Computers and Operations Research 000 (2017) 1–13

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

An exact decomposition method to save trips in cooperative pickup and delivery based on scheduled trips and profit distribution

Yang Yu[a,c], Qi Lou[a], Jiafu Tang[b], Junwei Wang[c,*], XiaoHang Yue[d]

[a] Institute of Systems Engineering, Northeastern University, Shenyang, 110816, PR China
[b] College of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian, 116023, PR China
[c] Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong
[d] Lubar School of Business, University of Wisconsin-Milwaukee, Milwaukee, WI 53201, United States

ARTICLE INFO

Article history:
Received 6 September 2015
Revised 20 May 2016
Accepted 17 February 2017
Available online xxx

Keywords:
Cooperation
Exact algorithm
Decomposition
Profit distribution
Exact Shapley value

ABSTRACT

Compared to the non-cooperative mode, the cooperative mode is a powerful way to reduce operational cost in pickup and delivery service. In order to protect business sensitive information, sometimes participants are unwilling to open the customer's detailed information. Thus, we utilize the publishable trip scheduled results to compute the saved trips brought by cooperation. A mathematical model minimizing trips of cooperation is proposed. To obtain the exact solution, we define the cooperative trip set. We prove that only when cooperative trip set exists it is possible to save trips by cooperation. For a two-trip cooperative trip set, we exactly obtain the saved trips by enumerating all feasible cooperative cases. For a *K*-trip cooperative trip set, we propose an exact method to obtain the saved trips by decomposing it to at most *K*-1 two-trip cooperative trip sets. Computational complexity of the based-on-decomposition exact algorithm is $O(N)$, where N is the total number of trips. Using the based-on-decomposition algorithm, we calculate the exact Shapley value to distribute profit. To empirically verify the exact method, we perform the extensive experiment cases of the real cooperative pickup and delivery service, i.e., "picking up and delivering customers to airport service" (PDCA).

1. Introduction

Cooperation is a powerful way to reduce operational cost of pickup and delivery service. Before participants agree to join in a cooperation scheme, an estimation of the profit brought by cooperation must be available. The problems in the transportation have been studied (Caputo and Mininno, 1996; Frisk et al., 2010; Audy et al., 2011; Lozano et al., 2013). These studies calculated cost saving brought by cooperation by integrating the original data of all participants. However, sometimes participants do not like to publish the customer's detailed information in order to maintain business sensitive information, but it is acceptable to open the scheduled results because the scheduled results do not show sensitive customer's information. As a consequence, a method should be developed to estimate the profit brought by cooperation based on scheduled results.

This paper is originally motivated by the cost reduction brought by a real cooperative pickup and delivery service, i.e., "picking up and delivering customers to airport service" (PDCA). In real PDCA, a case of customer's detailed information is shown in Table 1. The customer's detailed information was provided by the companies performing PDCA, such as Zhongshan, Shuntian, and Jiantong Inc. in Shenyang in China (Tang et al., 2008, 2014; Yu et al., 2014, 2016).

Location means the vertical and horizontal coordinates of customer's preferred location to pick up the customer.

Customer's detailed information can reveal company's business sensitive information. For example of Table 1, we can know location (50, 70) is the important customer point. Thus, other companies can lure customers in thus important customer points. However, a company may publish the scheduled result when joining in cooperation. A scheduled result of Table 1 can be shown in Table 2.

As shown in Table 2, the scheduled result can conceal some sensitive business information, such as location of picking up and the number of customers in location. Thus, it is acceptable for a company to open the scheduled trips to participate cooperation.

Before participating cooperation, each company wonders the exact profit distributed by cooperation. Therefore, the first objective of the study is to estimate the exact profit brought by cooperation based on trip scheduled results. The second objective is to obtain the fair profit distribution based on exact Shapley value to stabilize coalition.

* Corresponding author.
E-mail address: jwwang@hku.hk (J. Wang).

http://dx.doi.org/10.1016/j.cor.2017.02.015
0305-0548/

Table 1
A case of customer's detailed information.

Location	Airport arrival soft time window	Airport arrival hard time window	Number of customers
(50,70)	[9:20 9:30]	[9:10 9:40]	3
(50,60)	[9:30 9:40]	[9:20 9:50]	1
(50,70)	[9:50 10:00]	[9:40 10:10]	2

Table 2
A scheduled result of Table 1.

Trips	Airport arrival time window	Number of customers
1	[9:28 9:32]	4
2	[9:48 10:02]	2

The remainder of this research is organized as following. The literature review on cooperative cases in the transportation, profit distribution, and picking up and delivering customers to airport service (PDCA) is given in the second section. The third section constructs the mathematical model minimizing trips of cooperation. We define cooperative trip set and prove that if no cooperative trip set exists the trips cannot be saved by cooperation. For a two-trip cooperative trip set, we obtain the exact solution by enumerating all feasible cooperative cases. For a cooperative trip set with K trips, we propose a novel decomposition method to obtain the exact solution by decomposing it to at most K-1 two-trip cooperative trip sets. Section four develops a based-on-decomposition algorithm to accurately calculate saved trips by cooperation. The computational complexity of the exact algorithm is $O(N)$, where N is the total number of used trips in non-cooperative companies. The fifth section demonstrates the profit distribution based on Shapley value. The Shapley value can be easily obtained by the exact algorithm. Section six gives the extensive computational cases from PDCA and states how to compute exactly Shapley value based on the decomposition algorithm. In the last section conclusions and future research are given. The results of extensive experiments are given in Appendices A–C.

2. Literarily review

2.1. Cooperative case in transportation sector

Compared to the non-cooperative mode, cooperative pickup and delivery is a powerful way to reduce operational costs. Profit calculation and profit distribution are the two keys. The two problems in the transportation have been researched (Lozano et al., 2013), such as grocery distribution (Caputo and Mininno, 1996), distribution in rural areas (Hageback and Segerstedt, 2004), freight carriers (Krajewska et al., 2008), forest (Frisk et al., 2010), and railway transportation (Sherali and Lunday, 2011).

Other researchers have approached cost saving and profit distribution from theoretical points of view. Cruijssen et al. (2007) carry out extensive experiments in order to measure cost savings on a number of characteristics of the distribution problem and found that significant cost savings are achievable. Cruijssen et al. (2010) present an approach for the initiative entering the cooperation among logistics service providers.

These researches compute the saved cost by integrating the data of all participants in cooperation. To maintain business sensitive information, a participant is unwilling to open the customer's detailed information, but can publish the scheduled results. Therefore, our study focuses on how to estimate the cost saving brought by cooperation based on the publishable scheduled results of participants.

In addition, some prior studies used heuristic or meta-heuristic algorithms to estimate cost saving of cooperation due to the larger-scale instances caused by cooperation. Krajewska et al. (2008) use the heuristic proposed by Ropke and Pisinger (2006) to solve their problem. The heuristic is based on the large neighborhood search heuristic. To minimize execution costs for a coalition of freight forwarders, Ergun et al. (2007) use a greedy heuristic as well as set partitioning, sets of cycles to solve the instance. As a result, the profit brought by cooperation is not exact. So we focus on exactly computing the profit brought by cooperation.

2.2. Profit distribution

Most profit sharing cases were studied based on cooperative game theory. The set of solutions includes the kernel, the bargaining set, the stable set, the core, the Shapley value and the nucleolus (Ordeshook, 1986; Osborne and Rubinstein, 1994). Engevall et al. (1998) investigate the routes costs allocation among the customers based on a traveling salesman game. Krajewska et al. (2008) use the Shapley value for sharing profits in cooperative freight carriers in order to balance their request portfolios. The profits are estimated through a multi-depot Pickup and Delivery Problem with time windows. Özener and Ergun (2008) propose several cost-allocation schemes based on cooperative game concepts (such as stability and others) applied to a logistics network in which shippers collaborate. Frisk et al. (2010) study a cooperative forest transportation planning problem and investigate some classical cost-allocation methods (including the Shapley value and the nucleolus). Sherali and Lunday (2011) analyze four alternative schemes to apportion railcars to manufacturers and propose a new railroad allocation scheme.

The Shapley value (Shapley, 1953) is one of the solution methods most common in cooperative game theory. We shall retain here the Shapley value as the mechanism to share the dividend of cooperation among participants. The exact Shapley value for some special cases has been studied. Littlechild and Owen (1973) give a famous simple expression for the Shapley value for airport runway cost games. Kuipers et al. (2013) study the exact expression of Shapley value for the cost sharing in highways.

2.3. Picking up and delivering customers to airport service

Flight Ticket Sales Agency (short for FTSA) operates as a typical service company in China aviation service industry. The major services include ticket sales, flight lines design, and delivering tickets to customers. To win the competition, some new services and value-added program are proposed to facilitate customers in some FTSAs, among which 'picking up and delivering of customers to airport' (short for PDCA) is a recently provided new service in some FTSAs, e.g., Zhongshan Flight Ticket Sales Service Company Inc. With the PDCA service, the customers who bought the flight tickets have the rights to be picked up at his preferred time and position, and delivered to airport within his specified deadline. This new service can facilitate customers to airport in a way of more convenient, and thus earn high customer satisfaction. To diminish operational costs, the cooperative PDCA is proposed because the total operational costs can be saved by cooperation. Moreover, there are at least three FTSAs providing PDCA in Shenyang, such as Zhongshan, Shuntian, and Jiantong FTSAs.

PDCA is a real case of Pickup and Delivery Problem. Pickup and Delivery Problem has received widely attention (Cherkesly et al., 2015; Gouveia and Ruthmair, 2015; Iori and Riera-Ledesma, 2015; Madankumar and Rajendran, 2016). However, PDCA has the following distinctive characteristics: 1) the capacity of the vehicles is small, because the used vehicles in PDCA service are cars to comfort customers; and 2) the arrival airport time window is tight, because it's unacceptable for customers to arrive in airport too early or too late. Therefore, PDCA has been researched by academic from many perspectives. Regarding single-trip mode, Tang et al. (2008) establish a multi-objective model of minimizing costs and maximizing service quality, and solve it using a two-stage heuristic algorithm based on savings algorithm. Dong et al. (2008) study the model of minimizing costs by a permutation-based cluster priority heuristics. Dong et al. (2011) propose an exact algorithm for the single-trip mode of PDCA based on set-partitioning model. Regarding multi-trip mode, Tang et al. (2014) propose an exact algorithm based on the trip-chain-oriented set-partitioning (TCO-SP) model in order to reduce the operational cost. Regarding reducing carbon emission of PDCA, Yu et al. (2014) investigate the measure to reduce carbon emission of PDCA by multi-trip mode, and demonstrate that multi-trip mode can reduce 17% carbon emission.

Our study focuses on how to accurately estimate profit brought by cooperative PDCA using the trip scheduled results and how to obtain the exact Shapley value for reasonably distributing the profit.

3. Model minimizing trips of cooperation based on scheduled results of all participants

3.1. Notations

Indices

c	index of company ($c = 1,2,\ldots, \|S\|$)
i, k	index of feasible trips in cooperation of S (i, $k = 1,2,\ldots, \|FTS\|$)
j	index of scheduled trips ($j = 1,2,\ldots,J$).

Input parameters

Q	Capacity of vehicle. In the research, Q is set to 4 to protect customers' satisfaction degree in real PDCA (Tang et al., 2008, 2014)
S	Set of companies participating cooperation. $\|S\|$ is the number of companies in S
FTS	Set of the feasible trips in cooperation of S. $\|FTS\|$ is the number of feasible trips in S
S_c	company c in S, $S_c \in S$
$n(S_c)$	number of trips in scheduled result of S_c
fc	fixed cost of a rental vehicle
t_{cj}	the jth trip of company c
st_{cj}	the customer set of t_{cj}
$[et_{cj}, lt_{cj}]$	time window of t_{cj}
wt_{cj}	Number of customers in t_{cj}. Therefore, t_{cj} can be expressed as $\boxed{wt_{cj}}$.
t_i	the ith feasible trip in S, $t_i \in FTS$
st_i	the customer set of t_i
wt_i	the number of customers in t_i
$[et_i, lt_i]$	time window of t_i;

Decision variable

$$x_i = \begin{cases} 1, & \text{if } t_i \text{ is used in cooperation} \\ 0, & \text{otherwise} \end{cases}$$

r	Number of saved trips by cooperation
$v(S)$	saved cost through cooperation of S

3.2. Mathematics model minimizing trips of cooperation

Objective:

$$\min \sum_{i=1}^{|FTS|} x_i, \tag{1}$$

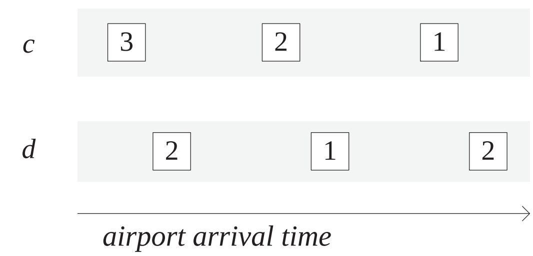

Fig. 1. A cooperative case unable to save trips.

Subject to:

$$wt_i x_i \le Q, \ \forall i, \tag{2}$$

$$\bigcup_{i=1}^{|FTS|} st_i = \bigcup_{c=1}^{|S|} \bigcup_{j=1}^{n(S_c)} st_{cj}, \ \ i|x_i = 1, \ \forall c, \ \forall j, \tag{3}$$

$$st_i \cap st_k = \emptyset, \ \ i, k|x_i = x_k = 1, \ \ i \ne k, \tag{4}$$

$$et_i = \max\{et_{cj}\}, \ t_{cj}|st_{cj} \cap st_i \ne \emptyset, \ \ i|x_i = 1, \tag{5}$$

$$lt_i = \min\{lt_{cj}\}, \ t_{cj}|st_{cj} \cap st_i \ne \emptyset, \ i|x_i = 1. \tag{6}$$

where Eq. (1) is to minimize the trips used in cooperation; Eq. (2) guarantees that the customers in each trip of cooperation cannot exceed the vehicle capacity; Eq. (3) states that the customers of all companies must be served in cooperation; Eq. (4) is the constraint that each customer in cooperation is visited once exactly; Eqs. (5) and (6) are the time windows constraints, i.e., et_i is the maximal earliest airport arrival time of all trips whose customers are served by cooperative trip t_i, and lt_i is the minimal latest airport arrival time of all trips whose customers are served by cooperative trip t_i.

Property. Cooperation cannot save any trip, if there is no time window intersection among the trips of all companies.

Explanation. According to Eqs. (5) and (6), no time window intersection among trips of all companies means that each trip (t_i) in cooperation is formed by only one trip (t_{cj}) of non-cooperation, i.e., $[et_i, lt_i] = [et_{cj}, lt_{cj}]$. As a consequence, customers of any trip cannot be reallocated to other trip(s). This suggests that the number of trips cannot be reduced by cooperation. Therefore, $\min \sum_{i=1}^{|FTS|} x_i = \sum_{c=1}^{|S|} n(S_c)$. See Example 1.

Example 1. Suppose that two companies and the trip scheduled results are shown in Fig. 1, where no trip can cooperate with other trips because their time windows do not overlap. Therefore, the total number of trips in cooperation (i.e., 6) equals that in non-cooperation (i.e., $3+3$). The cooperation cannot save any trip.

In Fig. 1, c and d represent company c and company d respectively, but $c \ne d$; a box represents a trip; the number in a box represents the number of customers; the left and right borders of a box are the earliest and latest airport arrival time of the trip respectively.

PROPERTY means that we can compute the saved trips (i.e., $\sum_{c=1}^{|S|} n(S_c)$-$\min \sum_{i=1}^{|FTS|} x_i$) by considering only the trips which overlap with other trip(s). Thus, we define cooperative trip set.

3.3. Cooperative trip set

Definition 1. In a cooperative trip set, the trips must satisfy:

(1) Belong to at least two different companies;

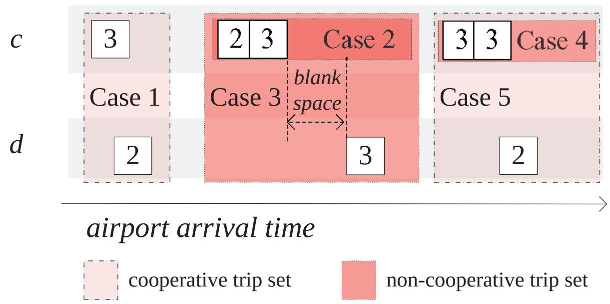

Fig. 2. Trip scheduled results of two companies and cooperative trip sets.

(2) The airport arrival time window of each trip overlaps with that of other trip(s);
(3) The union of the trips covers the whole space between the earliest airport arrival time of the trips and the latest airport arrival time of the trips. This means that no blank space exists in the whole space.

A cooperative trip set means that it is possible to save trips by reallocating customers in the trips.

Example 2. Two companies, and the trip scheduled results are shown in Fig. 2 where company c has 5 trips (i.e., $\boxed{3}$, $\boxed{2}$, $\boxed{3}$, $\boxed{3}$, and $\boxed{3}$) and company d has 3 trips ($\boxed{2}$, $\boxed{3}$, and $\boxed{2}$). According to Definition 1, in Fig. 2, we can know:

Case 1: Trip 1 of company c and trip 1 of company d can form a cooperative trip set.
Case 2: Trips 2 and 3 of company c cannot form a cooperative trip set because they both belong to company c, which violates condition (2).
Case 3: Trips 2-3 of company c and trip 2 of company d cannot form a cooperative trip set because condition (3) in Definition 1 is not satisfied (see the blank space in Case 3).
Case 4: Trips 4 and 5 of company c cannot form a cooperative trip set.
Case 5: Trips 4-5 of company c and trip 3 of company d can form a cooperative trip set. At this time, one trip can be saved.

The cooperative trip set in Case 1 cannot save trips, because the total number of customers is 5 and so at least 2 trips are used in cooperation. The cooperative trip set in Case 5 can save one trip, because 8 customers in the 3 trips can be allocated to 2 trips by cooperation.

3.4. Two-trip cooperative trip set and optimal solution by enumeration

Suppose a cooperative trip set has two trips labeled as $\boxed{wt_1}$ and $\boxed{wt_2}$ and contain wt_1 and wt_2 customers respectively. According to Definition 1, $[e_{\boxed{wt_1}}, l_{\boxed{wt_1}}] \cap [e_{\boxed{wt_2}}, l_{\boxed{wt_2}}] \neq \emptyset$. $e_{\boxed{wt_1}}$ and $l_{\boxed{wt_1}}$ are the earliest and latest airport arrival time of $\boxed{wt_1}$ respectively

Theorem 1. *Any two-trip cooperative trip set can be exactly solved by enumerating according to the sum of and wt_1 and wt_2.*

Proof. A two-trip cooperative trip set can be expressed as Fig. 3 after sorting trips in ascending order based on earliest airport arrival time ($e_{\boxed{wt_i}}$).

Subsequently, all three cases for the sum of wt_1 and wt_2 can be expressed as $wt1 + wt2\left\{\begin{matrix}<Q\\=Q\\>Q\end{matrix}\right.$.

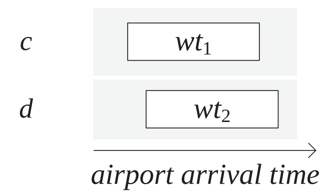

Fig. 3. A two-trip cooperative trip set.

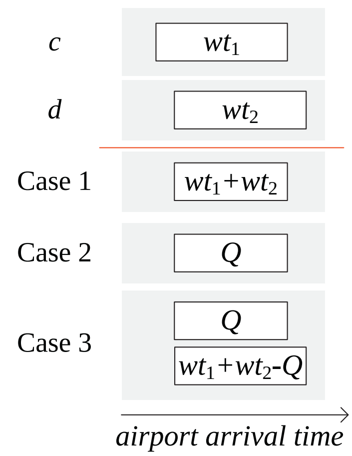

Fig. 4. All three feasible cooperative cases for a two-trip cooperative trip set.

Case 1: When $wt_1 + wt_2 < Q$, obviously it is optimal to form the cooperative trip $\boxed{wt_1+wt_2}$. At this time, one trip is saved.
Case 2: When $wt_1 + wt_2 = Q$, obviously it is optimal to form the cooperative trip $\boxed{Q}$. At this time, one trip is saved. The difference between Case 1 and Case 2 is that Case 1 can continue to cooperate with other trip(s).
Case 3: When $wt_1 + wt_2 > Q$, two trips must be used and no trip can be saved. At this time, it is optimal to form the first trip $\boxed{Q}$ and the second trip $\boxed{wt_1+wt_2\text{-}Q}$ because the first trip has the highest loading efficiency and second trip has the highest possibility of continuing to cooperate with other trip(s).

Thus, it is proved. □

Each case's optimal cooperative solution is shown in Fig. 4. Theorem 1 means that for a two-trip cooperative trip set we can easily obtain the optimal solution of trip number according to the sum of customer numbers in the two trips.

However, for a cooperative trip set with more trips, it is difficult to enumerate all feasible cooperative cases for several reasons. First, the types of the time window intersections among trips increase with the number of trips. Second, the number of feasible cooperative cases grows exponentially with the number of trips. For example of a three-trip cooperative trip set, we demonstrate the difficulty of enumeration.

3.5. Three-trip cooperative trip set and optimal solution by enumeration

Theorem 2. *A three-trip cooperative trip set can be exactly solved by enumerating the all feasible cooperative cases.*

Proof. Suppose three trips are labeled as $\boxed{wt_1}$, $\boxed{wt_2}$, and $\boxed{wt_3}$ respectively. They are sorted in ascending order based on $e_{\boxed{wt_i}}$. According to Definition 1, for a three-trip cooperative trip set, the time window intersections can be divided into the following two types.

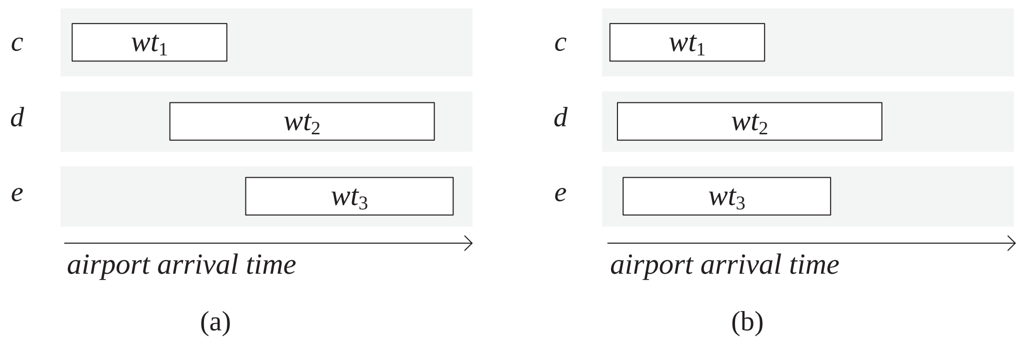

Fig. 5. All two types of three-trip cooperative trip set.

Table 3
All five cases for three-trip cooperative trip set when $[e\boxed{wt_1}, l\boxed{wt_1}] \cap [e\boxed{wt_2}, l\boxed{wt_2}] \neq \emptyset$, $[e\boxed{wt_2}, l\boxed{wt_2}] \cap [e\boxed{wt_3}, l\boxed{wt_3}] \neq \emptyset$, and $[e\boxed{wt_1}, l\boxed{wt_1}] \cap [e\boxed{wt_3}, l\boxed{wt_3}] = \emptyset$.

Sum of wt_1 and wt_2	Sum of $wt_1 + wt_2$-Q and wt_3	Cases	Optimal solution	Saved trips
$wt_1 + wt_2 < Q$		1	form $\boxed{wt_1+wt_2}$ and reserve $\boxed{wt_3}$	1
$wt_1 + wt_2 = Q$		2	form $\boxed{Q}$ and reserve $\boxed{wt_3}$	1
	$wt_1 + wt_2\text{-}Q + wt_3 < Q$	3	form $\boxed{Q}$ and $\boxed{wt_1+wt_2+wt_3\text{-}Q}$	1
$wt_1 + wt_2 > Q$	$wt_1 + wt_2\text{-}Q + wt_3 = Q$	4	form $\boxed{Q}$ and $\boxed{Q}$	1
	$wt_1 + wt_2\text{-}Q + wt_3 > Q$	5	form $\boxed{Q}$, $\boxed{Q}$, and $\boxed{wt_1+wt_2+wt_3\text{-}2Q}$	0

Table 4
All cases for three-trip cooperative trip set when $[ent_1, lnt_1] \cap [ent_2, lnt_2] \cap [ent_3, lnt_3] \neq \emptyset$.

Sum of wt_1, wt_2, and wt_3	Cases	Optimal solution	Saved trips
$wt_1 + wt_2 + wt_3 < Q$	6	form $\boxed{wt_1+wt_2+wt_3}$	2
$wt_1 + wt_2 + wt_3 = Q$	7	form $\boxed{Q}$	2
$Q < wt_1 + wt_2 + wt_3 < 2Q$	8	form $\boxed{Q}$ and $\boxed{wt_1+wt_2+wt_3\text{-}Q}$	1
$wt_1 + wt_2 + wt_3 = 2Q$	9	form $\boxed{Q}$ and $\boxed{Q}$	1
$2Q < wt_1 + wt_2 + wt_3$	10	form $\boxed{Q}$, $\boxed{Q}$, and $\boxed{wt_1+wt_2+wt_3\text{-}2Q}$	0

$$\begin{cases} [e\boxed{wt_1}, l\boxed{wt_1}] \cap [e\boxed{wt_2}, l\boxed{wt_2}] \neq \emptyset, [e\boxed{wt_2}, l\boxed{wt_2}] \cap [e\boxed{wt_3}, l\boxed{wt_3}] \neq \emptyset, \\ \text{and } [e\boxed{wt_1}, l\boxed{wt_1}] \cap [e\boxed{wt_3}, l\boxed{wt_3}] = \emptyset; \\ [e\boxed{wt_1}, l\boxed{wt_1}] \cap [e\boxed{wt_2}, l\boxed{wt_2}] \cap [e\boxed{wt_3}, l\boxed{wt_3}] \neq \emptyset. \end{cases}$$

According to the two types of the time window intersections, any three-trip cooperative trip set can be expressed as Fig. 5(a) or (b).

In Fig. 5, c–e cannot be identical at the same time.

(a) When $[e\boxed{wt_1}, l\boxed{wt_1}] \cap [e\boxed{wt_2}, l\boxed{wt_2}] \neq \emptyset, [e\boxed{wt_2}, l\boxed{wt_2}] \cap [e\boxed{wt_3}, l\boxed{wt_3}] \neq \emptyset$, and $[e\boxed{wt_1}, l\boxed{wt_1}] \cap [e\boxed{wt_3}, l\boxed{wt_3}] = \bullet$, according to the sum of wt_1 and wt_2 and the sum of $wt_1 + wt_2$-Q and wt_3, we can enumerate all cases, as shown in Table 3 and Fig. 6(a).

In Table 3, when $wt_1 + wt_2 > Q$, it is optimal to form the first trip $\boxed{Q}$ and the second trip $\boxed{wt_1+wt_2\text{-}Q}$. Subsequently, according to the sum of $wt_1 + wt_2$-Q and wt_3, we can enumerate the cases 3, 4, and 5.

(b) When $[ent_1, lnt_1] \cap [ent_2, lnt_2] \cap [ent_3, lnt_3] \neq \emptyset$, according the sum of wt_1, wt_2, and wt_3, we can enumerate the following cases, as shown in Table 4 and Fig. 6(b). □

Obviously, it is difficult to enumerate all feasible cooperative cases for the cooperative trip set with more than three trips. Fortunately, the cooperative trip set with K trips can be decomposed into at most K-1 two-trip cooperative trip sets.

3.6. Decomposition for any three-trip cooperative trip set

System decomposition divides a large system into many small ones. Gershwin (1987), Colledani and Gershwin (2013) and Tolio and Matta (1998) use the decomposition method to analyze a complex system with more than two machines by decomposing it to a set of two-machine lines. The idea of the decomposition in the study is to decompose the cooperative trip set with K trips into at most K-1 two-trip cooperative trip sets. The aim of the decomposition approach is to extend two-trip cooperative trip set to investigate the optimal solution of the cooperative trip set with more trips.

Theorem 3. *A three-trip cooperative trip set can be exactly solved by decomposing it to at most 2 two-trip cooperative trip sets.*

Proof. Suppose three trips are labeled as $\boxed{wt_1}$, $\boxed{wt_2}$, and $\boxed{wt_3}$ respectively. The decomposition for a three-trip cooperative trip set includes two steps.

The first step is to form the two-trip (i.e., $\boxed{wt_1}$ and $\boxed{wt_2}$) cooperative trip set. Then produce the all feasible cooperative cases according to the sum of wt_1 and wt_2, as shown in Fig. 7(1) same as Fig. 4.

The second step judges if the last trip (i.e., $\boxed{wt_1+wt_2}$ or $\boxed{wt_1+wt_2\text{-}Q}$) produced in the first step can cooperate with trip $\boxed{wt_3}$. If no, the final optimal solution is obtained; otherwise, a new two-trip cooperative trip set is formed and the final feasible cooperative cases are obtained by solving the new two-trip cooperative trip set, as shown in Table 5 and Fig. 7(2). □

As shown in Fig. 7, for a three-trip cooperative trip set, based on the decomposition method, we can obtain all feasible cases same as those obtained by enumeration. At most 2 two-trip cooperative trip sets are formed in the decomposition (see Cases 2 and 4 in Fig. 7(2)).

3.7. Decomposition for a four-trip cooperative trip set

Theorem 4. *A four-trip cooperative trip set can be exactly solved by decomposing it to at most 3 two-trip cooperative trip sets.*

Proof. Suppose $\boxed{wt_1}$, $\boxed{wt_2}$, $\boxed{wt_k}$, and $\boxed{wt_4}$ represent the four trips respectively. The decomposition for a four-trip cooperative trip set includes three steps.

The first step is to form the two-trip (i.e., $\boxed{wt_1}$ and $\boxed{wt_2}$) cooperative trip set. Then produce the all feasible cooperative cases according to the sum of wt_1 and wt_2, as shown in Fig. 8(1) same as Fig. 7(1) and Fig. 4.

Please cite this article as: Y. Yu et al., An exact decomposition method to save trips in cooperative pickup and delivery based on scheduled trips and profit distribution, Computers and Operations Research (2017), http://dx.doi.org/10.1016/j.cor.2017.02.015

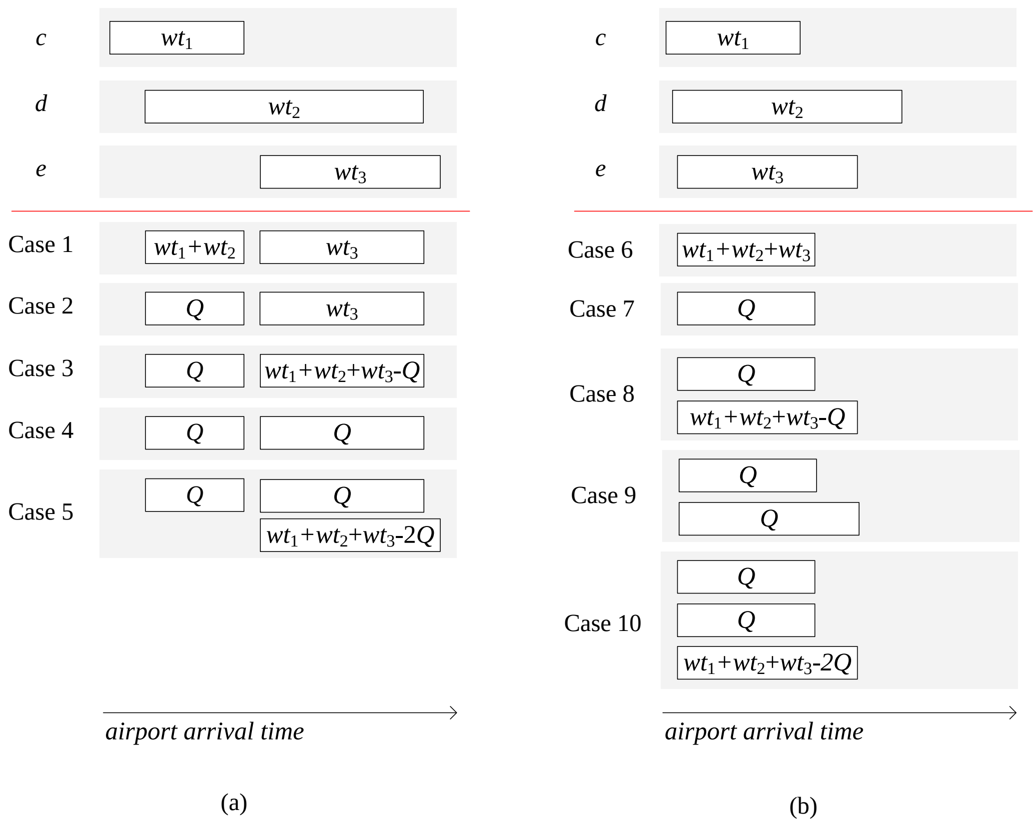

Fig. 6. Enumerating all feasible cooperative cases for three-trip cooperative trip set.

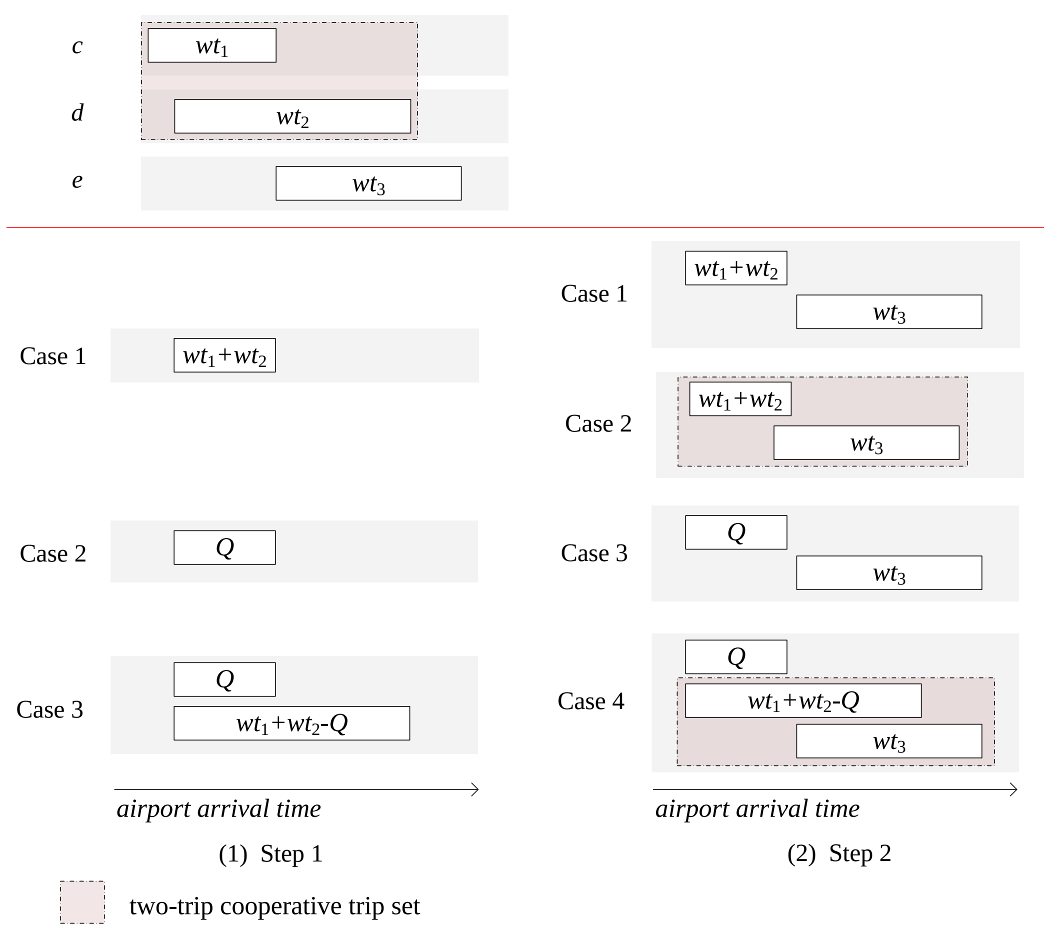

Fig. 7. Decomposition for a three-trip cooperative trip set.

Table 5
All cases for three-trip cooperative trip set in step 2 in decomposition.

Sum of wt_1 and wt_2	Can $\boxed{wt_1+wt_2}$ or $\boxed{wt_1+wt_2\text{-}Q}$ cooperate with $\boxed{wt_3}$?	Cases	Optimal solution or new two-trip cooperative trip set	Cases in Fig. 6
$wt_1+wt_2<Q$	No	1	form $\boxed{wt_1+wt_2}$ and reserve $\boxed{wt_3}$	Case 1
	Yes	2	form two-trip ($\boxed{wt_1+wt_2}$ and $\boxed{wt_3}$) cooperative trip set	Cases 6–8
$wt_1+wt_2=Q$		3	form $\boxed{Q}$ and reserve $\boxed{wt_3}$	Case 2
$wt_1+wt_2>Q$	Yes	4	form $\boxed{Q}$ two-trip ($\boxed{wt_1+wt_2\text{-}Q}$ and $\boxed{wt_3}$) cooperative trip set	Cases 3–5, 9, and 10

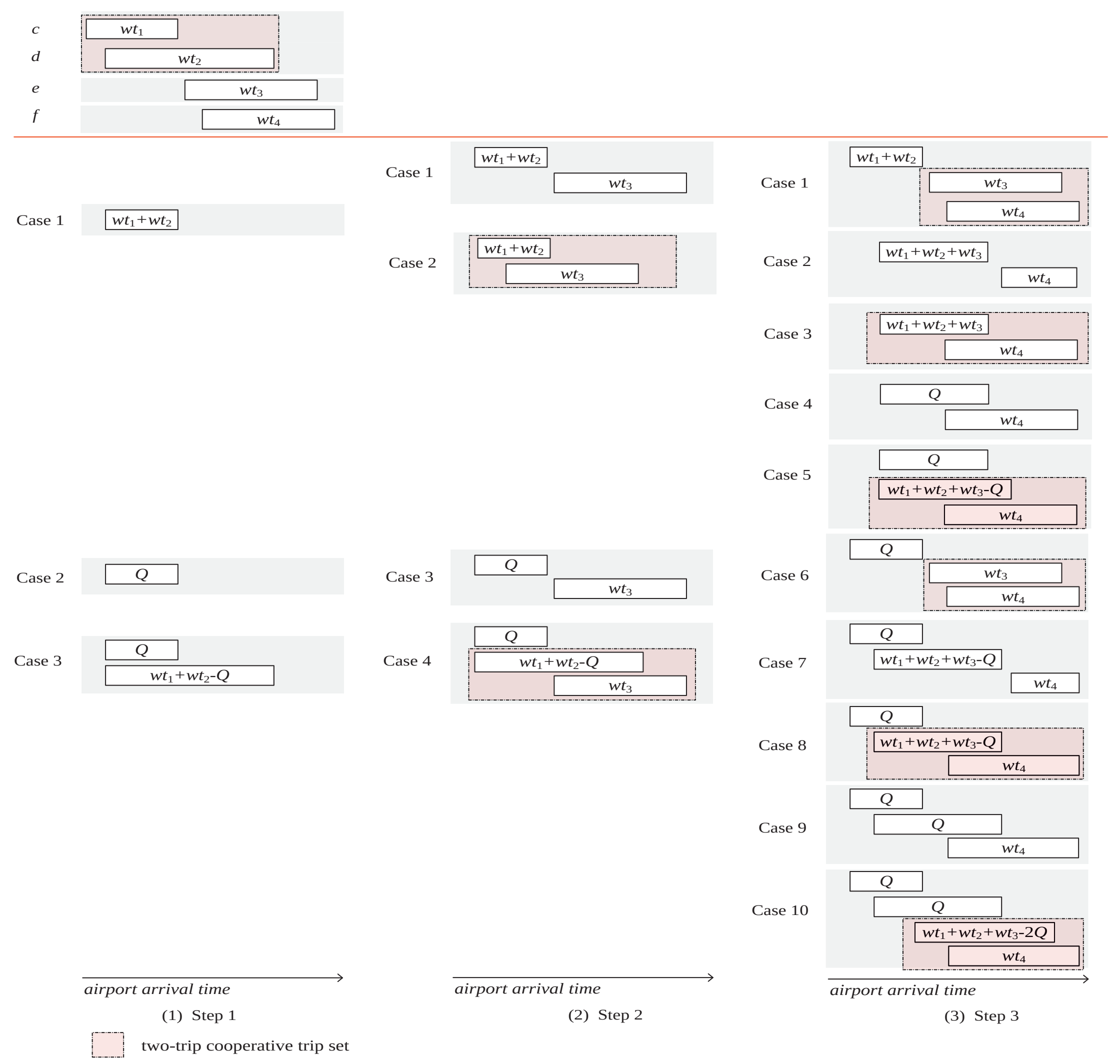

Fig. 8. Decomposition for a four-trip cooperative trip set.

The second step judges if the last trip (i.e., $\boxed{wt_1+wt_2}$ or $\boxed{wt_1+wt_2\text{-}Q}$) produced in the first step can cooperate with trip $\boxed{wt_3}$. If no, trip $\boxed{wt_3}$ is added into the second step as the last trip; otherwise a new two-trip cooperative trip set is formed and the all feasible cooperative cases are obtained by solving the new two-trip cooperative trip set, as shown in Fig. 8(2) same as Fig. 7(2).

The third step judges if the last trip produced in the second step can cooperate with trip $\boxed{wt_4}$. If no, the final optimal solution is obtained; otherwise, a new two-trip cooperative trip set is formed and the final all feasible cooperative cases are obtained by solving the new two-trip cooperative trip set, as shown in Fig. 8(3). □

As shown in Fig. 8, for a four-trip cooperative trip set, based on the decomposition method, we can easily obtain the optimal solution by all the feasible cooperative cases in step 3, i.e., Fig. 8(3). At most 3 two-trip cooperative trip sets are formed in the decomposition (see Cases 1, 3, 5, 6, 8, and 10 in Fig. 8(3)).

3.8. Decomposition for any K-trip cooperative trip set

Theorem 5. *Any K-trip cooperative trip set can be exactly solved by decomposing it to at most K-1 two-trip cooperative trip sets.*

Proof. Suppose $\boxed{wt_1}$, $\boxed{wt_2}$, ..., $\boxed{wt_k}$ represent the *K* trips respectively. The decomposition includes *K*-1 steps.

The first step is to form the two-trip (i.e., $\boxed{wt_1}$ and $\boxed{wt_2}$) cooperative trip set. Then produce the all feasible cooperative cases according to the sum of wt_1 and wt_2.

The *i*th (i = 2,3,...*K*−1) step judges if the last trip produced in the $(i-1)$th step can cooperate with trip $\boxed{wt_{i+1}}$. If no, trip $\boxed{wt_{i+1}}$ is added into the *i*th step as the last trip; otherwise a new two-trip cooperative trip set is formed and subsequently the cooperative cases are obtained by solving the new two-trip cooperative trip set.

Until the $(K-1)$th step is operated. □

4. Exact algorithm for saving trips based on decomposition

4.1. The procedure of exact algorithm for saving trips based on decomposition

Based on the decomposition method, we propose an exact algorithm to accurately solve saved trips by cooperation. The procedure can be described in detail as following.

Algorithm. Exact algorithm for saving trips based on decomposition.

```
Input: NT (set of all scheduled trips in non-cooperation).
Output: Number of saved trips by cooperation.
(1) Initialize.
        ST←sorting NT in ascending order based on earliest airport arrival time
        r (saved trips)←0
        dt (seed trip)←null
        ndt (the nearest following trip of dt)←null
        ct (produced cooperative trip)←null
(2) for (each st_j ∈ ST) do
        st_j.IfCooperated←false
    end for
(3) while (st_i ∈ ST) do
      (3-1) if (st_i.IfCooperated = true) then
                i←i+1
                continue
            end if
      (3-2) dt←st_i
      (3-3) ndt←st_{i+1}
      (3-4) if (ndt = null) then
                break
            end if
            else
(3-5)          if ([edt, ldt] ∩[endt, lndt] = ∅) then
               i←i+1
               continue
           end if
(3-6)          else
(3-6-1)        st_{i+1}.IfCooperated←true
(3-6-2)        if ((dt.Customers + ndt.Customers)<Q) then
                    ct.Customers ←dt.Customers + ndt.Customers
                    ct.EarliestTime←max[dt.EarliestTime, ndt.EarliestTime]
                    ct.LatestTime ←min[dt.LatestTime, ndt.LatestTime]
                    r←r+1
                    i←i+1
                    dt←ct
                    goto (3-3)
            end if
(3-6-3)        if ((dt.Customers + ndt.Customers) = Q) then
                    r←r+1
                    i←i+1
                  Continue
               end if
(3-6-4)        if ((dt.Customers + ndt.Customers)>Q) then
                      ct.Customers ←dt.Customers + ndt.Customers - Q
                      if (ndt.LatestTime > dt.LatestTime) do
                          ct.LatestTime←ndt.LatestTime
                          ct.EarliestTime←ndt.EarliestTime
                      end if
                      else
                            ct.LatestTime←dt.LatestTime
                            ct.EarliestTime←dt.EarliestTime
                    end else
                    i←i+1
                    dt←ct
                    goto (3-3)
                end if
            end else
        end if
    end while
(4) Output r.
```

Step (2) sets all trips' IfCooperated as false. The field records if a trip has cooperated with other trip.

Step (3) is to traverse all trips.

Step (3-1) judges if st_i has been processed as a cooperative trip. If yes, st_i will be skipped.

Step (3-2) sets *dt* (the seed trip) as st_i.

Step (3-3) sets *ndt* as the nearest following trip of *dt. dt* and *ndt* are used to judge if a two-trip cooperative trip set can be formed. The judge procedure is shown as steps (3-5) and (3-6).

Step (3-4) judges if *ndt* is null. If yes, then all trips have been processed, and so the loop stops and *r* is output.

Step (3–5) means that *dt* and *ndt* cannot form a two-trip cooperative trip set and so st_i is skipped.

Step (3–6) means that *dt* can cooperate with *ndt*, i.e., a two-trip cooperative trip set.

Step (3-6-1) sets st_{i+1}.IfCooperated as true to avoid st_{i+1} be processed as the seed trip when st_{i+1} has cooperated with the previous trip. According to the sum of the customer numbers in *dt* and *ndt*, three cases are enumerated as following:

Step (3-6-2) describes when *dt*.Customers + *ndt*.Customers < *Q* the cooperative trip (*ct*) is formed by merging customers and the time windows equals to the intersection of the time windows of *dt* and *ndt*. One trip is saved. *dt* is set as *ct* to continue loop.

Step (3-6-3) describes when *dt*.Customers + *ndt*.Customers = *Q* the cooperative trip (*ct*) is *Q*. One trip is saved. In the next loop, *dt* will be set as st_{i+1}.

Step (3-6-4) describes when *dt*.Customers + *ndt*.Customers > *Q* the first trip is *Q*, and second trip is the cooperative trip (*ct*) containing *dt*.Customers+*ndt*.Customers-*Q* customers. If *ndt*.LatestTime > *dt*.LatestTime, then the time window of *ct* equals that of *ndt*; otherwise, equals that of *dt*. Thus the cooperative trip has the highest possibility to cooperate with the following trips. *dt* is set as *ct* to continue loop.

Step (4) outputs *r* (number of saved trips by cooperation). Obviously, $r=\sum_{c=1}^{|S|} n(S_c)$-min $\sum_{i=1}^{|FTS|} x_i$.

4.2. Mathematical insights on the based-on-decomposition algorithm

Theorem 6. *The computational complexity of the based-on-decomposition algorithm is O(N), where* $N=\sum_{c=1}^{|S|} n(S_c)$, i.e., *the total number of trips in non-cooperation.*

Proof. The computational complexity of the proposed algorithm is $O(N)$ because each trip is processed once. For example of trip st_i, if st_i. IfCooperated is false then st_i will be processed once as the seed trip; otherwise, st_i will be skipped. Since each trip is processed only once, the computational complexity is $O(N)$. □

Theorem 7. *The solution obtained by the based-on-decomposition algorithm is optimal.*

Proof. Step (3-6) implements the decomposition, because *dt* and *ndt* are used to form a new two-trip cooperative trip set. The three feasible cases of the new two-trip cooperative trip set are enumerated by steps (3-6-2), (3-6-3), and (3-6-4). Therefore, the algorithm can obtain the optimal solution of saved trips in cooperation for any *K*-trip cooperative trip set. □

5. Profit distribution based on Shapley value obtained by the algorithm

5.1. Profit brought by cooperation

Let $N=\{1, 2, \ldots, n\}$ be the set of all companies, coalition *S* be any subset of $N(S \subseteq N)$. The profit (i.e., saved costs) brought by the

Table 6
Trip scheduled results of three companies alone operating PDCA.

c	t_{cj}	wt_{cj}	$[et_{cj},lt_{cj}]$	c	t_{cj}	wt_{cj}	$[et_{cj},lt_{cj}]$	c	t_{cj}	wt_{cj}	$[et_{cj},lt_{cj}]$
1	1	3	[6:13,6:27]	2	1	2	[6:16,6:30]	3	1	2	[6:19,6:33]
	2	2	[6:22,6:36]		2	3	[6:25,6:39]		2	3	[6:40,6:54]
	3	2	[6:46,7:00]		3	3	[6:43,6:57]		3	3	[6:49,7:03]
	4	3	[6:52,7:06]		4	2	[6:58,7:12]		4	3	[6:55,7:09]
	5	2	[7:13,7:27]		5	4	[7:16,7:30]		5	2	[7:28,7:42]
	6	2	[7:49,8:03]		6	3	[7:43,7:57]		6	2	[7:46,8:00]
	7	1	[8:07,8:21]		7	1	[8:04,8:18]		7	1	[8:01,8:15]
	8	2	[8:22,8:36]		8	1	[8:19,8:33]		8	1	[8:16,8:30]
	9	4	[8:46,9:00]		9	3	[8:40,8:54]		9	3	[8:37,8:51]
	10	3	[9:04,9:18]		10	3	[8:52,9:06]		10	3	[9:01,9:15]
	11	3	[9:10,9:24]		11	3	[9:16,9:30]		11	2	[9:19,9:33]
	12	2	[9:25,9:39]		12	3	[9:22,9:36]		12	3	[9:28,9:42]
					13	2	[9:43,9:57]		13	2	[9:46,10:00]

cooperation of any coalition S can be expressed as:

$$v(S) = \sum_{c=1}^{|S|} t(S_c) - t(S), \tag{7}$$

where $t(S_c)$ is the operational cost of company c in non-cooperation; $t(S)$ is the operational cost of coalition S.

In rental vehicle mode, $v(S)$ can be expressed as:

$$v(S) = fc * r = fc * \left(\sum_{c=1}^{|S|} n(S_c) - \min \sum_{i=1}^{|FTS|} x_i \right) \tag{8}$$

In cooperation, the cost of any coalition S should be lower than the sum of the individual operational costs of the members in coalition S. Therefore, $v(S)$ must satisfy the following two conditions:

$$v(\emptyset) = 0, \tag{9}$$

$$v(S \cup T) \geq v(S) + v(T),\ S \cap T = \emptyset,\ S, T \subset N, \tag{10}$$

where Eq. (9) is a convention in which a void coalition has a zero value; Eq. (10) states that when two coalitions cooperate, they can achieve at least the same profit as when acting separately.

5.2. Profit distribution solution

A vector $x = (x_1, \ldots, x_p)$ is a profit distribution solution (i.e., imputation) if it satisfies:

$$x_c \geq v(\{c\}),\ c \in N, \tag{11}$$

$$\sum_{c=1}^{n} x_c = v(N), \tag{12}$$

where x_c is the allocation profit of company c.

Eq. (11) states individual rationality, which means that any company cannot accept an allocation where the profit is less than that of it alone operating. Eq. (12) states group rationality, which means that the total cooperative gain, when the grand coalition forms, is fully shared.

From a negotiation perspective, the set of imputations (denoted X) can be seen as the set of feasible agreements. This set is seldom a singleton and therefore it is important to find the fair allocation of profit ($v(S)$) derived from cooperation. Shapley value is one fair method for sharing profit in cooperation (Kuipers et al., 2013).

5.3. Shapley value of cooperation based-on-decomposition algorithm

$\varphi(v) = (\varphi_1(v), \varphi_2(v), \cdots, \varphi_n(v))$ is the Shapley value of cooperation with n companies, where

$$\varphi_i(v) = \sum_{S_c \in S} \frac{(n-|S|)!(|S|-1)!}{n!} [v(S) - v(S - \{S_c\})],$$
$$i = 1, 2, \cdots, n. \tag{13}$$

The factor $v(S) - v(S - \{S_c\})$ corresponds to the marginal contribution of company c to coalition S. Therefore, the Shapley value allocates to each company the weighted sum of his contributions.

In rental vehicle mode, $\varphi_i(v)$ can be exactly solved as:

$$\varphi_i(v) = \sum_{S_c \in S} \frac{(n-|S|)!(|S|-1)!}{n!} fc[r(S) - r(S - \{S_c\})],$$
$$i = 1, 2, \cdots, n, \tag{14}$$

where $r(S)$ is the saved trips brought by coalition S, $r(S\text{-}\{S_c\})$ is the saved trips brought by the cooperation of $S\text{-}\{S_c\}$.

Therefore, using the based-on-decomposition algorithm, the exact Shapley value for profit distribution can be exactly obtained.

6. Computational results

6.1. Experiment instances

To test the real reductions of trips and costs brought by cooperative mode, we use a simple cooperative case to demonstrate the details of the decomposition algorithm. The detailed trip scheduled results are shown in Table 6. Q is set to 4 according to the feature of PDCA. In Table 6, companies 1, 2 and 3 have 12 trips, 13 trips and 13 trips respectively. They have 29 customers, 33 customers and 30 customers respectively. The total number of trips and customers are 38 and 92 respectively.

From Table 6, we can observe that it is possible of saving trips by cooperation because the vehicle loading efficiency of each company is lower (i.e., 60%, 63%, and 58% respectively) and there are many cooperative trip sets in the trips of the three companies.

Definition 2. Vehicle loading efficiency equals $\frac{\text{number of customers}}{\text{number of trips} * Q} * 100\%$.

For example, for companies 1, 2, and 3, the vehicle loading efficiency are 60% ($= \frac{29}{12*4} * 100\%$), 63% ($= \frac{33}{13*4} * 100\%$), and 58% ($= \frac{30}{13*4} * 100\%$).

6.2. Software and hardware specifications

The algorithm was coded in C# and executed on the computer with Intel Core(TM) i5-2400 processor at 3.10 GHz under the Microsoft Windows XP operation system using 4.00 GB RAM.

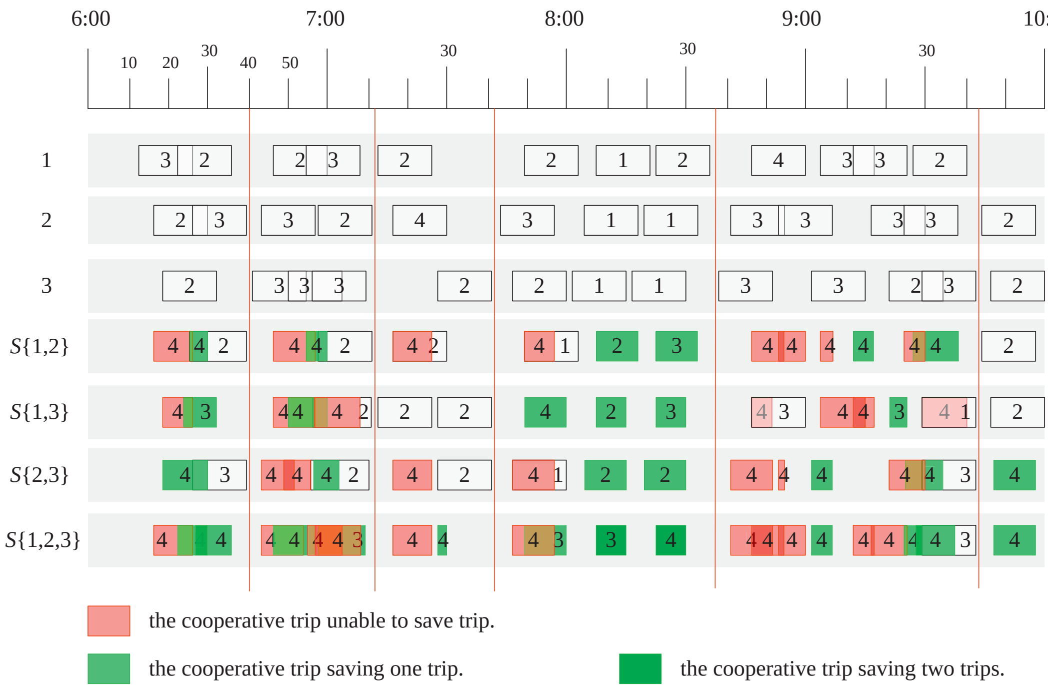

Fig. 9. Exact results of saved trips based on the decomposition algorithm.

Table 7
Exact Shapley value of profit distribution in cooperation.

Coalition	Trips	Exactly saved trips	Cost saving	Shapley value	Saved cost %
1	12	0	0	260	36%
2	13	0	0	290	37%
3	13	0	0	290	37%
1,2	19	6	360		
1,3	19	6	360		
2,3	19	7	420		
1,2,3	24	14	840		

6.3. Exact result of saved trips solved by the based-on-decomposition algorithm

Fig. 9 shows the exact result of saved trips in each coalition (i.e., $S\{1,2\}$, $S\{1,3\}$, $S\{2,3\}$ and $S\{1,2,3\}$) based on the decomposition algorithm. In coalition $S\{1, 2, 3\}$, 42 trips form 6 cooperative trip sets.

6.4. Exact Shapley value for profit distribution based on the decomposition algorithm

Based on the results of saved trips in Fig. 9, we can obtain the exact Shapley value of profit distribution in cooperative PDCA, as shown in Table 7. The rent cost of a trip is set to 60. Consequently, we get the exact Shapley value of companies 1, 2, and 3.

6.5. More experiments to evaluate the decomposition algorithm

To evaluate the proposed based-on-decomposition exact algorithm, we perform extensive experiments. The used trip scheduled results are shown in Table 8.

Table 8 shows fifteen trip scheduled results from three companies labeled as 1, 2 and 3. Nine trip scheduled results are obtained under $Q=4$, six other trip scheduled results are obtained under $Q=8$. The name of a trip scheduled result includes the number of used vehicles, the number of served customers, and the vehicle loading efficiency.

Table 8
Used data of trip scheduled results in the extensive experiments.

Q	Company		
	1	2	3
4	21_40_48%	30_72_60%	51_107_52%
	29_88_76%	36_112_78%	44_116_66%
	37_117_79%	25_80_80%	93_288_77%
8	21_88_52%	20_80_50%	46_107_29%
	22_117_66%	26_112_54%	40_116_36%

For example, "21_40_48%" means 21 used vehicles, 40 served customers and vehicle loading efficiency equal to 48% ($=\frac{40}{21*4}*$ 100% according to Definition 2). In addition, "21_88_52%" means 21 used vehicles, 88 served customers and vehicle loading efficiency equal to 52% ($=\frac{88}{21*8}*100\%$ according to Definition 2).

Based on the nine trip scheduled results from three companies under $Q=4$, we performed 54 experiments to save trips by using the exact algorithm based on decomposition. The results are in Appendix A.

Based on the saved trips of Appendix A, we can easily obtain the exact Shapley value for the cooperation with 3 companies. For a cooperative case with 3 companies, to compute the Shapley value of each company, we need to perform 4 experiments. Based on Appendix A, we can produce 27 cooperative cases. Therefore, we select 9 cooperative cases to show the detailed Shapley value of each company, as shown in Appendix B.

Based on the six trip scheduled results from three companies under $Q=8$, we performed 20 experiments. The results are in Appendix C.

Fig. 10 shows the saved trips in 54 experiments under Q equal to 4. The horizontal axis represents the experiment order and the vertical axis represents the saved trips in each experiment. Fig. 11 shows the saved trips in 20 experiments under Q equal to 8.

Fig. 10 shows that all 54 cooperation can save at least 6 and at most 37 trips. Fig. 11 shows that all 20 cooperation can save at least 4 and at most 36 trips. Therefore, the cooperative mode can be used to save the operational cost.

Moreover, we can observe the following insights from Figs. 10 and 11.

Please cite this article as: Y. Yu et al., An exact decomposition method to save trips in cooperative pickup and delivery based on scheduled trips and profit distribution, Computers and Operations Research (2017), http://dx.doi.org/10.1016/j.cor.2017.02.015

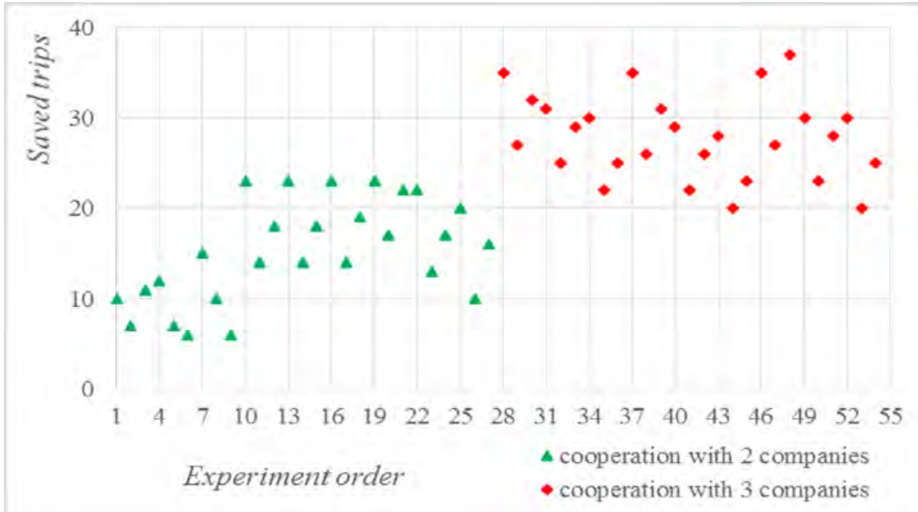

Fig. 10. Saved trips in 54 experiments under Q equal to 4.

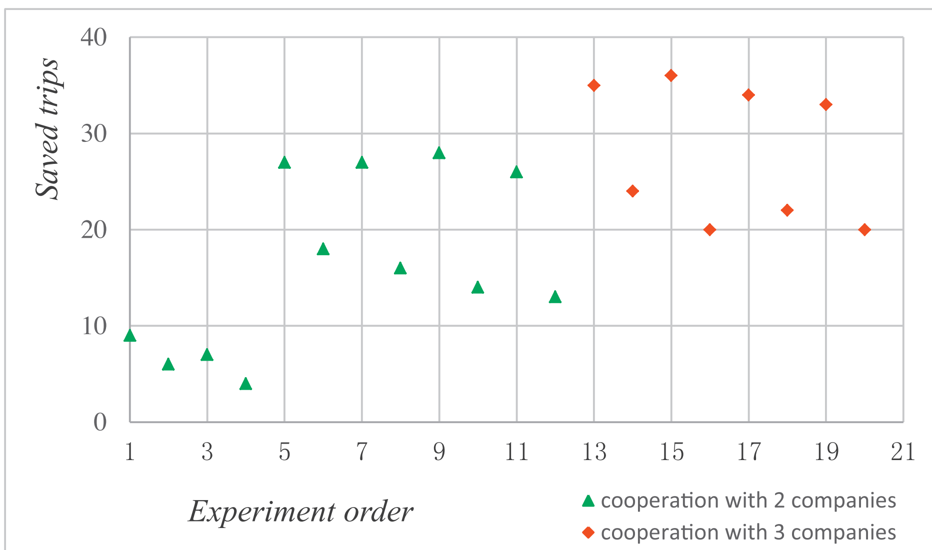

Fig. 11. Saved trips in 20 experiments under Q equal to 8.

1. Saved trips decrease with the increase of vehicle loading efficiency. For example, in Fig. 10, Experiments 4 to 6 and 7 to 9 (where vehicle loading efficiency of company 1 is fixed) shows that the saved trips decrease with the increase of vehicle loading efficiency of company 2 (i.e., 60%, 78%, and 80%). Also, in Fig. 11, when increasing vehicle loading efficiency of a company and fixing that of one or two other companies, the saved trips decrease, such as Experiments 1 to 2, 3 to 4, ..., 19 to 20.
2. Saved trips increase with the increase of the number of used vehicles. That is because the more used vehicles, the higher possibility for a trip cooperating with the other(s). For example, in Fig. 10, Experiments 11 to 12, 14 to 15, ..., 53 to 54, the vehicle loading efficiency increases from 66% to 77%, but the saved trips increase because the number of used vehicles increases from 44 to 93 in these experiments.
3. Saved trips increase with the number of cooperative companies. In Figs. 10 and 11, the saved trips of cooperation with 3 companies are always larger than those of cooperation with any 2 of the 3 companies. For example, in Fig. 10, saved trips in Experiment 28 are larger than that in Experiments 1, 10, and 19.
4. Saved trips depend on the number of cooperative trips. For example, for Experiments 2 and 3 in Fig. 10, trip scheduled result of company 2 changes from 36_112_78% to 25_80_80%. The vehicle loading efficiency increases and the number of used vehicles decrease, but saved trips increases. By investigating the detailed data of 36_112_78% and 25_80_80%, we find the cooperation with 21_40_48% and 25_80_80% has more cooperative trips than that of 21_40_48% and 25_80_80%.
5. We did not find the relationship between saved trips and the number of the customers to be server by cooperation based on the 74 experiments.
6. We did not find the relationship between saved trips and the vehicle capacity (Q) based on the 74 experiments.

Fig. 12 shows the computation times of the exact algorithm based on decomposition for 54 experiments under Q equal to 4 by using the data in Appendix A. Fig. 13 shows the computation times of the exact algorithm based on decomposition for 20 experiments under Q equal to 8 by using the data in Appendix C. From Figs. 12 and 13, we can see the exact algorithm has a good computation performance. For example, in Fig. 12, the worst running time is 2.32 seconds in Experiment 51 where three companies to cooperate to serve 517 customers. In Fig. 13, the worst running time is 0.88 seconds in Experiment 13 where three companies to cooperate to serve 275 customers.

Fig. 12. Running time of 54 experiments under Q equal to 4.

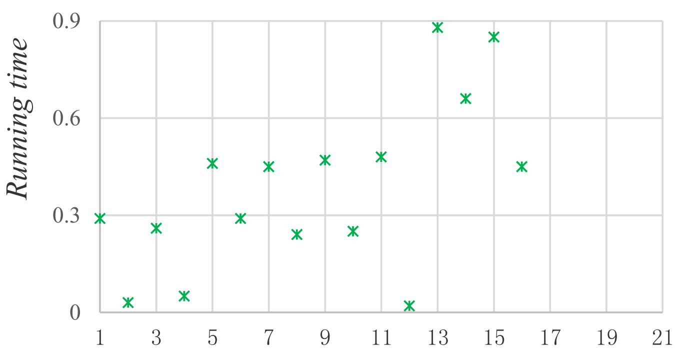

Fig. 13. Running time of 20 experiments under Q equal to 8.

Moreover, from Figs. 12 and 13, we can easily observe that the computation time of the exact algorithm increases linearly with the total number of trips in non-cooperative companies. That proves Theorem 6.

Picking up and delivering customers to airport service is a special case of Pickup and Delivery Problem with the identical delivery point, i.e., airport. As described in Algorithm of exact algorithm for saving trips based on decomposition, the proposed exact algorithm considers the airport arrival time window (i.e., delivery time window). Therefore, the proposed exact algorithm based on decomposition can solve saved trips brought by cooperative based on scheduled trips and profit distribution based on Shapley value for the Pickup and Delivery Problem under which the delivery point is identical and the delivery time window is considered.

7. Conclusion

Our contributions in this study can be summarized as following. First, a mathematical model minimizing trips of cooperation based on trip scheduled results is proposed. By defining cooperative trip set, we prove that cooperation can save trips only when cooperative trip set exists. Second, for the two-trip cooperative trip set, we obtain the optimal solution of saved trips by enumerating all feasible cooperative cases. Subsequently we propose a novel decomposition method to obtain the optimal solution of K-trip cooperative trip set by decomposing it to at most $K-1$ two-trip cooperative trip sets. Third, we develop a based-on-decomposition algorithm to accurately calculate saved trips by cooperation. Computational complexity of the exact algorithm is $O(N)$, where N is the total number of trips. Using the exact algorithm, we calculate the exact Shapley value for a real cooperative pickup and delivery case, i.e., PDCA.

Using the proposed decomposition algorithm, we can further investigate the other classical profit distribution method based on cooperative game theory, such as the kernel, the bargaining set, the stable set, the core and the nucleolus.

Acknowledgement

This research is supported by the National Natural Science Foundation of China (61203182, 71471034, 71571037 and 71571156), a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (T32-101/15-R), and the open project funded by State Key Laboratory of Synthetical Automation for Process Industries (PAL-N201505).

Appendix A

Results of 54 experiments under Q equal to 4.

No.	Name of trip scheduled results: Company 1	Company 2	Company 3	Cooperative result	Time (ss)	Saved trips
1	21_40_48%	30_72_60%	–	41_112_68%	0.27	10
2	21_40_48%	36_112_78%	–	50_152_76%	0.26	7
3	21_40_48%	25_80_80%	–	35_120_86%	0.2	11
4	29_88_76%	30_72_60%	–	47_160_85%	0.45	12
5	29_88_76%	36_112_78%	–	58_200_86%	0.45	7
6	29_88_76%	25_80_80%	–	48_168_88%	0.67	6
7	37_117_79%	30_72_60%	–	52_189_91%	0.62	15
8	37_117_79%	36_112_78%	–	63_229_91%	0.82	10
9	37_117_79%	25_80_80%	–	56_197_88%	0.67	6
10	21_40_48%	–	51_107_52%	49_147_75%	0.7	23
11	21_40_48%	–	44_116_66%	51_156_76%	0.47	14
12	21_40_48%	–	93_288_77%	96_328_85%	1.15	18
13	29_88_76%	–	51_107_52%	57_195_86%	0.92	23
14	29_88_76%	–	44_116_66%	59_204_86%	0.7	14
15	29_88_76%	–	93_288_77%	104_376_90%	1.03	18
16	37_117_79%	–	51_107_52%	65_224_86%	1.1	23
17	37_117_79%	–	44_116_66%	67_233_87%	0.94	14
18	37_117_79%	–	93_288_77%	111_405_91%	1.5	19
19	–	30_72_60%	51_107_52%	58_179_77%	0.73	23
20	–	30_72_60%	44_116_66%	57_188_82%	0.63	17
21	–	30_72_60%	93_288_77%	101_360_89%	1.33	22
22	–	36_112_78%	51_107_52%	65_219_84%	0.94	22
23	–	36_112_78%	44_116_66%	67_228_85%	0.66	13
24	–	36_112_78%	93_288_77%	112_400_89%	1.54	17
25	–	25_80_80%	51_107_52%	56_187_83%	0.72	20
26	–	25_80_80%	44_116_66%	59_196_83%	0.68	10
27	–	25_80_80%	93_288_77%	102_368_90%	1.11	16
28	21_40_48%	30_72_60%	51_107_52%	67_219_82%	1.12	35
29	21_40_48%	30_72_60%	44_116_66%	68_228_84%	0.92	27
30	21_40_48%	30_72_60%	93_288_77%	112_400_89%	1.54	32
31	21_40_48%	36_112_78%	51_107_52%	77_259_84%	1.14	31
32	21_40_48%	36_112_78%	44_116_66%	76_268_88%	1.12	25
33	21_40_48%	36_112_78%	93_288_77%	121_440_91%	1.72	29
34	21_40_48%	25_80_80%	51_107_52%	67_227_85%	1.12	30
35	21_40_48%	25_80_80%	44_116_66%	68_236_87%	1.12	22
36	21_40_48%	25_80_80%	93_288_77%	114_408_89%	1.53	25
37	29_88_76%	30_72_60%	51_107_52%	75_267_89%	1.32	35
38	29_88_76%	30_72_60%	44_116_66%	77_276_90%	1.12	26
39	29_88_76%	30_72_60%	93_288_77%	121_448_93%	1.91	31
40	29_88_76%	36_112_78%	51_107_52%	87_307_88%	1.54	29
41	29_88_76%	36_112_78%	44_116_66%	87_316_91%	1.31	22
42	29_88_76%	36_112_78%	93_288_77%	132_488_92%	2.12	26
43	29_88_76%	25_80_80%	51_107_52%	77_275_89%	1.32	28
44	29_88_76%	25_80_80%	44_116_66%	78_284_91%	1.13	20
45	29_88_76%	25_80_80%	93_288_77%	124_456_92%	1.91	23
46	37_117_79%	30_72_60%	51_107_52%	83_296_89%	1.51	35
47	37_117_79%	30_72_60%	44_116_66%	84_305_91%	1.32	27
48	37_117_79%	30_72_60%	93_288_77%	123_477_97%	2.13	37
49	37_117_79%	36_112_78%	51_107_52%	94_336_89%	1.70	30
50	37_117_79%	36_112_78%	44_116_66%	94_345_92%	1.52	23
51	37_117_79%	36_112_78%	93_288_77%	138_517_94%	2.32	28
52	37_117_79%	25_80_80%	51_107_52%	83_304_92%	1.51	30
53	37_117_79%	25_80_80%	44_116_66%	86_313_91%	1.31	20
54	37_117_79%	25_80_80%	93_288_77%	130_485_93%	2.11	25

Please cite this article as: Y. Yu et al., An exact decomposition method to save trips in cooperative pickup and delivery based on scheduled trips and profit distribution, Computers and Operations Research (2017), http://dx.doi.org/10.1016/j.cor.2017.02.015

Appendix B

Shapley value of 9 cooperation with 3 companies based on results of Appendix A.

		Name of trip scheduled results			trips	value
Cooperation	No.	Company 1	Company 2	Company 3	Saved	Shapley
1	1	21_40_48%	30_72_60%	–	10	1-570
	10	21_40_48%	–	51_107_52%	23	2-570
	19	–	30_72_60%	51_107_52%	23	3-960
	28	21_40_48%	30_72_60%	51_107_52%	35	–
2	2	21_40_48%	36_112_78%	–	7	1-450
	11	21_40_48%	–	44_116_66%	14	2-420
	23	–	36_112_78%	44_116_66%	13	3-630
	32	21_40_48%	36_112_78%	44_116_66%	25	–
3	3	21_40_48%	25_80_80%	–	11	1-470
	12	21_40_48%	–	93_288_77%	18	2-410
	27	–	25_80_80%	93_288_77%	16	3-620
	36	21_40_48%	25_80_80%	93_288_77%	25	–
4	4	29_88_76%	30_72_60%	–	12	1-440
	14	29_88_76%	–	44_116_66%	14	2-530
	20	–	30_72_60%	44_116_66%	17	3-590
	38	29_88_76%	30_72_60%	44_116_66%	26	–
5	5	29_88_76%	36_112_78%	–	7	1-430
	15	29_88_76%	–	93_288_77%	18	2-400
	24	–	36_112_78%	93_288_77%	17	3-730
	42	29_88_76%	36_112_78%	93_288_77%	26	–
6	6	29_88_76%	25_80_80%	–	6	1-450
	13	29_88_76%	–	51_107_52%	23	2-360
	25	–	25_80_80%	51_107_52%	20	3-870
	43	29_88_76%	25_80_80%	51_107_52%	28	–
7	7	37_117_79%	30_72_60%	–	15	1-640
	18	37_117_79%	–	93_288_77%	19	2-730
	21	–	30_72_60%	93_288_77%	22	3-850
	48	37_117_79%	30_72_60%	93_288_77%	37	–
8	8	37_117_79%	36_112_78%	–	10	1-440
	17	37_117_79%	–	44_116_66%	14	2-410
	23	–	36_112_78%	44_116_66%	13	3-530
	50	37_117_79%	36_112_78%	44_116_66%	23	–
9	9	37_117_79%	25_80_80%	–	6	1-490
	16	37_117_79%	–	51_107_52%	23	2-400
	25	–	25_80_80%	51_107_52%	20	3-910
	52	37_117_79%	25_80_80%	51_107_52%	30	–

The data in Column "Shapley value" expresses the company's Shapley value in the cooperation. For example, in cooperation 1, "1-570″ expresses Shapley value of company 1 is 570, and "3-960″ expresses Shapley value of company 3 is 960. In addition, the rent cost per trip is set to 60.

Appendix C

Results of 20 experiments under Q equal to 8

	Name of trip scheduled results				(ss)	trips
No.	Company 1	Company 2	Company 3	Cooperative result	Time	Saved
1	21_88_52%	20_80_50%	–	32_168_66%	0.29	9
2	21_88_52%	26_112_54%	–	41_200_61%	0.03	6
3	22_117_66%	20_80_50%	–	35_197_70%	0.26	7
4	22_117_66%	26_112_54%	–	44_229_65%	0.05	4
5	21_88_52%	–	46_107_29%	40_195_61%	0.46	27
6	21_88_52%	–	40_116_36%	43_204_59%	0.29	18
7	22_117_66%	–	46_107_29%	41_224_68%	0.45	27
8	22_117_66%	–	40_116_36%	46_233_63%	0.24	16
9	–	26_112_54%	46_107_29%	46_219_60%	0.48	26
10	–	26_112_54%	40_116_36%	53_228_54%	0.02	13
11	–	20_80_50%	46_107_29%	38_187_62%	0.47	28
12	–	20_80_50%	40_116_36%	46_196_53%	0.25	14
13	21_88_52%	20_80_50%	46_107_29%	52_275_66%	0.88	35
14	21_88_52%	20_80_50%	40_116_36%	57_284_63%	0.66	24
15	21_88_52%	26_112_54%	46_107_29%	57_307_67%	0.85	36
16	21_88_52%	26_112_54%	40_116_36%	67_316_59%	0.45	20
17	22_117_66%	20_80_50%	46_107_29%	54_304_70%	0.87	34
18	22_117_66%	20_80_50%	40_116_36%	60_313_65%	0.65	22
19	22_117_66%	26_112_54%	46_107_29%	61_336_69%	0.85	33
20	22_117_66%	26_112_54%	40_116_36%	68_345_63%	0.48	20

References

Audy, J.F., D'Amours, S., Rousseau, L.M., 2011. Cost allocation in the establishment of a collaborative transportation agreement-an application in the furniture industry. J. Oper. Res. Soc. 62 (6), 960–970.

Caputo, M., Mininno, V., 1996. Internal, vertical and horizontal logistics integration in Italian grocery distribution. Int. J. Phys. Distrib. Logist. Manage. 26 (9), 64–91.

Cherkesly, M., Desaulniers, G., Laporte, G., 2015. A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading. Comput. Oper. Res. 62, 23–35.

Colledani, M., Gershwin, S.B., 2013. A decomposition method for approximate evaluation of continuous flow multi-stage lines with general Markovian machines. Ann. Oper. Res. 209 (1), 5–40.

Cruijssen, F., Bräysy, O., Dullaert, W., Fleuren, H., Salomon, M., 2007. Joint route planning under varying market conditions. Int. J. Phys. Distrib. Logist. Manage. 37 (4), 287–304.

Cruijssen, F., Borm, P., Fleuren, H., Hamers, H., 2010. Supplier initiated outsourcing: a methodology to exploit synergy in transportation. Eur. J. Oper. Res. 207 (2), 763–774.

Dong, G., Tang, J., Kong, Y., Zhang, J., 2008. Minimizing costs model and algorithm of free pickup and delivery customers to airport service. Chin. J. Syst. Eng. 23 (4), 437–443.

Dong, G., Tang, J., Lai, K., Kong, Y., 2011. An exact algorithm for vehicle routing and scheduling problem of free pickup and delivery service in flight ticket sales companies based on set-partitioning model. J. Intel. Manuf. 22 (5), 789–799.

Engevall, S., Göthe-Lundgren, M., Värbrand, P., 1998. The travelling salesman game: an application of cost allocation in a gas and oil company. Ann. Oper. Res. 82 (0), 203–218.

Ergun, Ö, Kuyzu, G, Savelsbergh, M.W.P., 2007. Shipper collaboration. Comput. Oper. Res. 34, 1551–1560.

Frisk, M., Göthe-Lundgren, M., Jörnsten, K., Rönnqvist, M., 2010. Cost allocation in collaborative forest transportation. Eur. J. Oper. Res. 205 (2), 448–458.

Gershwin, S.B., 1987. An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. Oper. Res. 35 (2), 291–305.

Gouveia, L., Ruthmair, M., 2015. Load-dependent and precedence-based models for pickup and delivery problems. Comput. Oper. Res. 63, 56–71.

Hageback, C., Segerstedt, A., 2004. The need for co-distribution in rural areas: a study of Pajala in Sweden. Int. J. Prod. Econ. 89 (2), 153–163.

Iori, M., Riera-Ledesma, J., 2015. Exact algorithms for the double vehicle routing problem with multiple stacks. Comput. Oper. Res. 63, 83–101.

Krajewska, M.A., Kopfer, H., Laporte, G., Ropke, S., Zaccour, G., 2008. Horizontal cooperation among freight carriers: request allocation and profit sharing. J. Oper. Res. Soc. 59 (11), 1483–1491.

Kuipers, J., Mosquera, M.A., Zarzuelo, J.M., 2013. Sharing costs in highways: a game theoretic approach. Eur. J. Oper. Res. 228 (1), 158–168.

Littlechild, S.C., Owen, G., 1973. A simple expression for the Shapley value in a special case. Manage. Sci. 20 (3), 370–372.

Lozano, S., Moreno, P., Adenso-Díaz, B., Algaba, E., 2013. Cooperative game theory approach to allocating benefits of horizontal cooperation. Eur. J. Oper. Res. 229 (2), 444–452.

Madankumar, S., Rajendran, C., 2016. Mathematical models for green vehicle routing problems with pickup and delivery: a case of semiconductor supply chain. Comput. Oper. Res. online.

Ordeshook, P.C., 1986. Game Theory and Political Theory. Cambridge University Press, Cambridge.

Osborne, M.J., Rubinstein, A., 1994. A Course in Game Theory. MIT Press, Cambridge, MA.

Özener, O.Ö, Ergun, Ö., 2008. Allocating costs in a collaborative transportation procurement network. Transp. Sci. 42 (2), 146–165.

Ropke, S., Pisinger, D., 2006. An adaptive large neighbourhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40, 455–472.

Shapley, L.S., 1953. A value for n-person games. Ann. Math. Stud. 28 (11), 307–317.

Sherali, H.D., Lunday, B.J., 2011. Equitable apportionment of rail cars within a pooling agreement for shipping automobiles. Transp. Res. Part E 47 (2), 263–283.

Tang, J., Dong, G., Pan, Z.D., Kong, Y., 2008. Multi-objective model and algorithm of free pickup customer and delivery to airport service. J. Manage. Sci. China 11 (6), 35–42.

Tang, J., Yu, Y., Li, J., 2014. An exact algorithm for the multi-trip vehicle routing and scheduling problem of pickup and delivery of customers to the airport. Transp. Res. Part E 73 (2015), 114–132.

Tolio, T., Matta, A., 1998. A method for performance evaluation of automated flow lines. Ann. CIRP 47 (1), 373–376.

Yu, Y., Tang, J., Li, J., Sun, W., Wang, J.W., 2016. Reducing carbon emission of pickup and delivery using integrated scheduling. Transp. Res. Part D 47, 237–250.

Yu, Y., Tang, J., Zhang, Y., Yang, P., Li, J., 2014. Reducing carbon emission by multi-trip mode of vehicle. Transp. Res. Rec. 2467, 149–156.