Sequential Online Chore Division for Autonomous Vehicle Convoy Formation

Harel Yedidsion
University of Texas at Austin
harel@cs.utexas.edu

Shani Alkoby
Ariel University
shania@ariel.ac.il

Peter Stone
University of Texas at Austin and Sony AI
pstone@cs.utexas.edu

ABSTRACT

Chore division is a class of fair division problems in which some undesirable "resource" must be shared among a set of participants, with each participant wanting to get as little as possible. Typically the set of participants is fixed and known at the outset. This paper introduces a novel variant, called sequential online chore division (SOCD), in which participants arrive and depart online, while the chore is being performed: both the total number of participants and their arrival/departure times are initially unknown. In SOCD, exactly one agent must be performing the chore at any give time (e.g. keeping lookout), and switching the performer incurs a cost. In this paper, we propose and analyze three mechanisms for SOCD: one centralized mechanism using side payments, and two distributed ones that seek to balance the participants' loads. Analysis and results are presented in a domain motivated by autonomous vehicle convoy formation, where the chore is leading the convoy so that all followers can enjoy reduced wind resistance.

KEYWORDS

Chore Division; Mechanism Design; Multi Agent Coordination; Autonomous Vehicles; Convoy Formation; Platooning

1 INTRODUCTION

Autonomous vehicles are said to form a convoy when vehicles headed in the same direction follow each other in close proximity. This behavior has been proven to save energy, due to the reduction in aerodynamic drag, and is used by migrating bird flocks and in cyclist pelotons.[1] Autonomous vehicle technology offers a safe and accurate way of following with short inter-vehicle distances, even at high speeds, thanks to Vehicle-to-Vehicle (V2V) communication abilities, which alert all the followers immediately whenever any slowing is necessary. Empirical evaluations estimate that a follower can save over 10% of its fuel consumption [19]. However, since the leader sees little or no such gains, choosing the leader of such a convoy raises issues of fairness. Solving these issues is challenging since vehicles can dynamically join and leave the convoy.

This convoy formation problem is representative of an interesting class of previously unexplored fair division problems. Fair division is concerned with dividing a resource between several players, such that each one receives a fair share. One of the most notable fair division problems is cake cutting. Chore division is the dual problem, in which an undesirable task must be fairly divided among agents. Motivated by the details of the convoy formation problem, we define a novel and unique variation of chore division called **S**equential **O**nline **C**hore **D**ivision (SOCD), where agents arrive online, their number is not known a priori, and only one agent can handle the chore at any given time. We investigate how to design SOCD allocation mechanisms that guarantee fairness, maximize efficiency, and are strategy proof (SP) .

The notion of fairness has various interpretations such as proportionality, envy-freeness, and equability. Guaranteeing fairness in dynamic environments, where either resources or participants arrive online, is not always possible for a single game [27], for any interpretation of fairness, while in repeated games, fairness can be guaranteed in expectation.

An SP mechanism is designed to make self-interested agents choose to report their private information truthfully, out of their own self-interest. Enforcing the regulations in a distributed setting is challenging due to the lack of a central entity that can penalize agents or manage a reputation system.

To the best of our knowledge, none of the fair division literature in either game theory or multi-agent systems has considered the SOCD problem as we define it. Furthermore, no previous work has developed a mechanism for profit sharing or load balancing among the vehicles in a convoy.

This paper's contribution is twofold. First, in the area of fair division, it defines the general SOCD model, and second, in the area of convoy formation and platooning it introduces mechanisms that enable spontaneous formation of ad hoc heterogeneous convoys while maintaining fairness and efficiency.

We find that optimal fairness and efficiency can be guaranteed in a centralized setting. In a distributed setting, they can be guaranteed in expectation after participating in multiple games. However, for a single game in a distributed setting, only a weaker form of fairness, i.e., ex-ante proportionality, can be guaranteed with minimal efficiency loss. Furthermore, strategyproofness issues must be addressed for this solution to be applicable in real world scenarios involving self interested agents. We analyze the SP properties of our proposed mechanisms and provide an impossibility result for a single game distributed solution.

Following a review of related work in Section 2, a formal definition of the problem is given in Section 3. Issues of fairness in online and distributed mechanisms are discussed in Section 4. Section 5 models the convoy formation problem as an SOCD problem. Our proposed solution mechanisms are defined and analyzed in Section 6. Conclusions and future work are provided in Section 7.

[1] Another commonly used term for convoy formation is *platooning*. We use these two terms interchangeably.

Appears at the 2nd Games, Agents, and Incentives Workshop (GAIW 2020). Held as part of the Workshops at the 19th International Conference on Autonomous Agents and Multiagent Systems., May 2020, Auckland, New Zealand

https://doi.org/doi

2 RELATED WORK

Fair division is a long-standing and still very active field of research spanning multiple disciplines such as economics, sociology, game-theory and mathematics, and having numerous real-world applications [7, 14]. These applications consider the allocation of both goods and chores. Compared to goods, the literature on fair allocation of chores is relatively under-developed [3].

In dynamic environments, where either agents or goods arrive online, the problem becomes more complex and even the definition of fairness becomes challenging to specify. The following papers define modified notions of fairness in online settings [2, 4, 13, 16]. Similar to these papers , we also define dynamic fairness criteria for online chore division problems in Section 4.

In online cake cutting problems, where agents arrive online, and have heterogeneous valuation functions, it has been proven that no online cake cutting procedure is either proportional, envy-free, or equitable [27]. We adhere to this paper's call to continue investigating online chore division. We also extend its analysis to provide an impossibility result for both envy-free, and for equitable allocations in the single game distributed SOCD problem.

The SOCD model is relevant to applications such as assigning a guard to keep a lookout at a campground where travelers arrive online, or assigning a goal keeper in a drop-in soccer match. In this paper, we focus on convoy formation due to its social impact. Autonomous vehicle technology such as Cooperative Adaptive Cruise Control (CACC) [19] utilizes a combination of sensory data and V2V communication to enable vehicles to cooperate and follow each other closely, accurately, and safely, by synchronizing braking and accelerating.

Safe grouping of vehicles into convoys offers numerous advantages including: increased energy efficiency, improved road capacity, increased traffic safety, and decreased harmful emissions. As a result, major projects are being undertaken around the world in academia, private fleet companies, auto manufacturers, governments, and by individuals, to develop the applicability and regulation of convoys [11, 12, 21, 22, 25, 26].

While the percent of fuel saving varies with vehicle weight, size, speed, and inter-vehicle distance, one finding remains consistent across all studies; the leader's savings is significantly lower than that of the followers [1, 18, 19].

Many of the research projects and experiments in this area are geared towards single fleet convoys, owned and operated by the same organization [5, 6, 15, 17], and as a consequence do not put an emphasis on developing ways to fairly divide the otherwise unequal savings between the leader and the followers. As opposed to single fleet convoys where participants are not self-interested and are all motivated to maximize the social welfare, in heterogeneous ad-hoc convoys, individual participants are interested in maximizing their own energy savings. Consequently, to enable the spontaneous formation of ad-hoc heterogeneous convoys, it is imperative to design a mechanism that ensures a fair division of both benefits and duties among convoy participants.

Furthermore, such a mechanism ought to be SP by preventing any possible manipulation of the system. Classical cake cutting protocols are not necessarily SP, they are often very simple, elegant, and designed so that the agents can easily implement them by following a sequence of natural steps [8, 9]. There are numerous papers that study the problem of fair division; a small number of them also take into account self-interested agents and consequent strategic issues, but these papers focus on fairness and consider a strikingly weak notion of truthfulness [10]. Some papers simply assume that agents are truthful [24].

3 SEQUENTIAL ONLINE CHORE DIVISION

In this section we specify the definitions, assumptions and constraints of the SOCD problem, where a continuous chore must be divided among an a-priori unknown number of agents. The input to the problem is an online stream of agents A, one of which is tasked with performing the chore at any given time. Agent i, denoted as a_i, arrives at time t_arrive_i and leaves at time t_leave_i. During this period of time, $[t\_arrive_i, t\_leave_i]$, a_i is considered to be *available*, and able to perform the chore. When performing the chore, an agent is considered to be *active*. Every available agent, except for the active one, gains a positive utility u_i per unit time. The active agent gains nothing. [2]

In this initial treatment of SOCD, we make the following assumptions, each of which may be relaxed in future work:

- Agents are homogeneous and have the same utility per unit of time $\forall i \quad u_i = u$, the same cost per unit of time as a leader, $\forall i \quad c_i = cl$, and the same valuation for leading each section of the road, $\forall i, j, s \quad V_i(s) = V_j(s)$. We provide a short discussion on the challenges associated with considering heterogeneous agents in Appendix 9.1.
- Agents are rational and risk neutral.
- Each agent know its own arrival and departure times, and communicate this information to the agents that are present when it arrives.
- No two agents have the exact same arrival time.
- There always exists a positive probability that new agents will arrive. In order to keep the described model as general as possible, we do not make any assumptions regarding the nature of the agents' arrival distribution other than the agents being aware of it.

The time frame for an SOCD game, T, starts when there is at least one available agent, and ends where there are none left.

Within one availability period agents may have more than one active period. We denote the m-th time at which a_i is assigned to become *active* by $t\_start^i_m$ and the corresponding m-th time in which it is assigned to stop by $t\_stop^i_m$. M_i is the total number of times that a_i is assigned to become active.

Agent a_i's assigned share of the task, denoted as s_i, is the sum of all the periods that a_i is assigned to be active, $s_i = \sum_{m=1}^{m=M_i} t\_stop^i_m - t\_start^i_m$

We define a switch between an active agent a_i and an available agent a_j, to happen if $t\_stop^i_m = t\_start^j_n \quad m \in M_i, n \in M_j$, for any of their stop and start times respectively.

[2]We considered another representation by which all the agents gain nothing while the active agent gets a negative utility. We find that these two representations are identical in term or analysis, and decided to go with the one that more accurately represents the convoy formation problem. Note that receiving positive utility does not make this into a cake cutting problem since all the agents still prefer to get as little as possible from the divided chore.

A switch results in a cost to the system c. We intentionally leave this definition abstract as some applications may assign the cost to the outgoing agent while others to the incoming agent. In some applications this cost is fixed while in others if can be a function that depends on the state of the system, as is the case in convoy formation switching cost which is detailed in Section 5.

In SOCD there is a one-to-one mapping between times in T and agents in A. Hence, a *feasible solution* is an online assignment of one agent to be active, out of the available agents in A, at any given time $t \in T$.

Efficiency is defined as the total utility gained by all participating agents.

Problem Definition for SOCD:

Given an online input stream of agents, find a mechanism that produces a feasible solution while maximizing both efficiency and fairness.

With the assumption of homogeneity, maximizing efficiency is straightforward; simply reduce the number of switches as much as possible. Maximizing fairness on the other hand is more complex, as explained in Section 4.

Figure 1 illustrates the concepts that define an SOCD problem. It shows the availability periods of three agents as horizontal lines stretching from their arrival times to their respective leaving times. The period of this single game is between a_1's arrival time and a_3's leave time, and is marked with a darker background. Throughout this period, one agent is active, starting with a_1, followed by a_2, and then a_3, as denoted by the dashed red line. Note that this allocation is a feasible one, but is not fair. We will discuss what is considered fair in SOCD in Section 4.

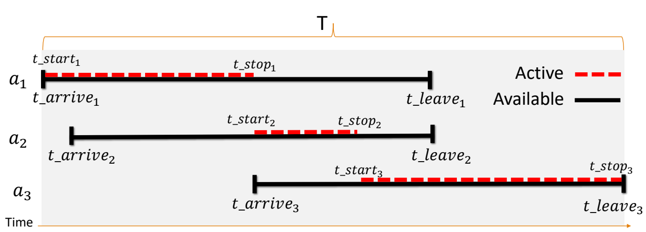

Figure 1: An example of the SOCD model. Three agents' availability periods are displayed on the time axis with agents 1, 2, and 3 acting as the active agent sequentially.

4 FAIRNESS DEFINITIONS AND PROPERTIES

In order to clarify the definition of fairness in the dynamic SOCD model, we begin by revisiting the static definitions of fairness, where X is the chore we aim to divide, n is the number of participants, and s_i is the share allocated to agent i. $V_i(s_i)$ is a_i's valuation of s_i.

The most commonly used **static** types of fairness are:

(1) *Proportionality* - Every agent assigns a value of at most $\frac{1}{n}$ of the total value to their share, i.e., $\forall i\ V_i(s_i) \leq \frac{V_i(X)}{n}$;
(2) *Envy-freeness* - Every agent gets a share that it values at most as much as all other shares i.e., $\forall i, j\ V_i(s_i) \leq V_i(s_j)$;
(3) *Equitability* - All the agents' valuations of their own shares are the same i.e., $\forall i, j\ V_i(s_i) = V_j(s_j)$;

In this paper all the agents are assumed to be homogeneous, having $\forall i, j, s\ V_i(s) = V_j(s)$, and as a result we drop the valuation notation. Additionally, due to the homogeneity assumption, equitability and envy-freeness are actually identical, i.e., if one exists then so does the other. Moreover, equitability implies proportionality (but not the other way around).

In the SOCD model we use the **dynamic** definitions of fairness which consider allocations of agents with overlapping availability periods, and distinguish between earlier arrivals and later arrivals, as in [16].

For a given agent, a_i, we define two notions of proportionality. The first is ex-ante proportionality which takes in account only the agents which are present at t_arrive_i. The second is ex-post proportionality which considers all the agents that were available during a_i's availability period.

In order to define a_i's proportional share in the dynamic SOCD model, it is necessary to separately analyze every segment of its availability period. In each segment there is a different subset of available agents from A. We calculate its proportional share for each segment, and finally sum up all of these shares.

We define EAS^i, as the set of segments, $seg_j^i \in EAS^i$ (j is simply an index of the segments) within a_i's availability period, which are known at t_arrive_i. This set does not consider future arrivals. The first segment, seg_1^i, starts with a_i's arrival, and ends at the t_leave of the first agent among the ones that are present at t_arrive_i. Each consecutive segment ends at the departure of another agent, until t_leave_i.

The ex-ante proportional share for a_i is the sum of known segments' sizes, each divided by the respective number of agents present at that segment, $n\_seg_j^i$, without considering future arrivals. we do not consider future arrivals in the calculation of the ex-ante proportional share since we only have estimates of future arrivals, and thus, in the worst case scenario, the estimate will not reflect the real outcome. In such a case, the sum of the proportional shares will not add up to cover the entire task.

$$ex\_ante\_prop_i = \sum_{j=1}^{j=|EAS^i|} \frac{|seg_j^i|}{n\_seg_j^i} + \frac{c}{u}$$

We add one switching cost to the proportional share of every agent since in the worst case, every agent except for the last one would have to switch at least once in order to divide the chore into n parts. The proportional share is expressed in terms of time. In order to keep the unit of measurement consistent, the switching cost, c, is divided by the utility per unit time, u.

The ex-post proportional share of a_i considers the actual segments, including future arrivals, that occurred during a_i's availability period. We define EPS^i as the set of actual segments within a_i's availability period, $seg_j^i \in EPS^i$. The first segment, seg_1^i, starts with a_i's arrival, and a new segment starts whenever there is a change in the number of available agents, until t_leave_i.

$$ex\_post\_prop_i = \sum_{j=1}^{j=|EPS^i|} \frac{|seg_j^i|}{n\_seg_j^i} + \frac{c}{u}$$

Note that ex-post proportionality can only be calculated in retrospect, while the ex-ante proportionality can be calculated immediately when the agent arrives. If no new agents arrive during a_i's availability period, its ex-ante and ex-post shares are the same. Also note that the ex-post proportional share is also envy-free and equitable since for each segment all the agents get equal shares.

Figure 2 provides an example which highlights the proportional shares of agents a_1, a_2 and a_3. The three horizontal lines represent the availability periods of the agents. Agent a_2's ex-ante proportional share is a half of its availability period which is shared with the existing agent a_1 and the total rest of its availability period. This share is highlighted in purple on top of a_2's timeline. Agent a_2's ex-post proportional share is a half of its availability period which is shared with just a_1, a third of its shared availability with both a_1 and a_3, a half of the time with just a_3, and the rest of the time alone. These shares are highlighted by the orange below a_2's timeline.

There are no new entrants after a_3 and so its ex-ante and ex-post shares are the same. Note that the switching costs are not depicted in this diagram. However, the cost (in terms of time) of one switch is added to the ex-ante proportional share of each agent.

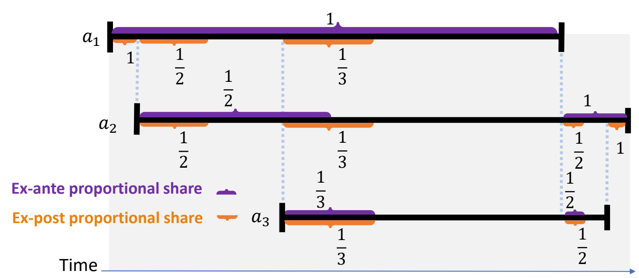

Figure 2: An example of the calculation of the proportional shares.

Theorem 1 outlines the limitations of guaranteeing equitability (and envy-freeness) in SOCD.

THEOREM 1. *In SOCD, no mechanism can guarantee ex-post proportionality for a single game, in a distributed setting.*

PROOF. Since the chore can only be performed sequentially by one agent at a time, any schedule has to assign one agent, a_{last}, to be the last to perform the task, and complete its equitable share s_{last}. By performing the last part of the task, a_{last} would satisfy the other agents' demands for equitability, i.e., doing the same as they did, $\forall i, j\ s_i = s_j$. Specifically having $\forall i\ s_i = s_{last}$. However, because we are dealing with an online arrival stream of agents, it is possible that a new agent, a_{new} would arrive just when agent a_{last} is about to perform its last portion of the task. The new agent would share that portion with a_{last} and would reduce a_{last}'s share. Thus, having $\forall i\ s_{last} < s_i$. Note that if a_{new} does not contribute anything then that would create inequitability with regards to a_{new}. Hence, equitability cannot be guaranteed ex-post (nor envy-freeness). □

5 MODELING CONVOY FORMATION AS AN SOCD PROBLEM

In this section we frame the convoy formation problem as an instance of the SOCD problem, and outline the application-specific assumptions that are relevant to convoy formation.

Initially, in convoy formation, the *chore* to be divided is *leading* the convoy and the *active* agent in the SOCD model is the *leading* agent in the convoy.[3]

Switching and Rotating. Unlike the general definition of switching an active agent in the SOCD model, in the convoy formation setting, we distinguish between two types of switching: rotating and joining/leaving. A *rotation* is when a leading agent finishes its share and moves to the back of the convoy. The rotation process requires the leading agent to switch lanes, slow down to let the convoy pass it, and rejoin from the back. The time it takes to rotate is proportional to n_r - the number of vehicles in the convoy when at the time of rotation. During the rotation process, the rotating agent is effectively out of the convoy and does not enjoy fuel savings. Therefore, this form of switching incurs a cost to the rotating agent. Other forms of switching, happen when a new agent joins the convoy at the front, or when the leader leaves entirely. We assume that these forms of switching do not incur any cost. Formally, in convoy formation, the switching cost, c_{cf}, incurred by outgoing leading agent a_i, when switching with a_j, is defined as:

$$c_{cf}(n_r) = \begin{cases} 0 & \text{if } a_j \text{ joins from the front} \\ 0 & \text{if } a_i \text{ leaves} \\ c \cdot n_r & \text{otherwise} \end{cases}$$

The constant speed assumption. For simplicity, we assume that the convoy is moving at a constant speed and so the time spent in the convoy is proportional to the length of the road traveled. We expect our results to easily generalize as long as speed limits are known for all road segments. This implies that we can refer to u which is the utility per unit time, also as the utility per unit length.

The ability to disconnect followers. The CACC technology that enables convoy formation requires V2V communication to alert of any speed change, thus enabling followers to safely drive at very small inter-vehicle distances, which cannot be achieved by sensors alone. By disabling backward communications, a vehicle can prohibit being followed. This ability comes in handy for preventing manipulations of the mechanisms as described in Section 6.

6 CONVOY FORMATION MECHANISMS

In this section we outline a number of possible convoy formation mechanisms, each geared toward a different set of environmental assumptions. The first mechanism is applicable when there is a central payment transfer system. The second mechanism assumes a distributed setting where there are no payment transfer abilities, and guarantees fairness in expectation through repeated games. The third mechanism aims to guarantee fairness for every single game by rotating the leader. This mechanism also tries to minimize the number of rotations and the resulting loss of efficiency.

[3] Despite evidence showing that the leading agent might get some reward, we assume that it gains nothing as in the general SOCD model since its gain is negligible compared to that of the followers and thus does not meaningfully affect our analysis.

the last case is the interesting one from a technical point of view. Although we are mainly interested in the distributed setting, and although the design and analysis of the centralized mechanism is fairly straightforward, we include it for the sake of completeness.

In addition to fairness and efficiency we will also analyze the SP of the proposed mechanisms. An SP mechanism is one that induces self-interested agents to report their private information truthfully, out of their own self-interest.

Definition 1. *In SOCD, a mechanism is SP if no agent, a_i, can gain any additional utility by falsely reporting its leave time, t_leave_i, over the utility it would gain from truthful reporting, regardless of the other agents' actions.*

6.1 Payment Transfers

Assuming the existence of a central payment transfer system among the agents, the Payment Transfer mechanism (Mechanism 1) assigns only one active agent while the followers transfer a share of their savings to the leader in order to keep fairness. They do so at every time there is a change in the number of agents in the convoy, i.e., at the end of every segment in EPS^i. We divide the chore X into segments where in each segment the number of convoy members is constant (i.e., a segment is a part of the road between two adjacent arrivals/departures). We denote the number of convoy members in segment seg as n_{seg}. For each segment, the amount that a following agent a_i needs to pay the leader of that segment is $p^i_{seg} = \frac{|seg| \cdot u}{n_{seg}}$.

This mechanism allows agents to join the convoy either from the rear or from the front, and does not require any rotations to be made at all, yielding an optimal solution in terms of efficiency. It is also optimal in terms of fairness since the agents equally share the savings for every segment. Each agent pays a cost which is equivalent to the loss of saving it would endure if it had lead for its equitable share, i.e., its ex-post proportional share. Furthermore, the leader gets payments which are equal to the saving it would have gotten if it had only led for its equitable share, as proved in Theorem 2.

Theorem 2. *The Payment Transfer mechanism is efficiently optimal, and equitable.*

Proof. The proof can be found in Appendix 9.2. □

Algorithm 1 The Payment Transfer Mechanism

- Agents can join either from the back or from the front.
- A switch happens when the leader leaves or when a new agent joins from the front.
- For every segment, each following agent transfers p^i_{seg} to the leader of that segment.

Theorem 3. *The Payment Transfer mechanism is SP.*

Proof. The Payment Transfer mechanism utilizes a central control system to bill the agents according to their time in the convoy as followers, or pay them for their time as leaders. The system does not rely at all on the self reporting of the agents and therefore, is not susceptible to any false information reported. Hence, it is SP. □

This mechanism can easily be generalized to any SOCD problem since it does not rely on any feature that is unique to convoy formation. Although we are mainly interested in the distributed setting, and although the design and analysis of this centralized mechanism is fairly straightforward, we include it for the sake of completeness.

6.2 Load-Balancing

The payment transfer mechanism's basic assumption is the availability of some payment system to each one of the convoy's participants. In real-world scenarios however, this is not always the case. Furthermore, even if such a system does exist, one motivation to still pursue a distributed solution that does not rely on an external central system is to avoid failures which can occur in centralized systems when communication with the central server is not reliable.

As a result, in this section we consider a load balancing mechanism aimed to distribute the load equally between all convoy's participants. Note that perfect load balancing is impossible to guarantee in a single game due to possible new entrants dynamically introduced into the convoy, thus disrupting any given scheduling sequence as explained in Theorem 1.

One way to increase equality is to rotate the leader frequently and by doing so, to maintain near-equal load balancing at every point in time. However, this frequent rotation solution has obvious drawbacks. Since rotations incur a cost, frequent rotations lead to reduced efficiency, and in addition to worse road utilization as the convoy frequently occupies two lanes. Consequently, there exists a clear trade-off between efficiency and fairness.

In addition to fairness and efficiency, the distributed mechanisms' design must also consider strategyproofness, so that no agent will be strictly better off by misreporting its departure time, or by leaving the convoy before it had contributed its share of the leading chore. These considerations were not relevant in the central case which does not rely on self reporting, or on balancing the leading load.

6.2.1 *Repeated Game Load Balancing.*

In the distributed setting, we start by analyzing a mechanism where agents do not make any costly rotations. Here there are two possibilities to consider, whether agents join from the front or from the back. In both cases the mechanism offers perfect efficiency as no costly rotations are made at all. In addition, it offers equitability in expectation, after participating in multiple games. However, there is a difference between joining from the back compared to joining from the front. When agents join from the back, the leading agent has no incentive to continue leading the convoy since it will not gain any utility until it leaves. It might even leave the convoy and rejoin it from the back as a follower again, and if all the leaders do that, the convoy's efficiency will deteriorate.

If a leading agent attempts to rotate and rejoin as a follower, it can be prevented by disconnecting communications to it, but this will require collective book-keeping of the participants in the convoy. An easier solution would be to enable new entrants to only join from the front of the convoy. This way the leader can potentially gain if another vehicle enters in front of it, so it has no incentive to leave prematurely. Moreover no vehicle is allowed to join from the back or middle, so there is no better alternative.

The Repeated Game Load Balancing mechanism that we propose (Mechanism 2) demands that each agent will first contribute its share and only then will enjoy the advantages of being a follower. Therefore, any new agent can only join the convoy from the front, i.e., become the leader until someone else joins, or until it leaves.

Algorithm 2 Repeated Game Load Balancing Mechanism

- Agents can join only from the front.
- A switch happens when the leader leaves or when a new agent joins.

This mechanism does not rely on self reporting of destinations and so is not prone to any manipulations from misreporting.

Theorem 4. *The Repeated Game Load Balancing mechanism is SP.*

Proof. The Repeated Game Load Balancing mechanism does not depend on any reports from the participants and thus is not prone to manipulation by false reporting of the destination. Additionally, disabling backward communications prevents the option to join from the rear. □

This mechanism has no fairness guarantees for a single game, only over an infinite time horizon where each agent can have multiple, non-overlapping availability periods. We assume that each agent's different availability periods are independent and identically distributed (i.i.d). We also assume that for each availability period, the arrival rate of other agents is i.i.d.

Theorem 5. *Given the i.i.d assumption of the availability periods and arrival rates, and assuming that each agent participates in an infinite number of convoys, if all agents use the Repeated Game Load Balancing mechanism, every agent a_i will lead for the expected ex-post proportional share.*

Proof. If all the agents use mechanism 2 in repeated games, due to the law of large numbers, and the assumption that availability periods and arrival rates are i.i.d, the average of the results obtained from a large number of convoy formations is equal to the expected value. Each agent has the same probability for leading as all the others, hence, in expectation they will all lead for the same amount of time and follow for the same amount of time in any size of convoys. □

The main advantage of this mechanism is that no switching is performed and thus c is not incurred.

Corollary 1. *Given the i.i.d assumption of the availability periods and arrival rates, the Repeated Game Load Balancing mechanism guarantees equability, i.e. ex-post proportionality, in expectation.*

Note that since this mechanism relies on the convoy being a repeated game, it is not subject to the impossibility result discussed in Theorem 1.

While the Repeated Game Load Balancing mechanism is fair in expectation, any individual participant may end up with a very unfair allocation until it has participated for many times. To study this effect, and in particular how many convoys an agent needs to participate in, in order for its ratio of actual to ex-post proportional lead time to converge to 1, we created a simulation environment in Java.

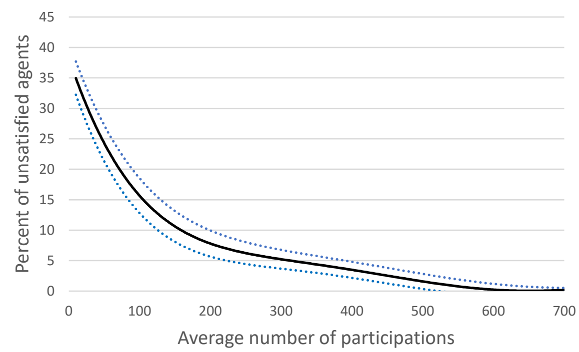

Figure 3: Simulation results for the Repeated Game Load Balancing mechanism.

The experimental setup has 100 vehicles randomly distributed over 100 stations (using a uniform distribution). A single convoy cycles through the stations, and when it reaches a station, the vehicles in that station have a 0.1 probability of joining the convoy. The distance they join for is also randomly generated from a uniform distribution [0, 100]. If multiple vehicles join at the same station, their order is randomized since the last one in becomes the leader. Leaders are replaced when another vehicle joins or when they reach their destination. For every section between consecutive stations, every vehicle in the convoy accumulates $\frac{1}{n}$ to its proportional share, and the leader also accumulates 1 to its actual share. Whenever a vehicle exits, the accumulated actual and ex-post proportional shares are recorded, and added to the list of convoys that the vehicle has participated in. We measure how many convoys it takes for the ratio of actual over ex-post proportional share to converge to 1. Specifically, we measure what is the percentage of vehicles that lead for 10% or more than what they should have (i.e., unsatisfied agents).

The results indicate that it takes at least 200 participations per vehicle on average to get fewer than 10% unsatisfied agents. After participating in 700 convoys on average, there are no vehicles who lead more than 10% of their ex-post proportional share. The results are shown in Figure 3. The dashed lines represent one standard deviation above and below the average.

6.2.2 Single Game Load Balancing. The repeated game mechanism promises fairness in expectation but, in practice, it requires agents to be part of numerous convoys in order to attain fairness. However, some agents may not travel on the highway often and thus may be interested in achieving fair division of the load in each individual game. This motivates our investigation of a mechanism that guarantees fairness for every single game, in a distributed setting. This is the most technically interesting and challenging scenario to analyze.

Creating a distributed single game mechanism that guarantees fairness, efficiency and SP is challenging and in some cases even impossible as we will show in this section. There are five important considerations to take into account when designing such a mechanism.

First, since for a distributed, single SOCD game, equitability (i.e., ex-post proportionality) cannot be guaranteed, as described in Theorem 1, we aim to guarantee ex-ante proportionality, while getting as close as possible to ex-post proportionality. The mechanism we propose requires each agent to lead the convoy for no more than its ex-ante proportional share.

Second, in terms of efficiency, the mechanism should not require more than one rotation per agent, as this is the minimum number of rotations needed to divide the chore into n parts. According to the mechanism we propose, new agents can only join the convoy from the front, and lead until someone else joins in, or until they finish their assigned leading share, after which they rotate to the back of the convoy. This guarantees that agents rotate at most once.

Third, agents can take "illegal" actions to reduce their share of the chore such as leaving the convoy when it is their turn to lead, or entering the convoy from the back. To solve this, we utilize the ability to disconnect followers, and prevent joining from the back. Consequently, leading agents have no incentive to leave before completing their assigned share since they will only get to become followers after they had completed leading their share. Agents who attempt to prematurely join from the back will be disconnected.

Fourth, in terms of SP, the problem is much harder. The previous two mechanisms did not rely on self reporting of destinations. In this setting, however, we have to deal with untruthful reporting. It is intuitive that agents can gain from misreporting a shorter destination, as this will cause the mechanism to assign them a shorter lead time. To solve this, agents who overstay as followers will be disconnected.

Surprisingly, we show that agents can also gain from misreporting a **farther** destination, and that this manipulation is not trivial, and maybe impossible to prevent. One example of this form of manipulation which can be overcome, happens between the first two agents when the one who goes second has an incentive to report a farther destination, possibly making the first leader's share larger than what it would have been otherwise. This forces the decision of who goes first to be randomized.

The fifth issue is deciding who goes first at the beginning of a convoy. Even if the decision is randomized, the agents would only accept the coin toss decision if the two shares that are offered are equal in expectation, similar to classic bargaining games [20, 23]. We call this problem the *"Who Goes First?" Problem*. In short, the second leader has a greater potential to lead less time due to the possibility of new entrants. A more detailed description of this problem in provided in Appendix 9.3.

We offer a solution that addresses this problem by offering the first leader a smaller share to compensate it for missing out on the potential contribution of new agents by taking the first lead. The second agent, on the other hand, receives a larger portion, but, has the potential of benefiting from the contribution of new agents. Since the two shares are equal in expectation, the agents will be indifferent and accept the decision of who takes on the role of the first leader which is made randomly.

The expected future contribution of new entrants, over the first two agents' mutual availability period is denoted $ec(x_{\alpha,\beta})$. The calculation of $ec(x_{\alpha,\beta})$ can be found in Appendix 9.4.

Formal definition of allocations- The allocations in the Single Game Load Balancing mechanism are:

- α commits to lead first for $s_\alpha = \frac{x_{\alpha,\beta}}{2} - \frac{1}{2} \cdot ec(x_{\alpha,\beta}) + \frac{c_{cf}(\bar{n}_r)}{2 \cdot u}$ where $\bar{n}_r$ is the expected number of available agents when α rotates.
- β commits to lead $s_\beta = \frac{x_{\alpha,\beta}}{2} + \frac{1}{2} \cdot ec(x_{\alpha,\beta}) + \frac{c_{cf}(\bar{n}_r)}{2 \cdot u}$
- Any other joining agent i will join the convoy from the front and commit to lead for no more than its proportional share, $ex\_ante\_prop_i$.

Algorithm 3 Single Game Load Balancing Mechanism

- The first two agents randomly decide who goes first.
- The first leader's ex-ante proportional share is reduced according to the estimate of future entrants and equals s_α.
- New agents can join only from the front and lead for their ex-ante proportional share, $ex\_ante\_prop_i$.
- A switch happens when the leader finishes its share or when a new agent joins in.
- When a leader finishes its share it rotates to the back of the convoy.

Theorem 6. *Given the defined allocations, any agent will be indifferent to being either the first or the second leader.*

Proof. We need to prove that in expectation α and β have equal shares. Since α commits to be the first leader and to complete its share before β becomes the leader, in expectation α does not benefit from additional agents' contributions. It does however get a smaller share to lead, in the first place. By construction, the difference between the allocated shares is equal to the expected contribution of new agents that only β would benefit from. The agents share the cost of the one rotation required to switch between them. Hence, since the agents are risk neutral, and their shares are equal in expectation, they are indifferent to receiving either share. □

Theorem 7. *In the Single Game Load Balancing mechanism, all the agents but β are guaranteed to lead for no more than their ex-ante proportional share, while β leads for less than its ex-ante proportional share in expectation.*

Proof. Since according to the mechanism all agents but α and β lead for no more than their ex-ante proportional share, all we need to show is that α is also doing no more than its ex-ante proportional share. Since α and β are the first two members of the convoy they should lead for no more than the ax-ante proportional share $\frac{T}{2} + \frac{c}{u}$. According to the mechanism, α is committed to lead for

$$\frac{T}{2} - \frac{1}{2} \cdot ec(T) + \frac{c}{2 \cdot u}$$

(i.e., we subtract positive value from α's ex-ante proportional share). Hence, α will lead for no more than its ex-ante proportional share. As shown in Theorem 6, β's share is equal in expectation to that of α, and so β is guaranteed to lead less than its ex-ante proportionality in expectation. □

The Single Game Load Balancing mechanism resolves all the issues we have described so far including the *who goes first* problem. It guaranties full ex-ante proportionality for all but one agent, who is guaranteed ex-ante proportionality in expectation.

Despite solving almost all of the issues for a single distributed SOCD game, there is still one SP issue that cannot be resolved. In some cases agents still have an incentive to misreport a farther destination as explained in the proof to Theorem 8.

THEOREM 8. *In a single game SOCD problem, no proportional distributed mechanism can guarantee SP.*

Preliminaries:

- α and β are the first two agents to start a convoy
- γ is the third entrant
- To guarantee ex-ante proportionality, each agent must lead for at most its ex-ante proportional share, i.e., its shared road with the existing agents, divided by the number of agents including itself. For example, for γ it would be $\frac{x_{\alpha,\beta,\gamma}}{3}$. Where $x_{\alpha,\beta,\gamma}$ is the shared section on the road for all three agents.

LEMMA 6.1. *Any distributed single game mechanism that does not assign equal probability of being the first leader to α and β is not SP.*

PROOF. W.l.o.g. we assume that α has a lower probability of leading first. If α falsely reports a farther destination than its truthful one, it will increase the first leader's proportional share and since α's probability of being first is lower than β 's, in expectation α will benefit from β 's increased share. Hence, if α and β do not have an equal probability of leading first, then the one with the lower probability of leading first, has an incentive to falsely report a farther destination. □

LEMMA 6.2. *If there is a zero probability of new entrants beyond α and β, and the mechanism assigns equal probability for α and β to lead first, then α and β are indifferent to falsely reporting a farther destination.*

PROOF. If the agent is one of the first two agents, it might report a farther destination in order to have the other agent believe that their mutual travel time is larger than it really is, and agree to lead for a longer stretch. However, since the decision of who goes first is random, there is a probability of 0.5 that the untruthful agent would be chosen to lead first and then it would lead for longer than its proportional share. Thus, in expectation, both agents will not gain or lose anything from giving false information in this case. □

LEMMA 6.3. *If there is a positive probability of new entrants and there is a positive probability that the new entrant, γ, will lead first or second as it joins, then α and β have an incentive to falsely report a farther destination than their truthful one.*

PROOF. If there is a positive probability of newcomers arriving, and there is a positive probability that the newcomer would lead first or second, then the indifference result no longer holds, and both α and β have an incentive to lie since they can only gain from γ's extended lead time. □

LEMMA 6.4. *If γ has a probability of more than 0.5 to be the last to lead, it will have an incentive to lie.*

PROOF. γ's proportional share is $\frac{x_{\alpha,\beta,\gamma}}{3}$. If γ reports $x'_\gamma > x_\gamma$ instead of x_γ, then if γ is last to lead, the first two leaders (α and β) will lead for $\frac{x'_{\alpha,\beta,\gamma}}{3} \geq \frac{x_{\alpha,\beta,\gamma}}{3}$, thus leaving γ a smaller share to lead than the proportional share it would have gotten had it provided its truthful destination.

On the other hand, if γ is the first or second to lead, it will lead for a larger share than its proportional share if it falsely reports a farther destination than its truthful one.

Therefore, if γ's probability of leading last is exactly 50%, γ will be indifferent to lying. However, if there is a probability of more than 50% that γ leads last, then it has an incentive to lie. □

PROOF. Hence, according to Lemma 1-4, in all cases at least one agent has an incentive to lie, which leads to the conclusion that there is no mechanism that can guarantee SP in a distributed single SOCD game where there is a positive probability of more than two agents. □

Thus, in order to apply the single game solution mechanism in practical settings, we need to either assume that agents are truthful, or assume the existence of some central system that can enforce the rules, e.g. by penalizing agents, or keeping a reputation log. Though ultimately a negative result, we note once again that these sorts of assumptions are common in past fair division problems [9, 24].

7 CONCLUSIONS AND FUTURE WORK

In this paper we define the SOCD problem, a novel sequential online variation of the chore division problem. We instantiate SOCD on a real-world problem of autonomous vehicle convoy formation. We propose three fair-division mechanisms to balance the load of the leader and equally share the energy savings of the followers among all the convoy's participants. The Payment Transfer mechanism assumes the existence of a central payment transfer system, and achieves optimal efficiency and fairness. The Repeated Game Load Balancing mechanism does not rely on a central payment system, yet offers optimal efficiency, and fairness in expectation, after participating in multiple repeated games. The Single Game Load Balancing mechanism is also distributed, and is able to achieve ex-ante proportionality for all but one agent for which it guarantees ex-ante proportionality in expectation, the highest attainable fairness criterion for a single game that solves the *who goes first* problem. It does so with a minimal number of divisions.

The mechanism also handles issues that arise when dealing with self-interested agents in a dynamic distributed environment except for misreporting a farther destination for which we prove that there is no possible solution.

For future work we plan to run experiments to measure how close in practice does our proposed single game solution come to the equitable offline solution offered by the payment transfer mechanism. Other possible threads for future work include relaxing the assumption that the agents' valuation functions are homogeneous and designing mechanisms that can support heterogeneous valuation functions as discussed in Appendix 9.1.

8 ACKNOWLEDGMENTS

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (CPS-1739964, IIS-1724157, NRI-1925082), ONR (N00014-18-2243), FLI (RFP2-000), ARO (W911NF-19-2-0333), DARPA, Lockheed Martin, GM, and Bosch. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

REFERENCES

[1] Assad Al Alam, Ather Gattami, and Karl Henrik Johansson. 2010. An experimental study on the fuel reduction potential of heavy duty vehicle platooning. In *Intelligent Transportation Systems (ITSC)*. 306–311.

[2] Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. 2015. Online Fair Division: Analysing a Food Bank Problem.. In *IJCAI*. 2540–2546.

[3] Haris Aziz. 2016. Computational Social Choice: Some Current and New Directions.. In *IJCAI*. 4054–4057.

[4] Gerdus Benade, Aleksandr M Kazachkov, Ariel D Procaccia, and Christos-Alexandros Psomas. 2018. How to Make Envy Vanish Over Time. In *ACM Conference on Economics and Computation*. 593–610.

[5] Carl Bergenhem, Steven Shladover, Erik Coelingh, Christoffer Englund, and Sadayuki Tsugawa. 2012. Overview of platooning systems. In *Proceedings of the 19th ITS World Congress*.

[6] Anirudh Kishore Bhoopalam, Niels Agatz, and Rob Zuidwijk. 2017. Planning of truck platoons: A literature review and directions for future research. *Transportation Research Part B: Methodological* (2017).

[7] Steven J Brams and Alan D Taylor. 1996. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press.

[8] Simina Brânzei and Peter Bro Miltersen. 2013. Equilibrium analysis in cake cutting. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 327–334.

[9] Simina Brânzei and Peter Bro Miltersen. 2015. A dictatorship theorem for cake cutting. In *IJCAI*.

[10] Yiling Chen, John K Lai, David C Parkes, and Ariel D Procaccia. 2013. Truth, justice, and cake cutting. *Games and Economic Behavior* 77, 1 (2013), 284–297.

[11] Arnaud De La Fortelle, Xiangjun Qian, Sébastien Diemer, Jean Grégoire, Fabien Moutarde, Silvère Bonnabel, Ali Marjovi, Alcherio Martinoli, Ignacio Llatser, Andreas Festag, et al. 2014. Network of automated vehicles: the AutoNet 2030 vision. In *ITS World Congress*.

[12] Cristofer Englund, Lei Chen, Jeroen Ploeg, Elham Semsar-Kazerooni, Alexey Voronov, Hoai Hoang Bengtsson, and Jonas Didoff. 2016. The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles. *IEEE Wireless Communications* 23, 4 (2016), 146–152.

[13] Eric Friedman, Christos-Alexandros Psomas, and Shai Vardi. 2015. Dynamic fair division with minimal disruptions. In *ACM conference on Economics and Computation*. 697–713.

[14] Eric J Friedman and David C Parkes. 2003. Pricing wifi at starbucks: issues in online mechanism design. In *ACM conference on Electronic commerce*. 240–241.

[15] GR Janssen, J Zwijnenberg, IJ Blankers, and JS de Kruijff. 2015. Truck platooning: Driving the future of transportation. *TNO Mobility and Logistics* (2015).

[16] Ian Kash, Ariel D Procaccia, and Nisarg Shah. 2014. No agent left behind: Dynamic fair division of multiple resources. *Journal of Artificial Intelligence Research* 51 (2014), 579–603.

[17] Pooja Kavathekar and YangQuan Chen. 2011. Vehicle platooning: A brief survey and categorization. In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 829–845.

[18] Michael P Lammert, Adam Duran, Jeremy Diez, Kevin Burton, and Alex Nicholson. 2014. Effect of platooning on fuel consumption of class 8 vehicles over a range of speeds, following distances, and mass. *SAE International Journal of Commercial Vehicles* 7, 2014-01-2438 (2014), 626–639.

[19] Michael P Lammert, Brian McAuliffe, Xiao-Yun Lu, Steven Shladover, Marius-Dorin Surcel, and Aravind Kailas. 2018. *Influences on Energy Savings of Heavy Trucks Using Cooperative Adaptive Cruise Control*. Technical Report. National Renewable Energy Lab.(NREL), Golden, CO (United States).

[20] John F Nash Jr. 1950. The bargaining problem. *Econometrica: Journal of the Econometric Society* (1950), 155–162.

[21] Peloton-tech. 2020. https://peloton-tech.com.

[22] Tom Robinson, Eric Chan, and Erik Coelingh. 2010. Operating platoons on public motorways: An introduction to the sartre platooning programme. In *17th world congress on intelligent transport systems*, Vol. 1. 12.

[23] Ariel Rubinstein. 1982. Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society* (1982), 97–109.

[24] Erel Segal-Halevi and Balázs R Sziklai. 2019. Monotonicity and competitive equilibrium in cake-cutting. *Economic Theory* 68, 2 (2019), 363–401.

[25] Steven E Shladover. 2007. PATH at 20—History and major milestones. *IEEE Transactions on intelligent transportation systems* 8, 4 (2007), 584–592.

[26] Sadayuki Tsugawa. 2013. An overview on an automated truck platoon within the energy ITS project. *IFAC Proceedings Volumes* 46, 21 (2013), 41–46.

[27] Toby Walsh. 2011. Online cake cutting. In *International Conference on Algorithmic DecisionTheory*. Springer, 292–305.

9 APPENDIX

9.1 Appendix for section 3 - Heterogeneous Agents

Considering heterogeneous agents is something we intend to continue investigating in future work, but presents new challenges that would require more space than is available in this initial paper that introduces this new problem. Just to give some insight, heterogeneity can manifest in different ways:

- Each agent a_i has an individual utility per unit of time u_i as a follower. However, the utility per unit of time is constant.
- Each agent a_i has an individual utility per unit of time $u_i(a_{front})$ as a follower. However, the utility per unit of time is a function of the vehicle in front a_{front}.
- Each agent a_i has an individual cost per unit of time for leading, c_i.
- Each agent a_i has an individual valuation v_i for leading different road segments.

When considering these options, the modeling of the problem changes and the notion of fairness becomes ill-defined. Other issues regarding strategyproofness come up when considering whether these individual parameters are known or self-reported.

In this paper we restrict our attention to considering homogeneous agents, and lay the foundations to the analysis of heterogeneous utility and cost functions in several possible avenues of future work.

9.2 Appendix for section 6.1 - Optimality of Payment Transfer Mechanism

Proof for Theorem 2:

The Payment Transfer mechanism is efficiently optimal, and equitable.

Proof. Since no rotations are made in the *Payment Transfer* mechanism, it achieves optimal efficiency. In terms of fairness, for each segment, all the followers save the same amount of energy and pay the same amount of money to the leader. All we need to prove is that the sum of payments that the leader receives per segment is equal to each of the followers' saving for that segment, calculated as:

$$|seg| \cdot u - p^i_{seg} = |seg| \cdot u - \frac{|seg| \cdot u}{n_{seg}} = \frac{(n_{seg} - 1) \cdot |seg| \cdot u}{n_{seg}}$$

The leader's received payment for every segment is equal to:

$$(n_{seg} - 1) \cdot p^i_{seg} = (n_{seg} - 1) \cdot \frac{|seg| \cdot u}{n_{seg}} = \frac{(n_{seg} - 1) \cdot |seg| \cdot u}{n_{seg}}$$

which is identical to the saving of each follower and thus the mechanism guaranties equitability, both ex-ante and ex-post, and allows agents to be indifferent between leading and following. □

9.3 Appendix for section 6.2 - The "Who Goes First?" Problem

In classic bargaining games, two agents have to agree on a division of some good between them, and delays are costly [20, 23]. In our model we face a similar problem. At the very beginning of an SOCD game, two agents a_i, and a_j must negotiate who will be the first active agent. Assuming they try to minimize the number of switches, they will split the shared time they have together so that each of them will lead for half of the time, i.e., their proportional share. W.l.o.g we assume that a_i is the first. Since new entrants are dynamically introduced, there is a chance that by the time it is a_j's turn, a_j would share its portion with other agents and would thus be active for less time than was initially assigned to it. Note that if new agents arrive when a_i is active, then both a_i and a_j can enjoy the additional savings. However, since performed chores are irrevocable, if a new agent were to arrive after a_i had done its part, then only a_j would benefit. This asymmetric advantage in favor of a_j creates an incentive for both agents to prefer to be the last to be active. Note that this issue only arises with the first two agents to start the convoy since they are symmetric and any delay will be equally costly to both sides. Other joining agents have no such bargaining power and will accept the shares offered to them.

9.4 Appendix for section 6.2 - Estimating expected future contribution for Single Game Mechanism

In order to estimate the potential contribution of new agents, we assume that the distribution of traffic density is known to the agents, and that they use it to estimate the portion of the road where new agents are expected to lead. For each section, the contribution of an agent is relative to the number of existing convoy members in that section. In order to calculate the total contribution, agents need to have an estimate of the length of the sections where the convoy is likely to have three members, four members, and so on. In addition, we make an assumption regarding the maximal number of agents in a given convoy:

Limiting the maximal number of agents- This mechanism imposes a maximal number of agents allowed in a convoy whose joint path is of length x. This maximal number is represented as $n_{max}(x)$. The limitation is introduced due to the requirement to rotate to the back of the convoy. If the time required to rotate, is greater than the time a prospective agent wants to join the convoy for as a follower, it would not be beneficial for that agent to join. In such cases we could consider allowing additional agents to join from the back without contributing as leaders. However, if the mechanism would allow joining for free, then current members, scheduled to lead, would have an incentive to leave and rejoin the convoy.

We refer to the first leader as α, and the second leader as β. Their joint road section will be denoted as $x_{\alpha,\beta}$. The distribution of traffic density for a given section of the road is modeled as the expected length of its subsections where there will be y expected additional convoy participants besides α and β. The length of the sections in which there are y new agents is denoted as $sec\_size(y)$, and is directly extracted from the traffic load distribution.

The sum of all the subsection lengths sums up to $x_{\alpha,\beta}$ and the number of new participants ranges from 0 to $n_{max}(x_{\alpha,\beta}) - 2$. Figure 4 presents an example of a traffic density distribution from which the agents estimate the expected section length with each number of new agents. The expected contribution for section $x_{\alpha,\beta}$,

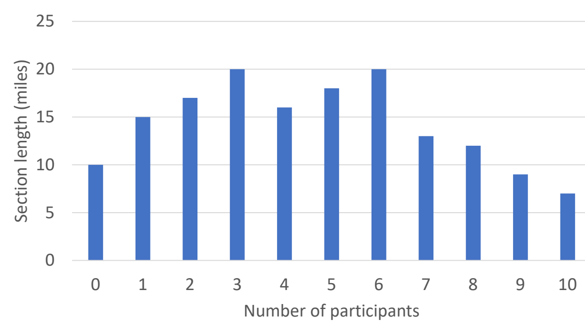

Figure 4: An example of the distribution of section lengths for each number of participants. In this example $n_{max} = 12$.

termed $ec(x_{\alpha,\beta})$, is a summation of all the contributions from all the subsections that comprise $x_{\alpha,\beta}$, according to the number of expected additional agents in each subsection from $y = 1$ until $y = n_{max(x_{\alpha,\beta})} - 2$. For each subsection, another summation is performed over the agents in that subsection from $z = 1$ until $z = y$. The contribution of each agent is calculated as the subsection length, divided by the number of agents that were in that subsection when the agent arrived. The first new agent sees two agents when it arrives so it contributes a third of the way, the second sees three agents present, and contributes a fourth of the way and so on. Finally, the total expected contribution is:

$$ec(x_{\alpha,\beta}) = \sum_{y=1}^{n_{max(x_{\alpha,\beta})}-2} \sum_{z=1}^{y} \frac{sec\_size(y)}{(z+2)}$$