^\sigma + \alpha_{\text{LR2\_NLCFL}}^\sigma$  for an NLCFL game according to Theorem 1. Furthermore, by Theorem 2, the corresponding LRB cost allocation value  $\sum_{k \in \text{MUN}} \alpha_{\text{LR\_NLCFL}}^{\sigma^*}(k)$  is equal to the Lagrangian relaxation lower bound  $c_{\text{LR\_NLCFL}}(M \cup N; \sigma^*)$  if  $\sigma^*$  is the optimal Lagrangian multiplier and  $(M \cup N; c_{\text{LR2\_NLCFL}}(\cdot; \sigma^*))$  has a nonempty 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  $h_1 = h_2 = \dots = h_m = n^2$ ,  $l_1 = l_2 = \dots = l_m = 1$ , and  $\theta_1 = \theta_2 = \dots = \theta_m = \theta$  to measure the operational cost, where  $\theta$  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 2,500.

To show the effectiveness of the LRB cost allocation, ideally we need to compare the total shared cost against the grand coalition cost  $c_{\text{NLCFL}}(M \cup N)$ . However, the grand coalition cost is only available in the benchmark data set for instances with  $\theta = 0$ . 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  $\theta = 0$ , which is available in Bachrach et al. (2009). The second feasible solution is derived from the optimal solution of  $c_{\text{LR2\_NLCFL}}(M \cup N; \sigma)$ . Note that this optimal solution might be infeasible for the centralized problem  $c_{\text{NLCFL}}(M \cup N)$ , 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.

In Table 4, "LRCA" represents the best found LRB cost allocation value  $\sum_{k \in \text{MUN}} \alpha_{\text{LR\_NLCFL}}^\sigma(k)$  under different  $\sigma$ , and "LRB" represents the best Lagrangian

**Table 4** Performance of LRB Cost Allocation Algorithm for the NLCFL Game

Capacity	$\theta$	LRCA/BFCS (%)			LRCA/LRB (%)			Total time(s)		
		Average	Max	Min	Average	Max	Min	Average	Max	Min
$Q = 10$	0.00	98.79	99.12	98.33	100.00	100	100.00	—	—	—
	0.01	99.64	99.70	99.55	100.00	100	100.00	5,683	6,838	4,987
	0.10	99.87	99.89	99.78	100.00	100	100.00	5,690	6,834	4,980
	0.50	99.90	99.92	99.87	100.00	100	100.00	5,742	6,814	5,036
	1.00	99.91	99.95	99.89	100.00	100	100.00	5,822	6,983	4,764
$Q = 20$	0.00	98.32	99.30	97.66	100.00	100	99.95	—	—	—
	0.01	99.61	99.76	99.48	100.00	100	100.00	9,925	10,478	9,485
	0.10	99.83	99.85	99.82	100.00	100	100.00	9,835	10,458	9,322
	0.50	99.85	99.88	99.84	100.00	100	99.99	9,825	10,487	9,315
	1.00	99.89	99.92	99.87	100.00	100	100.00	9,973	11,154	9,812
$Q = 30$	0.00	95.25	96.95	93.93	100.00	100.00	100	—	—	—
	0.01	99.02	99.15	98.82	100.00	100	99.99	11,686	12,831	10,410
	0.10	99.72	99.77	99.63	99.99	100	99.95	11,755	12,816	10,421
	0.50	99.81	99.87	99.78	100.00	100	100.00	11,485	13,064	10,277
	1.00	99.88	99.92	99.86	100.00	100	100.00	12,621	14,371	11,955

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  $\theta$ . 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  $\theta$  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 subgame 2 is not submodular, in general, column “LRCA/LRB” suggests that almost every NLCFL subgame 2 has a nonempty core. This indicates that even in cases where condition (2) of Theorem 2 does not hold, the LRB cost allocation still has a great chance to achieve the Lagrangian lower bound. As for time efficiency, we can see that the computational time tends to increase with  $Q$  and  $\theta$ . Among all instances, the longest computation time is around four hours.

For instances with  $\theta = 0$ , 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).

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  $\sigma^*$  in a UFL game, Theorem 2 ensures the optimality of the stable cost allocation corresponding to  $\sigma^*$  because UFL subgame 2 has a nonempty core. However, the NLCFL subgame 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  $\theta = 0$ , and compare LRB cost allocations that are obtained by using different  $\Lambda$  sets of Lagrangian multipliers. Table 6 reports

**Table 5 LPB vs. LRB Cost Allocations for the NLCFL Game with  $\theta = 0$  (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

**Table 6 Comparisons of LRB Cost Allocations Derived from Different  $\Lambda$  Sets of Lagrangian Multipliers**

Pairs of $\Lambda$ sets		Improved	Declined	Unchanged
$\Lambda_1$	$\Lambda_2$ (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

the number of instances whose LRB cost allocations are improved, declined, and unchanged, respectively, when  $\Lambda$  is changed from a baseline set  $\Lambda_2$  to another set  $\Lambda_1$ . Each  $\sigma^i$  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  $\Lambda$  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.

In summary, from the computational experiments, we can conclude that the LRB algorithm is 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 LP relaxation and duality techniques. We take the different approach of investigating Lagrangian relaxation techniques. Besides the competitiveness of the Lagrangian bound over the LP 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 LP-based algorithm. In fact, we have also successfully implemented our algorithm on other typical games, such as TSP games, e.g., see Liu (2015).

## Supplemental Material

Supplemental material to this paper is available at <http://dx.doi.org/10.1287/ijoc.2016.0707>.

## Acknowledgments

The authors appreciate the constructive and detailed comments by two anonymous reviewers. The work of Lindong Liu and Xiangtong Qi is partially supported by Hong Kong RGC GRF [Grant 618311], and the work of Zhou Xu is partially supported by the Hong Kong Polytechnic University [Grant G-SB12].

## References

- Ahuja RK, Magnanti TL, Orlin JB (1993) *Network Flows: Theory, Algorithms, and Applications* (Prentice-Hall, Englewood Cliffs, NJ).
- Aydinliyim T, Vairaktarakis GL (2010) Coordination of outsourced operations to minimize weighted flow time and capacity booking costs. *Manufacturing Service Oper. Management* 12(2): 236–255.
- Bachrach Y, Elkind E, Meir R, Pasechnik D, Zuckerman M, Rothe J, Rosenschein JS (2009) The cost of stability in coalitional games. Mavronicolas M, Papadopolou VG, eds. *Internat. Sympos. Algorithmic Game Theory SAGT 2009*, Lecture Notes in Computer Science, Vol. 5814 (Springer, Berlin), 122–134.
- Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* 46(3):316–329.
- Beresnev V, Kochetov Y, Pashchenko M, Kononov A, Goncharov E, Plyasunov A, Kochetova N, Alekseeva E, Ivanenko D (2006) Benchmark. Accessed June 24, 2016, <http://www.math.nsc.ru/AP/benchmarks/english.html>.
- Bläser M, Ram LS (2008) Approximately fair cost allocation in metric traveling salesman games. *Theory Comput. Syst.* 43(1):19–37.
- Cai X, Vairaktarakis GL (2012) Coordination of outsourced operations at a third-party facility subject to booking, overtime, and tardiness costs. *Oper. Res.* 60(6):1436–1450.
- Caprara A, Letchford AN (2010) New techniques for cost sharing in combinatorial optimization games. *Math. Programming* 124(1–2):93–118.
- Chen X (2009) Inventory centralization games with price-dependent demand and quantity discount. *Oper. Res.* 57(6):1394–1406.
- Chen X, Zhang J (2009) A stochastic programming duality approach to inventory centralization games. *Oper. Res.* 57(4):840–851.
- Deng X, Ibaraki T, Nagamochi H (1999) Algorithmic aspects of the core of combinatorial optimization games. *Math. Oper. Res.* 24(3):751–766.



- Edmonds J (1970) Submodular functions, matroids, and certain polyhedra. Guy R, ed. *Combinatorial Structures & Their Applications* (Gordon and Breach, New York), 69–87.
- Engvall S, Göthe-Lundgren M, Värbrand P (2004) The heterogeneous vehicle-routing game. *Transportation Sci.* 38(1):71–85.
- Faigle U, Kern W (1993) On some approximately balanced combinatorial cooperative games. *Zeitschrift für Oper. Res.* 38(2): 141–152.
- Faigle U, Kern W, Paulusma D (2000) Note on the computational complexity of least core concepts for min-cost spanning tree games. *Math. Methods Oper. Res.* 52(1):23–38.
- Faigle U, Fekete SP, Hochstättler W, Kern W (1998) On approximately fair cost allocation in Euclidean TSP games. *OR Spectrum* 20(1):29–37.
- Geoffrion A (1974) Lagrangian relaxation and its uses in integer programming. *Math. Programming Stud.* 2:82–114.
- Goemans MX, Skutella M (2000) Cooperative facility location games. *J. Algorithms* 50(2):76–85.
- Göthe-Lundgren M, Jörnsten K, Värbrand P (1996) On the nucleolus of the basic vehicle routing game. *Math. Programming* 72(1):83–100.
- Granot D, Huberman G (1981) Minimum cost spanning tree games. *Math. Programming* 21(1):1–18.
- Hartman BC, Dror M, Shaked M (2000) Cores of inventory centralization games. *Games Econom. Behav.* 31(1):26–49.
- He S, Zhang J, Zhang S (2012) Polymatroid optimization, submodularity, and joint replenishment games. *Oper. Res.* 60(1): 128–137.
- 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.
- Kern W, Paulusma D (2003) Matching games: The least core and the nucleolus. *Math. Oper. Res.* 28(2):294–308.
- Kolen A (1983) Solving covering problems and the uncapacitated plant location problem on trees. *Eur. J. Oper. Res.* 12(3):266–278.
- Li G, Li Y, Shu J, Xu D (2013) A cross-monotonic cost-sharing scheme for the concave facility location game. *J. Global Optim.* 56(4):1325–1334.
- Liu L (2015) To stabilize grand coalitions in unbalanced cooperative games. Doctoral dissertation, The Hong Kong University of Science and Technology.
- Liu Z (2009) Complexity of core allocation for the bin packing game. *Oper. Res. Lett.* 37(4):225–229.
- Mallozzi L (2011) Cooperative games in facility location situations with regional fixed costs. *Optim. Lett.* 5(1):173–181.
- Martínez-de Albéniz FJ, Rafels C, Ybern N (2013) A procedure to compute the nucleolus of the assignment game. *Oper. Res. Lett.* 41(6):675–678.
- Maschler M, Peleg B, Shapley LS (1979) Geometric properties of the kernel, nucleolus, and related solution concepts. *Math. Oper. Res.* 4(4):303–338.
- Meir R, Rosenschein JS, Malizia E (2011) Subsidies, stability, and restricted cooperation in coalitional games. *Proc. 22nd Internat. Joint Conf. Artificial Intelligence, IJCAI'11*, Vol. 1 (AAAI Press, Palo Alto, CA), 301–306.
- Owen G (1975) On the core of linear production games. *Math. Programming* 9(1):358–370.
- Potters JA, Curiel IJ, Tijs SH (1992) Traveling salesman games. *Math. Programming* 53(1–3):199–211.
- Puerto J, Tamir A, Perea F (2011) A cooperative location game based on the 1-center location problem. *Eur. J. Oper. Res.* 214(2): 317–330.
- Puerto J, Tamir A, Perea F (2012) Cooperative location games based on the minimum diameter spanning Steiner subgraph problem. *Discrete Appl. Math.* 160(7):970–979.
- Schulz AS, Uhan NA (2010) Sharing supermodular costs. *Oper. Res.* 58(4, Pt. 2):1051–1056.
- Schulz AS, Uhan NA (2013) Approximating the least core value and least core of cooperative games with supermodular costs. *Discrete Optim.* 10(2):163–180.
- Shapley LS (1953) A value for  $n$ -person games. *Contributions Theory Games* 2:307–317.
- Shapley LS (1971) Cores of convex games. *Internat. J. Game Theory* 1(1):11–26.
- Shapley LS, Shubik M (1971) The assignment game I: The core. *Internat. J. Game Theory* 1(1):111–130.
- Tamir A (1989) On the core of a traveling salesman cost allocation game. *Oper. Res. Lett.* 8(1):31–34.
- Xu D, Yang R (2009) A cost-sharing method for an economic lot-sizing game. *Oper. Res. Lett.* 37(2):107–110.
- Zhang J (2009) Cost allocation for joint replenishment models. *Oper. Res.* 57(1):146–156.