

# 协作车辆路径成本分摊问题的 B-T Shapley 方法<sup>①</sup>

饶卫振<sup>1,2</sup>, 朱庆华<sup>1\*</sup>, 金 淳<sup>3</sup>, 刘从虎<sup>1</sup>

(1. 上海交通大学中美物流研究院, 上海 200030; 2. 山东科技大学经济管理学院, 青岛 266590;  
3. 大连理工大学管理与经济学部, 大连 116024)

**摘要:** 多个企业协作配送能显著地降低物流配送成本和减少尾气排放, 研究协作配送模型与成本分摊方法是亟需解决的关键问题. 传统经典成本分摊方法需要计算所有子联盟的协作成本, 在本问题中等价于需要求解  $2^N - 1$  个 ( $N$  为企业数量) 复杂的车辆路径问题. 本文建立了多方协作车辆路径问题模型, 分析了协作配送成本分摊问题的属性. 基于经典的 Shapley 成本分摊方法, 提出了 B-T (Binary Tree) Shapley 近似方法, 不仅将成本分摊本身计算复杂度由  $O(N^2 2^N)$  降为  $O(N^2 \log_2 N)$ , 而且将需要求解的车辆路径问题数量由  $2^N - 1$  个锐减至  $2N - 1$  个, 从而能够在合理时间内完成协作配送问题的成本分摊. 通过求解算例和实际案例, 计算结果表明, B-T Shapley 的耗时与 Shapley 方法相比几乎可以忽略不计, 更重要的是 B-T Shapley 与 Shapley 的成本分摊结果之间仅有细微的偏差, 其平均准确度可以达到 95% 左右.

**关键词:** 协作车辆路径问题; 夏普利值; 成本分摊; 合作博弈

**中图分类号:** TP39 C93    **文献标识码:** A    **文章编号:** 1007-9807(2019)11-0107-20

## 0 引 言

协作配送是多个企业组成联盟进行合作运营的一种现代化配送方式. 协作配送能够有效地降低物流成本和尾气排放, 提升物流配送效率. 国务院在颁发的《物流业发展中长期规划(2014—2020年)》和 2018 国办发 1 号文件《国务院办公厅关于推进电子商务与快递物流协同发展的意见》中明确指出“鼓励物流企业采用协作联盟的方式提升物流业的集约化水平, 鼓励加强快递企业末端配送的协作”. 协作配送中一个需要研究的关键问题是如何有效计算参与协作企业的贡献, 合理分摊协作成本. 该问题主要涉及多配送中心车辆路径问题和成本分摊方法. 针对这两方面的研究较为丰富, 具有较多的相关文献.

针对多配送中心的车辆路径问题 (multi-dep-

ot vehicle routing problem, MDVRP), 国内外众多学者研究了其数学建模与求解算法. 现有研究中的 MDVRP 问题是假设某一个企业具有多个配送中心, 每个配送中心均具有配送车辆, 且要求完成配送后车辆回到出发的配送中心<sup>[1-3]</sup>. 围绕 MDVRP 问题的研究较多集中在有效算法设计方面, 现有比较有效的求解方法包括: Dondo 针对带时间窗的 MDVRP 问题设计减少搜寻空间的过滤规则<sup>[4]</sup>, 有效的多阶段算法<sup>[5,6]</sup>; 及精确算法<sup>[7]</sup>、分支定界法<sup>[8,9]</sup>、变邻域搜索算法<sup>[10]</sup>、自适应大邻域求解算法<sup>[11]</sup>. 针对 MDVRP 问题, 我国学者也做了深入的研究. 钟石泉等研究了求解该问题的禁忌算法<sup>[12]</sup>. 王晓博等在考虑多车场、多车型的基础上还考虑了带装卸混合因素的车辆路径问题<sup>[13]</sup>. 马建华等则针对该问题, 研究了考虑

① 收稿日期: 2017-05-12; 修订日期: 2018-04-27.

基金项目: 国家自然科学基金重点项目(71632007); 国家社会科学基金资助项目(16CGL016); 博士后基金资助项目(2018T110399); 山东省自然科学基金资助项目(ZR2018MG001); 教育部人文社科基金资助项目(15YJC630103).

通讯作者: 朱庆华(1970—), 女, 江苏太仓人, 教授, 博士生导师. Email: qhzh@sjtu.edu.cn

求解效率的变异蚁群算法<sup>[14]</sup>。马宇红等在2014年较为深入地研究了多车场多车型车辆调度问题及其遗传算法<sup>[15]</sup>。当前求解包含100个顾客规模的MDVRP问题,较为有效的算法也需要数分钟甚至十几分钟以上<sup>[8,10]</sup>。另外,在当前研究中,研究假设均基于单个企业拥有多个配送中心为背景,因此不存在货物在配送中心之间调运的情况。然而,协作配送中配送中心分属不同企业,协作运营中需要涉及到货物调运的情况。

针对成本分摊方法,在合作博弈领域中具有较丰富的研究。成本分摊问题是合作博弈领域中的重要部分,自1944年Von Neumann等的著作<sup>[16]</sup>确定合作博弈研究分支以来,众多学者研究了不同的成本分摊方法。比较典型的如Shapley提出的Shapley方法<sup>[17]</sup>、Schmeidler提出的核仁解<sup>[18]</sup>和Tijss提出的Nucleolus值法<sup>[19]</sup>等。目前为止,Guajardo等研究<sup>[20]</sup>发现国内外学者已经研究了多达40多种成本分摊方法。其中由于Shapley值和核仁解方法能够求解得到唯一成本分摊方案,两种方法被广泛应用<sup>[21~25]</sup>。另外,现有研究分析协作后的分摊结果等数据发现,在物流运输中采用协作运营的模式,能够有效地降低30%左右的配送成本和尾气排放<sup>[26~30]</sup>。

然而,在现有的经典成本分摊方法中,均要求获取 $2^N - 1$ 个子联盟 $S$ 的协作成本 $C(S)$ <sup>[20]</sup>。这在协作配送成本分摊问题中,等价于需要求解 $2^N - 1$ 个MDVRP问题。当参与协作的企业数量

$N$ 取值略大时,本问题的计算量就会非常巨大。然而实施物流协作配送的关键问题是建立一个值得信赖的第三方信息共享平台<sup>[31,32]</sup>,采取完全协作集中调度或部分合作的拍卖机制运作<sup>[33]</sup>。如果是完全协作的集中调度方式,则需要综合不同企业的配送中心、顾客位置和需求信息,计算出潜在协作企业通过协作能够节约的成本,并将该信息反馈至企业,从而促成潜在协作企业间的实际协作。因此,企业数量 $N$ 一般较大,针对协作配送中的成本分摊问题有必要研究计算效率更高的成本分摊方法。

## 1 问题描述

### 1.1 协作配送

在传统的物流配送问题中,企业根据顾客位置和需求,优化所需配送车辆数量和不同车辆的行驶路线,在保证完成配送任务的前提下,达到总耗时、总路程或总能耗最小的目标。然而,随着物流需求的迅速增加和城市规模的扩大,配送企业的顾客覆盖区域逐步扩大,这导致配送车辆行驶里程迅速增加,同时降低了配送车辆的利用率。如果处于城市不同区域的配送企业,进行协作配送,即交换部分自身配送中心较远而离对方配送中心较近的顾客,就能有效的降低协作配送企业双方的配送里程和车辆能耗,如图1所示。

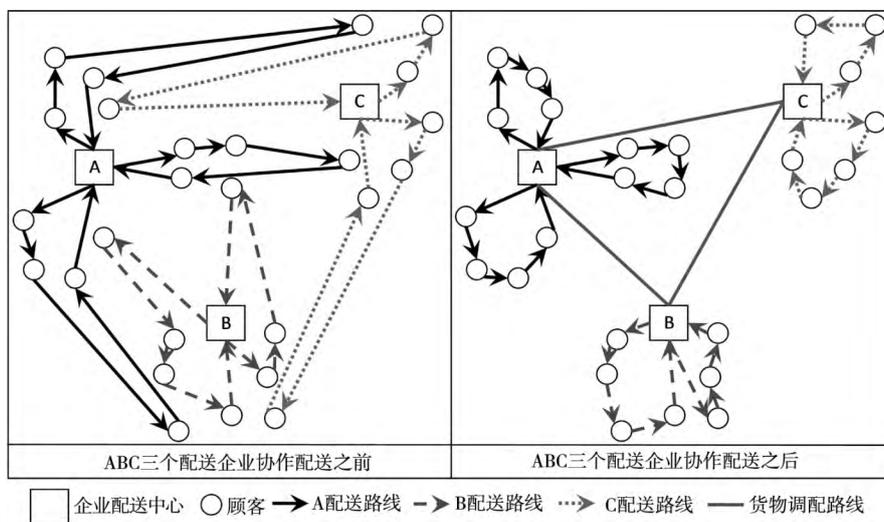


图1 三个企业协作前和协作后的配送示意图

Fig. 1 Comparison diagram of cooperative vehicle routing problem with three enterprises

因此,不同的配送企业组成一个企业联盟,统筹安排联盟内的配送资源所有企业的顾客服务,能够有效提高车辆装载率、降低物流配送成本、提高物流配送效率<sup>[34~37]</sup>。当然,配送企业组成联盟的前提条件是,每个联盟成员在协作配送前就能获悉通过协作配送可以节约的大概成本。因此,需要建立一个有效的用于量化协作配送成本的模型,并提出快速有效的成本分摊方法。

### 1.2 合作博弈中的成本分摊输入数据

公平合理的成本分摊结果是合作博弈成本分摊方法的最终目标。而什么样的成本分摊结果是公平的?显然,在计算成本分摊的过程中需要充分考虑在协作过程中不同成员所做的贡献。

在合作博弈中几乎所有的成本分摊方法,围绕的核心问题都是如何根据成员的贡献,设计运算规则得到成本分摊结果。为了确定不同成员的贡献,合作博弈成本分摊方法中的重要共同环节是,计算合作联盟  $N$  中不同成员不参与协作或联合部分成员,形成子联盟  $S$  ( $S$  包含于  $N, S \neq \emptyset$ ) 时协作的成本  $C(S)$ ,从而得到成本分摊方法的输入数据。

### 1.3 协作配送的成本分摊难题

根据上述分析可知,成本分摊方法所需要的输入数据是,  $2^N - 1$  个 ( $N$  为协作成员数量) 非空

子联盟  $S$  中成员协作所需要的成本  $C(S)$ 。然而对于求解协作配送成本分摊问题,协作的成员为配送企业,所以计算一个  $C(S)$  等价于求解一个如图 1 所示的复杂车辆路径问题。因此存在以下两个难题:

- 1) 如何构建一个协作配送模型,使其能够根据  $N$  个配送企业的配送数据,自动形成任意子联盟  $S$  对应的车辆路径问题模型?
- 2) 采用传统成本分摊方法,需要求解  $2^N - 1$  个复杂车辆路径问题。如何设计合理的成本分摊方法,从而有效减少需要求解的车辆路径问题数量?

## 2 MOCVRP 模型构建

本文研究协作车辆路径问题 (multi-owner collaborative vehicle routing problem, MOCVRP), 其思路是在多配送中心车辆路径问题 (multi-depot heterogeneous fleet vehicle routing problem, MDH-FVRP) 基础上,通过引入子联盟序列变量  $j$ ,将所有  $2^N - 1$  个子联盟  $S$  按照成员归属二进制数组对应的十进制,从小到大排序。表示方法如图 2 所示。

	1	2	...	$N-1$	$N$	
$j=1$	0	0	...	0	1	$S^1=\{N\}$
$j=2$	0	0	...	1	0	$S^2=\{N-1\}$
$j=3$	0	0	...	1	1	$S^3=\{N-1,N\}$
...	...	...	...	...	...	
$j=2^N-1$	1	1	...	1	1	$S^{2^N-1}=\{1,2,\dots,N-1,N\}$

图 2 子联盟  $S$  排序方法

Fig. 2 The sorting method sub-coalition  $S$

本部分的核心内容是结合上述表示子联盟方法,在考虑货物在配送中心之间调运前提下,建立能够自动计算任意多个企业协作配送所需成本的数学模型。

问题描述如下:

已知  $N$  个配送企业,每个企业拥有 1 个配送中心和  $K_i$  辆最大载量为  $Q$  的同型号车辆,共有  $n$  个需求量为  $q_i$  的顾客需要配送。现假设  $N$  个配送企业组成联盟进行协作配送,那么协作配送的车辆路径如何优化?如何有效计算配送成本的合理

分摊结果?

涉及的所有数学符号说明如下:

$N$ : 所有配送企业的数量或集合(设1个企业仅有1个配送中心);

$n$ : 所有顾客的数量;

$C_i$ : 第  $i$  ( $1 \leq i \leq N$ ) 个配送企业的顾客集合  $\{1, 2, \dots, n_i\}$ ;

$C$ : 所有  $N$  个配送企业的顾客集合,  $C = \bigcup_{i=1}^N C_i = \{1, 2, \dots, n\}$ ;

$D_i$ : 第  $i$  ( $1 \leq i \leq N$ ) 个配送企业的配送中心集合  $\{n + i\}$ ;

$D$ : 所有  $N$  个配送企业的配送中心集合,  $D = \bigcup_{i=1}^N D_i = \{n + 1, n + 2, \dots, n + N\}$ ;

$K_i$ : 第  $i$  ( $1 \leq i \leq N$ ) 个配送企业的车辆集合  $\{1, 2, \dots, k_i\}$ ;

$K$ : 所有  $N$  个配送企业的车辆集合,  $K = \bigcup_{i=1}^N K_i = \{1, 2, \dots, k\}$ ;

$Q$ : 配送车辆的载量;

$L$ : 配送车辆允许最大行驶距离;

$q_i$ : 第  $i$  ( $1 \leq i \leq n$ ) 个顾客需求量, 且  $0 \leq q_i \leq Q$ ;

$Q_{ij}$ : 需要从配送中心  $i$  ( $1 \leq i \leq n$ ) 向配送中心  $j$  ( $1 \leq j \leq n$ ) 调配的货物量;

$S^j$ : 第  $j$  ( $1 \leq j \leq 2^N - 1$ ) 个企业子联盟, 且  $S^j \neq \emptyset$ ;

$B2D(\cdot)$ : 表示将  $1 \times N$  维 0-1 向量“·”看成 2 进制数据, 并将其转化为十进制值;

$D2B(\cdot)$ : 为  $B2D(\cdot)$  的逆运算, 即将数值小于等于  $2^N - 1$  的十进制转化为  $N$  位的二进制, 并表示为  $1 \times N$  维 0-1 向量;

$B^j$ : 表示与  $S^j$  对应的  $1 \times N$  维成员归属向量, 其中当  $i \in S^j$  ( $1 \leq i \leq N$ ) 时  $B^j(i) = 1$ , 否则  $B^j(i) = 0$ , 且  $B2D(B^j) = j, D2B(j) = B^j$ ;

$C^j$ : 表示第  $j$  ( $1 \leq j \leq 2^N - 1$ ) 个企业子联盟  $S^j$  中所有企业成员的顾客集合,  $C^j = \bigcup_{i=1}^N (B^j(i) \times C_i)$ ;

$D^j$ : 表示第  $j$  ( $1 \leq j \leq 2^N - 1$ ) 个企业子联盟  $S^j$  中所有企业成员的配送中心集合,  $D^j = \bigcup_{i=1}^N (B^j(i) \times D_i)$ ;

$K^j$ : 表示第  $j$  ( $1 \leq j \leq 2^N - 1$ ) 个企业子联盟

$S^j$  中所有企业成员的车辆集合,  $K^j = \bigcup_{i=1}^N (B^j(i) \times K_i)$ ;

$c_{st}$ : 任意两顾客或配送中心  $s, t$  之间的配送成本;

$d_{st}$ : 任意两顾客或配送中心  $s, t$  之间的配送距离;

$x_{stk}$ : 为 0, 1 变量, 等于 1 表示配送车辆  $k \in K^j$  经过弧  $(s, t)$  ( $s, t \in C^j \cup D^j$ ), 否则等于 0.

$y_{sth}$ : 为 0, 1 变量, 等于 1 表示配送车辆  $k \in K^j$  经过弧  $(s, t)$  ( $s, t \in D^j$ ), 否则等于 0.

$z_{ih}$ : 为 0, 1 变量, 等于 1 表示顾客  $i$  ( $i \in C^j$ ) 由第  $h$  ( $h \in D^j$ ) 个配送中心派出车辆服务, 否则等于 0.

模型基于的假设条件:

- 1) 每个企业拥有一个配送中心;
- 2) 每辆车的起点和终点均为同一配送中心;
- 3) 每辆车均不允许超过最大载量  $Q$ ;
- 4) 每个顾客只能由 1 辆车服务 1 次;
- 5) 每辆车行驶的距离不超过  $L$  (当不考虑该约束时, 设  $L$  为足够大的数);
- 6) 每个配送中心的车辆数  $K_i$  ( $1 \leq i \leq N$ ), 均能满足对应顾客总需求, 即不考虑运力不够.

对于第  $j$  ( $1 \leq j \leq 2^N - 1$ ) 个子联盟  $S^j$  对应的 MOCVRP 模型如下:

$$\text{Min } C(S^j) = \sum_{s, t \in C^j \cup D^j} \sum_{k \in K^j} c_{st} x_{stk} + \sum_{s, t \in D^j} \sum_{k \in K^j} d_{st} y_{sth} \tag{1}$$

s. t.

$$\sum_{s \in C^j \cup D^j} \sum_{k \in K^j} x_{stk} = 1 \quad (t \in C^j) \tag{2}$$

$$\sum_{t \in C^j \cup D^j} \sum_{k \in K^j} x_{stk} = 1 \quad (s \in C^j) \tag{3}$$

$$\sum_{s \in C^j \cup D^j} x_{shk} = \sum_{t \in C^j \cup D^j} x_{thk} \quad (k \in K^j, h \in C^j \cup D^j) \tag{4}$$

$$\sum_{s \in D^j} y_{shk} = \sum_{t \in D^j} y_{thk} \quad (k \in K^j, h \in D^j) \tag{5}$$

$$\sum_{s \in C^j \cup D^j} \sum_{t \in C^j \cup D^j} q_t x_{stk} \leq Q \quad (k \in K^j) \tag{6}$$

$$Q_{st} = \sum_{u \in C_s \subset C^j} q_u z_{ut} \tag{7}$$

$$\sum_{s \in D^j} \sum_{t \in D^j} Q_{st} y_{sth} \leq Q \quad (k \in K^j) \tag{8}$$

$$\sum_{s \in C^j \cup D^j} \sum_{t \in C^j \cup D^j} d_{st} x_{stk} \leq L (k \in K^j) \quad (9)$$

$$\sum_{s \in D^j} \sum_{t \in C^j} x_{stk} \leq 1 (k \in K^j) \quad (10)$$

$$\sum_{t \in D^j} \sum_{s \in C^j} x_{stk} \leq 1 (k \in K^j) \quad (11)$$

$$y_s - y_t + (|C^j| + |D^j|) x_{stk} \leq |C^j| + |D^j| - 1 \quad (12)$$

$(k \in K^j, s, t \in C^j)$

$$x_{stk} \in \{0, 1\} \forall s, t \in C^j \cup D^j, k \in K^j \quad (13)$$

$$y_{stk} \in \{0, 1\} \forall s, t \in D^j, k \in K^j \quad (14)$$

$$y_s \in \{0, 1\} \forall s \in C^j \quad (15)$$

目标函数(1)是指最小化子联盟  $S^j$  中的配送企业协作配送的总成本; 约束条件(2)和约束条件(3)保证每个顾客被 1 辆车确切的服务一次; 约束条件(4)和约束条件(5)保证每个节点在配送和货物调运过程中流量净值为 0; 约束条件(6)、约束条件(7)和约束条件(8)分别表示每辆车在配送和货物调运过程中不超过载量; 约束条件(9)表示货物配送中车辆行驶距离不超过指定参数  $L$ ; 约束条件(10)和约束条件(11)表示配送车辆至多从配送中心被派出 1 次, 也至多回到配送中心 1 次; 约束条件(12)表示每辆车行驶的路径轨迹恰好为一个简单圈; 约束条件(13)、约束条件(14)和约束条件(15)是变量的取值范围.

### 3 MOCVRP 成本分摊问题性质分析

物流协作配送中的成本分摊问题, 属于合作博弈问题, 在合作博弈问题中其博弈均衡解为核心解, 然而, 在很多合作博弈问题中并不存在核心解. 但如果合作博弈问题属于凸博弈, 那么对合作博弈问题必定存在核心解, 并且 Shapley 值方法求解的结果必定为核心解. 通过证明下列定理 1 和定理 2 发现, 本文研究的问题在满足一定条件时为凸博弈问题.

**定理 1** 在一面积为  $A$  的矩形区域中, 假设均匀随机分布了  $n - 1$  个顾客和 1 个配送中心, 所有配送车辆行驶的最小总距离的期望长度为  $\alpha\sqrt{An}$ , 且  $\alpha$  为常数.

**证明** Beardswood 在研究通过均匀分布在固定矩形区域(面积为  $A$ )中  $n$  个点的最短路发现<sup>[38]</sup>, 其最短路的长度可近似表示为  $\alpha\sqrt{An}$ , 其中  $\alpha$  为常

数. Helsgaun 通过研究发现在旅行商问题中, 该结论完全成立并且通过大量的实验计算确定该常数  $\alpha \approx 0.7124$ <sup>[39]</sup>.

因此, 假设在面积为  $A$  的矩形区域中均匀随机分布的  $n - 1$  个顾客, 根据需求总量和车载量信息要求派送  $k$  辆车, 并且第  $i$  ( $1 \leq i \leq k$ ) 辆车服务的顾客数量为  $a_i n$ , 服务的顾客覆盖区域面积  $b_i A$ . 由于  $k$  辆车服务了所有的顾客, 故  $a_1 + a_2 + \dots + a_k = b_1 + b_2 + \dots + b_k = 1$ .

根据 Beardswood 和 Helsgaun 研究结论可知,  $k$  辆车的最短行驶路径期望长度可以用式(16)计算.

$$TL = \sum_{i=1}^k \alpha \sqrt{a_i n b_i A} = \alpha \sqrt{nA} \sum_{i=1}^k \sqrt{a_i b_i} \quad (16)$$

故要证明定理 1 成立, 等价于证明  $\sum_{i=1}^k \sqrt{a_i b_i} = 1$  成立. 因为假设顾客位置在面积为  $A$  的矩形区域中均匀随机分布, 所以从理论上  $a_i = b_i$  ( $1 \leq i \leq k$ ) 成立, 那么式(16)可以改写为式(17).

$$TL = \alpha \sqrt{nA} \sum_{i=1}^k \sqrt{a_i a_i} = \alpha \sqrt{nA} \sum_{i=1}^k a_i = \alpha \sqrt{nA} \quad (17)$$

据上述证明可知, 定理 1 成立. 证毕.

基于定理 1, 进一步可以证明定理 2 成立.

**定理 2** 假设每个企业的顾客和配送中心都均匀随机分布在面积为  $A$  的矩形区域内, 其中配送中心分布区域面积为  $B \leq A$ , 企业配送成本与路线距离成正比, 那么  $N$  ( $N \geq 3$ ) 个企业协作配送的成本分摊问题为凸博弈问题, 即假设任意两个企业子联盟  $S \subseteq N$  和  $T \subseteq N$  且均不为空集时, 有下列不等式  $C(S \cup T) \leq C(S) + C(T) - C(S \cap T)$  成立.

**证明情况 1.**  $S \cap T = \emptyset$ , 证明定理 2 中的不等式成立, 即证明此时的博弈具有超可加性的. 通过上述 MOCVRP 模型可知, 子联盟  $S$  的协作配送的最小成本即为  $C(S)$ , 那么根据目标极小的优化问题可知,  $S$  和  $T$  子联盟的最优协作方案合并, 即为联盟  $S \cup T$  的一个可行解, 显然,  $S \cup T$  成本等于  $C(S \cup T)$  的最优方案必定不高于其任何可行解的方案. 此时,  $C(S \cup T) \leq C(S) + C(T) - C(S \cap T) = C(S) + C(T)$  恒成立.

情况2  $S \cap T \neq \emptyset$ , 则不妨假设子联盟  $S = \{i, i+1, \dots, j\}$ ,  $T = \{h, h+1, \dots, k\}$ , 且  $1 \leq i \leq h \leq j \leq k \leq N$ . 那么根据定理1有式(18)~式(21)成立.

$$C(S \cup T) = \alpha\sqrt{(n_i+n_{i+1}+\dots+n_k)A} + \alpha\sqrt{(k-i+1)B} \quad (18)$$

$$C(S) = \alpha\sqrt{(n_i+n_{i+1}+\dots+n_j)A} + \alpha\sqrt{(j-i+1)B} \quad (19)$$

$$C(T) = \alpha\sqrt{(n_h+n_{h+1}+\dots+n_k)A} + \alpha\sqrt{(k-h+1)B} \quad (20)$$

$$C(S \cap T) = \alpha\sqrt{(n_h+n_{h+1}+\dots+n_j)A} + \alpha\sqrt{(j-h+1)B} \quad (21)$$

为了方便描述, 将式(18)~式(21)精简为式(22)~式(24).

$$C(S) = \alpha x_1 \sqrt{A} + \alpha x_2 \sqrt{B} \quad (22)$$

$$x_1 = \sqrt{n_i+n_{i+1}+\dots+n_j}, x_2 = \sqrt{j-i+1}$$

$$C(T) = \alpha y_1 \sqrt{A} + \alpha y_2 \sqrt{B}$$

$$y_1 = \sqrt{n_h+n_{h+1}+\dots+n_k}, y_2 = \sqrt{k-h+1} \quad (23)$$

$$C(S \cap T) = \alpha z_1 \sqrt{A} + \alpha z_2 \sqrt{B} \quad (24)$$

$$z_1 = \sqrt{n_h+n_{h+1}+\dots+n_j}, z_2 = \sqrt{j-h+1}$$

由于  $n_i$  代表的是第  $i$  个企业的顾客数量, 因此任意  $n_i \geq 1$ , 显然有  $1 \leq z_1 \leq \min\{x_1, y_1\}$ ,  $1 \leq z_2 \leq \min\{x_2, y_2\}$ , 且式(25)成立.

$$C(S \cup T) = \alpha\sqrt{(x_1^2 + y_1^2 - z_1^2)A} + \alpha\sqrt{(x_2^2 + y_2^2 - z_2^2)B} \quad (25)$$

由于系数和顾客分布区域面积  $A, B$  均大于0, 因此要证明定理2成立, 即证明式(26)和式(27)成立.

$$\sqrt{x_1^2 + y_1^2 - z_1^2} \leq x_1 + y_1 - z_1 \quad (26)$$

$$\sqrt{x_2^2 + y_2^2 - z_2^2} \leq x_2 + y_2 - z_2 \quad (27)$$

由于  $x_1, y_1, z_1$  和  $x_2, y_2, z_2$  均大于等于1, 证明式(26)和式(27)成立, 等价于证明式(28)和式(29)成立.

$$x_1^2 + y_1^2 - z_1^2 \leq (x_1 + y_1 - z_1)^2 \quad (28)$$

$$x_2^2 + y_2^2 - z_2^2 \leq (x_2 + y_2 - z_2)^2 \quad (29)$$

要证明式(28)和式(29)成立, 即证明, 当  $1 \leq z \leq \min\{x, y\}$  时, 式(30)成立

$$f(z) = (x+y-z)^2 - (x^2 + y^2 - z^2) \geq 0 \quad (30)$$

将式(30)中的  $f(z)$  可整理为

$$f(z) = z^2 - (x+y)z + xy = (z-x)(z-y) \quad (31)$$

对  $f(z)$  求一阶导数得

$$f'(z) = 2z - (x + y) \quad (32)$$

显然,  $1 \leq z \leq \min\{x, y\}$ ,  $f'(z) \leq 0$ , 所以  $z = \min\{x, y\}$  时,  $f(z) = 0$  的取值最小. 即  $1 \leq z \leq \min\{x, y\}$  时,  $f(z) \geq 0$ . 故定理2得证.

## 4 B-T Shapley 方法

### 4.1 Shapley 方法

合作博弈理论中, 利益分配或成本分摊的核心解是合作博弈的均衡解, 能保证联盟的稳定性. 但是并不是所有的合作博弈均存在核心解, 因此包括本文成本分摊在内的合作博弈问题, 以寻求核心解为目标可能无法得到最终解.

Shapley 法能够在任何情况下求解到唯一结果, 且当问题是凸博弈时能够保证结果为核心解, 因此被广泛用于解决不同领域的利益分配和成本分摊问题. Shapley 值的理论原理是根据每个合作个体对联盟的边际贡献大小, 确定其应该分享的利益或者承担的成本<sup>[17]</sup>. 本文中, 其第  $i$  ( $1 \leq i \leq N$ ) 个配送企业应该承担的成本, 计算式如式(33)所示.

$$\Phi(i) = \sum_{j=1}^{2^N-1} \frac{(|S^j|-1)! \times (N-|S^j|)!}{N!} \times [C(S^j) - C(S^j \setminus \{i\})] \quad (33)$$

$(1 \leq i \leq N)$

根据 Shapley 方法的性质, 式(33)计算的成本分摊方案必定为转归解, 即满足1) 每个企业的分摊成本大于等于0, 且必定不高于该企业单独配送的成本; 2) 所有  $N$  个企业分摊的成本之和等于  $N$  个企业协作配送的总成本.

### 4.2 Shapley 模型求解计算复杂度分析

由式(33)可知, Shapley 值法计算每个协作企业应分摊的成本时, 均需要遍历所有的  $2^N - 1$  个子联盟, 并且每个联盟需要用长度为  $N$  的0-1数组表示. 设1次加减乘除, 阶乘和判断计算量等同, 根据式(33), 可知其计算量所有  $N$  个参与企业分摊的成本量, 需要的计算量如式(34)所示.

$$\Omega(\text{Shapley}) = N^2 [(2^N - 1) + 8] = 2^N N^2 + 7N^2 \quad (34)$$

因此其计算复杂度为  $O(2^N N^2)$ , 属于指数级

计算复杂度. 另外值得注意的是, 本问题中所有的  $2^N - 1$  个子联盟  $S$  对应的成本  $C(S)$ , 均需要对应求解一个 MOCVRP 问题才能得到. 现假设每个协作企业具有相同顾客数量, 假设为  $n_0$  个. 那么复杂度为  $O(n^2)$  的算法求解  $2^N - 1$  个子联盟  $S$  对应的 MOCVRP 问题, 累计需要的计算量如式(35) 所示.

$$\Omega(\text{Shapley-CS}) = \sum_{i=1}^N C_N^i (i \times n_0)^2 = n_0^2 \sum_{i=1}^N C_N^i i^2 \quad (35)$$

表 1 Shapley 不同  $N$  和  $n_0$  下求解 MOCVRP 需要的耗时量(单位: h)

Table 1 The running time of computing MOCVRP in Shapley with different  $N$  and  $n_0$

$n_0 \backslash N$	20	40	60	80	100	150	200
2	1.20E-02	4.80E-02	1.08E-01	1.92E-01	3.00E-01	6.75E-01	1.20E+00
4	1.60E-01	6.40E-01	1.44E+00	2.56E+00	4.00E+00	9.00E+00	1.60E+01
6	1.34E+00	5.38E+00	1.21E+01	2.15E+01	3.36E+01	7.56E+01	1.34E+02
8	9.22E+00	3.69E+01	8.29E+01	1.47E+02	2.30E+02	5.18E+02	9.22E+02
10	5.63E+01	2.25E+02	5.07E+02	9.01E+02	1.41E+03	3.17E+03	5.63E+03
12	3.19E+02	1.28E+03	2.88E+03	5.11E+03	7.99E+03	1.80E+04	3.19E+04
14	1.72E+03	6.88E+03	1.55E+04	2.75E+04	4.30E+04	9.68E+04	1.72E+05
16	8.91E+03	3.57E+04	8.02E+04	1.43E+05	2.23E+05	5.01E+05	8.91E+05
18	4.48E+04	1.79E+05	4.03E+05	7.17E+05	1.12E+06	2.52E+06	4.48E+06
20	2.20E+05	8.81E+05	1.98E+06	3.52E+06	5.51E+06	1.24E+07	2.20E+07

通过表 1 可知, 随着  $N$  和  $n_0$  的增加, 本文研究的问题按照 Shapley 分摊成本的方法, 即使采用当前快速的启发式算法通过求解  $2^N - 1$  个 MOCVRP 问题, 获取  $C(S)$  信息需要的耗时量也会迅速增加. 如当  $N=10$  和  $n_0=60$  时获取所有的  $C(S)$  需要 507 h, 即需要 20 多天的运算耗时, 显然, 这已完全超出合理的耗时范围. 然而现实中的同一地区的物流配送企业数量和顾客数量还远大于该数量.

### 4.3 B-T Shapley 方法

由 4.2 部分可知, Shapley 分摊成本的方法需要计算所有的  $C(S)$ . 当参与协作的企业数量  $N$  和平均顾客数量  $n_0$  较大时, 计算量非常巨大. 而实际物流协作配送中的  $N$  和  $n_0$  数据值分别可达几十和几百, 因此, 本文研究求解效率更高的成本分摊方法.

当前有不少学者基于 Shapley 分摊方法, 从不同角度提出了新的分摊方法<sup>[40]</sup>. 本文从提高 Shapley 计算效率的角度, 基于 Shapley 计算方法, 采用二叉树 Binary Tree 分割法, 设计了求解效率更高的 B-T Shapley 方法. 其原理是: 首先将大联盟  $N$  一分为二, 形成两个子联盟; 然后将子

联盟再一分为二, 并不断重复该过程, 直到所有的子联盟只包含一个成员; 最后, 将每次一分为二的过程看成 1 个联盟由两个成员构成, 再基于 Shapley 方法求解包含两个成员的成本分摊问题, 并按此思路逐级得到每个成员的分摊成本.

考虑到输出结果的唯一性, 现设立 Binary Tree 分割的规则: 当联盟  $S$  的成员个数  $|S|$  为偶数时, 分成的左子联盟和右子联盟成员的个数恰好等于  $|S|/2$ , 如果  $|S|$  为奇数时, 定义左子联盟由  $S$  的前  $(|S| + 1)/2$  个成员构成, 右子联盟由后  $(|S| - 1)/2$  个成员构成. 如包含 7 个成员的联盟采用 Binary Tree 分割后如图 3 所示.

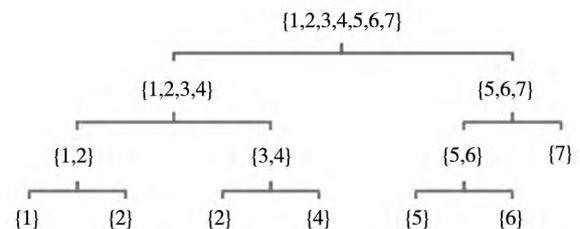


图 3 Binary Tree 分割包含 7 成员联盟示意图

Fig. 3 The Binary Tree structure of a coalition with 7 players

为了便于描述,在此引入下列数学符号

$L(N)$ : 一个包含  $N$  个成员的联盟采用二叉树分割产生的层数  $L(N) = \lceil \log_2 N \rceil + 1$  其中  $\lceil \cdot \rceil$  表示对“·”向上取整;

$k$ : 表示二叉树的第  $k$  层,显然  $1 \leq k \leq L(N)$ ;

$S_{ki}$ : 表示二叉树中第  $k$  层中的第  $i$  个节点对应的子联盟,由于第  $k$  层共有  $2^{k-1}$  个节点,则  $1 \leq i \leq 2^{k-1}$ ;

$F(S_{ki}) = S_{k-1, i_{k-1}}$ : 表示二叉树中子联盟  $S_{ki}$  的父节点对应的子联盟,不存在时等于 0;

$B(S_{ki}) = S_{k, \bar{i}}$ : 表示二叉树中子联盟  $S_{ki}$  的兄弟节点对应的子联盟  $S_{k, \bar{i}}$ , 其中  $\bar{i} = i + 1$  或  $\bar{i} = i - 1$ , 且  $B(S_{k, \bar{i}}) = S_{ki}$ ;

$C(S_{ki})$ : 表示二叉树中子联盟  $S_{ki}$  协作配送的实际成本;

$C'(S_{ki})$ : 表示二叉树中子联盟  $S_{ki}$  分摊的成本;

$\alpha_{ki}$ : 根据分摊前联盟成本信息,基于 Shapley 方法计算得到的子联盟  $S_{ki}$  分摊成本占父节点对应子联盟成本的比率,显然  $0 \leq \alpha_{ki} \leq 1$ .

基于上述二叉树分割联盟的方式可知,任何情况下  $S_{k, \bar{i}} \cap S_{ki} = \emptyset, S_{k, \bar{i}} \cup S_{ki} = F(S_{ki}) = F(S_{k, \bar{i}}) = S_{k-1, i_{k-1}} = S_{k-1, \bar{i}_{k-1}}$  均会成立. 且根据 Shapley 式(33) 计算包含两个成员的成本分摊问题,由式(36) 可计算  $\alpha_{ki}$ .

$$\alpha_{ki} = \frac{C(F(S_{ki})) + C(S_{ki}) - C(B(S_{ki}))}{2C(F(S_{ki}))} \tag{36}$$
$$= \frac{C(S_{k-1, i_{k-1}}) + C(S_{ki}) - C(S_{k, \bar{i}})}{2C(S_{k-1, i_{k-1}})}$$

根据式(36) 可知  $\alpha_{ki} + \alpha_{k, \bar{i}} = 1$ , 并且结合二叉树中子联盟的  $C(S_{ki})$  信息,可以计算出所有的  $\alpha_{ki}$ .

且令  $C'(S_{ki})$  由式(37) 计算.

$$C'(S_{ki}) = \alpha_{ki} \cdot C(F(S_{ki}))$$
$$= \alpha_{ki} \cdot C(S_{k-1, i_{k-1}}) \tag{37}$$

在此需要注意的是,对于大联盟  $N$  有  $C'(N) = C(N)$ , 即  $\alpha_{11} = 1$ . 根据该信息和式(37) 不难在二叉树中逐级计算出所有  $C'(S_{ki})$  的信息,其中二叉树中最后一层或倒数第二层中成员数量等于 1 的,即为各成员最终分摊成本,故式(37) 可以写成式(38).

$$C'(S_{ki}) = (\alpha_{2, i_2} \cdot \alpha_{3, i_3} \cdots \alpha_{k-1, i_{k-1}} \cdot \alpha_{ki}) \cdot C(N) \tag{38}$$

以图 2 中的二叉树分割包含 7 个成员的联盟为例,结合上述式,可得  $S_{44} = \{4\}$  的分摊成本,计算过程如下式所示:

$$C'(S_{44}) = (\alpha_{21} \cdot \alpha_{32} \cdot \alpha_{44}) \cdot C(\{1, 2, 3, 4, 5, 6, 7\})$$

#### 4.4 B-T Shapley 复杂度分析

由论文 4.3 部分可知 B-T Shapley 方法主要计算量包括:采用式(36) (包括 2 次减法计算 + 1 次除法计算 + 1 次乘法计算 = 4 次基本计算) 计算所有二叉树节点  $(2N - 1)$  对应的  $\alpha_{ki}$ , 采用式(38) (包括  $\log_2 N$  次乘法计算) 计算所有包括 1 个成员子联盟(共存在  $N$  个)的分摊成本. 另外,每个子联盟采用一个  $N$  维 0-1 向量表示. 因此, B-T Shapley 方法的计算复杂度如式(39) 所示.

$$\Omega(B-T Shapley) = 4(2N-1) \times N + N \times N \log_2 N \tag{39}$$
$$= N^2 \log_2 N + 8N^2 - 4N$$

结合  $\alpha_{ki} + \alpha_{k, \bar{i}} = 1$  可知,在计算出  $\alpha_{ki}$  后,其兄弟节点对应子联盟的  $\alpha_{k, \bar{i}} = 1 - \alpha_{ki}$  不再需要通过式(36) 计算得到,如前所述,任何情况下  $\alpha_{11} = 1$  不需要计算,即在计算  $\alpha_{ki}$  时其计算量可以进一步简化,因此,式(39) 所示的计算量超过了实际需要的计算量式(40).

$$\Omega(B-T Shapley) = 5(N-1)N + N \times N \log_2 N \tag{40}$$
$$= N^2 \log_2 N + 5N^2 - 6N$$

通过比较式(34) 和式(40) 可知,经典 Shapley 方法的计算复杂度达到  $N^2 2^N$ , 而 B-T Shapley 仅为  $N^2 \log_2 N$ .

针对求解本文研究的问题,由于 Binary Tree 分割包含  $N$  各成员的联盟只会形成  $2N - 1$  个节点,即只要求解  $2N - 1$  个子联盟对应的 MDVRP 问题. 同 4.2 部分假设每个协作企业具有类似顾客数量,假设为  $n_0$  个,那么根据 Binary Tree Shapley 方法可得,计算复杂度为  $O(n^2)$  的 VNS 算法求解  $2N - 1$  个子联盟对应的 MOCVRP 问题所需要的计算量,如式(41) 所示.

$$\Omega(B-T Shapley-CS) = \sum_{k=1}^{L(N)} \sum_{i=1}^{2^{k-1}} \Omega[C(S_{ki})] \tag{41}$$
$$= \sum_{k=1}^{L(N)} \sum_{i=1}^{2^{k-1}} \left(\frac{N}{2^{k-1}} n_0\right)^2$$
$$= (2N^2 - N) n_0^2$$

通过对比式 (35) 可知, B-T Shapley 相比 Shapley 在获取  $C(S)$  信息时, 能够节约巨大的计算量. 为了方便与 Shapley 计算量进一步直观比较, 同样以 Sali 的研究结果: 在电脑 CPU Pentium IV-M with 1GB 内存平台上, 采用 VNS 算法

求解一个 100 个顾客的 MDVRP 问题需要 3min (0.05 h) 为折算标准, 在表 2 中按折算标准计算了 B-T Shapley 在不同的  $N$  和  $n_0$  组合下获取需要的  $2N - 1$  个  $C(S)$  需要的耗时, 如表 2 所示.

表 2 B-T Shapley 不同  $N$  和  $n_0$  下求解 MOCVRP 需要的耗时量(单位: h)

Table 2 The running time of computing MOCVRP in B-T Shapley with different  $N$  and  $n_0$ (h)

$N \backslash n_0$	20	40	60	80	100	150	200
2	0.01	0.05	0.11	0.19	0.30	0.68	1.20
4	0.06	0.22	0.50	0.90	1.40	3.15	5.60
6	0.13	0.53	1.19	2.11	3.30	7.43	13.20
8	0.24	0.96	2.16	3.84	6.00	13.50	24.00
10	0.38	1.52	3.42	6.08	9.50	21.38	38.00
12	0.55	2.21	4.97	8.83	13.80	31.05	55.20
14	0.76	3.02	6.80	12.10	18.90	42.53	75.60
16	0.99	3.97	8.93	15.87	24.80	55.80	99.20
18	1.26	5.04	11.34	20.16	31.50	70.88	126.00
20	1.56	6.24	14.04	24.96	39.00	87.75	156.00

通过比较表 1 和表 2 可得, B-T Shapley 的耗时量远远低于 Shapley, 即使求解表中最大规模情况下的  $N = 20, n_0 = 200$  问题, 折算的耗时仅需要 156 h, 而 Shapley 求解对应问题需要耗时 2 200 万 h(约相当于 2 511 年). 因此, B-T Shapley 采用普通电脑就可以计算规模较大的协作配送成本分摊问题, 如果借助高性能计算机或分布型算法, 可能将耗时控制在更加合理时间范围内或求解规模更大的问题.

#### 4.5 B-T Shapley 求解质量分析

B-T Shapley 相比 Shapley 方法能够节约大量的计算, 那么其计算结果的质量是否能够保证在合理范围内? 本部分主要分析 B-T Shapley 的求解质量.

需要指出的是, B-T Shapley 求解结果与初始大联盟中成员排列顺序相关, 根据排列组合和对称关系可知, 对于恰好包括  $N = 2^l$  个合作成员的问题, 成员在大联盟中共有  $N! / 2^{N-1}$  种使 B-T Shapley 输出结果不同的排列方式. 通过研究发现成员在初始联盟中按任意顺序排列, B-T Shapley 方法的输出结果均为合作博弈中的转归解, 即有定理 3 成立.

**定理 3** B-T Shapley 求解协作配送成本分摊问题的结果必定为转归解, 即满足 1) 每个企业

的分摊成本大于等于 0, 且必定不高于该企业单独配送的成本; 2) 所有  $N$  个企业分摊的成本之和会等于  $N$  个企业协作配送的总成本  $C(N)$ .

**证明** 1) 由式 (38) 可知, 二叉树中任意节点对应的子联盟分摊成本会等于  $(\alpha_{i_2} \cdot \alpha_{i_3} \cdots \alpha_{i_{k-1}} \cdot \alpha_{i_k}) \cdot C(N)$ , 又任意  $\alpha_{i_k}$  均大于等于 0 小于等于 1, 且  $C(N) > 0$ , 因此, 每个企业最后的分摊成本必定大于等于 0.

另外, 协作配送成本分摊问题满足超可加性(定理 2 中情况 1 已证明). 那么现设  $S_{ki}$  为二叉树中仅包含 1 个成员的末端节点对应的子联盟. 根据二叉树结构可知, 该节点必定存在唯一的兄弟节点对应子联盟  $S_{k\bar{i}}$ , 且有  $S_{k\bar{i}} \cap S_{ki} = \emptyset, S_{k\bar{i}} \cup S_{ki} = F(S_{k-1}) = F(S_{k-1}) = S_{k-1, i_{k-1}} = S_{k-1, \bar{i}_{k-1}}$ . 由于问题满足超可加性, 则一定有不等式 (42) 成立.

$$C(S_{ki}) + C(S_{k\bar{i}}) \geq C(S_{k-1, i_{k-1}}) = C(F(S_{k-1})) = C(S_{k-1, i_{k-1}}) \quad (42)$$

结合不等式 (42) 有不等式 (43) 成立.

$$\alpha_{ki} = \frac{C(S_{k-1, i_{k-1}}) + C(S_{ki}) - C(S_{k\bar{i}})}{2C(S_{k-1, i_{k-1}})} \leq \frac{C(S_{ki})}{C(S_{k-1, i_{k-1}})} \quad (43)$$

因此, 不等式 (44) 必定成立.

$$\begin{aligned}
C'(S_{ki}) &= (\alpha_{2 i_2} \cdot \alpha_{3 i_3} \cdots \alpha_{k-1 i_{k-1}} \cdot \alpha_{ki}) \cdot C(N) \\
&\leq \frac{C(S_{2 i_2})}{C(N)} \cdot \frac{C(S_{3 i_3})}{C(S_{2 i_2})} \cdots \frac{C(S_{k-1 i_{k-1}})}{C(S_{k-2 i_{k-2}})} \cdot \\
&\quad \frac{C(S_{ki})}{C(S_{k-1 i_{k-1}})} \cdot C(N) \\
&= C(S_{ki})
\end{aligned} \tag{44}$$

故定理 3 第 1 部分成立.

2) 要证明定理 3 第 2 部分成立, 即证明等式 (45) 成立.

$$\sum_{S_{ki}: |S_{ki}|=1}^{2^{k-1}} C'(S_{ki}) = C(N) \tag{45}$$

由 B-T Shapley 原理可知,  $C'(N) = C(N)$ , 并且每个父节点对应子联盟的分摊成本, 按  $\alpha_{ki} + \alpha_{k\bar{i}} = 1$  的比率完全分配到对应 2 个子节点对应的子联盟. 所以二叉树中每一层(当  $N$  不等于  $2^l$ , 最后一层除外)节点对应子联盟分摊成本之和均等于  $C(N)$ . 即有等式(46)成立.

$$\sum_{i=1}^{2^{k-1}} C'(S_{ki}) = C(N) \quad (k=1, 2, \dots, L(N)-1) \tag{46}$$

根据二叉树结构和 B-T Shapley 分割联盟规则可知, 无论  $N$  取何值, 在分割后的二叉树倒数第二层  $k = \lceil \log_2 N \rceil = L(N) - 1$  的节点对应子联盟成员数量, 只有等于 2 和 1 两种可能. 且在  $k = \lceil \log_2 N \rceil$  层中的  $2^{\lceil \log_2 N \rceil - 1}$  个节点对应的子联盟中, 恰好有  $N - 2^{\lceil \log_2 N \rceil - 1}$  个的成员数量等于 2,  $2^{\lceil \log_2 N \rceil - 1}$  个成员数量等于 1. 那么在二叉树  $k = \lceil \log_2 N \rceil + 1 = L(N)$  最后一层中恰好有  $2N - 2^{\lceil \log_2 N \rceil}$  个节点, 且对应子联盟的成员个数均为 1. 所以, 所有单个企业分摊成本之和可用式 (47) 计算.

$$\begin{aligned}
\sum_{S_{ki}: |S_{ki}|=1} C'(S_{ki}) &= \sum_{S_{ki}: |S_{ki}|=1}^{k=L(N)} C'(S_{ki}) + \sum_{S_{ki}: |S_{ki}|=1}^{k=L(N)-1} C'(S_{ki}) \\
&= \sum_{S_{ki}: |S_{ki}|=2}^{k=L(N)-1} C'(S_{ki}) + \sum_{S_{ki}: |S_{ki}|=1}^{k=L(N)-1} C'(S_{ki}) \\
&= \sum_{i=1}^{2^{k-1}} C'(S_{ki}) \\
&= C(N)
\end{aligned} \tag{47}$$

因此, 定理 3 第 2 部分也成立, 所以 B-T Shapley 求解协作配送成本分摊问题的结果, 必定为转归解, 定理 3 成立. 证毕.

## 5 数据实验与结果分析

### 5.1 B-T Shapley 求解质量验证

由定理 2 可知, 当每个配送企业的顾客和配送中心均匀分布在面积为  $A$  的相同区域, 配送成本与路线长度成正比时, 本文研究的协作配送成本分摊问题将是一个凸博弈问题. 本部分为了验证 B-T Shapley 方法的准确性. 首先按照定理 2 的条件, 生成了规模为  $N = \{3, 4, 5, \dots, 20\}$  的 18 个属于凸博弈的协作配送成本分摊问题, 然后分别采用 Shapley 和 B-T Shapley 方法求解. 以验证 B-T Shapley 求解结果与 Shapley 的接近程度. 采用 MatlabR2012b 编程, 运行平台: CPU 为 Inter (R) core (TM) 2 Duo, 内存为 2.0G, 主频为 2.93GHz. 算例设计的具体规则为

1) 为了便于计算 MOCVRP 的期望最优解长度, 设  $\alpha\sqrt{A} = 1$ , 因此联盟特征函数值  $C(S)$  由下式计算;  $C(S) = \alpha\sqrt{A} \sum_{i \in S} n_i = \sqrt{\sum_{i \in S} n_i}$

2) 每个算例中, 协作配送企业的顾客数量如表 3 所示. 表 3 中的顾客数量为 100 ~ 300 之间的随机整数.

表 3 20 个协作企业的顾客数量

Table 3 The number of customer of 20 enterprises jointing cooperation distribution

协作企业	1	2	3	4	5	6	7	8	9	10
顾客数量	100	128	214	137	118	134	289	182	241	274
协作企业	11	12	13	14	15	16	17	18	19	20
顾客数量	167	139	244	249	219	145	114	135	285	123

B-T Shapley 方法的求解结果与大联盟  $N$  中的成员排列相关, 为了测试 B-T Shapley 求解质量的稳定性, 协作成员在联盟中的排序随机生成, 且 B-T Shapley 求解每个算例的次数为  $N^2$  ( $N \leq 10$ ) 次或  $N \times \lceil \log_2(N) \rceil$  ( $N > 10$ ) 次. 并将求解结果分别与 Shapley 方法的求解结果比较(用相关系数  $r$  表示接近程度), 分别计算 B-T Shapley 多次求解的  $Best-r$ ,  $Worst-r$  和  $Ave-r$ . 由于本规则生成的算例, 由定理 2 可知, 必定为凸博弈. 因此, Shapley 方法的解必定为核心解, 在此用  $\rho_{Core} = 100\% \times \text{核心解次数} / \text{求解次数}$ , 表示 B-T Shapley 求解的多个解为核心解的比率. 具体求解结果如表 4 和图 4 所示.

表 4 Shapley 和 B-T Shapley 算法求解结果比较

Table 4 Comparison results of Shapley and B-T Shapley on solving 18 instances

No.	$N$	SolveTimes	$\rho_{Core}\%$	Best- $\tau$	Ave- $\tau$	Worst- $\tau$
1	3	9	100	0.996 0	0.958 4	0.915 3
2	4	16	100	0.999 9	0.999 8	0.999 8
3	5	25	100	0.969 1	0.924 2	0.903 4
4	6	36	100	0.968 3	0.945 3	0.916 5
5	7	49	100	0.987 6	0.967 0	0.951 8
6	8	64	100	0.999 8	0.999 7	0.999 5
7	9	81	100	0.984 2	0.962 1	0.935 7
8	10	100	100	0.984 6	0.943 5	0.914 9
9	11	44	100	0.968 3	0.929 9	0.906 4
10	12	48	100	0.979 0	0.934 9	0.914 5
11	13	52	100	0.961 0	0.937 4	0.925 6
12	14	56	100	0.968 0	0.951 5	0.939 3
13	15	60	100	0.985 1	0.969 1	0.960 8
14	16	64	100	0.999 8	0.999 7	0.999 5
15	17	85	100	0.986 2	0.972 6	0.950 6
16	18	90	100	0.979 0	0.951 2	0.921 9
17	19	95	100	0.979 9	0.941 0	0.916 1
18	20	100	100	0.971 6	0.934 3	0.914 1
average			100	0.981 5	0.956 8	0.938 1

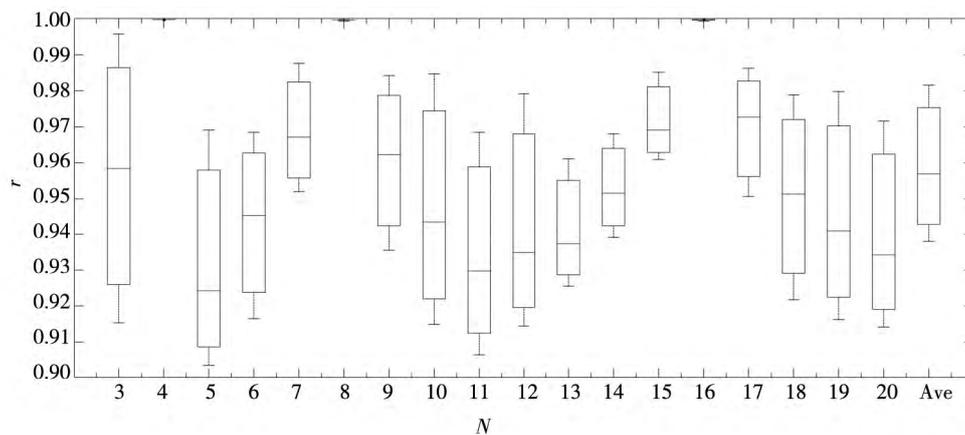


图 4 B-T Shapley 求解 18 个算例结果与 Shapley 结果的相关系数

Fig. 4 The correlation coefficient of cost sharing results of B-T Shapley and Shapley on solving 18 instances

由表 4 的求解结果可知, B-T Shapley 方法的求解结果与 Shapley 的结果高度相关, 18 个算例中最好结果的平均值达到 0.981 5, Ave- $\tau$  的平均值 0.956 8, 最差输出结果 Worst- $\tau$  的平均结果也高达 0.938 1. 特别是当  $N = 4, 8, 16$ , 时 B-T Shapley 能够精确的用二叉树将联盟分割, 因此求解的精确度非常高, 如表 4 和图 4 所示 B-T Shapley 求解这三个算例与 Shapley 结果相比最小的相关系数 Worst- $\tau$  也分别达到 0.999 8, 0.999 5 和 0.999 5.

综上, B-T Shapley 求解该类符合凸博弈特征的算例的结果高度接近 Shapley 方法. 另外值得注意的是, B-T Shapley 方法虽然在理论上不能保证求解凸博弈的结果一定为核心解, 但求解上述 18 个算例发现, 其求解结果均为核心解, 这说明该方法求解凸博弈的成本分摊问题, 结果为核心解的概率非常高.

### 5.2 B-T Shapley 求解速度验证

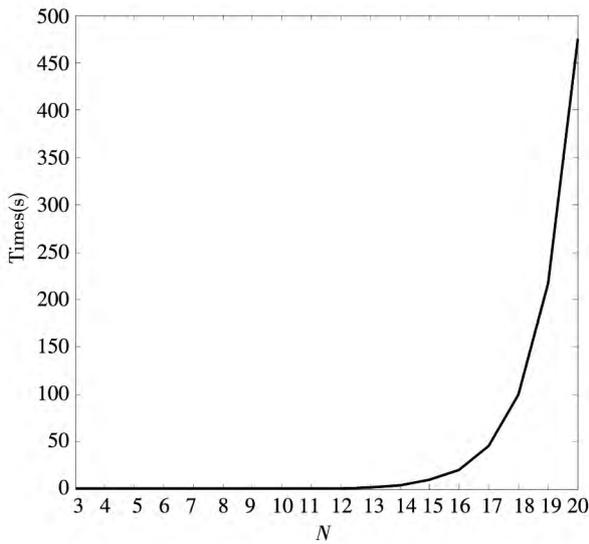
由论文 4.3 和 4.4 部分可知, Shapley 的计算

复杂度达到  $N^2 2^N$  指数级, 而 B-T Shapley 仅为  $N^2 \ln N$  多项式级. 为验证上述理论的合理性, 本部分通过实例求解来比较 Shapley 和 B-T Shapley

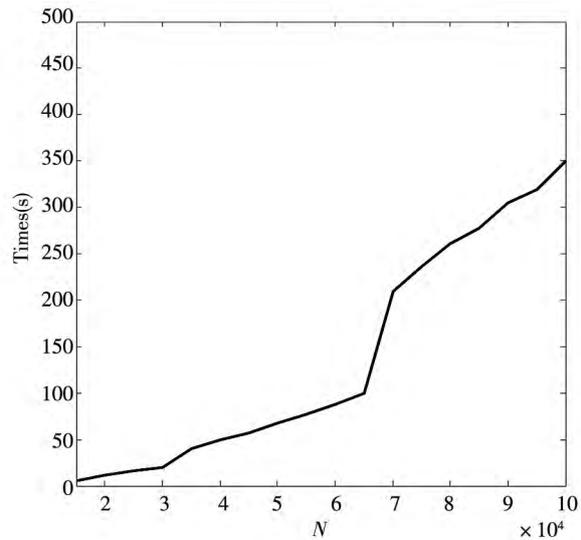
的耗时. 生成算例的规则如前文所述, 本部分侧重计算耗时, 因此假设协作的第  $i (1 \leq i \leq N)$  个企业的顾客数量为  $i$ . 求解耗时结果如表 5 和图 5 所示.

表 5 Shapley 和 B-T Shapley 算法求解耗时比较  
Table 5 The running time of Shapley and B-T Shapley

No.	Shapley				B-T Shapley			
	$N$	Times( s)	$K_{time}$	$K_{OS}$	$N$	Times( s)	$K_{time}$	$K_{OBTS}$
1	3	0.001 2	-	-	15 000	5.913 1	-	-
2	4	0.001 9	1.500 0	3.555 6	20 000	11.888 2	2.010 5	1.831 0
3	5	0.002 2	1.166 7	3.125 0	25 000	15.724 9	1.322 7	1.597 7
4	6	0.005 0	2.285 7	2.880 0	30 000	19.999 3	1.271 8	1.465 9
5	7	0.014 7	2.937 5	2.722 2	35 000	39.873 9	1.993 8	1.381 5
6	8	0.028 7	1.957 4	2.612 2	40 000	49.358 7	1.237 9	1.322 8
7	9	0.068 3	2.380 4	2.531 3	45 000	57.143 2	1.1577	1.279 7
8	10	0.156 0	2.283 1	2.469 1	50 000	67.501 6	1.181 3	1.246 7
9	11	0.374 4	2.400 0	2.420 0	55 000	77.392 1	1.146 5	1.220 7
10	12	0.811 2	2.166 7	2.380 2	60 000	88.187 4	1.1395	1.1996
11	13	1.856 4	2.288 5	2.347 2	65 000	99.840 6	1.132 1	1.182 1
12	14	4.165 2	2.243 7	2.319 5	70 000	209.727 7	2.100 6	1.167 5
13	15	9.422 5	2.262 2	2.295 9	75 000	235.826 7	1.124 4	1.155 1
14	16	20.576 5	2.183 8	2.275 6	80 000	260.833 7	1.106 0	1.144 3
15	17	45.396 3	2.206 2	2.257 8	85 000	277.494 6	1.063 9	1.135 0
16	18	99.216 6	2.185 6	2.242 2	90 000	304.779 2	1.098 3	1.126 8
17	19	216.841 4	2.185 5	2.228 4	95 000	319.318 4	1.047 7	1.119 5
18	20	475.491 0	2.192 8	2.216 1	100 000	350.269 0	1.096 9	1.113 0



(a)



(b)

图 5 Shapley (a) 和 B-T Shapley (b) 算法求解耗时

Fig. 5 Comparison of Shapley (a) and B-T Shapley (b) in the running time

其中表 5 中的  $K_{time}$  等于表中本行对应规模算例 Shapley 或 B-T Shapley 耗时除以上一行对应规

模算例耗时, 如表 5 左边的  $N = 15$  时的  $K_{time} = 9.422 5 / 4.165 2 = 2.206 2$ .  $K_{OS}$  等于 Shapley 求解

本行对应规模算例的理论计算复杂度除以上一行相应指标,  $K_{QBTS}$  等于 B-T Shapley 求解本行对应规模算例的理论计算复杂度除以上一行相应指标. 如表 5 中左边  $N = 15$  时的  $K_{QS} = 15^2 \times 2^{15} / (14^2 \times 2^{14}) = 2.2578$ , 表 5 中右边  $N = 75000$  时的  $K_{QBTS} = 75000^2 \times \log_2(75000) / (70000^2 \times \log_2(70000)) = 1.1551$ .

如表 5 所示, B-T Shapley 的求解速度非常快, Shapley 求解规模为 20 的合作博弈需要 475.491 0s, 而 B-T Shapley 算法求解规模为 100 000 的合作博弈问题仅要 350.269 0s. 且如图 5 所示, Shapley 的求解耗时随  $N$  的增加迅速升高, 而 B-T Shapley 的耗时随  $N$  的增加平缓增加. 另外, 由表 5 可知, 每个算例的  $K_{time}$  与  $K_{QS}$  或  $K_{QBTS}$  均很接近, 其中  $N = 70000$  的  $K_{time}$  与  $K_{QBTS}$  具有较大差别, 这与运行电脑内存达到一定比率后速度变慢具有一定关系, 以及当  $N = 3, 4, 5$  时计算量较小, 启动程序耗时占较大比率, 从而导致  $K_{time}$  与  $K_{QS}$  有较大差别. 根据实验数据可知论文第 4 部分的 Shapley 与 B-T Shapley 的复杂度分析较为准确.

因此, 根据 Shapley 复杂度和表 5 中的数据可推算当  $N = 30$  时, 在本实验平台上的耗时将达到  $475.4910s \times 30^2 \times 2^{30} / (20^2 \times 2^{20}) = 1095531.2640s = 12.6798d$ . 值得注意的是, 表 5 中是假设所有子联盟的特征函数已知情况下 Shapley 和 B-T Shapley 的求解耗时, 在求解本文研究的协作配送成本分摊问题时, 每一个子联盟的特征函数值等价于求解一个 MOCVRP 问题. 按照 Shapley 求解思路需要求解所有  $2^N - 1$  子联盟的特征函数(推算耗时参照表 1), 所以采用 Shapley 求解本文的成

本分摊问题,  $N$  较小时其对应计算量就会非常巨大.

### 5.3 MOCVRP 算例求解

由于当前尚不存在 MOCVRP 标准算例, 基于上海和青岛两地调研发现, 城市物流公司数量可达上百家, 并且配送区域高度重叠甚至几乎一致. 当前由于缺少用于协作配送的信息共享平台, 目前几乎不存在协作配送. 在此, 为了验证 B-T Shapley 求解 MOCVRP 的性能, 现基于部分企业历史配送数据, 设计 MOCVRP 算例和求解相关说明如下:

1) 企业数  $N \in \{4, 8, 10, 12, 32\}$ , 且每个企业的顾客数均为  $n_i = 20$ ; 2) 顾客的需求量均为  $q = 100$  kg, 车辆载量为  $Q = 1000$ ; 3) 所有顾客的位置假设随机分布在  $10 \text{ km} \times 10 \text{ km}$  的区域  $A$  中; 4) 每个企业有一个配送中心, 配送中心位置为将区域  $A$  均等分为  $N$  个小区域的中心位置, 分割方式  $N = 4: 2 \times 2, N = 8: 2 \times 4, N = 10: 2 \times 5, N = 12: 3 \times 4, N = 32: 4 \times 8$ ; 5) 配送车辆平均每千米的综合配送成本为  $p = 10$  元; 6) 求解 MOCVRP 算法的方法为聚类 Cluster + 节约 Saving + 变邻域搜索 Variable Neighborhood Search 三阶段算法(简称 CSV 算法).

分别采用 Shapley 与 B-T Shapley( Binary Tree 分解联盟时成员均按序号升序排列) 求解上述设计的算例, 求解结果如表 6 所示, 其中表中  $T_s$  表示成本分摊的耗时,  $T_c$  表示采用 CSV 算法求解相应 MOCVRP 所需耗时. 需要注意的是表 6 中  $T_s$  耗时单位为 s,  $T_c$  的单位为 h,  $r$  为 B-T Shapley 与 Shapley 成本分摊结果的相关系数.

表 6 Shapley 和 B-T Shapley 算法求解 5 个 MOCVRP 算例耗时与相关系数结果比较

Table 6 Comparison results of running time and correlation coefficient of Shapley and B-T Shapley on 5 instances

No.	N	Shapley		B-T Shapley		r
		$T_s$ (s)	$T_c$ (h)	$T_s$ (s)	$T_c$ (h)	
1	4	0.0019	0.1304	0.0000	0.0486	0.9816
2	8	0.0287	7.4909	0.0000	0.2017	0.9851
3	10	0.1560	49.7219	0.0000	0.3398	0.9537
4	12	0.8112	267.8792	0.0000	0.8427	0.9603
5	32	—	—	0.0000	3.3409	—

由于 Shapley 方法要求输入所有子联盟的特征函数值, 即要求解决  $2^N - 1$  个相应的 MOCVRP 问题, 因此耗时量非常高. 由上述求解结果可以

发现, 当协作企业数量达到  $N = 8$  时, 用于求解 MOCVRP 问题的耗时就需要 49.7219 h(约 2 d), 而当  $N = 10$  时相应的 MOCVRP 求解就需要

267.879 2 h(11.1 d). 本文提出的 B-T Shapley 仅需要解决  $2N - 1$  个相应的 MOCVRP 问题, 当  $N = 10, 12$  时的耗时在 1 h 之内, 甚至当参与协作的企业数达  $N = 32$  时, 求解相应 MOCVRP 问题的耗时也只需 3.340 9 h.

然而, 借助现有技术平台, 通过数据共享方式能够便捷的形成功能协作配送, 其潜在的协作成员数可以达到几十以上. 因此, Shapley 成本分摊方法所需要的计算量难以有效满足现实需求. B-T Shapley 相对于 Shapley 更加快速有效, 并且求解质量与 Shapley 仅有较小偏差. 特别是对于  $N = 2^t$  ( $t$  为任意大于等于 2 的整数), 由于二叉树能够均衡分解, 此时 B-T Shapley 与 Shapley 的求解结果更加接近, 如表 6 中  $N = 4$  和 8, B-T Shapley 与 Shapley 的求解结果相关系数分别 0.981 6 和 0.

985 1.

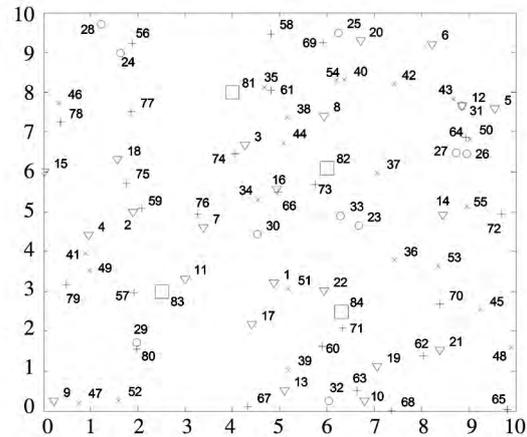
### 5.4 实例求解与相关分析

本部分完全基于 4 个实际物流配送企业的同一天历史配送任务数据, 研究了分别采取单独运营与协作运营的模式时, 配送方案中车辆行驶距离和派送车辆数的变化.

案例背景: 在青岛市区  $100 \text{ km}^2$  城市区域中选择了 4 个专门为便利店配送牛奶的企业, 4 个企业的配送中心和相应顾客位置信息如图 6, 更加具体的数据信息如表 7 所示(车辆载量 200 箱, 需求量  $q_i$  单位: 箱; 81 ~ 84 为配送中心). 其中图 6(a) 为实例配送区域的位置地图(边长为 10km 的方框), 图 6(b) 为以方框左下角为原点建立坐标系的 80 个顾客和 4 个配送中心坐标图(b).



(a)



(b)

图 6  $N = 4$  时的 MOCVRP 实例分布区域地图(a) 和顾客配送中心的坐标图(b)

Fig. 6 The spatial distribution (a) and coordinates (b) of depots and customers of a MOCVRP case with  $N = 4$

如表 7 所示, 本实例包括的 4 个企业具有几乎相同的顾客市场, 在当天的配送任务中企业 1 ~ 企业 4 分别有 22 个、11 个、22 个和 25 个顾客, 且顾客总需求量分别为 409、206、412 和 563 箱牛奶, 企业 1 ~ 企业 4 对应的配送中心为节点 81 ~ 84. 由图 6 所示, 企业 1 ~ 企业 4 的顾客 (“▽”表示企业 1 顾客, “○”表示企业 2 顾客, “×”表示企业 3 顾客, “+”表示企业 4 顾客, “□”表示配送中心) 较为均匀的分布在边长为 10km 的区域内.

由于 4 个企业的配送区域几乎相同, 通过协作能够大幅度的减少不同配送企业车辆路线交叉和重叠情况. 如图 7(a) 所示, 在没有协作之前企

业 1 ~ 企业 4 需要分别派送 3 辆、3 辆、2 辆、3 辆车, 如图 7(b) 所示, 通过协作后每个企业只需派送 2 辆车, 总行驶距离仅为 127.89 km, 距离节约了 44.12%.

在表 8 中, 列出了 4 个企业中任意 1 个及以上成员构成子联盟, 并计算了相应子联盟协作和不协作配送的距离、派送车辆数和装载率. 表 8 列标题符号解释如下:  $Len_c$  表示协作配送距离(单位: km, 包括货物调运路线距离);  $Len_{nc}$  表示不协作配送距离(单位: km);  $Len_s\% = 100\% \times (Len_{nc} - Len_c) / Len_{nc}$  表示通过协作节约的配送距离百分比;  $K_c$  表示协作需要的配送车辆;  $K_{nc}$  表示不协作需要的配送车辆;  $K_s\% = 100\% \times (K_{nc} -$

表 7  $N = 4$  的 MOCVRP 实例顾客和配送中心的位置及需求信息

Table 7 The coordinates and demand of customers and depots of MOCVRP instance with  $N = 4$

No.	$x_i$	$y_i$	$q_i$	所属企业	No.	$x_i$	$y_i$	$q_i$	所属企业	No.	$x_i$	$y_i$	$q_i$	所属企业
1	4.88	3.21	24	1	29	1.97	1.73	31	2	57	1.91	2.96	24	4
2	1.89	4.98	10	1	30	4.52	4.43	10	2	58	4.81	9.47	27	4
3	4.27	6.67	14	1	31	8.86	7.65	15	2	59	2.07	5.09	19	4
4	0.95	4.41	17	1	32	6.04	0.25	21	2	60	5.89	1.63	13	4
5	9.56	7.58	17	1	33	6.29	4.90	28	2	61	4.82	8.05	22	4
6	8.24	9.20	21	1	34	4.54	5.31	29	3	62	8.04	1.38	13	4
7	3.38	4.60	14	1	35	4.67	8.11	16	3	63	6.63	0.51	11	4
8	5.95	7.40	20	1	36	7.44	3.79	19	3	64	8.94	6.86	36	4
9	0.22	0.25	29	1	37	7.07	5.97	17	3	65	9.83	0.04	27	4
10	6.80	0.25	18	1	38	5.16	7.37	23	3	66	4.96	5.50	26	4
11	2.99	3.31	18	1	39	5.18	1.03	19	3	67	4.32	0.11	27	4
12	8.86	7.66	15	1	40	6.37	8.30	10	3	68	7.38	0.01	14	4
13	5.10	0.51	31	1	41	0.88	3.95	24	3	69	5.93	9.25	27	4
14	8.46	4.92	11	1	42	7.43	8.22	19	3	70	8.39	2.67	26	4
15	0.01	6.00	10	1	43	8.69	7.83	23	3	71	6.33	2.08	28	4
16	4.93	5.56	19	1	44	5.08	6.73	12	3	72	9.70	4.94	28	4
17	4.41	2.17	29	1	45	9.26	2.53	22	3	73	5.75	5.68	20	4
18	1.57	6.31	25	1	46	0.32	7.73	29	3	74	4.05	6.45	11	4
19	7.07	1.11	12	1	47	0.75	0.21	12	3	75	1.75	5.72	16	4
20	6.71	9.29	12	1	48	9.91	1.60	12	3	76	3.27	4.93	16	4
21	8.40	1.53	17	1	49	0.97	3.52	23	3	77	1.85	7.51	34	4
22	5.95	3.01	26	1	50	9.03	6.82	20	3	78	0.35	7.24	24	4
23	6.67	4.65	25	2	51	5.18	3.06	25	3	79	0.49	3.15	25	4
24	1.63	8.97	10	2	52	1.58	0.26	15	3	80	1.98	1.54	29	4
25	6.24	9.49	25	2	53	8.36	3.63	13	3	81	4.00	8.00	0	1
26	8.97	6.46	10	2	54	6.21	8.30	11	3	82	6.00	6.10	0	2
27	8.75	6.47	20	2	55	8.97	5.13	19	3	83	2.50	3.00	0	3
28	1.22	9.71	11	2	56	1.87	9.21	20	4	84	6.30	2.50	0	4

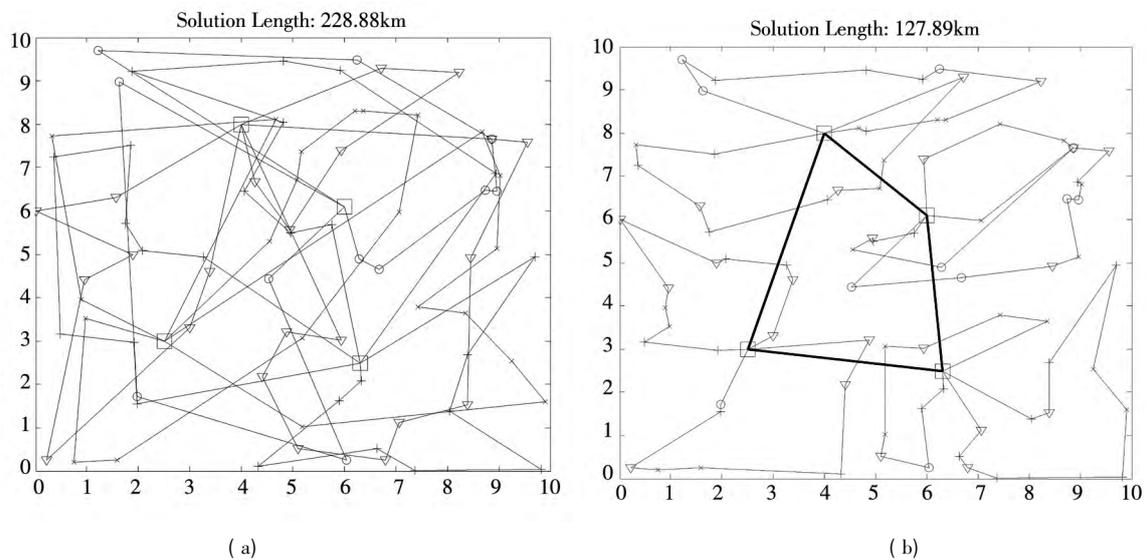


图 7  $N = 4$  时的 MOCVRP 算例协作配送前 (a) 后 (b) 车辆路线对比图  
Fig. 7 The vehicle tour of 4 enterprises before cooperation distribution (a) and after (b)

$K_c) / K_{nc}$ 表示通过协作减少的配送车辆百分比;  
 $LR_c\%$ 表示协作配送车辆平均装载率;  $LR_{nc}\%$ 表示  
 不协作时车辆平均装载率.  $N = 4$ 个成员合作的大联盟共有  $2^N - 1 = 2^4 - 1 = 15$ 个非空子联盟, 其中11个为2个及以上成员可以形成实际合作, 如表8所示所有11个包含2个及以上成员的子联盟通过协作后, 配送距离可以减少18.45% ~

44.12%, 配送车辆数量可以减少16.67% ~ 27.27%, 平均装载率具有显著的提高, 并且当子联盟成员越多时效益越显著. 需要指出的是, 在本实例中由于配送的牛奶属于同质物品, 如果能够建立协作配送平台, 对于该类物品协作配送不需要调配货物, 因此, 通过协作能够节约更多的配送距离和成本.

表8 15个子联盟通过协作或不协作时的配送距离车辆数和装载率

Table 8 The distance, number of vehicle, loading rate of 15 sub-coalitions with or without cooperation

$j$	$B^j$	$S^j$	$Len_c$	$Len_{nc}$	$Len_s\%$	$K_c$	$K_{nc}$	$K_s\%$	$LR_c\%$	$LR_{nc}\%$
1	[0, 0, 0, 1]	{4}	63.13	63.13	0.00	3	3	0.00	93.83	93.83
2	[0, 0, 1, 0]	{3}	63.79	63.79	0.00	3	3	0.00	68.67	68.67
3	[0, 0, 1, 1]	{3, 4}	90.51	126.92	28.68	5	6	16.67	97.50	81.25
4	[0, 1, 0, 0]	{2}	43.57	43.57	0.00	2	2	0.00	51.50	51.50
5	[0, 1, 0, 1]	{2, 4}	78.10	106.70	26.80	4	5	20.00	96.13	76.90
6	[0, 1, 1, 0]	{2, 3}	77.43	107.36	27.87	4	5	20.00	77.25	61.80
7	[0, 1, 1, 1]	{2, 3, 4}	102.09	170.49	40.12	6	8	25.00	98.42	73.81
8	[1, 0, 0, 0]	{1}	58.39	58.39	0.00	3	3	0.00	68.17	68.17
9	[1, 0, 0, 1]	{1, 4}	99.10	121.52	18.45	5	6	16.67	97.20	81.00
10	[1, 0, 1, 0]	{1, 3}	94.87	122.17	22.35	5	6	16.67	82.10	68.42
11	[1, 0, 1, 1]	{1, 3, 4}	110.26	185.31	40.50	7	9	22.22	98.86	76.89
12	[1, 1, 0, 0]	{1, 2}	80.82	101.96	20.74	4	5	20.00	76.88	61.50
13	[1, 1, 0, 1]	{1, 2, 4}	110.08	165.09	33.32	6	8	25.00	98.17	73.63
14	[1, 1, 1, 0]	{1, 2, 3}	104.51	165.75	36.95	6	8	25.00	85.58	64.19
15	[1, 1, 1, 1]	{1, 2, 3, 4}	127.89	228.88	44.12	8	11	27.27	99.38	72.27

表9  $N = 4$ 时实例采用4种不同分摊方法的结果比较

Table 9 Comparison results of cost sharing of Shapley, B-T Shapley,  $\tau$  value and Nucleolus

成本分摊方法	指标	$X(1)$	$X(2)$	$X(3)$	$X(4)$	$r$ with Shapley
Shapley	分摊值(元)	363.07	222.37	334.47	359.00	1.000 0
	节约成本%	37.82	48.96	47.56	43.14	
B-T Shapley	分摊值(元)	349.69	241.31	346.46	341.44	0.979 6
	节约成本%	40.11	44.62	45.68	45.92	
$\tau$ 值法	分摊值(元)	371.33	271.97	291.43	344.17	0.814 5
	节约成本%	36.40	37.58	54.31	45.48	
Nucleolus	分摊值(元)	366.18	284.47	286.28	341.97	0.718 4
	节约成本%	37.29	34.71	55.12	45.83	
单独配送成本(元)		583.89	435.71	637.85	631.32	—

通过对应求解 15 个 MOCVRP 问题后, 得到相应子联盟配送成本(假设每公里配送成本  $p = 10$  元). 基于此, 分别采用 Shapley, B-T Shapley,  $\tau$  值法和 Nucleolus 核仁法对该问题进行了成本分摊. 其中经典方法 Shapley、 $\tau$  值法和 Nucleolus 核仁法均需要用到所有的 15 个子联盟协作配送成本, 即表 8 中的  $Len_c \times p = Len_c \times p$ ; 此处 B-T Shapley 方法 Binary Tree 分割联盟的顺序为成员序号升序, 因此只需要求解  $2N - 1 = 7$  个 MOCVRP 问题, 即表 8 中  $j = 1, 2, 3, 4, 8, 12, 15$  时对应的子联盟  $\{4\}, \{3\}, \{3, 4\}, \{2\}, \{1\}, \{1, 2\}$  和  $\{1, 2, 3, 4\}$ , 而无需求解其他子联盟对应的 MOCVRP 问题. 通过分摊结果可以发现, 与之前的算例求解数据验证结果类似, B-T Shapley 成本分摊结果高度接近 Shapley 的结果, 相关系数达 0.979 6, 而其他两个经典成本分摊方法  $\tau$  值法和 Nucleolus 核仁法的求解结果与 Shapley 的相关系数仅为 0.814 5 和 0.718 4. 另外, 尽管不同的成本分摊方法, 结果具有一定差异, 但通过表 9 可知, 本实例通过协作配送每个企业的配送成本节约量可达 36.40% ~ 55.12%. 并且在成本分摊中, 不同方法均充分考虑了参与企业对协作的贡献, 如企业 3 的单独配送成本最高, 但不同方法下对企业 3 的成本分摊均不是最高, 这说明企业 3 在协作中的贡献较大, 其顾客分布与其他三个企业具有较高的互补性.

因此, 如果能够建立第三方物流平台, 根据潜在协作成员的配送中心、顾客位置和需求信息, 事先通过高效的成本分摊方法如本文的 B-T Shapley 方法计算出通过协作各企业能够节约的成本, 然后将该信息反馈至潜在协作成员, 将有利于促成企业间的实际合作.

## 6 结束语

本文基于多车场车辆路径问题模型, 构建了

能够同时运用于协作配送车辆路径优化和量子联盟协作配送成本的 MOCVRP 模型. 通过研究发现当配送企业的顾客和配送中心均匀分布在同一区域时, 协作配送的成本分摊问题为凸合作博弈问题. 通过分析经典成本分摊方法 Shapley 发现, 采用该方法求解协作配送的成本分摊问题具有非常高的复杂度, 求解过程中需要对应解决  $2^N - 1$  个 MOCVRP 问题. 鉴于此, 本文基于 Shapley 方法, 提出了一个 Binary-Tree Shapley (简称 B-T Shapley) 方法, 通过理论分析和实例求解发现该方法仅需要求解  $2N - 1$  个 MOCVRP 问题, 可以在合理时间内求解规模达几十个合作企业的协作配送成本分摊问题. 更重要的是, B-T Shapley 方法求解结果高度接近 Shapley 的求解结果, 对于接近或为凸博弈的成本分摊问题, 结果相关系数可达 0.95 以上. 在求解一个包含 4 个协作企业的实例发现, 通过协作能够有效的减少派送车辆, 缩短配送距离, 参与协作的配送企业可以减少 36.40% ~ 55.12% 的配送成本.

本文提出的 B-T Shapley 方法能够有效计算规模较大的协作配送成本分摊问题, 从而第三方信息共享平台可以根据潜在协作企业的相关数据快速计算出每个企业能够节约的成本, 并将其反馈至协作企业, 以促成企业间的实际合作. 与此同时, B-T Shapley 方法也存在有待进一步研究的问题, 如 B-T Shapley 虽然在实例求解中均获得了核心解, 但是目前只是理论证明了 B-T Shapley 的结果必定为转归解, 对于凸博弈问题, B-T Shapley 方法是否能保证得到核心解, 需要在以后的研究中进一步确定. 另外, 本文将协作配送中需要的货物调运成本近似为一个以配送中心为节点的旅行商问题, 然而当参与协作配送的企业较多时, 货物调运路线选择本身也是一个较为复杂的优化问题, 即 MOCVRP 问题可以看成 MD-VRP 和货物调运联合优化问题, 这要求求解算法更加高效, 是一个值得深入研究的方向.

## 参考文献:

- [1] Salhi S, Sari M. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem [J]. European Journal of Operational Research, 1997, 103(1): 95 - 112.
- [2] Irnich S. A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles [J]. European Journal of

- Operational Research ,2000 ,122( 2) : 310 – 328.
- [3]Wu T H , Low C , Bai J W. Heuristic solutions to multi-depot location-routing problems [J]. Computers & Operations Research ,2002 ,29( PII S0305 – 0548 ( 01) 00038 – 710) : 1393 – 1415.
- [4]Dondo R , Mendez C A , Cerda J. An optimal approach to the multiple-depot heterogeneous vehicle routing problem with time window and capacity constraints [J]. Latin American Applied Research ,2003 ,33( 2) : 129 – 134.
- [5]Dondo R G , Cerda J. A hybrid local improvement algorithm for large-scale multi-depot vehicle routing Problems with Time Windows [J]. Computers & Chemical Engineering ,2009 ,33( 2) : 513 – 530.
- [6]Dondo R , Cerda J. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows [J]. European Journal of Operational Research ,2007 ,176( 3) : 1478 – 1507.
- [7]Baldacci R , Mingozzi A. A unified exact method for solving different classes of vehicle routing problems [J]. Mathematical Programming ,2009 ,120( 2) : 347 – 380.
- [8]Bettinelli A , Ceselli A , Righini G. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows [J]. Transportation Research Part C-Emerging Technologies ,2011 ,19( 5SI) : 723 – 740.
- [9]Cornillier F , Boctor F , Renaud J. Heuristics for the multi-depot petrol station replenishment problem with time windows [J]. European Journal of Operational Research ,2012 ,220( 2) : 361 – 369.
- [10]Salhi S , Imran A , Wassan N A. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation [J]. Computers & Operations Research ,2014 ,52 , Part B: 315 – 325.
- [11]Mancini S. A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: formulation and adaptive large neighborhood search based matheuristic [J]. Transportation Research Part C-Emerging Technologies ,2016 ,70( SI) : 100 – 112.
- [12]钟石泉 , 贺国光. 多车场有时间窗的多车型车辆调度及其禁忌算法研究 [J]. 运筹学学报 ,2005 ,9( 4) : 67 – 73.  
Zhong Shiquan , He Guoguang. Study on multi-depot vehicle routing scheduling problem with time windows and multi-type vehicle limits and its Tabu search algorithm [J]. OR Transactions ,2005 ,9( 4) : 67 – 73. ( in Chinese)
- [13]王晓博 , 李一军. 多车场多车型装卸混合车辆路径问题研究 [J]. 控制与决策 ,2009 ,24( 12) : 1769 – 1774.  
Wang Xiaobo , Li Yijun. Study on multi-depot and multi-type vehicles vehicle routing problem with backhauls [J]. Control and Decision ,2009 ,24( 12) : 1769 – 1774. ( in Chinese)
- [14]马建华 , 房 勇 , 袁 杰. 多车场多车型最快完成车辆路径问题的变异蚁群算法 [J]. 系统工程理论与实践. 2011 , 31( 8) : 1508 – 1516.  
Ma Jianhua , Fang Yong , Yuan Jie. Mutation and colony algorithm for multi-depot multi-type vehicle routing problem with shortest finish time [J]. Systems Engineering: Theory & Practice ,2011 ,31( 8) : 1508 – 1516. ( in Chinese)
- [15]马宇红 , 姚婷婷 , 张芳芳. 多车场多车型车辆调度问题及其遗传算法 [J]. 数学的实践与认识. 2014 ,44( 2) : 107 – 114.  
Ma Yuhong , Yao Tingting , Zhang Fangfang. multi-depot multi-type vehicle routing scheduling problem and its genetic algorithm [J]. Mathematics in Practice and Theory ,2014 ,44( 2) : 107 – 114. ( in Chinese)
- [16] Von Neumann J , Morgenstern O. Theory of Games and Economic Behavior [M]. Princeton: Princeton University Press ,1944.
- [17]Shapley L S. A value for n-person games [J]. Contributions to the Theory of Games ( AM – 28) ,1953 ,28: 307 – 317.
- [18]Schmeidler D. The nucleolus of a characteristic function game [J]. SIAM Journal on Applied Mathematics ,1969 ,17( 6) : 1163 – 1170.
- [19]Tijs S H. Bounds for the Core and the Tau-Value [M]. Moeschlin O , Pallaschke D ,1981: 123 – 132.
- [20]Guajardo M , Ronnqvist M. A Review On cost allocation methods in collaborative transportation [J]. International Transactions in Operational Research ,2016 ,23( 3) : 371 – 392.
- [21]Gothelundgren M , Jornsten K , Varbrand P. On the nucleolus of the basic vehicle routing game [J]. Mathematical Program-

- ming, 1996, 72(1): 83 – 100.
- [22] Engevall S, Gothe-Lundgren M, Varbrand P. The heterogeneous vehicle-routing game [J]. *Transportation Science*, 2004, 38(1): 71 – 85.
- [23] Krajewska M A, Kopfer H, Laporte G, et al. Horizontal cooperation among freight carriers: Request allocation and profit sharing [J]. *Journal of the Operational Research Society*, 2008, 59(11): 1483 – 1491.
- [24] 陈雯, 张强. 模糊合作对策的 Shapley 值 [J]. *管理科学学报*, 2006, 9(5): 50 – 55.  
Chen Wen, Zhang Qiang. Shapley value for fuzzy cooperative games [J]. *Journal of Management Sciences in China*, 2006, 9(5): 50 – 55. (in Chinese)
- [25] 谭春桥, 陈晓红. 基于 Choquet 延拓  $n$  人模糊对策的 Shapley 值 [J]. *管理科学学报*, 2010, 13(2): 24 – 32.  
Tan Chunqiao, Chen Xiaohong. Shapley value for  $n$ -persons fuzzy games based on Choquet extension [J]. *Journal of Management Sciences in China*, 2010, 13(2): 24 – 32. (in Chinese)
- [26] Peng L, Yaohua W, Na X. Allocating collaborative profit in less-than-truckload carrier alliance [J]. *Journal of Service Science and Management*, 2010, 3(1): 143 – 149.
- [27] Cruijssen F, Borm P, Fleuren H, et al. Supplier-Initiated Outsourcing: A Methodology to Exploit Synergy in Transportation [J]. *European Journal of Operational Research*, 2010, 207(2): 763 – 774.
- [28] Caprara A, Letchford A N. New techniques for cost sharing in combinatorial optimization games [J]. *Mathematical Programming*, 2010, 124(1–2): 93 – 118.
- [29] Dahl S, Derigs U. Cooperative planning in express carrier networks: An empirical study on the effectiveness of a real-time decision support system [J]. *Decision Support Systems*, 2011, 51(3): 620 – 626.
- [30] Zakharov V V, Shchegryaev A N. Stable cooperation in dynamic vehicle routing problems [J]. *Automation and Remote Control*, 2015, 76(5): 935 – 943.
- [31] Gansterer M, Hartl R. Collaborative vehicle routing: A survey [J]. *European Journal of Operational Research*, 2018, 268: 1 – 12.
- [32] Wang Y, Ma X, Li Z, et al. Profit distribution in collaborative multiple centers vehicle routing problem [J]. *Journal of Cleaner Production*, 2017, 144: 203 – 219.
- [33] Xiu X, Geroge H, Meng C. Truthful, budget-balanced bundle double auctions for Carrier Collaboration [J]. *Transportation Science*, 2017, 51(4): 1365 – 1386.
- [34] Cruijssen F, Cools M, Dullaert W. Horizontal cooperation in logistics: Opportunities and impediments [J]. *Transportation Research Part E: Logistics and Transportation Review*, 2007, 43(2): 129 – 142.
- [35] Liu R, Jiang Z, Liu X, et al. Task selection and routing problems in collaborative truckload transportation [J]. *Transportation Research Part E: Logistics and Transportation Review*, 2010, 46(6): 1071 – 1085.
- [36] Nadarajah S, Bookbinder J H. Less-Than-Truckload carrier collaboration problem: Modeling framework and solution approach [J]. *Journal of Heuristics*, 2013, 19(6): 917 – 942.
- [37] Pérez-Bernabeu E, Juan A A, Faulin J, et al. Horizontal cooperation in road transportation: A case illustrating savings in distances and greenhouse gas emissions [J]. *International Transactions in Operational Research*, 2015, 22(3): 585 – 606.
- [38] Beardwood J, Hammersley J M. The shortest path through many points [J]. *Proceedings of the Cambridge Philosophical Society*, 1959, 55: 299 – 327.
- [39] Helsgaun K. An effective implementation of the Lin-Kernighan traveling salesman heuristic [J]. *European Journal of Operational Research*, 2000, 126: 106 – 130.
- [40] 胡勋锋, 李登峰, 刘家财, 等. 基于分量差的线性及匿名合作对策值的简化算法 [J]. *管理科学学报*, 2017, 20(6): 32 – 41.  
Hu Xunfeng, Li Dengfeng, Liu Jiakai, et al. A component differences based simplified algorithm for linear and anonymous values of cooperative games [J]. *Journal of Management Sciences in China*, 2017, 20(6): 32 – 41. (in Chinese)

## A Binary Tree Shapely method for cost sharing of the collaborative vehicle routing problem

RAO Wei-zhen<sup>1,2</sup>, ZHU Qing-hua<sup>1\*</sup>, JIN Chun<sup>3</sup>, LIU Cong-hu<sup>1</sup>

1. Sino-US Global Logistics Institute, Shanghai Jiao Tong University, Shanghai 200030, China;
2. College of Economics and Management, Shandong University of Science and Technology, Qingdao 266590, China;
3. Faculty of Management and Economics, Dalian University of Technology, Dalian 116024, China

**Abstract:** Cooperation in logistics distribution among enterprises can significantly reduce the cost and emission of logistics delivery. Key research questions are how to model cooperation in logistics distribution and develop a cost sharing method. Traditional cost sharing methods need to calculate the cooperative cost for all sub-coalitions, which is equivalent to solving  $2^N - 1$  complex vehicle routing problems for calculating the cooperative cost of  $2^N - 1$  sub-coalitions ( $N$  is the number of enterprises). This paper analyzes the properties of the cost sharing problem in collaborative distribution by modeling a multi-cooperation vehicle routing. Based on the classical Shapley cost sharing method, a Binary Tree (B-T) Shapley method is proposed. The new method can not only reduce the computational complexity of cost sharing from  $O(N^2 2^N)$  to  $O(N^2 \log_2 N)$ , but also can lower the number of vehicle routing problems to be solved from  $2^N - 1$  to  $2N - 1$ . As a result, the cost allocation of collaborative delivery can be completed within a reasonable time. Calculation results of numerical examples and actual cases show that the time needed for the B-T Shapley method is almost negligible compared with that for the Shapley value method. Moreover, the cost allocation results using the B-T Shapley method only have minor deviations compared with those using the Shapley value method, with an average accuracy of about 95%.

**Key words:** collaborative vehicle routing problem; Shapley; cost sharing; cooperative games