

基于拉格朗日松弛理论的非平衡博弈费用分摊算法

刘林冬、齐向彤、徐宙

摘要

针对核为空的合作博弈费用分摊问题,本文研究了在满足大联盟稳定条件下将分摊给所有参与者的总费用最大化的算法。此类问题的应用之一就是找到稳定大联盟所需的最低补贴费用。为求解此问题,我们提出了基于拉格朗日松弛理论的非平衡博弈费用分摊算法。与现有的基于线性松弛理论的算法相比,本文提出的算法可以得到更好结果,并且其适用范围也更加广泛。为证明该算法的有效性,本文研究了两种不同的设施选址博弈。结果表明,该算法不仅保证了新算法的效用,而且很大程度的拓展深化了费用分摊问题的研究广度与深度。

关键词: 博弈论、合作博弈、费用分摊、拉格朗日松弛、设施选址博弈

1 引言

合作博弈理论解决了多个独立决策人之间的合作问题。它在经济、金融、运筹学 (OR) 和电信等多个领域都有应用。在降低成本的问题中, (具有可转移效用的) 合作博弈可以大致表述为: 有 n 个参与者, 每个参与者都需要利用其资源以最小的费用完成既定的任务目标。为了降低总成本, 有些 (或所有) 参与者可能会通过结盟的方式集中资源实现任务目标。所有参与者的集合被称为大联盟。主要的问题是如何以“公平”的方式分摊大联盟的费用, 使得任何一位参与者不会退出。定义费用分摊的“公平”有不同的方法, 但是一个根本的概念就是联盟的稳定性。保证联盟的稳定性需满足分摊给每个联盟的成本 (分摊给联盟中每个参与者的成本之和) 不超过联盟成员未加入联盟时的最低成本。另外, 为保证公平, 还需满足分摊给所有参与者的总成本等于大联盟最低成本的预算平衡约束。合作博弈的核心是联盟的费用分配集满足 (1) 联盟稳定性和 (2) 预算平衡约束。如果至少存在一个这样的分配, 那该博弈的核心就不为空, 大联盟就是稳定的。目前已有许多方法可以判断各种合作博弈的核是否为空。然而许多合作博弈问题都是空核。对于空核博弈, 已经有人提出了用替换的概念来给出解决方案。其基本思想是松弛核心定义的两个条件之一。例如, 松弛联盟稳定性的最小核概念 *least core* (e.g., Maschler et al. 1979, Kern and Paulusma 2003, Schulz and Uhan 2010, 2013)。在最小核概念中, 分摊给每个联盟的费用不超过 z 与联盟的最小成本之和, 其中 z 是一个最小化的参数。在另一个被称为 γ -核 (e.g., Jain and Mahdian 2007) 的概念中, 预算平衡约束被替换为 γ 预算平衡约束。 γ 预算平衡约束指分摊给所有参与者的总成本不小于 γ 乘以大联盟的最低费用, 其中 $0 < \gamma \leq 1$ 。在数学领域 γ -核等同于另一个概念 ϵ -近似核 (e.g., Faigle et al. 1998, Bläser and Ram 2008)。这一概念约束预算平衡, 松弛联盟稳定性, 使分配给每个联盟的成本之和不超过 $(1 + \epsilon)$ 倍的联盟的最低费用。总之, 研究 γ -核或 ϵ -近似核的关键点是在具体的博弈中找到 γ 或

ϵ 的连续边界。在本文中，我们探究了 γ -核。但与原文的关注点不同，对于由 Caprara 和 Letchford 提出的最优费用分摊问题 (OCAP)，我们设计了一种算法来精确计算任意给定博弈的最佳 γ 值，而不是只寻找 γ 的连续边界。特别地，OCAP 问题尝试最大化分摊给所有参与者的总费用 Caprara and Letchford (2010)，这可以等价地看作计算大联合政府为保持社会稳定最优状态的“费用”。在大联合政府中，代表社会福利的第三方愿意为大联合政府的稳定提供补助。在这里，第三方可能是政府机构，而参与者是一群私营公司。或者，第三方是一个大公司的总部，而参与者是不同的分公司。在这种情况下，OCAP 的目标相当于最小化分配给所有参与者的总费用与大联盟总费用之间的差值，而这一差值则由第三方提供补助。简单地说，找到 OCAP 的解至少有两个困难。首先，普通线性规划 (LP) 公式的约束数量随着参与者数量的增加而指数级增加，即 n 个参与者需要 2^n 个约束。其次，对于给定的费用分摊方案，仅验证其中一个 LP 约束是否被满足，就需要解 NP-hard 的优化问题。因此，直接用 LP 公式求解 OCAP 是非常困难的。对于空核的合作博弈问题，首先想到的就是求解 OCAP。然而，该问题一直未得到充分的研究，直到 Caprara and Letchford (2010) 提出了一个基于线性松弛理论的算法 (LPB 算法)。其基本思想是建立一个稳定的费用分摊方法。该方法可以得到大联盟成本的线性松弛下界。从理论上讲，LPB 算法是可以最好地解决 OCAP，但前提是识别并添加 ILP 公式中的所有“可分配”约束。然而，有时很难识别所有可分配的约束。如果没有多项式时间间隔算法来处理可分配约束的指数数量，即使在所有可分配的约束都可以被识别的情况下，LPB 算法仍然是不适用的。例如，研究的无根旅行商博弈。本文基于拉格朗日松弛理论，提出了一个新的算法处理 OCAP。该方法 (LRB 算法) 虽然也试图找一种费用分摊方案使大联盟费用达到更低，但与 LPB 算法相比具有以下优势。首先，它是一个通用算法，可以应用于比 Caprara and Letchford (2010) 所描述的更广泛的合作博弈。而且与 LPB 算法只能解决线性目标函数不同，新算法也适用于具有非线性目标函数的问题。其次，由于拉格朗日松弛界不比松弛 LP 解提供的界差，对于模型相同的大联盟问题 LRB 算法可以找到比 LPB 算法更好的解。在一定程度上，这避免了 LPB 算法中识别所有“可分配”约束的要求。此外，即使在某些可找到所有可分配约束情况下，LRB 算法仍然是有价值的。因为它可以为参与者提供可替代的最优费用分摊方案，从而提供更多的评估选择。第三，在求解 OCAP 时，LRB 算法利用分解的方法将原博弈问题分解为两个子博弈。子博弈 1 可以得到用闭型表示的最优解。子博弈 2 可以具有一些原博弈所缺的特性。在许多情况下与原博弈相比，子博弈中每个联盟产生的最小费用更容易计算。在某些情况下，子博弈的最优成本分配是多项式可解的，而原始博弈的最优成本分配不是多项式可解的。最后，LRB 算法是依赖于几十年来求解拉格朗日对偶问题的大量研究。在将它应用到 OCAP 时，我们可以充分利用各种加速收敛和生成更清晰界限的方法。这些结果可以在 LRB 算法的第一步中进行合并。

2 文献综述

自 Shapley (1952) 的开创性工作以来，合作博弈的研究一直在深入研究。与运筹学应用相关的领域最为突出，主要包括分配博弈 (Shapley and Shubik 1971, Martínez-de Albéniz et al. 2013)、装箱博弈 (Faigle and Kern 1993, Liu 2009)、线性生产博弈 (Owen 1975)、最小生成树博弈 (Granot and Huberman 1981)、旅行商博弈 (Tamir 1989, Potters et al. 1992)、车辆路径规划博弈 (Göthe-Lundgren et al. 1996,

Engevall et al. 2004)、库存博弈 (Hartman et al. 2000, Chen 2009, Chen and Zhang 2009, He et al. 2012, Zhang 2009)、生产外包博弈 (Aydinliyim and Vairaktarakis 2010, Cai and Vairaktarakis 2012) 以及一些图形包装和覆盖博弈 (Deng et al. 1999) 等。这些博弈的研究重点通常是核心的存在性。本文以设施选址博弈为例。Kolen (1983) 给出了一个早期的重要结果。他指出, 对于无容量设施选址 (UFL) 博弈, 参与者的最大分摊费用等于大联盟优化问题的经典 LP 松弛成本。后来, Goemans and Skutella (2000) 在此基础上扩展了这一研究, 并证明了在一些特殊的设施选址博弈中的核心非空, 如地址在一条线上、一个循环和一个树形网络上。其他人也研究了这个问题的变形。如 Puerto et al. (2011, 2012) 分别介绍了最小半径选址博弈和最小直径选址博弈。Xu and Yang (2009), Mallozzi (2011), Li et al. (2013) 等人研究了考虑各种成本要素的设施定位博弈, 如服务安装成本, 区域固定成本, 以及凹形设施选址成本。如前文所述, 最小核是一种可用于处理空核博弈的松弛方法。Faigle et al. (2000) 指出, 计算最小生成树博弈的最小核分配是 NP-hard 问题。Kern and Paulusma (2003) 基于基数匹配博弈中最小核的多项式描述研究了核仁。Schulz and Uhan (2010) 的研究表明求一个具有超模成本单机调度博弈的最小核值是弱 NP-hard 问题, 并为计算最小核值的 3-approximate 算法 (界为 3) 提供了方法 Schulz and Uhan (2013)。关于 γ -core 和 ϵ -approximate core, 使用后者进行研究的更多。例如, Faigle et al. (1998) 提出了一种基于 LP 的算法, 为 Euclidean TSP 博弈得出了一个 $\frac{1}{3}$ -近似核。Bläser and Ram (2008) 提出了一种多项式时间算法, 该算法可以在一个 $(\log_2(n-1)-1)$ -近似核中得到非对称 TSP 博弈的费用分摊方法。我们建议读者参考 Jain and Mahdian (2007) 的研究以对先前的概念进行更全面的回顾。现在仅有几篇论文直接研究 OCAP。Bachrach et al. (2009) 提出了在稳定合作博弈中大联盟的前提下, 如何确定补贴的最小值并且得到最小值合适的上、下界的问题之后, Meir et al. (2011) 进行了类似的关于联合博弈中限制合作的研究。目前唯一解决 OCAP 的算法是由 Caprara and Letchford (2010) 提出的。他们提出了一种基于 LP 松弛和对偶理论的算法。在他们的方法中, 通常需要通过引入具有特殊结构的约束来重新表达优化问题。详细内容将在下一节中介绍。

3 正文前述

可转移效用的合作博弈用二元向量 (V, c) 表示, 其中 $V = \{1, 2, \dots, v\}$ 表示参与者集合, $c: S \rightarrow \mathbb{R}$ 表示特征函数, $S = 2^V \setminus \emptyset$ 表示参与者非空的联盟集合。特征函数为每个联盟 $s \in S$ 分配了一个值 $c(s)$, 代表了 s 中的成员在合作时需要支付的最小总成本。本文研究费用分摊问题是在 V 中的参与者之间分摊大联盟的成本 $c(V)$ 使得任何一个小联盟的参与者都没有动机脱离大联盟。一个博弈 (V, c) 的稳定费用分摊是一个向量 $\alpha \in \mathbb{R}^v$, 其满足联盟稳定性: $\sum_{k \in s} \alpha(k) \leq c(s), \forall s \in S$ 。理想中的费用分摊另外满足预算平衡约束: $\sum_{k \in V} \alpha(k) = c(V)$ 。博弈 (V, c) 的核心定义为:

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

众所周知，并不是每个博弈 (V, c) 都有非空的核心。要求解核心为空的博弈，我们的目标是找到一个稳定的费用分摊方案，尽可能多地接近大联盟成本 $c(V)$ 。这就是最优费用分摊问题，定义如下：

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

通常很难直接求解 (1)，因为它包含一个指数级的约束条件，而且计算特征函数 $c(s)$ 可能是 NP-难问题。本文的重点是为 OR 博弈求得稳定的最优费用分摊方案。该类博弈的核心可能为空，特征函数由整数规划定义，是 Caprara and Letchford (2010) 研究的整数最小化 (IM) 博弈的推广。与只能使用 ILP 来定义特征函数的 IM 博弈不同，OR 博弈允许使用非线性整数规划来定义特征函数。

如果一个博弈 (V, c) 满足以下条件，则该博弈被称为 OR 博弈。

- 正整数 r, r' 和 t ,
 - 左边矩阵 $A \in \mathbb{R}^{r \times t}$ 和 $A' \in \mathbb{R}^{r' \times t}$,
 - 右边矩阵 $B \in \mathbb{R}^{r \times v}$ 和 $B' \in \mathbb{R}^{r' \times v}$,
 - 右边非负列向量 $D \in \mathbb{R}^r$ and $D' \in \mathbb{R}^{r'}$,
 - 线性或非线性目标函数 $f(x)$,
 - 对于所有 $s \in S$, 示性列向量 $\gamma^s \in \{0, 1\}^v$ 满足：若 $k \in s$ 且 $\gamma_k^s = 0$, 则 $\gamma_k^s = 1$ 否则, $\forall k \in V$,
- 则特征函数 $c(s)$ 用整数规划表示为：

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

注意，在式 (2) 中，为方便使用拉格朗日松弛，约束被分为两部分。根据 Caprara and Letchford (2010) 的研究，很容易证明每个 OR 博弈的非负列向量 D 和 D' 都具有次可加性，即对于任意 $s_1, s_2 \in S$, $c(s_1 \cup s_2) \leq c(s_1) + c(s_2)$, 且 $s_1 \cap s_2 = \emptyset$ 。IM 博弈是 OR 博弈的一种特殊类型，它的特征函数 $c(s)$ 是由整数线性规划表示为：

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

其中 C 是 t 维行向量。我们用 $c_{LP}(V)$ 表示 (3) 中 $c(V)$ 的 LP 下界，其中 $x \in \{0, 1\}^{t \times 1}$ 松弛为 $0 \leq x \leq 1$ 。Caprara and Letchford (2010) 给出了通过使用列生成、行生成或二者结合解决 IM 博弈的 OCAP 方法。我们在 online supplement 中总结了这些方法的要点。列生成方法有一个简单的公式，但由于优化过程中需要极大的解空间，与其相关的定价问题通常很难处理。行生成方法需要通过识别被称为可分配约束的集合 $\{Ex \geq F\gamma\}$ 来重新确定 ILP(3)。然后，通过求解仅具有可分配约束的 LP 松弛 $c_{LP}^{EF}(V) = \min\{Cx : Ex \geq F\gamma\}$ 得到费用分摊方案，其中总费用等于 $c_{LP}^{EF}(V)$ —为 $c(V)$ 的下界之一。注意， $c_{LP}^{EF}(V)$ 可能与 $c_{LP}(V)$ 不同。对于 IM 博弈，LPB 费用分摊方案的好坏很大程度上依赖于已确定的可分配的约束。从理

论上讲，如果所有可分配的约束都能被识别和添加，LPB 算法就能找到一个最优的稳定成本。然而，对于不同 IM 博弈，识别可分配约束的方法通常不同。对于一些可分配的约束，没有已知的多项式时间间隔算法又增加了计算的难度。尽管存在这些问题，LPB 算法依然可以作为一种有效的启发式算法，在只添加可分配约束的子集的情况下，找到良好的稳定的费用分摊方案。

4 基于拉格朗日松弛理论的费用分摊算法

在本节给出了基于拉格朗日松弛理论的费用分摊算法。对于 OR 博弈 (V, c) ，当 (2) 中的特征函数 $f(x)$ 为线性时，LPB 算法和 LRB 算法都适用；当特征函数 $f(x)$ 为非线性时，则只有 LRB 算法可以求解该问题。本节首先给出了 LRB 算法并证明了其有效性，最后给出了该算法详细的展开过程。

4.1 拉格朗日松弛理论与博弈分解

在拉格朗日松弛过程中，通过松弛式 (2) 中的约束 $\{A'x \geq B'\gamma^s + D'\}$ ，将该约束乘以非负的拉格朗日乘子 λ 后代入目标函数，可以推导出 OR 博弈的拉格朗日特征函数 $c_{LR}(\cdot; \lambda)$ ：

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

其中， λ 是一个非负的 r' 维行向量，即 $\lambda \in \mathbb{R}_+^{1 \times r'}$ 。特别地，大联盟 V 的拉格朗日特征函数为：

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

与经典的拉格朗日松弛相同，需要仔细挑选约束 $\{A'x \geq B'\gamma^s + D'\}$ 使得 $c_{LR}(s; \lambda)$ 相对容易求解，即对于任意 $s \in S$ 在多项式或者拟多项式时间可以求解。显而易见，对于任意的 λ ， $c_{LR}(V; \lambda)$ 是 $c(V)$ 的下界。为了找到最大下界，拉格朗日对偶问题 $d_{LR}(V)$ 可以找到使 $c_{LR}(V; \lambda)$ 最大化的最优拉格朗日乘子 λ ，即，

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

通过次梯度方法 (e.g., see Ahuja et al. 1993)，可以计算得到 $d_{LR}(V)$ 的最优的拉格朗日乘子 λ^* 。借助任一非负拉格朗日乘子 λ^* ，可以将拉格朗日特征函数 $c_{LR}(\cdot; \lambda)$ 分解为两个子特征函数 $c_{LR1}(\cdot; \lambda)$ 和 $c_{LR2}(\cdot; \lambda)$ 。对于任意 $s \in S$ ，两个子特征函数满足 $c_{LR}(s; \lambda) = c_{LR1}(s; \lambda) + c_{LR2}(s; \lambda)$ 其中，

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

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

将 $(V, c_{LR1}(\cdot; \lambda))$ 定义为子博弈 1，其特征函数为 $c_{LR1}(s; \lambda)$ ；类似地，将 $c_{LR2}(\cdot; \lambda)$ 定义为子博

弈 2。在某些具体的应用中，可以继续将子博弈 2 进行分解，使得 LRB 算法更加高效。在 5.2.1 节中给出了一个例子。

定理 1: 对于任意非负的拉格朗日乘子 λ ，若 α_{LR1}^λ 和 α_{LR2}^λ 是子博弈 $(V, c_{LR1}(\cdot; \lambda))$ 和 $(V, c_{LR2}(\cdot; \lambda))$ 严格意义上的稳定费用分摊解，则 $\alpha_{LR}^\lambda = \alpha_{LR1}^\lambda + \alpha_{LR2}^\lambda$ 是 OR 博弈 (V, c) 的一个稳定费用分摊解。证明：对于任意 $s \in S$ ， α_{LR1}^λ 和 α_{LR2}^λ 的稳定性满足：

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

因此，推导出 $\sum_{k \in s} \alpha_{LR}^\lambda(k) = \sum_{k \in s} [\alpha_{LR1}^\lambda(k) + \alpha_{LR2}^\lambda(k)] \leq c(s)$ 。□ 根据定理 1，可以设计算法通过求解每个子联盟的稳定费用分摊解得到 OCAP 的稳定解。实际上，根据具体的 λ ，可以得到每个子博弈的最优稳定解。在介绍求解两个子博弈的具体方法之前，首先对比了 LPB 算法和 LRB 算法的有效性。对于同一 IM 博弈，具有不同整数线性规划公式的特征函数可以得到不同的解。定理 2 给出了 LRB 解优于或等同于 LPB 解的充分条件。

定理 2: 对于特征函数具有相同 ILP 公式的 IM 博弈，当满足以下两个条件时，LRB 解优于或等同于 LPB 解。(1) $d_{LR}(V)$ 中的拉格朗日乘子是最优的，即 $\lambda = \lambda^*$ ，(2) $\alpha_{LR1}^{\lambda^*}$ 和 $\alpha_{LR2}^{\lambda^*}$ 存在于子博弈 $(V, c_{LR1}(\cdot; \lambda^*))$ 和 $(V, c_{LR2}(\cdot; \lambda^*))$ 的核中。证明：对于相同的式子，拉格朗日松弛的下界不会比线性松弛的下界差，即 $c_{LR}(V; \lambda^*) \geq c_{LP}(V)$ 。根据条件 2 可得：

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

另外，LPB 费用分摊解 $\sum_{k \in V} \alpha_{LP}(k)$ 明显不超过线性松弛下界 $c_{LP}(V)$ 。因此 $\sum_{k \in V} \alpha_{LR}^{\lambda^*}(k) = c_{LR}(V; \lambda^*) \geq c_{LP}(V) \geq \sum_{k \in V} \alpha_{LP}(k)$ 。□ 关于定理 2 有以下注释：注释 1：定理 2 表明了 LRB 算法相较于 LPB 算法的竞争优势。通过增加合适的 ILP 公式到特征函数 $c(s)$ 可以使 LRB 算法和 LPB 算法的解更优。但是根据定理 2，对具有相同 ILP 公式的特征函数 $c(s)$ ，如果同时满足条件 (1) 和 (2)，那么 LRB 解不小于 LPB 解。即使在某些例子中 LPB 解是最优解，LRB 算法可以提供具有不同特征的最优解。见 5.1.3 节。注释 2：定理 2 中的两个条件是充分条件，不是必要条件。实际上，即使不满足条件 (1)，如果对于一个非最优的拉格朗日乘子 $\bar{\lambda}$ ，满足 $c_{LR}(V; \bar{\lambda}) \geq c_{LP}(V)$ ，定理 2 的结论依然成立。注释 3：条件 (2) 要求两个子博弈的核心非空。在接下来的引理 1 中，子博弈 1 $(V, c_{LR1}(\cdot; \lambda))$ 的核心通常非空，但子博弈 2 $(V, c_{LR2}(\cdot; \lambda))$ 的核心可能为空集。因此 LRB 算法的有效性取决于子博弈 2，尤其是可以分摊给博弈 $(V, c_{LR2}(\cdot; \lambda))$ 的费用 $\sum_{k \in V} \alpha_{LR2}^\lambda(k)$ 。计算结果表明，解释子博弈 2 的核心为空，LRB 算法依然具有竞争性。实例见 5.2.2 节。注释 4：当松弛 $c(V)$ ， $d_{LR}(V)$ 具有整数松弛性时（即松弛掉 x 的整数约束后 $d_{LR}(V)$ 不增加），拉格朗日下界 $c_{LR}(V; \lambda^*)$ 等于线性松弛下界 $c_{LP}(V)$ 。另外，如果 $c_{LP}(V)$ 的所有约束可分配，由于 $\sum_{k \in V} \alpha_{LP}(k) = c_{LP}(V) = c_{LR}(V; \lambda^*) \geq \sum_{k \in V} \alpha_{LR}^{\lambda^*}(k)$ ，LRB 解不会优于 LPB 解。当子博弈 2 的核可能为空集时，注释还与 LRB 算法的收敛性有关。在这种情况下，首先需要有一个在求解其拉格朗日对偶问题过程中可以进行大量迭代的拉格朗日下界 $c_{LR2}(V; \lambda)$ ；然而一个更紧的边界 $c_{LR2}(V; \lambda)$ 不能确保得到一个值更大的解 $\sum_{k \in V} \alpha_{LR2}^\lambda(k)$ 。换言之，最优的拉格朗日乘子 λ^* 不一定能得

到对应的最优解。计算实例见 5.2.2 节。基于这些情况，提出以下算法：算法 1：求解 OR 博弈 (V, c) 的 LRB 费用分摊算法。第一步：为问题 (2)，构造一个如式 (4) 的拉格朗日松弛。利用次梯度法求解拉格朗日对偶问题 $d_{LR}(V)$ (5) 的最优解 λ^* 。在迭代过程中将拉格朗日乘子存入集合 $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_p\}$ 。第二步：根据 (6) 和 (7)，利用每个 $\lambda \in \Lambda$ ，将拉格朗日特征函数 $c_{LR}(\cdot; \lambda)$ 分解为两个子特征函数 $c_{LR1}(\cdot; \lambda)$ 和 $c_{LR2}(\cdot; \lambda)$ 。第三步：求解两个子博弈 $(V, c_{LR1}(\cdot; \lambda))$ 和 $(V, c_{LR2}(\cdot; \lambda))$ 的最优稳定费用分摊解 α_{LR1}^λ 和 α_{LR2}^λ 。详细内容在后两小节的引理 1 和引理 2。第四步：计算博弈 (V, c) 的稳定费用分摊解 $\alpha_{LR}^\lambda = \alpha_{LR1}^\lambda + \alpha_{LR2}^\lambda$ 。在 $\lambda \in \Lambda$ 对应的 α_{LR}^λ 中找到分摊成本最多的稳定分摊解。注意，如果具有最优拉格朗日乘子 λ^* 的子博弈 2 的核心非空，只需利用 Λ 中的最后一个拉格朗日乘子 λ^* 计算即可。对于其他情况，根据算法 1 之前的讨论，可以考虑使用更多的拉格朗日乘子。在选择拉格朗日乘子方面并没有理论研究，根据经验从不同的迭代中选择 5 或者 6 个值比较有效。

4.2 求解子博弈 1

本节介绍如何求解子博弈 1 $(V, c_{LR1}(\cdot; \lambda))$ 在核中的费用分摊解。在该博弈中，每个参与者 $k \in V$ 会减少一个费用 $(\lambda B')_k$ 。 $(\lambda B')_k$ 表示向量 $\lambda B'$ 中第 k 个元素的值。据此得到如下引理：

引理 1: 若向量 α_{LR1}^λ 满足 $\{\alpha_{LR1}^\lambda(k) = (\lambda B')_k : \forall k \in V\}$ ，则 α_{LR1}^λ 在博弈 $1(V, c_{LR1}(\cdot; \lambda))$ 的核中。证明：根据博弈 $(V, c_{LR1}(\cdot; \lambda))$ 的定义可知对于任意联盟 $s \in S$ ，减少的总费用满足 $c_{LR1}(s; \lambda) = \sum_{k \in s} (\lambda B')_k$ ，并且 $\sum_{k \in s} \alpha_{LR1}^\lambda(k) = \sum_{k \in s} (\lambda B')_k$ 。□ 引理 1 说明可以直接求解的博弈 $(V, c_{LR1}(\cdot; \lambda))$ 核中的费用分摊解。

4.3 求解子博弈 2

子博弈 2 $(V, c_{LR2}(\cdot; \lambda))$ 与子博弈 1 不同，核心可能为空。我们的目标是在核心非空的条件下求解核中的最优稳定费用分摊解。为求解一般博弈 $(V, c_{LR2}(\cdot; \lambda))$ 问题的最优稳定费用分摊解，本节首先给出了一个通用的基于列生成 (CGB) 的算法。本文的 CGB 算法与 Caprara and Letchford (2010) 提出的算法相比计算更加方便。与 Caprara 和 Letchford(2010) 提出的将优化问题的每个可行解作为一系列来处理，重新构造主规划问题 (详见附录 LP (A.2)) 方法不同，本文将每个联盟作为如 LP(9) 所定义的主规划问题的一系列，。求解每个联盟 s 的 $c(s)$ 时，为了避免解强 NP 难问题，对其进行重新构造是有必要的。但这使得在生成列时搜索空间大大增大。考虑到与 $c(s)$ 相比，在拉格朗日松弛后的 $c_{LR2}(s; \lambda)$ 变得更容易求解，本文直接选择将每个联盟作为一系列。子博弈 2 的最优稳定费用分摊是对应 OCAP 问题的最优解。

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

在该假设下，拉格朗日松弛会更易求解。对于任意 $s \in S$ ，其 $c_{LR2}(s; \lambda)$ 在计算上更易处理。为求解式 (8) 的 OCAP，考虑求解其对偶问题：

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

其中， $\{\beta_s : \forall s \in S\} \in \mathbb{R}^{(2^v-1) \times 1}$ 是决策变量。根据强对偶理论，式 (8) 中的 OCAP 等价于式 (9) 中的 LP 问题。根据标准的列生成算法，本文提出了可以求解式 (9)、得到子博弈 2 最优费用分摊解的算法 2。算法 2：为求解博弈 $(V, c_{LR2}(\cdot; \lambda))$ 最优稳定费用分摊解 α_{LR2}^λ 的 CGB 算法包含以下四步：第一步：求解主规划问题 (9) 的最优对偶解 π^* 。设集合 $S' \subset S$ 含有多项式数目的元素。第二步：找到 the pricing problem 的最优联盟 s^*

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

第三步：若存在代入式 (10) 的值为负数的联盟 s^* ，将其添加到集合 S' ，然后返回第一步；若不存在，则对偶问题 (9) 已求得最优解，继续第四步。第四步：根据更新的联盟集合 S' 核其对应的特征函数值 $\{c_{LR2}(s; \lambda) : \forall s \in S'\}$ ，下面的 LP 可以求得博弈 $(V, c_{LR2}(\cdot; \lambda))$ 的最优稳定费用分摊解 α_{LR2}^λ ：

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

在 CGB 算法中，重难点是求解 the pricing problem(10)，对具体的博弈需要具体分析。本文在 5.2 节给出了实例。最后，本节给出了 CGB 算法的引理。

引理 2：通过 CGB 算法求得的向量 α_{LR2}^λ 是子博弈 2 $(V, c_{LR2}(\cdot; \lambda))$ 的最稳定费用分摊解。上面的 CGB 算法给出了求解子博弈 2 的通用方法。但是，对于特殊的问题，可以用更简便、更快的方法求解。比如以下两种情况：第一种，如果 $c_{LR2}(V; \lambda)$ 的线性松弛包含了完整的可分配约束集，则可以用 Caprara 和 Letchford(2010) 提出的 LPB 算法求解。第二种，如果子博弈 2 的特征函数为次模函数，则可以通过贪心算法求解，即对参与者进行排序并计算边际成本向量。定义 2：设 a 和 b 为大联盟 V 中的两个参与者。如果对于任意 $s \in V \setminus \{a, b\}$ 满足 $c_{LR2}(s \cup \{a\}; \lambda) + c_{LR2}(s \cup \{b\}; \lambda) \geq c_{LR2}(s; \lambda) + c_{LR2}(s \cup \{a, b\}; \lambda)$ ，则特征函数 $c_{LR2}(\cdot; \lambda)$ 为次模函数。本节特别强调这两种情况的原因是当原始博弈不存在这两种情况时，子博弈 2 仍可能含有完整可分配约束或具有次模性。这正是 LRB 算法的潜在优势，在 5.1 节本文给出了一个例子。但是，探究子博弈 2 是否含有完整可分配约束或是否具有次模性依然需要复杂的分析。

5 设施选址博弈的实现

我们将在两种不同的设施选址博弈中说明 LRB 成本分配算法，即无容量限制的设施选址博弈（下称 UFL 博弈）和非线性有容量限制设施选址合作博弈（下称 NLCFL 博弈）。UFL 博弈是利用 LPB 和 LRB 算法来计算最优成本分配的一种情况；此外，得到的 UFL 子博弈 2 是次模的，其核心成本分配可以在多项式时间内得到。NLCFL 博弈具有不同的联盟定义和非线性特征函数，显示了我们的 LRB 算法的全部能力。

5.1 UFL 博弈

在一个 UFL 博弈中，有一个双向网络图定义为 $G = (M, N, E)$ ，其中 M 为潜在工厂开设的集合， N 为必须被服务顾客的集合， E 为连接工厂和顾客所形成边的集合。每一个工厂潜在开设点 $i \in M$ 都有一个固定开启成本 f_i ，并且每一条边 $(i, j) \in E$ 都有一个运输成本 c_{ij} 。在 UFL 博弈中，顾客分享工厂开启和运输的成本，即在博弈中的参与者是顾客。我们在表中列出了在 UFL 中需要用到的记号。表 1。

表 1: UFL 博弈中使用的记号

M	潜在设施位置集合, $M = \{1, 2, \dots, m\}$.
N	顾客即博弈参与者集合, $N = \{1, 2, \dots, n\}$.
c_{ij}	从设施 i 到顾客 j 的转移成本, $\forall i \in M, j \in N$.
f_i	设施 i 开启的固定成本, $\forall i \in M$.
s	参与者联盟, $s \subseteq N$.
γ^s	示性向量 $[\gamma_1^s, \gamma_2^s, \dots, \gamma_n^s]^T$, 如果参与者 j 在联盟 s 中则有 $\gamma_j^s = 1$ 否则 $\gamma_j^s = 0$
v_i	决策变量, 如果设施 i 开启则有 $v_i = 1$ 否则 $v_i = 0$, $\forall i \in M$.
u_{ij}	决策变量, 如果参与者 j 被设施 i 服务则有 $u_{ij} = 1$ 否则 $u_{ij} = 0$
$\forall i \in M \text{ and } j \in N$.	

UFL 博弈 (N, c_{UFL}) 的定义为，在集合 N 中的顾客为参与者并且特征函数为 $c_{UFL}(s)$ 由如下的整数规划决定，

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

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

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

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

在上述的整数规划中，目标函数 (11) 是为了最小化对于一个联盟 s 中工厂开启和运输的总成本，限制条件 (12) 需要在联盟 s 中的每一位顾客被服务，而限制条件 (13) 确保了只有一个开启的工厂可以服务顾客。

整数规划 (11)-(14) 是对于一个无容量限制的设施选址问题的常规表达式。基于定义，我们可以看到 UFL 博弈 (N, c_{UFL}) 是一个 OR 博弈 (V, c) 其中 $V = N$, $c = c_{UFL}$ 。在 c 中的决策变量 x 现在变为在 c_{UFL} 中的 $[v; u]$ ，并且矩阵 C, A, A', B, B', D, D' 的具体表达式也可以通过用矩阵写出 c_{UFL} 而得到。特别地， D 和 D' 现在都是 $\mathbf{0}$ ，所以博弈 (N, c_{UFL}) 是次可加的。这一点对于我们接下来要研究的 ULCFL 博弈来说也是正确的。

5.1.1 对于 UFL 博弈线性规划解的成本分配

Kolen (1983) 和 Goemans and Skutella (2000) 证明了，对于一个 UFL 博弈，最大稳定成本分配值与 $c_{UFL}(N)$ 的 LP 下界重合。为了做进一步的分析，我们给出了使用 LPB 算法来计算最优稳定成本分配的更多细节。

在 $c_{UFL}(s)$ 中，限制条件 (12) 和 (13) 已经是可分配的。通过加入可分配限制条件 $\{u_{ij} \geq 0 : i \in M, j \in N\}$ 来松弛二元限制条件 (14)，我们可以得到一个对于大联盟最优解问题 $c_{UFL}(N)$ 的 LP 松弛如下：

$$c_{LP_UFL}(N) = \min_{v, u} \sum_{i \in M} f_i v_i + \sum_{i \in M} \sum_{j \in N} c_{ij} u_{ij} \quad (15)$$

$$s.t. \quad \sum_{i \in M} u_{ij} \geq \gamma_j^N, \quad \forall j \in N, \quad (16)$$

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

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

我们有顺序地对限制条件 (15), (16) 和 (17) 分别从 1 到 n , $n+1$ 到 $n+mn$ 和 $n+mn+1$ 到 $n+2mn$ 进行标号。对于 $c_{LP_UFL}(N)$ ，我们考虑它的对偶线性规划。让 μ_k 为 $c_{LP_UFL}(N)$ 的第 k 个限制条件所对应的对偶变量，并且 μ^* 为对偶式的最优解。根据在附录中的行生成方法，我们有如下的引理：

引理 3 对于一个 UFL 博弈，LPB 成本分配 α_{LP_UFL} 由下式给出

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

是最优的，同时总的共享成本为 $c_{LP_UFL}(N)$ 。

我们记一种对于 UFL 博弈得到最优稳定成本分配的简单方式是直接解 $c_{LP_UFL}(N)$ 并且通过计算限制条件的影子价格得到最优对偶变量。然而，解对偶线性规划有助于找到替代的最优解，因为在对偶

LP 有多个最优解的情况下, 并不是所有的都对应于原始的影子价格。

5.1.2 对于 UFL 博弈拉格朗日松弛解的成本分配

接下来我们将演示如何应用 LRB 算法来获得 UFL 博弈的最优成本分配。我们将证明 UFL 博弈的子博弈 2 是次模的。我们还将通过计算实验表明, LRB 算法所获得的该博弈的最优成本分配与 LPB 算法所获得的成本分配不同, 从而为评价和比较提供更多的选择。

在 $c_{UFL}(s)$ 中, 我们加入了一系列新的限制条件

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

然后引入限制条件 $\{\sum_{i \in M} u_{ij} \geq \gamma_j^s : j \in N\}$ 到加入了非负拉格朗日乘子 σ 的目标函数中, 从而得到 UFL 的拉格朗日特征函数。

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

约束条件 (18) 的增强是为了加强 $c_{UFL}(s)$ 的拉格朗日下界, 这有可能因此得到更好的 LRB 成本分配。它禁止设置 $u_{ij} = 1$ 对于任何不在联盟 s 中的参与者, 即使计算 $c_{LR_UFL}(s; \sigma)$ 时, 系数 $c_{ij} - \sigma_j < 0$ 。很容易看出 (18) 的增强只相当于在目标函数 $c_{LR_UFL}(s; \sigma)$ 中替换术语 $\sum_{i \in M} \sum_{j \in N} (c_{ij} - \sigma_j) u_{ij}$ by $\sum_{i \in M} \sum_{j \in s} (c_{ij} - \sigma_j) u_{ij}$ 。

在算法 1 下, 一般的 LRB 成本分配算法对于任何 $s \in S$ 和非负拉格朗日乘子 σ , 我们可以分解 $c_{LR_UFL}(s; \sigma)$ 为 $c_{LR1_UFL}(\cdot; \sigma)$ 和 $c_{LR2_UFL}(\cdot; \sigma)$ 使得 $c_{LR_UFL}(s; \sigma) = c_{LR1_UFL}(s; \sigma) + c_{LR2_UFL}(s; \sigma)$, 并且定义 UFL 子博弈 1 $(N, c_{LR1_UFL}(\cdot; \sigma))$ 和 UFL 子博弈 2 $(N, c_{LR2_UFL}(\cdot; \sigma))$ 。

对于 UFL 的子博弈 1, 它的特征函数是

$$c_{LR1_UFL}(s; \sigma) = \sum_{j \in N} \sigma_j \gamma_j^s. \quad (19)$$

根据引理 1, 最优稳定成本分配 $\alpha_{LR1_UFL}^\sigma$ 位于由 $\alpha_{LR1_UFL}^\sigma(j) = \sigma_j, j \in N$ 给出的博弈 $(N, c_{LR1_UFL}(\cdot; \sigma))$ 的核中。

对于 UFL 的子博弈 2，它的特征函数为

$$\begin{aligned}
c_{LR2_UFL}(s; \sigma) &= \min_{v, u} \sum_{i \in M} f_i v_i + \sum_{i \in M} \sum_{j \in N} (c_{ij} - \sigma_j) u_{ij} \\
s.t. \quad &u_{ij} - v_i \leq 0, \forall i \in M, j \in N, \\
&u_{ij} \leq \gamma_j^s, \forall i \in M, j \in N, \\
&v_i, u_{ij} \in \{0, 1\}, \forall i \in M, j \in N.
\end{aligned} \tag{20}$$

为了解 $c_{LR2_UFL}(s; \sigma)$ ，我们可以将其分解为各种条件，并且得到一个由下式给出的最优目标函数值 $c_{LR2_UFL}(s; \sigma) = \sum_{i=1}^m \min \{0, f_i + \sum_{j \in s} \min \{0, c_{ij} - \sigma_j\}\}$ 。

引理 4 UFL 子博弈 2 $(N, c_{LR2_UFL}(\cdot; \sigma))$ 是次模的。

证明：记 a 和 b 为 N 中的两个参与者。为了说明次模性，我们需要证明，对于任何联盟 $s \in N \setminus \{a, b\}$,

$$c_{LR2_UFL}(s \cup \{a\}; \sigma) - c_{LR_UFL2}(s; \sigma) \geq c_{LR2_UFL}(s \cup \{a, b\}; \sigma) - c_{LR2_UFL}(s \cup \{b\}; \sigma). \tag{21}$$

即对于每一个 $i \in M$ ，令 $\Delta_i(s; \sigma) = \min \{0, f_i + \sum_{j \in s} \min \{0, c_{ij} - \sigma_j\}\}$ 。为了证明 (21)，由式

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

令 $\rho(x) = \min \{0, x\}$ ，并且定义 $x_{\hat{s}} = f_i + \sum_{j \in \hat{s}} \min \{0, c_{ij} - \sigma_j\}$ for each $\hat{s} \in \{s, s \cup \{a\}, s \cup \{b\}, s \cup \{a, b\}\}$ 。可以看出 $x_s + x_{s \cup \{a, b\}} = x_{s \cup \{a\}} + x_{s \cup \{b\}}$ ，和 $x_{s \cup \{a, b\}} \leq \min \{x_{s \cup \{a\}}, x_{s \cup \{b\}}\} \leq \max \{x_{s \cup \{a\}}, x_{s \cup \{b\}}\} \leq x_s$ 。因此，既然 $\rho(x)$ 是一个关于 x 的凹函数，我们有

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

从中我们可以直接得到 (22)，并且完成了引理的证明。 \square

由于 UFL 子博弈 2 的次模性，我们能够很容易地计算它的核成本分配，记为 $\alpha_{LR2_UFL}^\sigma$ ，通过子博弈章节中的贪心算法。在最优拉格朗日乘子 σ^* 下，我们可以得到由 $\alpha_{LR_UFL}^{\sigma^*} = \alpha_{LR1_UFL}^{\sigma^*} + \alpha_{LR2_UFL}^{\sigma^*}$ 给出的最优 UFL 的 LRB 成本分配。既然 UFL 的子博弈 1 和子博弈 2 都有非空核，通过定理 2 最优 LRB 成本分配值达到了拉格朗日下界 $c_{LR_UFL}(N; \sigma^*)$ ，即不小于 LP 的下界 $c_{LP_UFL}(N)$ 。

下面的定理说明了 UFL LRB 成本分配的最优性，并揭示了 LRB 和 LPB 成本分配的等价性。

定理 3 对于一个 UFL 博弈，LRB 的成本分配 $\alpha_{LR_UFL}^{\sigma^*} = \sigma^* + \alpha_{LR2_UFL}^{\sigma^*}$ 是最优的。并且，LRB 成本分配集和 LPB 成本分配集都包含所有最优 UFL 成本分配。

对于 UFL 博弈，我们首先证明了 LRB 的成本分配是最优的。如前所述，最优 LRB 成本分配值达到拉格朗日下界 $c_{LR_UFL}(N; \sigma^*)$ ，其不小于 LP 下界 $c_{LP_UFL}(N)$ 。我们知道对于 UFL 博弈的 LP 下界

等于最大总共享成本 (Kolen 1983, Goemans and Skutella 2000)。因此, LRB 成本分配一定是一个最优 UFL 成本分配, 并且 $c_{LR_UFL}(N; \sigma^*) = c_{LP_UFL}(N)$ 。

然后, 我们证明了 LRB 成本分配集和 LPB 成本分配集都包含所有的最优 UFL 成本分配。众所周知, LPB 成本分配集由所有最佳 UFL 成本分配 (Goemans and Skutella 2000) 组成。这意味着每个 LRB 成本分配必须属于 LPB 成本分配集。因此, 仍然需要证明, 每个 LPB 成本分配都属于 LRB 成本分配集。

考虑到每一个 LPB 成本分配 $\alpha_{LP_UFL}(j) = \mu_j^*$ for $j \in N$, 这里 μ^* 和一些 δ^* 构成一个最优解对于下面的关于 $c_{LP_UFL}(N)$ 的对偶问题:

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

对每一个 $i \in M$, 可以看出 $f_i = \sum_{j \in N} \delta_{ij}^*$, 并且 $\delta_{ij}^* \geq \max\{0, \mu_j^* - c_{ij}\}$ 对于 $j \in N$, 这表明 $f_i \geq \sum_{j \in N} \max\{0, \mu_j^* - c_{ij}\} = -\sum_{j \in N} \min\{0, c_{ij} - \mu_j^*\}$ 。因此, 我们有

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

既然 $c_{LR2_UFL}(N; \sigma) = \sum_{i=1}^m \min\{0, f_i + \sum_{j \in N} \min\{0, c_{ij} - \sigma_j\}\}$ 对于任何非负 σ 成立, 由 (23) 我们有 $c_{LR2_UFL}(N; \mu^*) = 0$ 。同时加上 $c_{LR1_UFL}(N; \mu^*) = \sum_{j \in N} \mu_j^*$ 该式, 表明 $c_{LR_UFL}(N; \mu^*) = \sum_{j \in N} \mu_j^* = c_{LP_UFL}(N) = c_{LR_UFL}(N; \sigma^*)$ 。因此 μ^* 是一个最优拉格朗日乘子。由此产生的 LRB 成本分配如下: $\alpha_{LR_UFL}^\mu(j) = \mu_j^* + \alpha_{LR2_UFL}^\mu(j)$ for $j \in N$ 。注意到对于每一个 $s \in S$, 因为 $c_{LR2_UFL}(s; \mu^*) \leq 0$ 和 $c_{LR2_UFL}(s; \mu^*) \geq c_{LR2_UFL}(N; \mu^*) = 0$, 我们有 $c_{LR2_UFL}(s; \mu^*) = 0$, 这将得到 $\alpha_{LR2_UFL}^\mu(j) = 0$ 对于所有 $j \in N$ 。因此我们得到 $\alpha_{LR_UFL}^\mu = \mu^*$, 表明每一个 LPB 成本分配 μ^* 属于 LRB 成本分配集。这就完成了定理 3 的证明。 \square

5.1.3 替代最优稳定成本分配

对于一个 UFL 博弈, 我们知道每一个最优成本分配对应于一个 LPB 解, 它必须是对偶式 $c_{LP_UFL}(N)$ 的所有基本最优解的凸组合。然而, 只应用一般的 LP 求解方法于对偶式 $c_{LP_UFL}(N)$ 是很难得到所有基本最优解的。

通过定理 3, 我们知道 LRB 算法提供了对于 UFL 博弈的一种替代最优成本分配。所获得的 LRB 解有可能被排除在一般 LP 求解产生的 LPB 解之外。我们通过下面的例子来说明这一点。

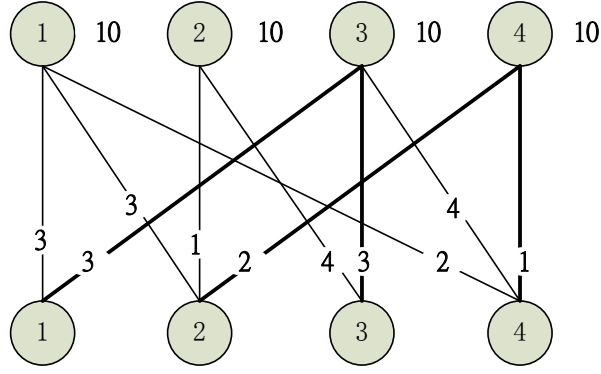


图 1: UFL 博弈的一个例子

在图 1 中所示的 UFL 博弈中，这里有四个设施和四个参与者。每一个设施有一个固定的启用成本 10。在连接线上的数字代表从设施到参与者的运输成本。对于大联盟的一个最优解是开启设施 3 和 4，并且画粗的连接线是最优路径。因此，大联盟所需要的成本是 $10+10+3+3+2+1=29$ 。

在这个例子中，我们使用了两种 LP 求解方法，单纯形法和内点法，分别在 MATLAB 发行版本 2011a 上计算 LPB 分配。表 2 显示了在不同方法下的每一个参与者需要承担的成本。

表 2: 不同方法下的最优稳定 UFL 成本分配

方法	参与者 1	参与者 2	参与者 3	参与者 4	总共享成本
单纯形法	5.00	6.50	8.50	6.50	26.5
内点法	6.58	6.50	8.50	4.92	26.5
LRB	6.87	6.50	8.50	4.63	26.5

实例表明，LRB 算法可以产生不同于一般 LP 方法的最优稳定成本分配。LRB 解超出了两个 LPB 解的凸组合范围。这说明了 LRB 算法在提供替代成本分配方面的价值。

为了研究 LRB 算法在一般设置下的能力，我们测试了由 Beresnev et al. (2006) 开发的 30 个无容量限制的设施位置基准实例，所有实例均有 $M = N = 100$ 。我们在一台 Windows7 个人电脑上进行了所有的计算实验，该电脑的 CPU 型号为英特尔酷睿 i7-2600，主频 3.4GHz 和内存为 16G。所有算法均在 Matlab 2011a 版本中实现。在 30 个实例中，有 22 个实例的 LRB 解超出了两个 LPB 解的凸组合范围。同样，这显示了 LRB 算法在计算上寻找替代最优稳定成本分配方面的价值，即使在 LPB 成本分配显示为最优的情况下也是如此。

5.2 非线性有容量限制设施选址博弈

在一个非线性有容量限制设施选址博弈 (以下简称 NLCFL 博弈) 中，存在一个和无容量限制设施选址博弈 (以下简称 UFL 博弈) 中定义类似的二部网络 $G = (M, N, E)$ 。现每一个潜在的设施点 $i \in M$ 有

容量 Q_i , 每个客户 $j \in N$ 有需求 q_j 。每一个客户只能由一个设施点提供服务。除了开业成本外, 每一间设施 i 也有运营成本, 该成本随着其服务客户数量的增加而增加。为了模拟经济中的规模效应, 我们使用二次函数 $\theta_i[h_i n_i - l_i(n_i)^2]$ 去衡量运营成本, 其中 n_i 为设施 i 服务的顾客数量, θ_i , h_i 和 l_i 是特征函数的参数, 该特征函数是凹函数, 并且随着 $n_i \in [0, n]$ 的增加而增加。在 UFL 博弈中, 参与者仅仅局限于 N 中, 与之不同的是, 在 NLCFL 博弈中, 参与者包括 M 中的设施参与者和 N 中的客户参与者。类似的参与者设置可以在装箱博弈中找到。我们的 LRB 算法也可以处理只有客户参与者参与的情况。然而, 如果参与者中包括设施点参与者, 并且特征函数为非线性的, 我们则可以证明我们的 LRB 算法应用的广泛性。我们需要表 3 中的名词解释, 去定义 NLCFL 博弈。

表 3: 在 NLCFL 博弈中使用的新记号

记号	含义
Q_i	设施 i 的容量, $\forall i \in M, Q_i \in \mathbb{Z}^+$.
q_j	顾客 j 的需求, $\forall j \in N, q_j \in \mathbb{Z}^+$.
s	一个参与者联盟, $s = s_f \cup s_c$.
s_f	联盟 s 中的设施集合, $s_f \subseteq M$.
s_c	联盟 s 中的顾客集合, $s_c \subseteq N$.
γ^s	示性向量 $[\gamma_1^{s_f}, \gamma_2^{s_f}, \dots, \gamma_m^{s_f}, \gamma_1^{s_c}, \gamma_2^{s_c}, \dots, \gamma_n^{s_c}]^T$, 如果 $i \in s_f$ 则 $\gamma_i^{s_f} = 1$ 否则 $\gamma_i^{s_f} = 0$; 如果 $j \in s_c$ 则 $\gamma_j^{s_c} = 1$ 否则有 $\gamma_j^{s_c} = 0$, $\forall j \in N, s_f \subseteq M, s_c \subseteq N$.

定义 4. 一个 NLCFL 博弈 $(M \cup N, c_{NLCFL})$ 由 $M \cup N$ 中的参与者定义, 其中 M 是设施参与者集合, N 是客户参与者集合, 特征函数 $c_{NLCFL}(s)$ 由 NLP 定义:

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

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

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

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

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

与 $c_{UFL}(s)$ 相比, $c_{NLCFL}(s_f \cup s_c)$ 有一些新约束。约束 (26) 代表设施的容量限制; 约束 (27) 确保只有当一个设施参与者在联盟中, 相应的设施才可以用于服务客户。由于 $c_{NLCFL}(s_f \cup s_c)$ 的目标函数在衡量设施运营成本的时候, 具有非线性的部分, LPB 算法不再适用计算成本分配。接下来, 我们将说明 LRB 算法在 NLCFL 博弈上的应用。

5.2.1 NLCFL 博弈中的 LRB 成本分配

在 NLCFL 博弈中的特征函数 $c_{NLCFL}(s_f \cup s_c)$ 中，通过增加一组新的约束

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

，并将约束 $\{\sum_{i \in M} u_{ij} \geq \gamma_j^{s_c} : \forall j \in N\}$ 乘上非负的拉格朗日乘子 σ 加入目标函数中，我们可以得到 NLCFL 博弈中的拉格朗日特征函数，

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

与 c_{UFL} 中的约束 (18) 类似，约束 (29) 的作用是增强 $c_{NLCFL}(s_f \cup s_c)$ 的下界。利用 LRB 算法，我们能把 $c_{LR_NLCFL}(\cdot; \sigma)$ 分解成 $c_{LR1_NLCFL}(\cdot; \sigma)$ 和 $c_{LR2_NLCFL}(\cdot; \sigma)$ ，即 $c_{LR_NLCFL}(s_f \cup s_c; \sigma) = c_{LR1_NLCFL}(s_f \cup s_c; \sigma) + c_{LR2_NLCFL}(s_f \cup s_c; \sigma)$, for all $s_f \subseteq M$ and $s_c \subseteq N$ 并且定义 NLCFL subgame 1 $(M \cup N; c_{LR1_NLCFL}(\cdot; \sigma))$ 和 NLCFL subgame 2 $(M \cup N; c_{LR2_NLCFL}(\cdot; \sigma))$ 。对于 NLCFL subgame 1，特征函数是

$$c_{LR1_NLCFL}(s_f \cup s_c, \sigma) = \sum_{j \in N} \sigma_j \gamma_j^{s_c}. \quad (30)$$

根据引理 1，我们可以得出博弈 $(M \cup N, c_{LR1_NLCFL}(\cdot; \sigma))$ 的核心成本分配，其中每个客户参与者 $j \in N$ 都被分配一个成本，准确等于服务她的拉格朗日对偶价格，没有设施参与者被分配任何费用，因为他们不需要服务。成本分配为 $\alpha_{LR1_NLCFL}^\sigma(j) = \sigma_j$ for all $j \in N$, 和 $\alpha_{LR1_NLCFL}^\sigma(i) = 0$ for all $i \in M$ 。

对于 NLCFL 子博弈 2，特征函数为

$$\begin{aligned}
c_{LR2_NLCFL}(s_f \cup s_c; \sigma) = \min_{v, u} & \sum_{i \in M} f_i v_i + \sum_{i \in M} \sum_{j \in N} (c_{ij} - \sigma_j + \theta_i h_i) u_{ij} - \sum_{i \in M} \theta_i l_i \left(\sum_{j \in N} u_{ij} \right)^2 \\
s.t. & \sum_{j \in N} u_{ij} q_j - Q_i v_i \leq 0, \quad \forall i \in M, \\
& u_{ij} \leq \gamma_i^{s_f}, \quad \forall i \in M, j \in N, \\
& v_i \leq \gamma_i^{s_f}, \quad \forall i \in M, \\
& u_{ij} \leq \gamma_j^{s_c}, \quad \forall i \in M, j \in N, \\
& v_i, u_{ij} \in \{0, 1\}, \quad \forall i \in M, j \in N.
\end{aligned}$$

为了解决 $c_{LR2_NLCFL}(s_f \cup s_c; \sigma)$ ，我们能够依照设施来分离它， $c_{LR2_NLCFL}(s_f \cup s_c; \sigma) = \sum_{i \in s_f} \psi^i(s_c; \sigma)$ ，其中对于每一个 $i \in s_f$ ，

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

可以看出，对于每个问题 $\psi^i(s_c; \sigma)$ 都符合背包问题的变体，该背包问题的目标函数是最小化非线性总值，其中 Q_i 是背包容量， s_c 是物品的集合，每个物品 $j \in s_c$ 都具有权重 q_j 和值 $(c_{ij} - \sigma_j + \theta_i h_i)$ 。除装在背包中的物品的总价值外，参与者可以获得一个额外的价值 $-\theta_i l_i \left(\sum_{j \in s_c} u_{ij} \right)^2$ ，它是装在背包中的物品数量的二次方。当所有权重 q_j 都是整数，我们可以用动态规划在伪多项式时间 $O(Q_i n^2)$ 内解决 $\psi^i(s_c; \sigma)$ 。具体来说，我们将 $F_i^{s_c}(j, k, q)$ 定义为 $\sum_{j' \in s_c, j' \leq j} (c_{ij'} - \sigma_{j'} + \theta_i h_i) u_{ij'}$ 的最小值，以至于 $\sum_{j' \in s_c, j' \leq j} u_{ij'} = k$ 并且 $\sum_{j' \in s_c, j' \leq j} q_{j'} u_{ij'} \leq q$ 。换句话说， $F_i^{s_c}(j, k, q)$ 中的自变量包括从 $\{1, 2, \dots, j\}$ 中取出的 k 个值以及容量 q ，其代表着装入背包中物品的最小价值。动态规划递归方程如下：

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

其初始条件为： $F_i^{s_c}(0, 0, q) = 0$ for $q \geq 0$ 其边界条件为： $F_i^{s_c}(j, k, q) = +\infty$ for $q < 0$ 从而， $\psi_i(s_c; \sigma)$ 可以通过 $\psi_i(s_c; \sigma) = \min_{k \leq |s_c|} \{0, f_i + F_i^{s_c}(n, k, Q_i) - \theta_i l_i k^2\}$ 找到，并且我们有 $c_{LR2_NLCFL}(s_f \cup s_c; \sigma) = \sum_{i \in s_f} \psi_i(s_c; \sigma)$ 。现在，我们已准备通过 CGB 算法计算 NLCFL 子博弈 2 的最佳稳定的成本分配 $\alpha_{LR2_NLCFL}^{\sigma}$ ，此处我们需要解决一个定价问题。在这种特殊情况下，定价问题是用最小的降低成本找到一个联盟（或

列) $s = s_f \cup s_c$, 此处每个 $s = s_f \cup s_c$ 的降低成本为

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

其中, π^* 是 NLCFL 子博弈 2 的相应大规模问题的最佳对偶。对于每个给定的 $s = s_f \cup s_c$, 我们可以直接获得 (32) 的最佳目标值, 它等于 $\sum_{i \in s_f} \psi^i(s_c; \sigma) - \sum_{k \in M \cup N} \gamma_k^s \pi_k^*$, 其中对于每个 $\psi^i(s_c; \sigma)$, 如前所示, 可以通过动态规划来计算。然而, 由于指数数量的联盟, 用枚举法去寻找一个最负减少成本的列 s , 这在计算上是难以处理的。因此, 我们通过考虑以下两种情形, 尝试在一开始定义一个最负减少成本的列 \bar{s} : 情形 1. 至少存在一个 k 使得 $\pi_k^* > 0$ 。在这种情况下, 可以通过包括 k with $\pi_k^* > 0, k \in M \cup N$ 来定义联盟 \bar{s} , 对于 \bar{s} 的降低成本是负的, 因为通过让所有的 u 和 v 都趋向于 0, \bar{s} 至多为 $-\sum_{k \in M \cup N} \max\{0, \pi_k^*\}$ 。情形 2. 对于所有 $k \in M \cup N, \pi_k^* \leq 0$ 。这种情况更加的复杂。为了有效地找到一个负成本的联盟 $\bar{s} = \bar{s}_f \cup \bar{s}_c$, 我们可以考虑以下的 ILP, 其中二进制变量 v_i 和 γ_j 分别表示 \bar{s} 是否包括设施点参与者 i 和客户点参与者 j , 这里 $f'_i = f_i - \pi_i^*, c'_{ij} = c_{ij} - \sigma_j + \theta_i h_i, l'_i = \theta_i l_i$ 并且 $\pi'_j = -\pi_j^*$ 。

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

因此, 可以看出负的目标值得 ILP 的可行解, 是和负的减少成本的联盟 $\bar{s} = \bar{s}_f \cup \bar{s}_c$ 一一对应的。而且, 通过利用以下属性, 这样的联盟 \bar{s} 可以有效获得。

引理 5. 对于 (33), 在不改变最佳目标值的情况下, 我们可以通过以下步骤将一些变量依次固定为零: (1) 对于每个 $(i, j) \in M \times N$, 如果 $c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2] > 0$, 则 $u_{ij} = 0$, 其中 n_i 是集合 $\{c'_{ij} < \infty : \forall j \in N\}$ 中元素的数量。之后, 设置 $c'_{ij} = \infty$ 。(2) 对于每个 $j \in N$, 如果 $\pi'_j + \sum_{i \in M} \min\{c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2], 0\} \geq 0$, 则 $\gamma_j = 0, u_{ij} = 0, \forall i \in M$ 。(3) 对于每个 $i \in M$, 求解非线性背包问题类似于 (31), 其中 Q_i 是背包容量, N 是物品集合, 每个物品 $j \in N$ 具有权重 q_j 和值 c'_{ij} 。让 ω^i 是这个背包问题的最优目标函数值。如果 $\omega^i + f'_i \geq 0$, 则 $v_i = 0, u_{ij} = 0, \text{ for all } j \in N$ 。证明: 首先, 如果存在一对索引 (i, j) , 使得 ILP(33) 的可行解中的 $u_{ij} = 1$ 和 $c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2] > 0$, 我们可以直接设置 $u_{ij} = 0$, 并推导出另一个可行解, 在此基础上, 目标函数值将至少减少 $c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2]$ 。第二, 如果存在客户 j , 使得 ILP(33) 的可行解 $\gamma_j = 1, \pi'_j + \sum_{i \in M} \min\{c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2], 0\} \geq 0$ in a feasible solution of ILP (33), 则对于所有的 $i \in M$ 设置 $\gamma_j = 0, u_{ij} = 0$, 这将推导出另一个可行解, 此处的目标函数值将至少减少 $\pi'_j + \sum_{i \in M} \min\{c'_{ij} - l'_i [n_i^2 - (n_i - 1)^2], 0\}$ 。第三, 如果存在设施 i , 使得 ILP(33) 的可行解中, $v_i = 1$,

$\omega^i + f'_i \geq 0$, 则对于所有的 $j \in N$, 通过设置 $v_i = 0$, $u_{ij} = 0$ 的解也是可行的, 并且目标函数值将减少至少 $\omega^i + f'_i$ 。上述引理只想说明, 做出上述改动并不能增加 $\min_{v,u;\gamma} R(v,u,\gamma)$ 。虽然理论上没有确保多项式时间复杂度, 但是引理 5 中的步骤确实可以大大减少求解 (33) 时的问题规模。在得出 NLCFL 子博弈 1 and 2 最优稳定成本分配 $\alpha_{LR1_NLCFL}^\sigma$ 和 $\alpha_{LR2_NLCFL}^\sigma$ 之后, 我们可以根据定理 1 计算 NLCFL 博弈的稳定成本分配 $\alpha_{LR_NLCFL}^\sigma = \alpha_{LR1_NLCFL}^\sigma + \alpha_{LR2_NLCFL}^\sigma$ 。更重要的是, 通过定理 2, 如果 σ^* 是最优的拉格朗日乘子, 并且 $(M \cup N; c_{LR2_NLCFL}(\cdot; \sigma^*))$ 有一个非空的核心, 则相应的 LRB 成本分配数值 $\sum_{k \in M \cup N} \alpha_{LR_NLCFL}^{\sigma^*}(k)$ 都是等于拉格朗日松弛下界 $c_{LR_NLCFL}(M \cup N; \sigma^*)$ 。

5.2.2 非线性有容量限制设施选址博弈的计算结果

为了进行数值实验, 我们用 20 个由 Beresnev 等人 (2006) 开发的单一资源设施位置基准实例。每个实例均有一个二部网络 $G = (M, N, E)$, 其中 $m = n = 100$, 并且对于所有的 $i \in M$, $f_i = 100$ 。对于每个例子, 都有三个容量级别 10, 20 和 30。另外, 我们使用 $h_1 = h_2 = \dots = h_m = n^2$, $l_1 = l_2 = \dots = l_m = 1$, $\theta_1 = \theta_2 = \dots = \theta_m = \theta$ 去衡量运营成本, 这里 θ 表示运营成本的相对权重。当通过次梯度法去解决拉格朗日对偶问题时, 我们设置的最大迭代次数是 2500。为了显示 LRB 成本分配的有效性, 理想情况下, 我们需要比较总共享成本和大联盟成本 $c_{NLCFL}(M \cup N)$ 。然而, 大联盟的成本只能在 $\theta = 0$ 的基准实例数据集中可以利用。因此, 对于一般比较, 我们需要通过启发式解去替换中心最优解, 该启发式解叫做最好的被发现的中心解 (BFCS), 这被定义为下列两个可行的解决方案中的较好的那个。第一个可行解是 $\theta = 0$ 的 NLCFL 问题的初始基准实例的最优解, Bachrach 等人 (2009 年) 研究过该解。第二个可行解可由 $c_{LR2_NLCFL}(M \cup N; \sigma)$ 的最优解推导出来。提醒一下, 这种最优解对于中心化问题 $c_{NLCFL}(M \cup N)$ 可能是不可行的, 因为有些客户可能没有被服务。如果是这样, 那就是对于给定的不可行解得出一个可行解, 我们可以继续进行如下。对于每个未得到服务的客户, 我们选择一个已开办的设施点, 其有足够的剩余容量, 以及服务这个客户的最小运输成本; 如果没有这样的设施, 我们就开辟拥有最小运输成本的新的可行性设施去服务这个客户。表 4 展示了在 20 个实例中, 设施容量均一致等于 10, 20, 30 的情形下, LRB 成本分配算法应用的效果和计算效率。

表 4: LRB 对于 NLCFL 博弈的成本分配算法表现

容量	θ	LRCA / BFCS (%)			LRCA / LRB(%)			总时间 (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

在表 4 中，“LRCA”表示在不同的 σ 下，找到的最佳 LRB 成本分配值 $\sum_{k \in M \cup N} \alpha_{LR_NLCFL}^\sigma(k)$ ，而“LRB”代表用次梯度法求得的最好的拉格朗日下界 $c_{LR_NLCFL}(M \cup N; \sigma^*)$ 。对于每个容量，我们列出了在不同的 θ 值下的计算结果。从列“LRCA / BFCS”可以看出，对于所有检查的实例，我们的 LRB 成本分配算法可以产生稳定的成本分配，至少达到 BFCS 的 93.93%。当 θ 增加，以至于设施运营成本增加了更多的权重，我们的 LRB 成本分配可以分享 BFCS 超过 99%。这些发现表明了 LRB 成本分配的高品质。而且，尽管 NLCFL 子博弈 2 通常没有次模性，列“LRCA / LRB”表明几乎每个 NLCFL 子博弈 2 都有一个非空的核心。这表明甚至在定理 2 的条件 (2) 不满足的情况下，LRB 成本分配仍有很大可能达到拉格朗日下界。至于时间效率，我们可以看到计算时间趋于随着 Q 和 θ 的增加而增加。在所有情况下，最长的计算时间大约是四个小时。对于 $\theta = 0$ 的情形，NLCFL 博弈在其特征函数中没有非线性项，我们通过展示在表 5 中的成本分配值和基准实例中的大联盟成本之比，即比较 LPB 和 LRB 的成本分配。这里，是 LPB 和 LRB 成本分配是基于在约束 (29) 中提到的相同的特征函数 $c_{NLCFL}(s_f \cup s_c)$ 的 ILP 算式计算的。

表 5: 当 $\theta = \mathbf{0}$ 时, LPB vs. LRB 对于 NLCFL 博弈的成本分配结果 (in %)

容量	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

为了研究约束 (29) 的影响, 我们用新的拉格朗日下界 LRB' 和新的 LRB 成本分配值 $LRCA'$ 去比较 LRB 和 LRCA, 其中 LRB' 和 $LRCA'$ 可从约束 (29) 被松弛后的修订特征函数 $c_{LR_NLCFL}(s; \sigma)$ 中获得。而且, 因为 LPB 的成本分配值和 LP 下界是相等的, 他们都被放在表 5 的“LPCA”栏中, 从表格中, 我们可以观察到以下结论。首先, 平均来看, 如前两个“平均值”下的列所示, 拉格朗日下界比 LP 下界更紧。这意味着 LRB 成本分配相对 LPB 成本分配有潜在的优势。另外, 如“LRCA – LPCA”下的列所示, 平均而言, LRB 成本分配是确实优于 LPB, 特别是对于容量较小的案例。其次, 如“LRB, LRCA, LRB', LRCA'”栏中所示, “将约束 (29) 添加到 $c_{NLCFL}(s_f \cup s_c)$ 确实可以提高拉格朗日下界和 LRB 成本分配值。另外, 通过比较列“LPCA”和“LRCA'”, 我们发现, 平均来看, 即使没有额外的限制条件, LRB 成本分配结果平均仍然优于 LPB。这进一步意味着我们 LRB 算法的竞争力。我们接下来研究 NLCFL 博弈中 LRB 算法的收敛性。这不是 UFL 博弈的情形。在一个 UFL 博弈中, 只要用次梯度法解拉格朗日对偶问题收敛于 σ^* , 因为 UFL 子博弈 2 具有非空的核心, 定理 2 确保了符合 σ^* 的稳定成本分配的最优性。然而, NLCFL 子博弈 2 可能有一个空核, 这就暗示拉格朗日下界和能被分配的总成本之间可能存在的差距。虽然我们可能认为一个一般化的趋势是更好的拉格朗日下界将会带来更好的成本分配, 但是当拉格朗日下界增加时, 并不能保证成本分配的值严格增加。为了检验这一点, 我们将算法 1 应用于 NLCFL 博弈上, 此实例中, $\theta = \mathbf{0}$, 并且通过使用不同的拉格朗日乘子集合 Λ , 比较 LRB 成本分配。表 6 展示了当 Λ 从一个基准集 Λ_2 变到另一个集 Λ_1 , LRB 成本分配分别得到改善, 倒退和没有发生改变的实例数。每个 σ^i 代表在算法 1 的步骤 1 用次梯度法的第 i 次迭代中, 找到的最好的拉格朗日乘子。结果显示, 随着拉格朗日下界的改善, 成本分配可能会变得更加糟糕, 尽管在以后的迭代中, 变坏的可能性非常小。例如, 在一百个实例中, 当 Λ 选择的是 $\{\sigma^{2500}\}$ 而不是 $\{\sigma^{500}, \sigma^{800}, \sigma^{1000}, \sigma^{1500}, \sigma^{2000}\}$ 时, 有七个实例的 LRB 成本分配会下降。这个发现印证了在算法 1 中使用多个拉格朗日乘数的必要性。

表 6: 从不同的 Λ 拉格朗日乘子集合生成的 LRB 成本分配的比较

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

总之, 从计算实验来看, 我们可以得出结论, LRB 算法在解决 NLCFL 博弈的 OCAP 时, 是高效的。

6 结论

本文的研究重点是核心可能为空的合作博弈。我们提出一个通用算法来计算得到一个最优的稳定费用分摊方案, 使其在满足联盟的稳定性的条件下尽可能地覆盖大联盟的成本。在以往地研究中, 这类问题主要通过 LP 松弛和对偶理论求解。我们采用与其不同的拉格朗日松弛理论求解。该方法的拉格朗日松弛下界不仅比现行松弛下界更具有竞争力, 还打破了依赖于可分配约束和线性目标函数的局限。在两种代表典型的设施选址博弈应用该算法, 结果表明该算法能够为这些博弈提供近乎最优的费用分摊方案, 优于现有的 LPB 算法。

参考文献

- 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., Papadopoulos, V.G.(eds.) SAGT 2009. LNCS, vol. 5814, pp. 122–134. Springer, Heidelberg*.
- 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.
- Engvall, 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.
- 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, 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, 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.