Incentive Mechanism Design for Federated Learning: Hedonic Game Approach

Cengis Hasan
University of Luxembourg
SnT - Interdisciplinary Centre for Security, Reliability and Trust
cengis.hasan@uni.lu

ABSTRACT

Incentive mechanism design is crucial for enabling federated learning. We deal with clustering problem of agents contributing to federated learning setting. Assuming agents behave selfishly, we model their interaction as a stable coalition partition problem using hedonic games where agents and clusters are the players and coalitions, respectively. We address the following question: is there any utility allocation method ensuring a Nash-stable coalition partition? We propose the Nash-stable set and analyze the conditions of non-emptiness. Besides, we deal with the decentralized coalition partition. We formulate the problem as a non-cooperative game and prove the existence of a potential.

KEYWORDS

Federated Learning, Hedonic Games, Optimal Clustering

1 INTRODUCTION

Data protection is a major concern. If we do not trust someone withholding our data, we may opt for federated learning by privately developing intelligent systems to create privacy-preserving AI. Federated learning enables privacy-preserving machine learning in a decentralized way [10]. It is used in situations where data is distributed among different agents and training is impossible due to the difficulty to collect data centrally. All data is kept on device while a shared (global) learning model is trained in each device and aggregated (combined) centrally. Formally, we consider the following setting: i) data owner agents which locally trains the shared learning model, and ii) model aggregating entity (MAE) which combines learning model of its own with the agents. MAE and agents contribute to the same shared learning model. Federated learning has been identified as a distributed machine learning framework which sees rapid advances and broad adoption in next generation networking and edge systems [4, 6, 8–13]. Obviously, the motivation to implement federated learning is to reduce the variance in a learned model by accessing more data.

A very crucial question is how would MAE motivate the agents to participate in federated learning. Designing the mechanism of agents’ incentives can be performed by utilizing various frameworks such as game theory, auction theory, etc [11]. Any clustering among agents (players) being able to make strategic decisions becomes a coalition formation game when the players—for various individual reasons—may wish to belong to a relative small coalition rather than the grand coalition—the set of all players. Players’ moves from one to another coalition are governed by a set of rules. Basically, an agent (player) will move to a new coalition when it may obtain a better utility from this coalition. We shall not consider any permission requirements, which means that a player is always accepted by a coalition to which the player is willing to join. Based on those rules, the crucial question in the game context is how a stable partition exists. This is essential to enable federated learning.

We study the hedonic coalition formation game model of the agents and analyze the Nash stability [7]. A coalition formation game is called hedonic if each player’s preferences over partitions of players depend only on the members of his/her coalition. Finding a stable coalition partition is the main question in a coalition formation game. We refer to [1] discussing the stability concepts associated to hedonic conditions. In the sequel, we concentrate on the Nash stability. The definition of the Nash stability is quite simple: a partition of players is Nash stable whenever no player deviates from its coalition to another coalition in the partition.

In this work, the following problem is studied: having coalitions associated with their utilities, we seek the answer of how must the coalition utilities be allocated to the players in order to obtain a stable coalition partition. Clearly, the fundamental question is to determine which utility allocation methods may ensure a Nash-stable partition. We first propose the definition of the Nash-stable set which is the set of all possible utility allocation methods resulting in Nash-stable partitions. We show that additively separable and symmetric utility allocation always ensures Nash-stable partitions. Moreover, our work aims also at finding the partitions in a decentralized setting. We model the problem of finding a Nash-stable partition by formulating it as a non-cooperative game and show that such a game is a potential game. A recent work that considers the clustering of agents in the form of hedonic games can be found in [5] where the authors study the agents decisions to participate in a biased learning setting in case of a biased global model.

2 MOTIVATION AND PROBLEM DESCRIPTION

We consider a set of agents denoted by $N = \{1, 2, \ldots, n\}$ that can participate in the federated learning setting, and a model aggregating entity (MAE) which aggregates (combines) learning model of its own with the agents. MAE and agents contribute in the same global learning model. The parameters of learning model of MAE and
We assume that agents may not agree to be in the same cluster. Therefore, we shall formulate the problem using the utility term due to the fact that multiple disjoint clusters may occur. In Figure 1, we illustrate with probability $p_i$ that from the agents’ point of view, this corresponds to the earnings of the agents. Then, we come up with the question how to design such an example scenario in which two disjoint clusters, i.e. 1, 2, 3 and 4, 5, create two different aggregated learning models, i.e. $(\theta^{1,2,3};x)$ and $(\theta^{4,5};x)$, respectively.

From the perspective of agents, we assume that MAE assigns a utility to all possible clusters. Note that from the MAE point of view, this is the cost that must be paid to the cluster. By this way, MAE evaluates the contribution to the aggregated model. However, we shall formulate the problem using the utility term due to the fact that from the agents’ point of view, this corresponds to the earnings of the agents. Then, we come up with the question how to design the incentives in order that the agents are willing to participate in federated learning setting taking into account their preferences.

In this work, we consider the following linear aggregation method:

$$\theta^S = A(\theta_1, \theta_2, \ldots, \theta_{NS};x), \quad (n_S = |S|)$$  

(1)

where $A(\cdot)$ shows the aggregation function given $\theta_1, \theta_2, \ldots, \theta_{NS}$ and $x = (x_1, x_2, \ldots, x_N)$ in which $x_i = 1$ if MAE receives successfully information of $\theta_i$. This is nothing more than choosing agent $i$ with probability $p_i$.

### 2.1 Incentives of Agents

We assume that agents may not agree to be in the same cluster depending on their preferences. Thus, we come up with the case where multiple disjoint clusters may occur. In Figure 1, we illustrate such an example scenario in which two disjoint clusters, i.e. 1, 2, 3 and 4, 5, create two different aggregated learning models, i.e. $(\theta^{1,2,3};x)$ and $(\theta^{4,5};x)$, respectively.

From the perspective of agents, we assume that MAE assigns a utility to all possible clusters. Note that from the MAE point of view, this is the cost that must be paid to the cluster. By this way, MAE evaluates the contribution to the aggregated model. However, we shall formulate the problem using the utility term due to the fact that from the agents’ point of view, this corresponds to the earnings of the agents. Then, we come up with the question how to design the incentives in order that the agents are willing to participate in federated learning setting taking into account their preferences.

In this work, we consider the following linear aggregation method:

$$\theta^S = \frac{\sum_{i \in S} x_i m_i \theta_i}{\sum_{i \in S} x_i m_i}, \quad x_i = \begin{cases} 1, & \text{reception successful} \\ 0, & \text{otherwise.} \end{cases}$$  

(2)

which essentially corresponds to the weighted average of the learning models within the cluster. Furthermore, we represent by $L(\theta^S;x)$ the loss of learning model given by parameters $\theta^S$. The expected value of loss function is given by

$$E_x [L(\theta^S)] = \sum_{x \in A} L(\theta^S;x)P[x]$$  

(3)

$$P[x] = \prod_{i \in N} p_i^{x_i} (1 - p_i)^{1-x_i}$$  

(4)

where $X$ with $|X| = 2^n$ is the set of all possible combinations of $x$ vectors. On the other hand, given $x$ and cluster $S$, the loss of aggregated model due to $S$ is lower than the loss averaged over the agents in $S$:

$$L(\theta^S;x) \leq \frac{\sum_{i \in S} x_i L(\theta_i)}{\sum_{i \in S} x_i}$$  

(5)

due to the fact that when the disjoint agents are merged, the amount of data used to train the model increases which results in lower training error. If two disjoint clusters $S$ and $T$, i.e. $S \cap T = \emptyset$, are federated, then we denote the new parameters as $\theta^{S\cup T}$. It is reasonable to assume that the minimum of loss of $\theta^{S\cup T}$ is lower than the minimum of average loss of $\theta^S$ and $\theta^T$:

$$L(\theta^{S\cup T};x) \leq \frac{\sum_{i \in S} x_i L(\theta_i)}{\sum_{i \in S} x_i} + \frac{\sum_{i \in T} x_i L(\theta_i)}{\sum_{i \in T} x_i}$$  

(6)

Moreover, we consider that there exists a communication cost, denoted $c$, when MAE receives the learning model’s parameters’ data; note that this data increases with the size of cluster. On the other hand, MAE earns a monetary gain $f : R \rightarrow R$ by utilizing the aggregated model and commits a monetary value which can be paid to agents before deducting the communication cost $c$. We represent by $u$ the utility which assigns a real value for every subset of $N$, i.e. $u : 2^N \rightarrow R$ where $2^N$ is the collection of all possible non-empty subsets of $N$ and empty set $\emptyset$, and we set $u(\emptyset) = 0$. MAE determines the utility of any cluster $S \in 2^N$ as follows

$$u(S) = f \left( \frac{1}{E_x [L(\theta^S)]} - c(S) \right)$$  

(7)

where $f : R \rightarrow R$ is a monotonically increasing function and inversely proportional to $E_x [L(\theta^S)]$ meaning that the less loss the more gain. Note that this is the monetary value that MAE can pay to cluster $S$.

Any agent $i$ can join a cluster if guaranteed to be paid at least $u(i) = \pi_i$ price which is given by

$$\pi_i = f \left( \frac{p_i}{E_x [L(\theta_i)]} \right)$$  

(8)
Besides, $p_i$ is the second parameter which has an impact on the price asked by agent. It corresponds to the fact that as $p_i$ has a poor value, the agent asks lower price to participate in the federation. We define the utility received by agent $i$ being in cluster $S$ as following:

$$\text{utility of agent } i = \pi_i + \phi_i^S$$  \hspace{1cm} (9)

where $\phi_i^S \in \mathbb{R}$ is the gain of agent $i$ by joining cluster $S$, and we set $\phi_i^S = 0$ for all $i \in N$.

### 2.2 Optimal Clustering

MAE aims at finding the clustering that results in minimal cost. Optimal clustering problem is defined through the agents’ set $N$ and a clustering set $\Pi$ which partitions the players’ set $N$ such that $\bigcup_{\pi \in \Pi} \pi = N$. All clusters in $\Pi$ are disjoint clusters, i.e., $S \cap T = \emptyset$ for all $S, T \in \Pi$. Given $\mathcal{P}$, the set of all possible clustering structures, the optimal clustering problem is to find a clustering $\Pi \in \mathcal{P}$ which minimizes the objective while satisfying the constraints of agents:

$$\min_{\pi \in \Pi} \sum_{S \in \pi} u(S) \text{ subject to } \sum_{i \in S} \pi_i \leq u(S), \ \forall S \in \Pi,$$  \hspace{1cm} (10)

where the constraints in eq. (10) ensure that the demand of agents are satisfied.

Considering that the agents strategically decide to which cluster to join, we can define the problem of clustering as a coalition formation game. We then change the language of problem formulation using game theoretic terms, i.e.

$$\text{agents } \rightarrow \text{ players, cluster } \rightarrow \text{ coalition}$$

Players may not be always fully cooperative and behave selfishly. We then face the problem of finding coalition structures which are stable under selfishness. In the sequel, we deal with such a setting.

### 3 HEDONIC GAME

A hedonic coalition formation game (in short, hedonic game) is given by a pair $\langle N, \succ \rangle$, where $\succ = (\succ_1, \succ_2, \ldots, \succ_n)$ denotes the preference profile, specifying for each player $i \in N$ his preference relation $\succ_i$, i.e. a reflexive, complete and transitive binary relation.

Given $\Pi$ and $i$, $\succ_i(i)$ denotes the set $S \in \Pi$ such that $i \in S$. Moreover, $\mathcal{P}$ is the set of all possible clustering partitions over $N$. In its partition form, a coalition formation game is defined on the set $N$ by associating a utility value $u(S|\Pi)$ to each subset of any partition $\Pi$ of $N$. In its characteristic form, the utility value of a set is independent of the other coalitions, and therefore, $u(S|\Pi) = u(S)$. The games of this form are more restrictive but present interesting properties to reach an equilibrium. Practically speaking, this assumption means that the gain of a group is independent of the other players outside the group. Hedonic games fall into this category with an additional assumption:

**Definition 3.1.** A coalition formation game is hedonic if

- the gain of any player depends solely on the members of the coalition to which the player belongs, and
- the coalitions arise as a result of the preferences of the players over their possible coalitions’ set.

### 3.1 Preference Relation

The preference relation of a player can be defined over a preference function. We consider the case where the preference relation is chosen to be the utility allocated to the player in a coalition. Thus, player $i$ prefers the coalition $S$ to $T$ iff,

$$\phi_i^S \geq \phi_i^T \Rightarrow S \succeq_i T.$$  \hspace{1cm} (11)

### 3.2 The Nash Stability

The stability concepts for a hedonic game are various. In the literature, a hedonic game is said individually stable, Nash stable, core stable, strict core stable, Pareto optimal, strong Nash stable, or, strict strong Nash stable. We refer to [1] for a thorough definition of these different stability concepts. In this paper, we are only interested in the Nash stability because the players do not cooperate to take their decisions jointly.

**Definition 3.2 (Nash Stability).** A partition of players is Nash-stable whenever no player has incentive to unilaterally change his or her coalition to another coalition in the partition which can be mathematically formulated as: partition $\Pi^{NS}$ is said to be Nash-stable if no player can benefit from moving from his coalition $S_{\Pi^{NS}}(i)$ to another existing coalition $T \in \Pi^{NS}$, i.e.:

$$S_{\Pi^{NS}}(i) \succeq_i T \cup i, \ \forall T \in \Pi^{NS} \cup \emptyset; \forall i \in N.$$  \hspace{1cm} (12)

which can be similarly defined over preference function as following:

$$\phi^S_{\Pi^{NS}}(i) \succeq \phi^{T,ij}_i, \ \forall T \in \Pi^{NS} \cup \emptyset; \forall i \in N.$$  \hspace{1cm} (13)

Nash-stable partitions are immune to individual movements even when a player who wants to change does not need permission to join or leave an existing coalition [3].

**Remark 3.1.** Stability concepts being immune to individual deviation are Nash stability, individual stability, contractual individual stability. Nash stability is the strongest within above. The notion of core stability has been used already in some models where immunity to coalition deviation is required [7].

**Remark 3.2.** In [2], the authors propose some set of axioms which are non-emptiness, symmetry pareto optimality, self-consistency, and they analyze the existence of any stability concept that can satisfy these axioms. It is proven that for any game $|N| > 2$, there does not exist any solution which satisfies these axioms. In this work, we show how non-emptiness can be guaranteed using Nash stability as a solution concept.

### 3.3 Aggregated Learning Model Parameters

When a stable partition exists, then this means that all the players (agents) are agreed to participate to federation. As a result of this, MAE utilizes the following aggregation of learning model parameters:

$$\theta^F = w \theta_{\text{MAE}} + (1 - w) \frac{\sum_{i \in N} \chi_i m_i \theta_i}{\sum_{i \in N} \chi_i m_i}$$  \hspace{1cm} (14)

where $0 \leq w \leq 1$ is a weighting parameter showing how much MAE favors the aggregated learning model parameters of agents (players), $\theta_{\text{MAE}}$ shows the learning parameters of MAE’s local model. In summary, we have the following procedure:
Given \( \theta^F \), the expected value of loss function in federation can be calculated as following:

\[
\mathbb{E}_x[\mathcal{L}(\theta^F)] = \sum_{x \in \mathcal{X}} \mathcal{L}(\theta^F; x)\mathbb{P}[x] \\
\geq \mathcal{L}(\mathbb{E}_x[\theta^F]) \quad \text{(Jensen’s inequality)} \tag{15}
\]

where

\[
\mathbb{E}_x[\theta^F; x] = w\theta_{MAE} + (1 - w) \sum_{i \in \mathcal{X}} \frac{\sum_{i \in \mathcal{X}} x_i m_i \theta_i}{\sum_{i \in \mathcal{X}} x_i m_i} \mathbb{P}[x] \tag{16}
\]

### 4 THE NASH-STABLE SET

As the utility \( u \) associated with all possible coalitions are known, we are interested in finding a utility distribution to ensure Nash stability. We thus define a utility allocation method \( \phi \in \mathbb{R}^\kappa \) where \( \kappa = n2^{n-1} \) as following:

\[
\phi = \{ \phi_i^S : \forall i \in S, S \in 2^N \} \tag{17}
\]

which directly sets up a preference profile. The set of all possible utility allocation methods is denoted by \( \mathcal{F} \subset \mathbb{R}^\kappa \). Let us define the mapping \( M : \mathcal{F} \rightarrow \mathcal{P} \). Clearly, for any utility allocation method \( \phi \), mapping \( M \) finds corresponding partition, i.e. \( M(\phi) \in \mathcal{P} \). We define the Nash-stable set which includes all those efficient allocation methods that build the following set:

\[
\mathcal{N}_{stable} = \{ \phi \in \mathbb{R}^\kappa : SM(\phi) (i) \geq T \cup i, \forall T \in M(\phi) \cup \emptyset; \forall i \in N \}. \tag{18}
\]

Let us define the set of constraints stemming from the preference function in order to check if the Nash-stable set is non-empty. Due to the utility bound, for any utility allocation method \( \phi \), we have \( \sum_{i \in S}(\pi_i + \phi_i^S) \leq u(S) \) for all \( S \in 2^N \) called as ‘budged balanced’ utility allocation which further can be given by

\[
\sum_{i \in S} f \left( \frac{p_i}{\mathcal{L}(\theta_i)} \right) + \sum_{i \in S} \phi_i^S \leq f \left( \frac{1}{\mathcal{L}(\theta_i)} \right) - c(S), \forall S \in 2^N. \tag{19}
\]

For simplicity, let us define marginal utility as following:

\[
\Delta_\theta(S) = \begin{cases} 
\frac{f \left( \frac{1}{\mathcal{L}(\theta_i)} \right) - c(S) - \sum_{i \in S} f \left( \frac{p_i}{\mathcal{L}(\theta_i)} \right)}{s} & \text{if } S \in 2^N \setminus i, \forall i \in N \\
0 & \text{if } i \in N.
\end{cases} \tag{20}
\]

which results in the following constraints:

\[
\varphi_i^S(\phi) = \left\{ \sum_{i \in S} \phi_i^S \leq \Delta_\theta(S), \forall S \in 2^N \right\}. \tag{21}
\]

On the other hand, for any \( \phi \), the constraints that ensure the Nash stability are given by

\[
\varphi_i^T(\phi) = \left\{ \sum_{i \in S} \phi_i^S(i) \geq \delta_i^{Tij}, \forall T \in M(\phi) \cup \emptyset; \forall i \in N \right\}. \tag{22}
\]

Then, the Nash-stable set is non-empty, iif:

\[
\mathcal{N}_{stable}(\theta) = \{ \phi \in \mathbb{R}^\kappa : \varphi_i^1(\phi) \text{ and } \varphi_i^2(\phi) \}, \tag{23}
\]

which allows us to conclude:

**Theorem 4.1.** The Nash-stable set can be non-empty.

**Proof.** We can check if the Nash-stable is non-empty by solving the following optimization problem:

\[
\max \phi \sum_{i \in S} \sum_{i \in T} \phi_i^S \text{ subject to } \varphi_i^1(\phi) \text{ and } \varphi_i^2(\phi).
\]

If there exists any feasible solution of this problem, then we conclude that there is at least one utility allocation method which provides a Nash-stable partition. \( \square \)

However, searching in an exhaustive manner over the whole partitions is NP-hard as the number of partitions grows according to the Bell number. Typically, with only 10 players, the number of partitions is as large as 115, 975.

#### 4.1 Superadditive Utility

If the utility function \( u \) is superadditive, then it is trivial to check that the marginal utility is also superadditive: \( \Delta_\theta(S \cup T) \geq \Delta_\theta(S) + \Delta_\theta(T) \) for all possible \( S \) and \( T \) such that \( S \cap T = \emptyset \). Due to eq. (19), we have

\[
\sum_{i \in S \cup T} \phi_i^{S \cup T} = \sum_{i \in S} \phi_i^S + \sum_{i \in T} \phi_i^T \leq \Delta_\theta(S \cup T)
\]

\[
\sum_{i \in S} \phi_i^S + \sum_{i \in T} \phi_i^T \leq \Delta_\theta(S) + \Delta_\theta(T)
\]

\[
\Rightarrow \sum_{i \in S} \phi_i^{S \cup T} + \sum_{i \in T} \phi_i^{S \cup T} \geq \Delta_\theta(S) + \Delta_\theta(T)
\]

This result means that any player is better off in a larger coalition which ultimately all players have the most gain in the grand coalition. This is obvious from eq. (22) where for every player \( i \in N \), \( \phi_i^S \geq \phi_i^T \) for all \( S \in 2^N \).

#### 4.2 Additively Separable and Symmetric Utility

Preferences of a player are additively separable whenever the preference can be stated with a function characterizing how a player prefers another player in each coalition. This means that the player’s preference for a coalition is based on individual preferences. This can be formalized as follows:

**Definition 4.2.** The preferences of a player are said to be additively separable if there exists a function \( v_i : N \rightarrow \mathbb{R} \) such that

\[
\sum_{j \in T} v_i(j) \geq v_i(j) \Leftrightarrow S \geq T, \forall S, T \in 2^N. \tag{24}
\]

Then, \( v_i(i) \) is normalized and set to \( v_i(i) = 0 \). A profile of additively separable preferences satisfies symmetry if \( v(i) = v(j) = v(i, j) \) for all \( i, j \in N \). The meaning of \( v(i, j) \) is the mutual gain of player \( i \) and \( j \) when they are in the same coalition. Let \( V(S) \) be the all possible bipartite coalitions which can occur in coalition \( S \) such
We then define \( v \in \mathbb{R}^{V(N)} \) which shall serve as a utility allocation method to generate additively separable and symmetric preferences: 
\[
v = \{ (i, j) : \forall (i, j) \in V(N) \}
\]
and mapping \( M \) shall find corresponding partition, i.e. \( M(v) \in \mathcal{P} \). The constraints that define the Nash-stable set are then defined over \( v \):
\[
C^1(\theta) \rightarrow C^2(\theta) \text{ and } C^2(\theta) \rightarrow C^3(\theta). 
\]
Further, note that the utility that player \( i \) gains in coalition \( S \) is given by
\[
\pi_i + \phi_i^S = \frac{1}{2}(\sigma_i) + \sum_{j \in S} \hat{v}(i, j) 
\]
\[
\Rightarrow \phi_i^S = \sum_{j \in S} v(i, j) \quad (25)
\]
On the other hand, due to the symmetry property of mutual gain, we have the following:
\[
\sum_{i, j \in S} v(i, j) = 2 \sum_{i, j \in V(S)} v(i, j).
\]
For example, if \( S = (1, 2, 3) \), then \( \sum_{i, j \in S} v(i, j) = 2[\hat{v}(1, 2) + \hat{v}(1, 3) + \hat{v}(2, 3)] \).

**Theorem 4.3.** Additively separable and symmetric preferences always admit a Nash-stable partition. Therefore, constraints in \( C^3(\theta) \) are always satisfied [7].

Based on this theorem, we only need to satisfy the constraints given by \( C^3(\theta) \). Thus, we define the Nash-stable set which generates additively separable and symmetric preferences \( \mathcal{N}^A_{\text{stable}}(\theta) \subset \mathcal{N}_{\text{stable}}(\theta) \) as following:

\[
\mathcal{N}^A_{\text{stable}}(\theta) = \left\{ v \in \mathbb{R}^{V(N)} : \sum_{i \in V(S)} v(i, j) \leq \frac{\Lambda v(S)}{2}, \forall S \in 2^N \right\}
\]

Finding the values of \( v(i, j) \) in eq. (26) satisfying \( C^3(\theta) \) conditions can be done straightforward. However, we propose to formulate as an optimization problem for finding the values of \( v(i, j) \). A feasible solution of the following linear program guarantees the non-emptiness of \( \mathcal{N}^A_{\text{stable}}(\theta) \):

\[
\max_{v} \sum_{(i, j) \in V(N)} v(i, j) \text{ subject to } \\
\sum_{(i, j) \in V(N)} v(i, j) \leq \frac{\Lambda v(S)}{2}, \forall S \in 2^N, \quad (27)
\]
where note that any feasible solution \( v^* \) is upper bounded by \( \sum (i, j) \in V(N) v^*(i, j) \leq \Lambda v(N)/2 \). Furthermore, the coalition partition that stems from \( v^* \) is given by \( \Pi^\text{NS} := M(v^*) \) which is Nash-stable.

### 5 Decentralized Coalition Partition

In this section, we study finding a Nash-stable partition in a decentralized setting. We, in fact, model the problem of finding a Nash-stable partition as a non-cooperative game. A hedonic coalition formation game is equivalent to a non-cooperative game. Denote as \( \Sigma \) the set of strategies. We assume that the number of strategies is equal to the number of players, i.e. \( |\Sigma| = |N| \). This is sufficient to represent all possible choices. Indeed, the players that select the same strategy are interpreted as a coalition. For example, if every player chooses different strategies, then this corresponds to the coalition partition comprised of singletons.

Consider the best-reply dynamics where in a particular step \( l \), only one player chooses its best strategy. A strategy tuple in step \( l \) is denoted as \( \sigma^l = \{ \sigma_1^l, \sigma_2^l, \ldots, \sigma_n^l \} \), where \( \sigma_i^l \) is the strategy of player \( i \) in step \( l \). In every step, only one dimension is changed in \( \sigma^l \). We further denote as \( \Pi(\sigma^l) \) the partition in step \( l \). Define as \( S_{\sigma^l}(i) = \{ j : \sigma_j^l = \sigma_i^l, \forall j \in N \} \) the set of players that share the same strategy with player \( i \). Thus, note that \( \cup_{i \in N} S_{\sigma^l}(i) = N \) for each step. The utility of player \( i \) is \( \phi_i(\sigma^l) \) and verifies the following equivalence:

\[
\phi_i(\sigma^l) \geq \phi_i(\sigma^{l-1}) \Leftrightarrow S_{\sigma^l}(i) \geq_i S_{\sigma^{l-1}}(i), \quad (28)
\]
where player \( i \) is the one that takes its turn in step \( l \). Any sequence of strategy-tuple in which each strategy-tuple differs from the preceding one in only one coordinate is called a path, and a unique deviator in each step strictly increases the utility he receives is an improvement path. Obviously, any maximal improvement path which is an improvement path that can not be extended is terminated by stability.

#### 5.1 Stability Analysis

The Nash stability is defined as following:

\[
\tilde{\sigma}_i^{\text{NS}} \in \arg \max_{\sigma_i \in \Sigma} \phi_i(\sigma_i, \sigma_i^{-i}), \forall i \in N. \quad (29)
\]

Moreover, Nash-stable coalition partition is given by collection:

\[
S_{\sigma^{\text{NS}}}(j) = \{ k \in N : \sigma_k^{\text{NS}} = j \}, \quad \forall j \in \Sigma \\
\Pi^{\text{NS}} = \{ S_{\sigma^{\text{NS}}}(j) : \forall j \in \Sigma \}. \quad (30)
\]

In the sequel, we prove that the additively separable and symmetric utilities result in a potential game.

**Theorem 5.1.** Any additively separable and symmetric utility results in a potential game with potential:

\[
P_v(\sigma) = \sum_{S \in \Pi(\sigma)} \sum_{(i, j) \in V(S)} v(i, j). \quad (31)
\]

**Proof.** A non-cooperative game is a potential game whenever there exists a function \( P_v \) such that:

\[
P_v(\sigma_1, \sigma_i^{-i}) - P_v(\sigma_i', \sigma_i^{-i}) = \phi(\sigma_i, \sigma_i^{-i}) - \phi(\sigma_i', \sigma_i^{-i})
\]
where \( \sigma_i, \sigma_i^{-i} = \sigma \). This means that when player \( i \) switches from strategy \( \sigma_i \) to \( \sigma_i' \) the difference of its utility can be given...
by the difference of a function $P$. We choose the following potential function:

$$P_v(\sigma) = \sum_{S \in \Pi(\sigma)} \sum_{i,j \in V(S)} v(i, j)$$

(32)

Let us denote as $i \in S$ and $i \notin S'$ the coalitions when player $i$ switches from strategy $\sigma_i$ to $\sigma'_i$, respectively. Potential function is given as following before and after switching

$$P_v(\sigma_i, \sigma_{-i}) = \sum_{(i,j) \in V(S)} v(i, j) + \sum_{(k,j) \in V(S')} v(k, j)$$

$$+ \sum_{T \in \Pi(\sigma) \setminus \{S,S'\}} \sum_{(k,j) \in V(T)} v(k, j)$$

$$P_v(\sigma'_i, \sigma_{-i}) = \sum_{(k,j) \in V(S')} v(k, j) + \sum_{(i,j) \in V(S \cup i)} v(i, j)$$

$$+ \sum_{T \in \Pi(\sigma) \setminus \{S,S'\}} \sum_{(k,j) \in V(T)} v(k, j)$$

where note that we have $S \rightarrow S \setminus i$ and $S' \rightarrow S' \cup i$ after switching. Thus, we have

$$P_v(\sigma_i, \sigma_{-i}) - P_v(\sigma'_i, \sigma_{-i}) =$$

$$= \sum_{(i,j) \in V(S)} v(i, j) + \sum_{(k,j) \in V(S')} v(k, j)$$

$$- \sum_{(k,j) \in V(S')} v(k, j) - \sum_{(i,j) \in V(S \cup i)} v(i, j)$$

$$+ \sum_{T \in \Pi(\sigma) \setminus \{S,S'\}} \sum_{(k,j) \in V(T)} v(k, j)$$

On the other hand, the utility shift before and after strategy switch is given by

$$\phi(\sigma_i, \sigma_{-i}) - \phi(\sigma'_i, \sigma_{-i}) = \sum_{j \in S} v(i, j) - \sum_{j \in S' \cup i} v(i, j)$$

which concludes the proof that $P_v(\sigma_i, \sigma_{-i}) - P_v(\sigma'_i, \sigma_{-i}) = \phi(\sigma_i, \sigma_{-i}) - \phi(\sigma'_i, \sigma_{-i})$.

In a potential game, a Nash equilibrium shall result in an optimum in potential $P_v$. Therefore, $\sigma^* \in \arg \max_{\sigma} P_v(\sigma)$ corresponds to a coalition partition $\Pi(\sigma^*) = M(\psi)$ which is Nash-stable.

### 6 CONCLUSIONS

We analyzed stable clustering problem in federated learning setting. Clusters are made up of the agents contributing to federated learning. We considered that every agent gains a utility when switching from one cluster to another one. We modeled the decisions of agents in the framework hedonic games which is a widely used cooperative game model for this type of problems. A fundamental question in hedonic games is to analyze the conditions how stable coalition partitions can occur. We studied the existence of stable coalition partitions by introducing the Nash-stable set, and analyzed the existence of decentralized coalition partitions.