

Social Aware Assignment of Passengers in Ridesharing

Chaya Levinger, Noam Hazon, Amos Azaria

Department of Computer Science, Ariel University, Ariel, Israel 4070000
chaya.levinger@msmail.ariel.ac.il, noamh@ariel.ac.il, amos.azaria@ariel.ac.il

Abstract

We analyze the assignment of passengers in a shared ride, which considers the social relationship among the passengers. Namely, there is a fixed number of passengers in each vehicle, and the goal is to recommend an assignment of the passengers such that the number of friendship relations is maximized. We show that the problem is computationally hard, and we provide an approximation algorithm.

Introduction

Coalition formation is an important research branch within multiagent systems (?). It analyses the outcome that results when a set of agents is partitioned into coalitions. For example, consider a group of travelers (passengers) who want to reach a similar destination. Each passenger has a preference related to who will be with her in the vehicle. Namely, each passenger would rather share a vehicle with as many of her friends during the ride, and thus the utility of each passenger is the number of friends traveling with her. However, the vehicles have a limited capacity. The goal is to assign the passengers to vehicles while maximizing the social welfare (the sum of all passengers' utilities).

We formulate the described problem as the social aware assignment problem, which assumes that the agents' utilities depend on a social network that represents the social relationships among the agents. The social network is modeled as an unweighted graph where the vertices are agents and the edges indicate friendship among the agents. The utility function of an agent is the number of friends she has within the coalition to which she is assigned. In addition, there is a hard constraint on the maximal size of each coalition. Actually, our model is a special case of simple Additively Separable Hedonic Games (ASHGs) (?). In this paper, we show that the social aware assignment problem is computationally hard, and we provide an approximation algorithm.

Related Work

Sless et al. (?) tackle a problem similar to ours. Similar to our work, they assume a friendship graph and attempt

to maximize the number of friends in each group. However, in their setting the agents must be partitioned into exactly k groups without any restriction on each group's size. The graph partitioning problem, introduced by Hyafil and Rivest (?), is a more restricted problem, in which the vertices of a graph must be partitioned into exactly k groups of equal sizes.

Wright and Vorobeychik (?) also study a model of ASHG where there is a restriction on the size of each coalition. Within their model, they propose a strategyproof mechanism that achieves good and fair experimental performance, despite not having a theoretical guarantee.

The Social Aware Assignment Problem

We analyze the problem of the assignment in ride-sharing problem, while maintaining the human-centric approach. Specifically, our goal is to assign the users to vehicles such that each user will be matched with as many friends as possible in the same vehicle, while each vehicle is limited to a number of passengers, k . Formally,

Definition 1 (Social aware assignment). *We are given a number k and an undirected friendship graph $G = (U, E)$ where $(u_i, u_j) \in E$ if the user u_i and the user u_j are friends of each other. The goal is to find an assignment P , which is a partition of the set U , such that $\forall S \in P, |S| \leq k$, and the value of P , $V_P = |\{(u_i, u_j) \in E: \exists S \in P \text{ where } u_i \in S \text{ and } u_j \in S\}|$ is maximized.*

For example, given the graph in Figure 1a and a vehicle size limit $k = 3$, the value of the partition $P = \{\{v_1, v_3, v_6\}, \{v_2, v_4, v_7\}, \{v_5, v_8\}\}$, shown in Figure 1b, equals 7. Indeed, this is the optimal partition since there is no other partition with higher value. Clearly, the decision variant of the social aware assignment problem is to decide whether there exists a partition with a value of at least v .

The Hardness of the Social Aware Assignment Problem

The social aware assignment problem when $k = 2$ is equivalent to the maximum matching problem, and thus it can be computed in polynomial time (?). However, our problem becomes intractable when $k \geq 3$. For the hardness proof, we define for each $k \in \mathbb{N}$ the *Cliques_k* problem, which is as follows.

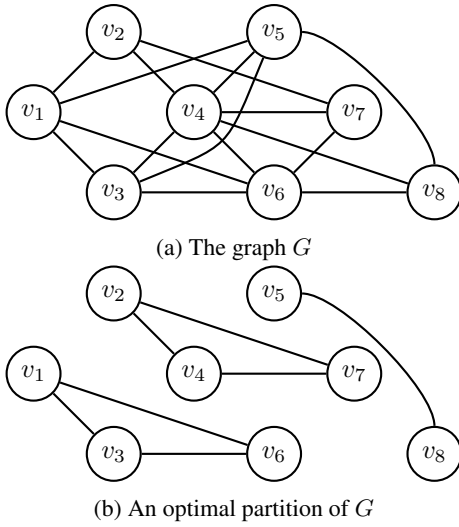


Figure 1: An example for the social aware assignment problem where $k = 3$.

Definition 2 ($Cliques_k$). Given an undirected graph $G = (V, E)$, decide whether V can be partitioned into disjoint cliques, such that each clique is composed of exactly k vertices.

Clearly, $Cliques_2$ can be decided in polynomial time by computing a maximum matching of the graph G , M , and testing whether $|M| = \frac{|V|}{2}$. However, $Cliques_k$ becomes hard when $k \geq 3$.

Lemma 1. $Cliques_k$ is in NP-Complete for every $k \geq 3$.

Theorem 1. The decision variant of the social aware assignment problem is in NP-Complete.

Approximation of the Social Aware Assignment Problem

Since we showed that the social aware assignment problem is in NP-Complete, we first provide an approximation algorithm where $k = 3$. The algorithm works as follows. It first computes a maximum matching, M , in the given graph G . It then creates a graph G' that includes all the unmatched nodes and a union node for each pair of matched nodes in M . Finally, it computes a maximum pair matching in G' and returns the partition, P , of all the matched sets.

Theorem 2. For $k = 3$, Algorithm 1 provides a solution for the social aware assignment problem with an approximation ratio of 0.5.

Similarly to the approximation algorithm where $k = 3$, we provide the Match and Merge (MnM) algorithm, which is an approximation algorithm for any $k \geq 3$. The algorithm consists of $k - 1$ rounds. Each round is composed of a matching phase followed by a merging phase. Specifically, in round l MnM computes a maximum matching, $M_l \subseteq E_l$, for G_l (where $G_1 = G$). In the merging phase, MnM creates a graph G_{l+1} that includes a merged node for each pair of matched nodes. G_{l+1} also includes all unmatched nodes,

Algorithm 1: Approximation algorithm for $k=3$

```

1 Input: A graph  $G = (V, E)$ 
   Result: A partition  $P$  of  $G$  where  $k = 3$ .
2  $M \leftarrow$  maximum matching in  $G$ 
3  $G' = (V', E') \leftarrow$  an empty graph
4 for every  $(v_i, v_j) \in M$  do
5   | Add vertex  $v_{i,j}$  to  $G'$ 
6   | remove  $v_i, v_j$  from  $V$ 
7 for every  $v_i \in V$  do
8   | Add vertex  $v_i$  to  $G'$ 
9 for every  $(v_i, v_j) \in M$  do
10  | for every  $v_k \in V$  do
11    |   if  $(v_i, v_k) \in E$  OR  $(v_j, v_k) \in E$  then
12      |     Add edge  $(v_{i,j}, v_k)$  to  $G'$ 
13  $M' \leftarrow$  maximum matching in  $G'$ 
14  $P \leftarrow$  an empty partition
15 for every  $(v_{i,j}, v_k) \in M'$  do
16  | add the set  $\{v_i, v_j, v_k\}$  to  $P$ 
17  | remove  $(v_i, v_j)$  from  $M$ 
18  | remove  $v_k$  from  $V$ 
19 for every  $(v_i, v_j) \in M$  do
20  | add the set  $\{v_i, v_j\}$  to  $P$ 
21 for every  $v_i \in V$  do
22  | add the set  $\{v_i\}$  to  $P$ 
23 return  $P$ 

```

along with their edges to the merged nodes. Finally, MnM returns the partition, P , of all the matched sets.

Theorem 3. For $k > 3$, MnM provides an approximation ratio of $\frac{1}{k-1}$ for the social aware assignment problem.

Acknowledgments

This research was supported in part by the Ministry of Science, Technology & Space, Israel.

Quidem eius eum et nostrum dolorem dicta error alias rem sed, eaque ad dolor, possimus modi fuga molestias aspernatur ut nulla enim, odit labore molestias sed quae, temporibus aperiam esse accusamus quaerat illo odit accusantium voluptatum voluptatem. Ipsum ipsam fugit, quos consequatur quas in, repellat placeat atque et deleniti impedit officiis ullam molestias, tenetur rerum nulla magni, perspicatis tenetur consectetur magnam quas ratione culpa libero repellendus? Quos similique odit, quaerat quia eum repudiandae iste aspernatur aliquam odit, expedita beatae tempora temporibus laborum modi ullam ipsam rerum, cum repudiandae iure sapiente nemo. Earum soluta officiis ad, ipsum odio assumenda iusto hic earum porro necessitatibus? Quia quibusdam blanditiis exercitationem, accusamus perspicatis non debitis pariatur, itaque maiores a iste dolore ea, rem dignissimos voluptas reiciendis quo exercitationem fugit quasi. Voluptatibus nemo minus unde ex ducimus quos incidunt velit rem ea nisi, dolorum rerum molestiae nesciunt pariatur quidem cumque quo omnis quibusdam hic, minus non nam nostrum eos rem odio cumque ad amet, optio natus deserunt vitae commodi molestiae, impedit sapiente vel? Excepturi consequuntur enim earum quod dolores tene-

tur quibusdam dicta illo recusandae aut, vel officia omnis
iure soluta voluptas fugit?