Transportation Research Part B 121 (2019) 41–55

Contents lists available at ScienceDirect

Transportation Research Part B

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

Fair cost allocation for ridesharing services – modeling, mathematical programming and an algorithm to find the nucleolus

Wei Lu [a,1], Luca Quadrifoglio [b,*]

[a] Amazon.com, Inc., United States
[b] Zachry Department of Civil Engineering, Texas A&M University, TX 77843, United States

ARTICLE INFO

Article history:
Received 26 October 2017
Revised 1 January 2019
Accepted 2 January 2019

ABSTRACT

This paper addresses one of the most challenging issues in designing an efficient and sustainable ridesharing service: ridesharing market design. We formulate it as a fair cost allocation problem through the lens of the cooperative game theory. A special property of the cooperative ridesharing game is that its characteristic function values are calculated by solving an optimization problem. Several concepts of fairness are investigated and special attention is paid to a solution concept named nucleolus, which aims to minimize the maximum dissatisfaction in the system. Due to its computational intractability, we break the problem into a master-subproblem structure and two subproblems are developed to generate constraints for the master problem. We propose a coalition generation procedure to find the nucleolus and approximate nucleolus of the game. Experimental results showed that when the game has a non-empty core, in the approximate nucleolus scheme the coalitions are computed only when it is necessary and the approximate procedure produces the actual nucleolus. And when the game has an empty core, the approximate nucleolus is close to the actual one. Regardless of the emptiness of the game, our algorithm needs to generate only a small fraction (1.6%) of the total coalition constraints to compute the approximate nucleolus. The proposed model and results nicely fit systems operated by autonomous vehicles.

1. Introduction

Ridesharing services whose aim is to gather travelers with similar itineraries and compatible schedules, are able to provide substantial environmental and social benefits through reducing the use of private vehicles. When the operations of a ridesharing system are optimized, it can also save travelers a significant amount of transportation cost. The economic benefits associated with ridesharing in turn attract more travelers to participate in ridesharing services and thereby improve the utilization of transportation infrastructure capacity.

* Corresponding author.
E-mail addresses: theweilu@amazon.com (W. Lu), lquadrifoglio@civil.tamu.edu, quadrifo@tamu.edu (L. Quadrifoglio).
[1] This work was done prior to the author joining Amazon.

https://doi.org/10.1016/j.trb.2019.01.001
0191-2615/

An optimized ridesharing service is usually designed to minimize the system-wide travel cost. This is beneficial in the society's point of view, assuming each agent accepts the system's assignment. This is, however, a strong assumption considering agents might form their own ridesharing groups if they find doing so is more of their own interest.

Recall that the agents of ridesharing system participate in this system in the hope of saving travel cost. So it is up to the ridesharing service provider to decide how the travel cost would be shared among customers after a ridesharing plan is proposed and accepted by the customers. This is a non-trivial task because if the agents find the cost allocation scheme unfair, they may leave the system and form their own ridesharing group in the long run. This fair cost-allocation situation is critical to the sustainability of a ridesharing system and thus is the motivation of the study in this paper.

The ridesharing cost allocation problem is modeled as a cooperative game. Cooperative game theory, due to its close relation to combinatorial optimization, has drawn significant attention of the operations research community. Since its introduction by von Neumannn and Morgenstern (1944), cooperative game theory has developed several solution concepts that aim to resolve the benefits (cost) allocation issues among cooperative players. In this paper, we are primarily concerned with a particular cost allocation solution concept - the nucleolus. The nucleolus of a cooperative game has several nice properties. Intuitively, it is a solution to the cost allocation problem that minimizes the maximal dissatisfaction among the customers.

The concept of nucleolus was first suggested by Schmeidler (1969) and since then was developed by Shapley (1967) and Maschler et al. (1979). Although the nucleolus has several game theoretic virtues, the computation of nucleolus is very difficult. In fact, for a n-player game, as the size of the characteristic function grows exponentially with the number of players, any enumeration algorithm that computes the nucleolus that requires the entire information of the characteristic function takes $O(2^n)$ time, assuming the characteristic function is readily available. Moreover, as will be shown in later section, finding the characteristic function value of ridesharing game involves solving an optimization problem related to the Traveling Salesman Problem (TSP), which is NP-hard itself. This means the computation of the nucleolus of ridesharing game can easily become intractable and more efficient algorithm needs to be developed.

In this paper, we propose a nucleolus-finding algorithm for the ridesharing game by successively solving a number of linear and integer programs. The linear programming (LP) problem for nucleolus calculation was first studied by Kopelowitz (1967) and stimulated several LP-based algorithms for nucleolus computation. Dragan (1981) suggested an algorithm for computing the nucleolus by generating the minimal balanced sets of the player set. Our nucleolus-finding procedure combine the LP-based algorithm with the constraint generation framework proposed in Hallefjord et al. (1995), such that the explicit information of the characteristic function of a ridesharing coalition is only computed when it is "dissatisfied". In this way the computational burden is significantly reduced.

Note that the constraint generation approach was first proposed in Gilmore and Gomory (1961) and was successfully applied to solving the cutting stock problem. Utilizing a similar idea, Göthe-Lundgren et al. (1996) studied the basic vehicle routing game (VRG) in which a fleet with homogeneous capacity are available. The authors analyzed the properties of this game and proposed a nucleolus-finding procedure based on coalition generation. Engevall et al. (2004) generalized the model of Göthe-Lundgren et al. (1996) to consider vehicles with heterogeneous capacities and studied a real-world case based on their model.

The recent prosperity of ridesharing services has spurred a growing attention from the research community. There are some game-theoretic studies, either from a cooperative or non-cooperative perspective, that focus on the mechanism and stability of ridesharing recommendations. Shen et al. (2016) proposed an online ridesharing mechanism that satisfy ex-post incentive compatibility, individual rationality, and budget-balance in a non-cooperative context. Zhang et al. (2016) designed a double auction based discounted trade reduction mechanism for dynamic ridesharing pricing that is individual rational, incentive compatible, budget balancing and has a larger trading volume. Gopalakrishnan et al. (2016) studied the costs and benefits of dynamic ridesharing by introducing the notion of sequential individual rationality and sequential fairness. Wang et al. (2018) introduced the concept of stable matches, understanding and addressing the gap and trade-off between the wholistic optimal matchings of and the optimal matchings from individual rider's perspective. From a cooperative game theoretic perspective, Bistaffa et al. (2017, 2015a, 2015b) tackled the coalition formation and payment allocation for the so-called social ridesharing problem, in which the feasible coalitions of a set of commuters are restricted by a social network represented by a graph. The authors focused on the solution concept of kernel-stable payments.

To sum up, this paper advances the state of the art as the following. In contrast with the previous related work in which the existence of several restrictions (fixed driver/rider roles, possible coalitions limited by social network, etc.) significantly reduces the search space, our model has no such restrictions and therefore is more general. Second, to our best knowledge, our work is the first attempt that computes the nucleolus which is provably the most stable payment allocation scheme, compared to other concepts such as kernel and core, in the context of ridesharing cost allocation.

This paper is organized as follows. In Section 2, a formulation of the ridesharing cost allocation problem is developed from a game theory perspective and the properties of the characteristic function are analyzed. Section 3 discusses the fairness issues in the ridesharing game regarding the core and the nucleolus. A coalition generation scheme is then developed to compute the nucleolus. The constraint generation subproblem is explicitly formulated by a mathematical formulation related to the ridesharing optimization problem. In Section 4, numerical experiments are conducted and the performance of the proposed nucleolus procedure is evaluated. Finally, conclusions and future research ideas are presented in Section 5.

2. Ridesharing optimization problem from a game theory perspective

Consider a set of ridesharing participants and denote it by N. Each participant wants to travel from her origin s_i to her destination t_i. Each participant can potentially be the driver. Denote the capacity of a vehicle by Q. Consider the subsets of participants that do not exceed the vehicle capacity, i.e. $|S| \leq Q$. For each such participant subset s, assume the *feasible* route with minimum cost is known. Here by feasible it means the following conditions are met

1. the route r starts from an agent $d's$ origin and ends at his destination.
2. Let $s_{-d} = s \backslash \{d\}$. For every agent $i \in s_{-d}$, s_i precedes t_i in r.

Denote by c_r the cost of such a feasible route and by R the set of feasible routes with minimal cost. Let $a_{ir} = 1$ if participant i (both s_i and t_i) is served by route r and 0 otherwise. The ridesharing optimization problem (RSP) can be formulated as

$$\text{(RSP)} \quad z = \min \sum_{r \in R} c_r x_r \tag{1}$$

$$\text{s.t.} \quad \sum_{r \in R} a_{ir} x_r = 1, \quad i \in N \tag{2}$$

$$x_r \in \{0, 1\}, r \in R \tag{3}$$

In the formulation $x_r = 1$ if feasible route r is selected and 0 otherwise. Constraints (2) guarantee that each participant is covered by exactly one route. Note that the coefficient c_r in the objective function is obtained by finding the minimal cost route that covers the participants for which $a_{ir} = 1$, that is, by finding the solution to the corresponding TSP with precedence constraints.

It is noted that this formulation is characterized by its large number of columns. Therefore, this formulation is practically solvable by a column generation solution method. Similar approaches were successfully applied to the vehicle routing problems (VRP) (Balinski and Quandt, 1964; Desrochers et al., 1992, see). When we solve the RSP with a column generation approach, it is of our interest to reduce the number of columns. We show this is possible as follows.

We first introduce the definition of the *profitable* ridesharing route.

Definition 1 (profitability)**.** Denote by $r(S)$ the corresponding minimum cost feasible route of participant subset S. $r(S)$ is *non-profitable* if there exists two non-empty subsets $S_1 \cup S_2 = S, S_1 \cap S_2 = \emptyset$ such that $c_{r(S)} > c_{r(S_1)} + c_{r(S_2)}$. A route is defined profitable otherwise.

Intuitively, a shared-ride route becomes non-profitable if by ridesharing the participants end up spending more money on the transportation cost.

The following proposition shows that we only need to consider a subset of the columns when solving RSP.

Proposition 1. *Let $X = \{x_{r_1}, x_{r_2}, \ldots, x_{r_m}\}$ be an optimal solution to RSP, i.e. $x_{r_i} = 1, i = 1, \ldots, m$. Then $r_i, \forall i = 1, \ldots, m$ must be a profitable route.*

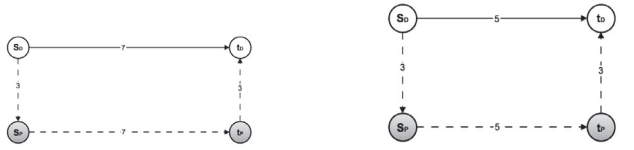
Proof. Proof by contradiction. Let $X = \{x_{r_1}, x_{r_2}, \ldots, x_{r_m}\}$ be an optimal solution to RSP. Suppose there exists i^* such that r_{i^*} is a non-profitable route. Let S^* e the corresponding participants that are covered by this route. Then by definition there must be two non empty subsets $S_1^* \cup S_2^* = S^*, S_1^* \cap S_2^* = \emptyset$ such that $c_{r(S^*)} > c_{r(S_1^*)} + c_{r(S_2^*)}$. Since all the customers that are covered by r_{i^*} are also covered by $r(S_1^*)$ and $r(S_2^*)$, we can get a new feasible solution X' to RSP by substituting $x_{r_{i^*}} = 1$ with $x_{r(S_1^*)} = 1$, $x_{r(S_2^*)} = 1$ and $x_{r_{i^*}} = 0$ while keep all the other x variables unchanged. This feasible solution has a strictly less cost than X. Contradiction. □

From a game theory perspective, we denote each ridesharing participant, $i \in N$, by a *player* and each subset of participants, $S \subseteq N$, by a *coalition*.

The ridesharing cost allocation problem is the problem of finding a "fair" cost allocation scheme for the ridesharing optimization problem (RSP).

A cooperative ridesharing game is defined by specifying a travel cost for each coalition. The game is defined by a ridesharing group S, and a *characteristic function* $c(S) : 2^S \to \mathbb{R}$ from the set of all possible coalitions (sub-ridesharing group) of players in S to a set of payment schemes satisfying $c(\emptyset)=0$. Here 2^S denotes the power set of S. In the context of RSP game the characteristic function can be seen as the travel cost occurring if coalition $S \subseteq N$ is formed. Each coalition can be defined by a binary vector s as

$$s_i = \begin{cases} 1, & \text{if customer } i \text{ is a member of the coalition,} \\ 0, & \text{otherwise,} \end{cases}$$



(a) Profitable Coalition, $c(S) = 13$ (b) Nonprofitable Coalition, $c(S) = 11$

Fig. 1. Profitable vs. nonprofitable coalition.

Define c as the objective value of a mathematical program. For all coalitions $S \subseteq N$, $S \neq \emptyset$, let $c(S)$ be the solution to the following mathematical program

$$c(S) = \min \sum_{r \in R} c_r x_r \tag{4}$$

$$\text{s.t.} \sum_{r \in R} a_{ir} x_r = s_i, \quad i \in N \tag{5}$$

$$x_r \in \{0, 1\}, \quad r \in R \tag{6}$$

Intuitively, $c(S)$ represents the cost of an optimal route that covers the players in S, i.e. the players for which $s_i = 1$.

When studying a cooperative game, it is of great interest to study the properties of its characteristic function. Assuming that the singleton coalitions have a positive cost, we prove that the characteristic function of the RSP game is *monotonic* and *subadditive*. Interested readers can find the proof in the Appendix. It is noteworthy that subadditivity implies larger coalitions save more. So it is always beneficial to include more people to participant in ridesharing and this is a desirable property of the RSP game.

Denote a coalition S whose cardinality is smaller than the vehicle capacity ($|S| \leq Q$) as a *feasible coalition* and otherwise as an *infeasible coalition*. Denote by $\mathbb{S}$ the set of feasible coalitions. Then we get

$$c(S) = c_r, \quad \forall r \in R \text{ and } S \in \mathbb{S} \text{ such that } a_{ir} = s_i, i \in N.$$

In addition, denote a coalition such that $\Sigma_{i \in S} c(i) \geq c(S)$ by a *profitable coalition* and otherwise by a *nonprofitable coalition*. Fig. 1 gives an example where forming a coalition will not always produce desirable results: instead of reducing total transportation cost as Fig. 1(a) and (b) actually increases the total cost, meaning it doesn't make much sense to form such a coalition. In this case the players are better off on their own. Note that the profitability of forming a coalition in a large extent depends on the relative geo-locations of the players.

3. Fairness and stability in RSP game

3.1. The core and the nucleolus

3.1.1. The core

Let y_i be the cost allocated to agent i, $i \in N$. The *core* of the RSP game is the set of the cost allocation plans y, such that

$$\sum_{i \in N} y_i = c(N) \tag{7}$$

$$\sum_{i \in S} y_i \leq c(S), \forall S \subset N. \tag{8}$$

The above inequalities can be interpreted as no single player or coalition should make a payment that is greater than their cost on their own. A cost allocation scheme that is in the core is a good allocation as no coalition has the incentive to leave the grand coalition. An inequality in (8) is called a *core defining inequality* (CDI).

It is observed that the number of CDIs is in the scale of $O(2^N)$. As will be shown in later sections, in order to find the core and the nucleolus efficiently, it is important and of our great interest to reduce the number of CDIs. This is possible through the following propositions.

Proposition 2. *Any CDI with a nonprofitable coalition S, $S \neq N$, is not needed in (8).*

Proof. Consider any nonprofitable coalition $\hat{S}$, $\hat{S} \neq N$. Denote by $\{1, 2, \ldots, m\}$ the players in $\hat{S}$. By definition of nonprofitable coalition we have $\sum_{i\in\hat{S}} c(i) < C(\hat{S})$. Note that all individual players are also singleton coalitions. It follows that

$$y_i \leq c(i), \forall i \in \hat{S} \quad \Longrightarrow \quad \sum_{i\in\hat{S}} y_i \leq \sum_{i\in\hat{S}} c(i) < c(\hat{S}). \qquad \square$$

Proposition 3. *For a RSP game with non-empty core, any CDI with an infeasible coalition S, S≠N, is not needed in (8).*

Proof. Similar to Göthe-Lundgren et al. (1996), let $\hat{S}$, $\hat{S} \neq N$ be an infeasible coalition. Denote by $\{r_1, \ldots, r_m\}$ the corresponding optimal routes and $\{s_1, \ldots, s_m\}$ the disjoint feasible coalitions corresponding to the optimal routes. Since we have $\sum_{j=1}^{m} \sum_{i\in S_j} y_i = \sum_{i\in\hat{S}} y_i$ and $\sum_{j=1}^{m} c(S_j) = c(\hat{S})$, then we have the following

$$\sum_{i\in S_j} y_i \leq c(S_j), \forall j = 1, \ldots, m \quad \Longrightarrow \quad \sum_{i\in\hat{S}} y_i \leq c(\hat{S}). \qquad \square$$

From Proposition 2 and Proposition 3 we have

$$C = \left\{ y \mid \sum_{i\in S} y_i \leq c(S), S \in \mathbb{S}; \quad \sum_{i\in N} y_i = c(N) \right\}.$$

Thus, when the core of the RSP game is non-empty, the only characteristic function values of our interests are those corresponding to profitable and feasible coalitions. This, as will be stated in later sections, reduces the size of the coalition-generating subproblem dramatically. Note that the calculation of $c(S)$ for a coalition S is equivalent to solving the corresponding traveling salesman problem with pick-up and drop-off constraints (TSPPD) for the customers for which $s_i = 1$.

3.1.2. The nucleolus

In a cooperative ridesharing game, in which players share travel cost allocations are the payments each player need to pay. That is to say, cooperative ridesharing game (RSP game) is a cost game.

Let $c : 2^S \to \mathbb{R}$ denote the cost characteristic function of a cooperative ridesharing game. Then the function gives the amount of collective cost a group of players need to pay through forming a coalition. In an RSP game, the *excess* of y for a coalition $S \subseteq N$ is defined as $e(y, S) = c(S) - \sum_{i\in S} y_i$ and measures the amount of cost-saving of coalition S in the allocation y, compared to $c(S)$. Note that when $e(y, S)$ is negative, it means the sum of the cost of S in the allocation y must exceed $c(S)$. Thus $e(y, S)$ measures the dissatisfaction of S in the allocation y. Recall that the core is defined as the set of imputations such that $c(S) \geq \Sigma_{i\in S} y_i$ for all coalitions S, then we have that an imputation y is in the core if and only if all its excesses are positive or zero. Denote by $\theta(y) \in \mathbb{R}^{2^N}$ the excess vector of y whose elements $c(S) - \sum_{i\in S} y_i$ are arranged in non-decreasing order, that is, $\theta_i(y) \leq \theta_j(y)$, $\forall i < j$. Then a cost allocation vector y is in the core if and only if it is efficient and $\theta_1(y) \geq 0$. Consider the lexicographic ordering of excess vectors: for two payment vectors x, y, we say $\theta(x)$ is lexicographically greater than $\theta(y)$ if $\exists k$ such that $\theta_i(x) = \theta_i(y)$, $\forall i < k$ and $\theta_k(x) > \theta_k(y)$. Denote this ordering by $\theta(x) \succ \theta(y)$.

Definition 2 (Nucleolus)**.** The nucleolus of a RSP game is the lexicographically maximal imputation. Denote the nucleolus by y and let $\bar{Y}$ be the set of imputations, then we have

$$\theta(y) \succeq \theta(y'), \quad \forall y' \in \bar{Y} \backslash y.$$

Intuitively, the nucleolus is by definition the fairest cost allocation plan because it minimizes the maximal dissatisfaction of all the coalitions in the ridesharing system. Remember that a coalition is a subset of all the ridesharing groups. As a result, on the condition that the core is nonempty, the nucleolus is the *center* of the core, and a ridesharing system that implements the nucleolus as the cost sharing plan is provably the most stable. It is of our interest to investigate the nonemptiness of the core of RSP game because the definitions of nucleolus and the core are related. In fact, the following example shows that the core of RSP game may be empty.

The transportation costs of three players 1,2,3 are given in Fig. 2. Assuming that each player's vehicle has a capacity of one extra passenger seats, the characteristic function of this 3-player game is then defined by $c(\{1\}) = c(\{2\}) = c(\{3\}) = 5$, $c\{1, 2\} = c(\{2, 3\}) = 7$ (e.g., $1 - 2 - 2' - 1'$), $c(\{1, 3\}) = 9$ (e.g., $1 - 3 - 3' - 1'$) and $c(\{1, 2, 3\}) = 12$. The optimal route configuration is, for example, $2 - 3 - 3' - 2'$ and $1 - 1'$. We show the calculation of nucleolus of this example in Table 1.

As an initial guess, we try (4, 4, 4). In Table 1, we find that the minimum excess happens at coalition (1, 2) and (2, 3). These are the coalitions with maximum dissatisfaction. To improve this, we must increase both y_1 and y_3. This involves decreasing y_2, and will decrease the excess for (1, 3) at the same rate. Also note that player 1 and player 3 have symmetrical roles in this game, thus we can conclude that the best scenario occurs when the excesses for (1, 2), (2, 3) and (1, 3) are all equal. Solving the equations,

$$y_3 - 5 = y_1 - 5 = y_2 - 3$$
$$y_1 + y_2 + y_3 = 12,$$

we find the nucleolus of this game is $y = (4\frac{2}{3}, 2\frac{2}{3}, 4\frac{2}{3})$ and $C = \emptyset$ (note that $c(1, 2) < y_1 + y_2$).

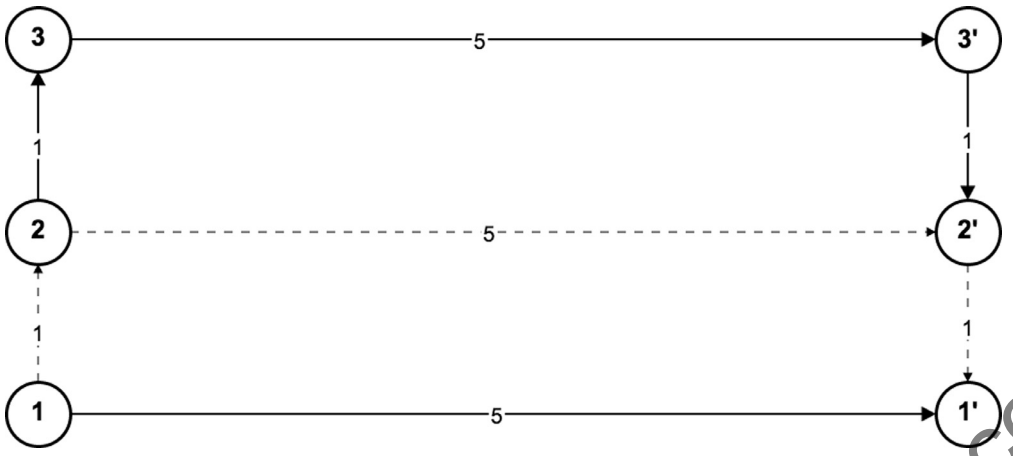


Fig. 2. A three player example.

Table 1
Calculation of nucleolus - empty core.

S	c(S)	e(y, S)	(4, 4, 4)	$(4\frac{2}{3}, 2\frac{2}{3}, 4\frac{2}{3})$
1	5	$5 - y_1$	1	$\frac{1}{3}$
2	5	$5 - y_2$	1	$2\frac{1}{3}$
3	5	$5 - y_3$	1	$\frac{1}{3}$
1 and 2	7	$7 - y_1 - y_2 = y_3 - 5$	−1	$-\frac{1}{3}$
2 and 3	7	$7 - y_2 - y_3 = y_1 - 5$	−1	$-\frac{1}{3}$
1 and 3	9	$9 - y_1 - y_3 = y_2 - 3$	1	$-\frac{1}{3}$

Table 2
Calculation of nucleolus - nonempty core.

S	c(S)	e(y, S)	(3, 3, 3)	$(\frac{11}{3}, \frac{5}{3}, \frac{11}{3})$
1	5	$5 - y_1$	2	4/3
2	5	$5 - y_2$	2	10/3
3	5	$5 - y_3$	2	4/3
1 and 2	7	$7 - y_1 - y_2 = y_3 - 2$	1	5/3
2 and 3	7	$7 - y_2 - y_3 = y_1 - 2$	1	5/3
1 and 3	9	$9 - y_1 - y_3 = y_2$	3	5/3

Interestingly, if we increase the capacity of the vehicles to two extra seat, then we obtain a game with a nonempty core. In this case, the characteristic function is defined in the following fashion. For the singleton coalitions and the 2-coalitions, characteristic function values remain the same. However, for the grand coalition $c(\{1, 2, 3\}) = 9$. The optimal route configuration is, for example, $1 - 2 - 3 - 3' - 2' - 1'$. The calculation of nucleolus of this example is shown in Table 2.

Initially, we try (3, 3, 3). In Table 2, we find that the minimum excess happens at coalition (1, 2) and (2, 3). These are the coalitions with maximum dissatisfaction. To improve this, we must increase both y_1 and y_3. This involves decreasing y_2, and will decrease the excess for (1, 3) at the same rate. Also note that player 1 and player 3 have symmetrical roles in this game, therefore we can conclude that the best scenario we can achieve happens when the excesses for (1, 2), (2, 3) and (1, 3) are all equal. Solving the equations,

$$y_3 - 2 = y_1 - 2 = y_2$$
$$y_1 + y_2 + y_3 = 9,$$

we find the nucleolus of this game is $y = (11/3, 5/3, 11/3)$. Here, $C \neq \emptyset$ and the nucleolus of this game is the center of the core.

3.2. An algorithm to find the nucleolus

As discussed before, finding a nucleolus will ensure the ridesharing system implements the provably fairest cost allocation plan to the users, which also ensures the stability of the system. A procedure to calculate the nucleolus is developed in this paper and with sufficient amount of computational resources, a system that is able to handle realistically large-scale ridesharing service system can be developed. In a nutshell, the proposed algorithm starts with the least core, and contin-

ues with lexicographic optimization through iterating between the master problem and the subproblems. We detail this procedure below.

3.2.1. The master problem

Since the nucleolus is in the least core, which minimizes the maximal dissatisfaction, we start with the solution to the following maximin problem

$$\max_{y\in Y}\min_{\forall S\subset N}\left(c(S)-\sum_{i\in S}y_i\right),$$

which can be transformed to a linear program

$$(P^1)\quad \max\quad w \tag{9}$$

$$\text{s.t.}\quad w\le c(S)-\sum_{i\in S}y_i,\quad S\in\mathbb{S} \tag{10}$$

$$\sum_{i\in N}y_i=c(N). \tag{11}$$

Notice that the LP program has $O(|\mathbb{S}|)$ constraints, and computing $c(S), S\in\mathbb{S}$ involves solving the corresponding TSPPD. So the LP program can easily become intractable. We therefore approach this problem with a constraint generation procedure. Hallefjord et al. (1995) has suggested such an approach for linear programming games. Göthe-Lundgren et al. (1996) has used a similar approach to solve the vehicle routing problem (VRP) game.

Since before searching for the nucleolus, we should already know the solution to the corresponding RSP, thus the optimal route configuration, we can start (P_1) with the coalitions corresponding to the optimal routes. Besides, the singleton coalitions' cost values are readily available. Denote by $\Omega\in\mathbb{S}$ the available coalitions, then (P_1) can be replaced by the following relaxed problem

$$(P_M^1)\quad \max\quad w \tag{12}$$

$$\text{s.t.}\quad w\le c(S)-\sum_{i\in S}y_i,\quad S\in\Omega \tag{13}$$

$$\sum_{i\in N}y_i=c(N). \tag{14}$$

If the solution to (P_M^1) is unique, let it be (y^*, w^*), i.e. $\theta_1(y^*)>\theta_1(y'), \forall y'\in Y\backslash\{y^*\}$, then y^* is the nucleolus of the game. If the solution to (P_M^1) is not unique, we continue to find the greatest $\theta_2(y)$ among the $y\in Y$ with $\theta_1(y)=w^*$. We continue this process until the solution to the linear program is unique. At stage t the master LP problem to be solved is

$$(P_M^t)\quad \max\quad w_t \tag{15}$$

$$\text{s.t.}\quad w_t\le c(S)-\sum_{i\in S}y_i,\quad S\in\mathbb{S}\backslash\bigcup_{\tau=1}^{t-1}\Gamma_\tau, \tag{16}$$

$$w_\tau=c(S)-\sum_{i\in S}y_i,\quad S\in\Gamma_\tau,\tau=1,\dots,t-1, \tag{17}$$

$$\sum_{i\in N}y_i=c(N). \tag{18}$$

The solution to the last program in this series is the nucleolus of this game. Let $\Pi_{t,S}$ be the dual variable corresponding to constraint $w\le c(S)-\sum_{i\in S}y_i$. Let Γ_t denote the set of coalitions whose corresponding constraints are *binding*, that is, $\Gamma_t=\{S\in\mathbb{S}\bigcup_{\tau=1}^{t-1}\Gamma_\tau \mid \Pi_{t,S}^*>0\}$.

The essential idea of constraint generation approach is trying to find the nucleolus with explicit information of only a small portion of the entire coalition set. This goal is realized by finding the most violated constraint that is not yet included in Ω via a subproblem after the master problem is solved at each stage. Denote the optimal solution to (P_M^1) by

$y^* = (y_1^*, \ldots, y_n^*)$. The constraint that is violated the most, aka the most unhappy coalition given the cost allocation scheme y^*, is obtained through solving the following subproblem

$$(P_S) \min_{S \in \mathbb{S} \setminus \Omega} c(S) - \sum_{i \in S} y_i^* - w^*$$

This nucleolus-finding procedure for a ridesharing game is developed based on the theories and techniques proposed in Dragan (1981) and Kopelowitz (1967) and a general constraint generation framework proposed in Hallefjord et al. (1995). The pseudocode of this procedure is given in Algorithm 1. First, at stage t the master LP problem P_M^t is solved and both

Algorithm 1: Procedure of finding the nucleolus of a cooperative ridesharing game.

input : Geolocations of customers
output: Nucleolus of the ridesharing game

```
t := 1 ;
Ω_INEQ ;
STOP := false ;
while !STOP do
    Solve a master problem P_M^t ;
    Solve a subproblem P_S ;
    c* = min_S c(S) − Σ_{i∈S} y_i^* − w* ;
    s* = arg min c(S) − Σ_{i∈S} y_i^* − w* ;
    if c* ≤ 0 then
        SP.addCut(s*) ;
        Ω_INEQ := Ω_INEQ ∪ {s*} ;
    end if
    else
        STOP := true ;
        for every active and binding constraint s do
            STOP := false ;
            Ω_INEQ := Ω_INEQ \ {s} ;
            Ω_EQ := Ω_EQ ∪ {s} ;
        end for
        t := t + 1 ;
    end if
end while
```

the primal and dual solutions are returned. Second, a subproblem P_S is solved and the least satisfied constraint (s^*) that is not yet included is identified. If $c^* \leq 0$, then we include s^* in Ω_{INEQ} and resolve P_M^t with newly included constraint $w_t \leq c(s^*) - \sum_{i \in s^*} y_i^*$. This stage iterates between the master problem and the subproblem until no coalition violates the rationality constraints of the master problem (i.e. $c^* \geq 0$). When this is achieved, we identify the active and binding constraints, reformulate the master problem (modify Ω_{INEQ} and Ω_{EQ}) and proceed to the next stage ($t = t + 1$). This process continues until the solution y^* o the master problem is unique. And this last solution is the nucleolus of the RSP game. Note that in the procedure SP.addCut(s^*), a cut of the type of inequality (42) is added to the subproblem to prevent the duplication of row associated with coalition s^*.

The following two subsections will discuss two formulations of subproblem P_S. In particular, we will discuss how the second formulation reduces the complexity of the overall procedure by utilizing the aforementioned propositions.

3.3. Coalition generation subproblem – general

Recall in the nucleolus-finding scheme described in Algorithm 1, it involves finding the most violated constraint in the subproblem. This is equivalent to finding the "least satisfied" subset of customers with a given allocation proposal. A general formulation of the subproblem is thus

$$(P_S^0) \min_{S \in \mathbb{S} \setminus \Omega} c(S) - \sum_{i \in S} y_i^* - w^* \tag{19}$$

$$\sum_{\{i | s_i^j = 0\}} s_i + \sum_{\{i | s_i^j = 1\}} (1 - s_i) \geq 1, \quad j \mid S_j \in \Omega \tag{20}$$

$$s_i \in \{0, 1\}, i \in N \tag{21}$$

Constraints (20) are preventing the re-generation of constraints.

Note that calculating $c(S)$ is equivalent to solving the RSP model for customers $i \in S$, i.e. those $s_i = 1$. This implies that we can formulate the subproblem P_S^0 explicitly. Denote by $G = (V, E)$ the graph of the RSP game with vertex set $V = V_O \cup V_D \cup \{0\}$ and edge set $E = \{(i, j) \mid i, j \in V, i \neq j\}$. Here vertex 0 is the "dummy" depot such that any edge incident with it has a cost of 0. $V_O(V_D)$ is the origin (destination) vertex set of players in N. Each player is associated with a profit (prize) equal to y_i^*. The subproblem of the constraint generation procedure is to find a subset of customers in N which maximizes the total prize minus the total cost, while conforming to certain constraints.

$$(P_S^1) \quad \pi = \max \quad \sum_{k \in N} y_k^* s_k - \sum_{i \in V} \sum_{j \in V} c_{ij} \lambda_{ij} + w^* \tag{22}$$

$$\sum_{\{i | s_i^j = 0\}} s_i + \sum_{\{i | s_i^j = 1\}} (1 - s_i) \geq 1, \quad j \mid S_j \in \Omega \tag{23}$$

$$\lambda_{0i} - \lambda_{(i+n)0} = 0, \quad i = 1, 2, \ldots, n \tag{24}$$

$$\sum_{i=0}^{2n} \lambda_{ik} = s_k, \quad k \in N \tag{25}$$

$$\sum_{i=0}^{2n} \lambda_{ki} = s_k, \quad k \in N \tag{26}$$

$$\sum_{i=0}^{2n} \lambda_{i,k+n} = s_k, \quad k \in N \tag{27}$$

$$\sum_{i=0}^{2n} \lambda_{k+n,i} = s_k, \quad k \in N \tag{28}$$

$$u_i - u_j + p\lambda_{ij} \leq p - 1, \quad 1 \leq i \neq j \leq 2n \tag{29}$$

$$u_i < u_{i+n}, \quad i = 1, 2, \ldots, n \tag{30}$$

$$\sum_k y_{ik} = 1, \quad i = 1, \ldots, 2n; \quad k = 1, \ldots, n \tag{31}$$

$$y_{ik} = y_{(i+n)k}, \quad i = 1 \ldots, n; \quad k = 1, \ldots, n \tag{32}$$

$$\lambda_{0i} = y_{ii}, \quad i = 1, \ldots, n \tag{33}$$

$$\lambda_{ij} \leq M_1(1 - z_{ij}), \quad 1 \leq i \neq j \leq 2n \tag{34}$$

$$-M_2 z_{ij} \leq y_{ik} - y_{jk} \leq M_2 z_{ij}, \quad 1 \leq i \neq j \leq 2n; \quad k = 1, \ldots, n \tag{35}$$

$$\lambda_{ij} \in \{0, 1\}, \quad 0 \leq i \neq j \leq 2n \tag{36}$$

$$y_{ik} \in \{0, 1\}, \quad i = 1, \ldots, 2n; \quad k = 1, \ldots, n \tag{37}$$

$$z_{ij} \in \{0, 1\}, \quad 1 \leq i \neq j \leq 2n \tag{38}$$

$$s_k \in \{0,1\}, \quad k \in N \tag{39}$$

This problem can be termed as prize-collecting RSP (see Balas, 1989 for an analogy of TSP and prize-collecting TSP). Constraints (24) make sure that a tour starts at origin and ends at destination. Constraints (25)–(28) are the continuity constraints. Constraints (29) are a group of subtour-elimination constraints (SECs) first proposed by Miller et al. (1960). Here u_i are continuous variables called *node potentials* that indicate the visit order of node i in the tour, while p denotes the maximal number of nodes a vehicle can visit in a tour. This parameter can be used to specify the seat capability of vehicles. Generally p won't exceed 10. Constraints (30) ensure that a customer's origin precedes his destination. Constraints (31) ensure that each node is visited by exactly one vehciel. Constraints (32) make sure that a customer's origin and destination are visited by the same vehicle. The intuitive meaning of constraints (34) and (35) is that if edge (i, j) is selected in the solution then nodes i, j must be served by the same vehicle. This set of constraints serve as a bridge between λ variables (indicating whether an edge is selected) and y variables (indicating whether a node is served by a particular vehicle). Here M_1 and M_2 are large numbers.

3.4. Coalition generation subproblem – non-empty core

Recall that when the RSP game has a non-empty core, the only coalitions that are non-redundant are the feasible coalitions (Proposition 3). Notice that $c(S), S \in \mathbb{S}$ is the minimum cost of a feasible route that covers the origin and destination of all the players in S, that is, those $s_i = 1, i \in N$. This inspires us to formulate the subproblem explicitly. Let $G = (V, E)$ be the graph with vertex set $V = V_O \cup V_D \cup \{0\}$ and edge set $E = \{(i, j) \mid i, j \in V, i \neq j\}$. Here vertex 0 is the "dummy" depot such that any edge incident with it has a cost of 0. $V_O(V_D)$ is the origin (destination) vertex set of players in N. Each player is associated with a profit (prize) equal to y_i^*. The constraint generation subproblem finds a feasible route in G which maximizes the total prize minus cost, while conforming to the ridesharing problem constraints, such as capacity, precedence and maximum travel distance per passenger.

Denote the edge selection variable in the graph by λ. This problem is represented by

$$(P_S^2) \quad \pi = \max \sum_{k\in N} y_k^* s_k - \sum_{i\in V}\sum_{j\in V} c_{ij}\lambda_{ij} + w^* \tag{40}$$

$$\text{s.t.} \quad \sum_{k\in N} s_k \leq Q \tag{41}$$

$$\sum_{\{i|s_i^j=0\}} s_i + \sum_{\{i|s_i^j=1\}} (1 - s_i) \geq 1, \quad j^* | S_j \in \Omega \tag{42}$$

$$\lambda_{0i} - \lambda_{(i+n)0} = 0, \quad i = 1, 2, \ldots, n \tag{43}$$

$$\sum_{k\in N} \lambda_{0k} = 1 \tag{44}$$

$$\sum_{k\in N} \lambda_{k+n,0} = 1 \tag{45}$$

$$\sum_{i=0}^{2n} \lambda_{ik} = s_k, \quad k \in N \tag{46}$$

$$\sum_{i=0}^{2n} \lambda_{ki} = s_k, \quad k \in N \tag{47}$$

$$\sum_{i=0}^{2n} \lambda_{i,k+n} = s_k, \quad k \in N \tag{48}$$

$$\sum_{i=0}^{2n} \lambda_{k+n,i} = s_k, \quad k \in N \tag{49}$$

Table 3
Data and approximate nucleolus of prob10c.

Customer number	Pickup coordinates	Drop-off coordinates	Nucleolus cost
1	(387, 137)	(918, 786)	346.2
2	(595, 4)	(852, 236)	267.1
3	(514, 483)	(9, 481)	627.5
4	(342, 655)	(609, 55)	729.7
5	(715, 887)	(372, 215)	434.3
6	(111 , 687)	(777, 91)	250.4
7	(692, 933)	(203, 173)	97.5
8	(791, 847)	(488, 312)	226.1
9	(702, 762)	(928, 755)	520.8
10	(543, 443)	(90, 700)	92.4

$$u_i - u_j + p\lambda_{ij} \leq p - 1, \quad 1 \leq i \neq j \leq 2n \tag{50}$$

$$u_i < u_{i+n}, \quad i = 1, 2, \ldots, n \tag{51}$$

$$\lambda_{ij} \in \{0, 1\}, \quad i, j \in V, i \neq j \tag{52}$$

$$s_k \in \{0, 1\}, \quad k \in N \tag{53}$$

Constraints (42) put the restriction that a constraint that is generated before is not generated again. Constraints (41) are the capacity constraints and (43) forces a tour to start at origin and end at destination. Constraints (46), (47), (48) and (49) are the flow balancing constraints for each vertex. Constraints (50) are the subtour elimination constraints and (51) are the precedence constraints.

It is noteworthy that the propositions in Section 3.1 help reduce the complexity of Algorithm 1 significantly. This contribution is twofold. First, when the RSP game has a non-empty core, only feasible coalitions need to be considered in the subproblem (P_S^2). Second, although both prize-collecting RSP (P_S^1) and prize-collecting TSPPD (P_S^2) are NP-hard, prize-collecting RSP involves rider partitioning, therefore is much more difficult to solve than prize-collecting TSPPD, which concerns a single route with limited number of stops.

4. Experiments

We have implemented the nucleolus algorithm (with both P_S^1 and P_S^2 as subproblem) in Java with CPLEX 12.6 and the Concert library. In this section, we first show the results of nucleolus algorithm with P_S^2 as subproblem. Because P_S^2 is much easier to solve than P_S^1, and the coalitions needed are much fewer in P_S^2 than in P_S^1, nucleolus algorithm with P_S^2 can not only calculate the nucleolus when the RSP has a non-empty core, but can also be used to find an approximate nucleolus when the corresponding RSP has an empty core. Next, we show a comparison between the nucleolus and the approximate nucleolus, obtained by using the nucleolus algorithm with P_S^1 and P_S^2 as the subproblem, respectively.

4.1. Approximate nucleolus

We report results for two instances of the 10-player problem, which is the largest problem we have solved. As will be shown later, the computational bottleneck is not at the nucleolus algorithm but at solving the corresponding ridesharing optimization problem (RSP). The data set[2] we used in the experiments were selected from Dumitrescu et al. (2010). The origins and destinations of customers were randomly generated in the square [0, 1000] × [0, 1000]. The Euclidean distances were used. Table 3 shows the geographical locations of the players. After solving the RSP MIP model, the optimal ridesharing plan is {1}, {3, 5}, {4, 6, 7}, {8}, {2, 9}, {10}. These, along with other singleton coalitions are used to generate the initial constraints of the master LP problem.

At stage 1 three constraints are generated by solving the subproblem. They correspond to the coalitions of {3, 4, 6}, {4, 6}, and {2, 3, 4, 5, 9}. Active constraints corresponding to coalitions {8}, {10} and {1} are then identified at the end of stage 1. At stage 2, we notice that the solution to the master problem (P_M^2) is unique, thus we find the approximate nucleolus. The approximate nucleolus of this game is listed in the last column of Table 3.

[2] The data sets can be downloaded from http://www.diku.dk/~sropke/

Table 4
Data and approximate nucleolus of prob10d.

Customer number	Pickup coordinates	Drop-off coordinates	Nucleolus cost
1	(60, 742)	(34, 697)	52.0
2	(730, 471)	(390, 845)	444.1
3	(964, 151)	(39, 78)	808.7
4	(336, 763)	(11 , 332)	481.2
5	(330, 593)	(570, 862)	326.9
6	(496, 333)	(88 , 346)	374.1
7	(168 , 403)	(432, 341)	271.2
8	(343, 502)	(525, 846)	173.3
9	(600, 534)	(585, 615)	83.0
10	(18 , 952)	(494, 605)	589.0

Table 5
Nucleolus vs. approximate nucleolus – problem8a.

Customer number	Nucleolus cost	Approximate nucleolus cost
1	442.1	449.9
2	505.0	512.8
3	639.2	636.6
4	409.3	406.7
5	527.6	525.0
6	465.8	463.2
7	228.4	225.8
8	519.4	516.8

Table 6
Nucleolus vs. approximate nucleolus – problem8b.

Customer number	Nucleolus cost	Approximate nucleolus cost
1	366.5	366.5
2	507.1	457.7
3	594.1	617.1
4	387.6	456.7
5	1012.5	1035.6
6	258.1	235.1
7	545.3	571.8
8	235.9	166.8

In total, $10+3+3=16$ out of $2^{10}-2=1022$ constraints are needed to compute the approximate nucleolus, which is only a very small fraction (1.6%).

In our second experiment of prob10d (see Table 4), the optimal ridesharing configuration is {1}, {2, 3, 4, 6}, {7}, {5, 8}, {9} and {10}. At stage 1, four constraints are generated after via solving the subproblem. They correspond to the coalitions of {2, 3, 4, 9}, {3, 6}, {2, 9} and {2, 3, 4, 5, 6}. At stage 2, the master LP problem is found to have a unique solution. This solution is thus the approximate nucleolus of this game. The approximate nucleolus is listed in the last column of Table 4. In total, $(10+2+4)/1022\approx 1.6\%$ constraints were needed to compute the approximate nucleolus.

It is noteworthy to point out that the computational time for both instances is very small (less than 10s), indicating the bottleneck is the optimization solution method.

4.2. Nucleolus vs. approximate nucleolus

In this subsection we conduct two experiments to compare the actual nucleolus and the approximate nucleolus. Finding the actual nucleolus by using Algorithm 1 with P_S^1 as subproblem is a very time-consuming process. In our first example, problem8a, it takes 5 h to find the actual nucleolus. In our second example, problem8b, the time it takes to find the actual nucleolus goes up to 20 h.

The actual nucleolus cost and approximate nucleolus cost are summarized in Tables 5 and 6. As we can see, the solutions obtained by the approximate nucleolus algorithm are a close approximation for the actual nucleolus in both cases.

It is of our interest to see the computational performance of Algorithm 1 using P_S^1 and P_S^2. In problem8a, a total of 193 constraints are generated by P_S^1, comparing to a total of 9 constraints generated by P_S^2. In problem10d, a total of 184 coalitions are generated by P_S^1, comparing to a total of only 9 coalitions generated by P_S^2. Note that the total number of coalitions (not including the empty set and the universal set) is $2^8-2=254$ for problem8a and problem8b. Therefore, Algorithm 1 using P_S^1 generated $193/254=76\%$ and $184/254=72\%$ of the total constraints to find the nucleolus in problem8a and prob-

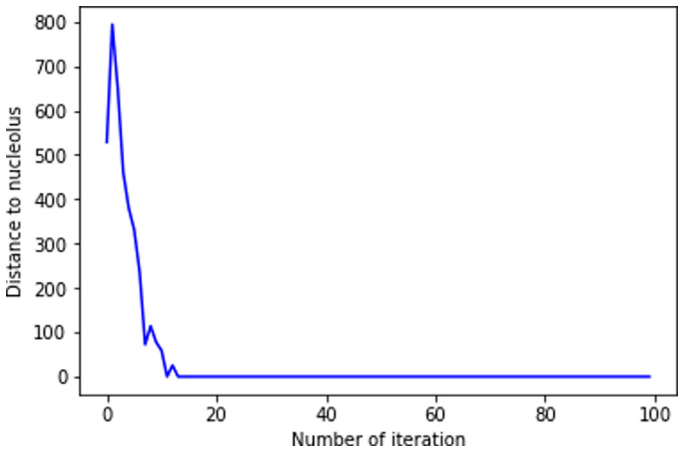


Fig. 3. Solution path – problem8a.

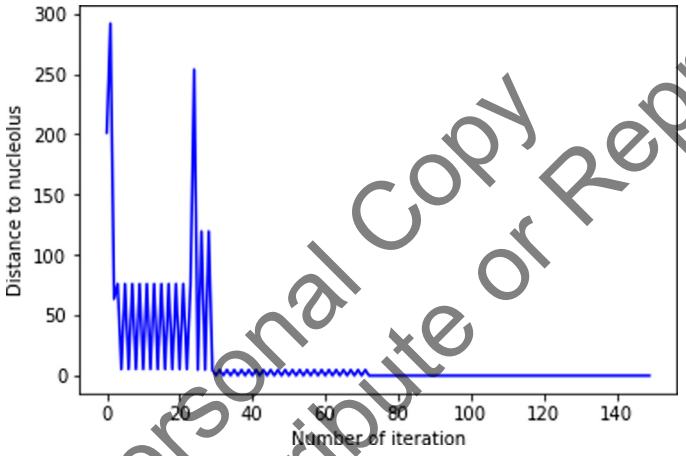


Fig. 4. Solution path – problem8b.

lem8b, respectively. So Algorithm 1 using P_S^1 is more like an enumeration procedure. Thus P_S^2 is much more computationally efficient than P_S^1, since the number of constraints generated by P_S^1 are significantly higher than that of P_S^2. This along with the fact that P_S^1 is much harder to solve than P_S^2 explains the significant time difference between Algorithm 1 using P_S^1 and P_S^2.

In Figs. 3 and 4 we measure the Euclidean distance between the incumbent nucleolus and the actual nucleolus ($\sqrt{\sum_{i\in N}(y_i^* - y_i)^2}$) as the algorithm iterates. These two figures show the solution path of the algorithm in both cases. As can be seen, in both cases, the algorithm found the nucleolus before it stopped. This happened before the 20th iteration in problem8a, and before 75 constraints were generated in problem8b. It means the majority of the running time of this algorithm is consumed after the actual nucleolus is found.

5. Conclusions

We studied an important problem faced by ridesharing service provider: how to allocate cost among ridesharing participants to ensure sustainability and fairness. This fair cost allocation problem was modeled as a cooperative game. A special property of the cooperative ridesharing game is that its characteristic function values are calculated by solving an optimization problem. To better understand this game, we further studied the characteristic function and proved it to be monotone, subadditive, but non-convex (meaning the core can be empty). The most fair allocation plan is identified by the nucleolus of the RSP game. We then proposed an iterative constraint-generation algorithm (Algorithm 1) for calculating it in two situations – the game has an empty core and the game has a non-empty core. In both cases the algorithm utilizes an explicitly formulated MIP as the subproblem to generate constraints. When the game has an empty core, the algorithm uses P_S^1 as the subproblem and becomes an enumeration procedure to find the nucleolus of the game. When the game has a non-empty core, this algorithm uses P_S^2 as the subproblem which utilizes the special properties of the RSP game such that the characteristic function values are computed only when they are needed. Therefore the number of subproblems (an NP-hard

optimization problem) that need to be solved is significantly reduced. Experiments showed that by adopting this algorithm with P_S^2 only a small fraction (1.6%) of the coalition constraints were needed to find the nucleolus. It is also found in the experiments where the emptiness of the RSP game is unclear, the algorithm with P_S^2 can be used to find an approximate nucleolus that is close to the actual one. This indicates that our proposed algorithm is promising in finding nucleolus of dynamic, large-scale RSP game and that since a cooperative game theory modeling approach does not necessarily differentiate drivers and riders explicitly, our model, the mathematical programs and the algorithm proposed in this paper also have very promising application in an autonomous vehicle ridesharing systems. We can see a few interesting and promising future research directions related to this study. First, efficient heuristics for the subproblems can be developed. Second, with such efficient algorithms in hand, we can further investigate the interaction between the vehicle capacity and the cost allocation, which we believe will provide insight on the intrinsic nature of ridesharing game. Besides, we started our study with the motivation of designing mathematical models and algorithms for the most general case of ridesharing scenario with the least amount of assumptions on the coalition etc. However in the real world, the existence of certain situations can significantly simplify the calculation. For instance, in the case some of the ridesharing participants have formed a coalition on their own, we can exploit these structural properties to simplify the coalition generation scheme and expedite the nucleolus calculation.

Acknowledgement

The authors would gratefully acknowledge the kind support from the Dissertation Fellowship of Texas A&M University. We are grateful to anonymous reviewers whose valuable suggestions have led to a considerable improvement in the organization and presentation of this manuscript.

Appendix A

We show the RSP game has the following properties. From here on we denote by $C(\cdot)$ the mathematical program that defines the characteristic function value of $\cdot$, i.e. $c(\cdot)$.

Proposition 4 (Monotonicity)**.** *The characteristic function of the RSP game is monotone, that is,* $c(S) \leq c(T)$, $S \subset T \subset N$.

Proof. Proof by contradiction. Suppose there exists $S \subset T \subset N$ and $c(S) > c(T)$. Let $X = \{x_r \mid r \in R\}$ be an optimal solution to $C(T)$. Let $R_T = \{r_i \mid x_{r_i} = 1\}$.

For $r \in R$, we construct a feasible solution to $C(S)$ in the following manner. Let

$$x_r = \begin{cases} 1, & \text{if } \exists i \in S \text{ such that } a_{ir} = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Let X' be the solution constructed in the above way. Denote by R_S the set of selected routes. Intuitively, we keep those routes in R_T that covers at least one player in S and discard those don't.

It is known that X must satisfy

$$\sum_{r \in R} a_{ir} x_r = 1, \quad i \in T$$

$$\sum_{r \in R} a_{ir} x_r = 0, \quad i \in N - T$$

Because $S \subset T$, then X' must satisfy

$$\sum_{r \in R} a_{ir} x_r = 1, \quad i \in S$$

$$\sum_{r \in R} a_{ir} x_r = 0, \quad i \in T - S$$

$$\sum_{r \in R} a_{ir} x_r = 0, \quad i \in N - T$$

This is equivalent to

$$\sum_{r \in R} a_{ir} x_r = s_i, \quad i \in N$$

So X' is a feasible solution to $C(S)$. In addition, since the cost matrix $\{c_{ij}\}$ is positive, the route cost is also positive. Therefore the cost of X' is less than $c(T)$, which is less than $c(S)$. Note that $c(S)$ is the optimal cost, so this is a contradiction. □

Proposition 5 (Subadditivity)**.** *The characteristic cost function of RSP game is subadditive, i.e.,* $c(S) + c(T) \geq c(S \cup T), S, T \subset N, S \cap T = \emptyset$.

Proof. Let R_S, R_T be the optimal solution to $C(S)$, $C(T)$, respectively. Because S, $T \subset N$ and $S \cap T = \emptyset$, $R_{S,T} = R_S \cup R_T$ must cover all the players in $S \cup T$, i.e. $R_{S,T}$ is a feasible solution to $C(S \cup T)$. Since this solution has an objective value equal to $c(S) + c(T)$, we have $c(S) + c(T) \geq c(S \cup T)$. □

References

Balas, E., 1989. The prize collecting traveling salesman problem. Networks 19 (6), 621–636.

Balinski, M.L., Quandt, R.E., 1964. On an integer program for a delivery problem. Oper. Res. 12 (2), 300–304.

Bistaffa, F., Farinelli, A., Chalkiadakis, G., Ramchurn, S.D., 2015a. Recommending fair payments for large-scale social ridesharing. In: Proceedings of the 9th ACM Conference on Recommender Systems. ACM, New York, NY, USA, pp. 139–146.

Bistaffa, F., Farinelli, A., Chalkiadakis, G., Ramchurn, S.D., 2017. A cooperative game-theoretic approach to the social ridesharing problem. Artif. Intell. 246, 86–117.

Bistaffa, F., Farinelli, A., Ramchurn, S.D., 2015b. Sharing rides with friends: a coalition formation algorithm for ridesharing. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI Press, pp. 608–614.

Desrochers, M., Desrosiers, J., Solomon, M., 1992. A new optimization algorithm for the vehicle routing problem with time windows. Oper. Res. 40 (2), 342–354.

Dragan, I., 1981. A procedure for finding the nucleolus of a cooperativen person game. Zeitschrift für Oper. Res. 25 (5), 119–131.

Dumitrescu, I., Ropke, S., Cordeau, J.-F., Laporte, G., 2010. The traveling salesman problem with pickup and delivery: polyhedral results and a branch-and-cut algorithm. Math. Program. 121 (2), 269–305.

Engevall, S., Göthe-Lundgren, M., Värbrand, P., 2004. The heterogeneous vehicle-routing game. Transp. Sci. 38 (1), 71–85.

Gilmore, P.C., Gomory, R.E., 1961. A linear programming approach to the cutting-stock problem. Oper. Res. 9 (6), 849–859 http://dx.doi.org/10.1287/opre.9.6.849.

Gopalakrishnan, R., Mukherjee, K., Tulabandhula, T., 2016. The costs and benefits of ridesharing: sequential individual rationality and sequential fairness. CoRR abs/1607.07306. URL http://arxiv.org/abs/1607.07306.

Göthe-Lundgren, M., Jörnsten, K., Värbrand, P., 1996. On the nucleolus of the basic vehicle routing game. Math. Program. 72 (1), 83–100.

Hallefjord, Å., Helming, R., Jørnsten, K., 1995. Computing the nucleolus when the characteristic function is given implicitly: a constraint generation approach. Int. J. Game Theory 24 (4), 357–372.

Kopelowitz, A., 1967. Computation of the Kernels of Simple Games and the Nucleolus of N-person Games. Defense Technical Information Center Hebrew Univ Jerusalem (Israel) Dept of Mathematics.

Maschler, M., Peleg, B., Shapley, L.S., 1979. Geometric properties of the kernel, nucleolus, and related solution concepts. Math. Oper. Res. 4 (4), 303–338.

Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer programming formulation of traveling salesman problems. J. ACM 7 (4), 326–329.

von Neumannn, J., Morgenstern, O., 1944. Theory of Games and Economic Behavior. Princeton University Press, Princeton, New Jersey.

Schmeidler, D., 1969. The nucleolus of a characteristic function game. SIAM J. Appl. Math. 17 (6), 1163–1170.

Shapley, L.S., 1967. On balanced sets and cores. Nav. Res. Logist. Q. 14 (4), 453–460.

Shen, W., Lopes, C.V., Crandall, J.W., 2016. An online mechanism for ridesharing in autonomous mobility-on-demand systems. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. AAAI Press, pp. 475–481.

Wang, X., Agatz, N., Erera, A., 2018. Stable matching for dynamic ride-sharing systems. Transp. Sci. 52 (4), 850–867.

Zhang, J., Wen, D., Zeng, S., 2016. A discounted trade reduction mechanism for dynamic ridesharing pricing. IEEE Trans. Intell. Transp. Syst. 17 (6), 1586–1595.