The Sequential Online Chore Division Problem - Definition and Application

Extended Abstract

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
pstone@cs.utexas.edu

ABSTRACT

This paper defines a novel formulation of chore division, called **S**equential **O**nline **C**hore **D**ivision (SOCD), in which participants arrive and depart online, while the chore is being performed. In SOCD, there exists some uncertainty both regarding the total number of participants and their arrival/departure times. Moreover, only one agent can perform the chore at any given moment, and switching the performer incurs a cost.

This novel variant of chore division can model real world problems such as the *autonomous vehicle convoy formation* problem, which has significant social implications. 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. Empirical evaluations estimate that a follower can save over 10% of its fuel consumption [1]. However, since the leader sees little or no such gains, choosing the leader of such a convoy raises issues of fairness, and efficiency. Solving these issues is challenging since vehicles can dynamically join and leave the convoy.

To address this problem, we propose three mechanisms for fair chore division. The first mechanism is centralized and uses side payments while the other two are distributed and seek to balance the participants' loads. We show that the payment-transfer mechanism, which requires a centralized server, results in optimal fairness and efficiency. For the cases where a central server is not available, we show that the repeated-game mechanism produces allocations which are efficiently-optimal and fair in expectation.

For the single-game case, we first prove that optimal fairness is impossible to guarantee. We then show that our proposed single-game mechanism, which offers minimal efficiency loss, achieves ex-ante proportionality.

KEYWORDS

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

ACM Reference Format:
Harel Yedidsion, Shani Alkoby, and Peter Stone. 2020. The Sequential Online Chore Division Problem - Definition and Application. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand. © 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

1 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.

We make the following assumptions:

- 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, $\forall i, j, s \quad V_i(s) = V_j(s)$.
- Each agent know its own arrival and departure times, and communicate this information to the agents that are present when it arrives.
- There always exists a positive probability that new agents will arrive.
- A switch between an active agent and an available agent results in a cost to the system c.

The time frame for an SOCD game, T, starts when there is at least one available agent, and ends where there are non left.

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.

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.

2 FAIRNESS DEFINITIONS AND PROPERTIES

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 a_i's proportional share for each segment, and then sum up the shares over all segments.

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. Also note that the ex-post proportional share is also envy-free and equitable since for each segment all the agents get equal shares.

Limitations of Fairness in SOCD - 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 [2]. We adhere to that paper's call to continue investigating online chore division, and provide an impossibility result for fairness in SOCD.

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

3 CONVOY FORMATION MECHANISMS

Convoy formation can be modeled as an SOCD problem. 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. 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.

In this section we outline a number of possible convoy formation mechanisms, each geared toward a different set of environmental assumptions.

The Payment Transfer mechanism is applicable when there is a central payment transfer system. It assigns only one active agent while the followers transfer a share of their savings to the leader in order to keep fairness. 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.

The Repeated Game Load Balancing mechanism assumes a distributed setting where there are no payment transfer abilities. It 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. This mechanism is perfectly efficient because it does not require any rotations to be made. It also guarantees equability, i.e. ex-post proportionality, in expectation through repeated games.

However, some agents may be interested in achieving fair division in each individual game.

The Single Game Load Balancing mechanism aims to guarantee fairness for every single game by rotating the leader. It requires each agent to lead the convoy for no more than its ex-ante proportional share. 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 mechanism guarantees the minimal number of rotations for any fair chore division, as each agent rotates at most once. Moreover, it guarantees ex-ante proportionality, the highest attainable fairness criterion for a single game. [1]

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), ARL, 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. We acknowledge the contribution of Mr. Vinay Shukla to a preliminary version of this work.

[1] For an in-depth analysis of the proposed mechanisms please refer to the full version of the paper at: https://www.dropbox.com/s/kd578wffqx6o71z/Convoy-aamas20Full.pdf?dl=0

REFERENCES

[1] 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).

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