An Efficient and Decentralised User Association Scheme for Multiple Technology Networks

Mannat Singh and Prasanna Chaporkar
Department of Electrical Engineering
Indian Institute of Technology Bombay
Powai, Mumbai, 400076, India
Email: mannat@iitb.ac.in, chaporkar@ee.iitb.ac.in

***Abstract*—Wireless networks have emerged as the most popular access technologies. Multiple technologies like WiFi, WiMAX, LTE, along with traditional cellular networks augmented with pico and femto base stations, are available for a user to access the Internet. Many emerging user devices, e.g. smart phones, allow for Internet connectivity using most of the available technologies. Thus, at a given point in time, a user can be in the communication range of many base stations, potentially using different access technologies. This naturally gives a user a choice as to which base station he should select for network access. The association chosen by a user determines not only his own performance, but also the performance perceived by other users. Though user performances are coupled, association decisions can seldom be taken in a centralised manner. This is because different access technologies may not share the required control information and cooperate. Also, different user devices may not want to divulge their own information, for privacy or security reasons, and also on account of additional power requirement for control message exchange. Thus, one needs to devise distributed association schemes that do not require any message passing. In this paper, we consider the problem of maximising network utility subject to constraints on user requirements. To this end, we propose a distributed association scheme in which a user chooses his association based only on his own past association and the utility he obtained. We prove that it is indeed possible to achieve maximum network utility while satisfying individual user requirements in a completely distributed manner. We also evaluate the performance of the proposed scheme using simulations.**

I. Introduction

Wireless networks are emerging as the preferred access networks. In addition to the traditional WiFi networks, WiMAX and LTE networks are becoming available for Internet access. Moreover, cellular networks are evolving into HetNets with the addition of picocells and femtocells. Ubiquitous networking ensures that user devices have the ability to connect to the Internet through any of the available access technologies. Thus, at a given point in time, a user may lie in the range of multiple base stations, using possibly multiple technologies. This naturally leads to the question, which base station should a user connect to (*user association*)? The decision may depend on the performance that a user can get and the cost he incurs by connecting to various available base stations. In this paper, *our aim is to propose a completely distributed user association algorithm that maximises network utility while satisfying individual user requirements, whenever possible.*

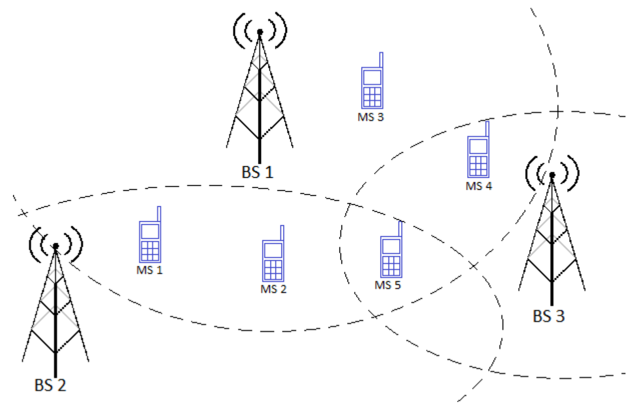


Fig. 1. A sample network topology

Let us first demonstrate the trade-offs that a user association scheme needs to address. The demonstration is through the following examples: consider a network topology with three base stations BS_k for $k = 1, 2, 3$ and five users (mobile stations) MS_i for $i = 1, \dots, 5$, as shown in Fig. 1. The dotted circle centered at BS_k denotes the boundary of its coverage area, i.e., MS_i can associate and get nonzero rate from BS_k only if it lies within the dotted circle centered at BS_k. We assume that the time is slotted and multiple access control (MAC) used at all the base stations is Time Division Multiple Access (TDMA). Also, consider a saturated system, i.e., users always have data to send/receive.

Example 1: Here, let us assume that the base stations do not have rate adaptation capability. Thus, each user can communicate 1 packet/slot, when he is scheduled by the base station to which he chose to associate. Now, consider the following two user associations: (1) All users associate to BS_1, and (2) users MS_1 and MS_2 associate with BS_2, user MS_3 associates with BS_1 and the remaining associate with BS_3. Note that the total system throughput in case of association (1) is 1, whereas it equals 3 in case of association (2). This example, though simplistic, demonstrates the importance of load balancing in the association in order to achieve good system performance. Also, though the example seems contrived, the association in (1) can happen in practice on account of user preference for a certain technology. For example, in case of Unlicensed Mobile Access (UMA), priority is always given to WiFi over

Universal Mobile Telecommunications System (UMTS) [1]. Thus, if BS_1 were a WiFi access point and the remaining were UMTS base stations, then the user association will be given by (1).

Example 2: Here, let us consider a scenario in which the base stations have rate adaptation capability. Thus, based on the user's channel quality, his rate can be appropriately chosen. We assume slow fading, i.e., the channel quality does not vary with time. Let $\boldsymbol{r}_k = [r_{k1} \ \cdots r_{k5}]$ denote the rate vector for BS_k, where r_{ki} is the transmission rate (in packets/slot) to user i, if it associates to BS_k. We assume the following rate vectors: $\boldsymbol{r}_1 = [4 \ 6 \ 6 \ 6 \ 6]$, $\boldsymbol{r}_2 = [6 \ 2 \ 0 \ 0 \ 1]$, and $\boldsymbol{r}_3 = [0 \ 0 \ 0 \ 5 \ 5]$. Consider the following two associations: (1) All users except MS_1 associate with BS_1 and MS_1 associates with BS_2, and (2) user MS_1 associates with BS_2, users MS_2 and MS_3 associate with BS_1 and the remaining associate with BS_3. Observe that the total system throughput in case of association (1) is 12, whereas it equals 17 in case of association (2). This example shows that associating to the base station with the best channel condition does not guarantee the best performance. Note that in conventional Wifi networks based on IEEE 802.11 standards, a user connects to the access point with the best Received Signal Strength Indication (RSSI) [2].

Example 3: Consider the same setting as in Example 2. In addition, let us assume that each user has certain minimum throughput requirements. Specifically, we assume that the minimum rate requirements of the users MS_1 through MS_5 are 3, 3, 6, 2 and 2 respectively. In this case, observe that both the association schemes proposed in Example 2 do not satisfy the users' requirements. However, an association scheme in which MS_1 and MS_2 associate to BS_1, MS_3 associates to BS_2, and the rest associate to BS_3 satisfies the requirements of all the users. Thus, user requirements play a significant role in determining a user association.

Our aim in this paper is to propose a user association scheme that maximises the total network utility subject to satisfying individual user requirements, whenever doing so is feasible. Key features of our scheme are that it is completely distributed and does not need any message passing. Specifically, we treat each user as a separate entity who periodically decides his association based only on the performance he has observed in the past. Thus, our proposed algorithm does not require any information from the network. This feature is crucial, as, unlike in traditional cellular networks, heterogeneity in the available access technologies means that there may not exist a backbone connecting all the base stations. Thus, the network wide state is not known at each base station. Moreover, depending on the underlying access technology, the base station may or may not be able to assist users with information that aids choosing an appropriate association. Observe that optimally addressing the above trade-offs in a distributed manner without any assistance from the network is a challenging task. Indeed, prima facie, it seems magical for the users to be able to find an association that maximises network utility without requiring any information about the remaining network (other users, their past associations and the utilities they have obtained). However, in their seminal work, the authors, in [3], have proposed a randomised algorithm for optimising social utility in games in which each player knows nothing about the actions and utilities of the other players. Our proposed algorithm is motivated by the algorithm in [3]. Here, we generalise the algorithm of [3] to account for individual user constraints. At this point we would like to mention that most of the existing work in the area of user association is either a centralised scheme (e.g. see [4], [5]) or a scheme that takes inputs from the network (e.g. see [1], [6], [7]). To the best of our knowledge, ours is the first optimal distributed algorithm that does not require any message passing.

The rest of the paper is laid out as follows: In Section II, we describe our network model and problem formulation. In Section III, we describe the proposed algorithm and the intuition behind it. In Section IV, we provide the numerical simulations. In Section V, we provide the literature survey. Finally, we conclude in Section VI.

II. System Model and Problem Formulation

We consider a network with N users and K base stations. Let $\mathcal{N} = \{1, 2, 3, ..., N\}$ and $\mathcal{K} = \{1, 2, 3, ..., K\}$ be the sets of all the users and all the base stations, respectively. For each user $i \in \mathcal{N}$, let $\mathcal{K}_i \subseteq \mathcal{K}$ denote the set of base stations to which i can associate. We do not put any restrictions on the technology employed by any of the base stations. All the base stations can potentially employ different technologies for providing connectivity to the users. Let $\mathcal{B}$ denote the set of all possible associations in the network. We denote an element of $\mathcal{B}$ as N-dimensional vector $\boldsymbol{b}_\ell = [b_{1\ell} \ \cdots \ b_{N\ell}]$, where $b_{i\ell} \in \mathcal{K}_i$. We denote by $r_{i\ell}$ and $c_{i\ell}$ the throughput obtained and cost incurred, respectively, by user i under association $\boldsymbol{b}_\ell$. Cost can be on account of access fee for the technology used by $b_{i\ell}$. Next, we assume that a user has requirements on the achieved throughput and also has constraints on the incurred cost. We capture these through a function $f_i : \mathbb{R}^2 \to \{0, 1\}$. If $f_i(r_{i\ell}, c_{i\ell}) = 1$, then we say that the user requirements are met (user is *satisfied*) under association $\boldsymbol{b}_\ell$, otherwise the user is said to be *unsatisfied*. Note that the function $f_i(\cdot)$ can model any user requirement. For example, if user i desires his throughput to be at least ξ_i and cost to be at most ζ_i, then we can can choose $f_i(r_{i\ell}, c_{i\ell}) = 1$ only when $r_{i\ell} \geq \xi_i$ and $c_{i\ell} \leq \zeta_i$. Finally, let $u_{i\ell}$ denote the utility for user i under association $\boldsymbol{b}_\ell$. Without loss of generality, we assume that $u_{i\ell} \in [0, 1)$ for every i and ℓ. Thus, network utility under association $\boldsymbol{b}_\ell$ is $\sum_{i \in \mathcal{N}} u_{i\ell}$. Let $\mathcal{U}_i$ denote the set containing all the possible utilities the player i can receive, i.e., for every configuration $\boldsymbol{b}_\ell \in \mathcal{B}$, the utility the player i receives using $\boldsymbol{b}_\ell$, $u_{i\ell}$, lies in $\mathcal{U}_i$. Let $\boldsymbol{u}_\ell$ denote the N-dimensional vector $[u_{1\ell} \ \cdots \ u_{N\ell}]$. Our aim is to find a user association that maximises network utility subject to satisfying throughput and cost constraints. Formally, we solve the following optimisation:

$$\begin{aligned} \max_{\boldsymbol{b}_\ell \in \mathcal{B}} \quad & \sum_{i \in \mathcal{N}} u_{i\ell} \\ \text{s.t.} \quad & f_i(r_{i\ell}, c_{i\ell}) = 1 \quad \forall i \in \mathcal{N}. \end{aligned}$$

Note that when $u_{i\ell} = \log(r_{i\ell})$ for every i, then the above optimisation program obtains proportionally fair user association, subject to the throughput and cost constraints. We can recover other notions of fairness like α-fairness (see [8]) by choosing an appropriate utility function.

We require that the notion of interdependence holds in such a scenario. We first define what we mean by interdependence.

Definition 1: By interdependence, it is meant that for every association $\boldsymbol{b}_\ell \in \mathcal{B}$, and every proper subset of users $\mathcal{J} \subset \mathcal{N}$, there exists a user $i \notin \mathcal{J}$, and a configuration of associations for the subset $\mathcal{J}$ of users, $\boldsymbol{b'}_{\mathcal{J}\ell} \in \prod_{m\in\mathcal{J}} \mathcal{K}_m$, such that the utility of the agent i does not stay unaffected, i.e. $u_i(\boldsymbol{b'}_{\mathcal{J}\ell}, \boldsymbol{b}_{-\mathcal{J}\ell}) \neq u_i(\boldsymbol{b}_{\mathcal{J}\ell}, \boldsymbol{b}_{-\mathcal{J}\ell})$ [3].

Note that this notion of interdependence seems to be quite valid in the case of wireless networks. Since, for a given profile, if we take any proper subset of players $\mathcal{J} \subset \mathcal{N}$ and change only their strategies, that would affect the utilities of at least one player not in this subset. The throughput received by some player $i \notin \mathcal{J}$ might change due to the alteration of the associations of the base station i is associated to. Even if this does not happen, the interferences observed by the players will get affected due to a change in strategy of the players in $\mathcal{J}$, thus resulting in a change in the throughputs of at least one player $i' \notin \mathcal{J}$. Also note that if the set $\mathcal{N}$ can be partitioned into two disjoint sets of mutually non-interacting users, the two sets can form two independent games, in each of which, interdependence holds.

In the following section, we describe our proposed algorithm.

III. Association Algorithm For Constrained Optimum Network Utility

In our proposed algorithm, users periodically assess their performance with their current association and may decide to change it. We assume that the users update their associations synchronously. Let $\{T_j\}_{j\geq 1}$ denote the monotone increasing sequence of update time epochs, i.e., for every user, T_j denotes the instance of j^{th} association update. Let $b_{i\ell(j)} \in \mathcal{K}_i$ denote the base station to which user i has chosen to associate at T_j. Thus, $\boldsymbol{b}_{\ell(j)}$ determines the user associations in the network for the time interval $[T_j, T_{j+1})$. We refer to $[T_j, T_{j+1})$ as the j^{th} update interval. Let $u_{i\ell(j)}$ denote the utility that user i gets in the interval $[T_j, T_{j+1})$. Each user chooses his next association randomly as per a distribution that depends on the performance perceived by the user. Next, we explain the algorithm in detail.

A. Explanation of the Algorithm

The algorithm is based on randomised search. Here, each user decides his association at random at every T_j. If it were a centralised random search, then upon finding a new random association $\boldsymbol{b}_\ell$, the algorithm will check for its feasibility ($f_i(r_{i\ell}, c_{i\ell}) = 1$ for every i) and whether the new association has a higher value of network utility than that of the previous one. If the new association is feasible and improves upon the previous then we retain it and generate a new association at random. In finite networks, this procedure clearly converges to the optimal association if the set of feasible associations is non-empty. However, the same algorithm can not be used in a distributed manner as users cannot check for overall feasibility or utility improvement based on only the knowledge of their own feasibility and obtained utility. The solution to this challenging problem is addressed as follows:

Each player can have two moods, viz., content or discontent, and these moods affect the decisions taken by a user. Broadly, a user is content only if he strongly believes that the network has reached an optimal association. A content player is very inclined to stay associated to the same base station. But occasionally, he may leave his association in order to look for better opportunities. To change his association, he chooses from the set of base stations, $\mathcal{K}_i$, randomly and uniformly. A discontent player, on the other hand, keeps on trying new associations very frequently in order to become content. He also chooses from $\mathcal{K}_i$, randomly and uniformly. Thus, when all the players are content, the system stays in the same state with a high probability. Thus, at T_j, depending on the mood of a user, he updates his association and gets associated to his chosen base station at T_j. For this chosen association each user measures his utility, throughput and cost in the j^{th} update interval. Thus, $(b_{i\ell(j)}, u_{i\ell(j)})$ is known to each user i at the end of the j^{th} interval. Now, using $(b_{i\ell(j-1)}, u_{i\ell(j-1)}, m_{i\ell(j-1)})$ and $(b_{i\ell(j)}, u_{i\ell(j)})$ user i computes his mood $m_{i\ell(j)}$. The computation is described below.

Let us explain how the mood for a user is updated. First, we consider a scenario in which $m_{i\ell(j-1)} = D$ and explain conditions under which the mood for the user will become content at end of j^{th} update interval. For $m_{i\ell(j)} = C$ to happen, first we need that $f_i(r_{i\ell(j)}, c_{i\ell(j)}) = 1$, otherwise the user will remain discontent. Given that the user constraints are satisfied at the end of the j^{th} update interval, the user changes his mood to content with probability (w.p.) $\epsilon^{1-u_{i\ell(j)}}$, where ϵ is a small parameter for the algorithm. Note that the user is more likely to become content and stop exploring aggressively for higher values of utility obtained. Next, we explain how a user's mood gets updated from content to discontent. A user may update his mood to discontent in any one of the following two scenarios: (i) he changes its own association ($b_{i\ell(j-1)} \neq b_{i\ell(j)}$), or ($ii$) his utility gets changed ($u_{i\ell(j-1)} \neq u_{i\ell(j)}$). In both the scenarios, if $f_i(r_{i\ell(j)}, c_{i\ell(j)}) = 0$, then the user changes his mood to discontent w.p. 1, otherwise the change happens w.p. $1 - \epsilon^{1-u_{i\ell(j)}}$. Note that both ($i$) and ($ii$) indicate that the system configuration has changed. Thus, if the user's own requirements are violated in the new configuration, then the user becomes discontent and explores aggressively. On the other hand, if the user himself is satisfied in the new configuration, then, he decides with a small probability to become content and stops exploring. Else, he stays discontent and continues his search.

Pseudo code for the proposed algorithm is given in Algorithm 1. In the pseudo code, lines 1 to 7 show how user association is updated at T_j. Lines 8 to 18 explain how mood for a user is updated at the end of j^{th} update interval, i.e. at T_{j+1}^{-}.

Algorithm 1

Each user i updates his association at T_j as follows:
if $m_{i\ell(j-1)} = C$ **then**
  $b_{i\ell(j)} = k$ w.p. $\frac{\epsilon^c}{|\mathcal{K}_i|-1}$ for $k \in \mathcal{K}_i \setminus \{b_{i\ell(j-1)}\}$
  $b_{i\ell(j)} = k$ w.p. $1-\epsilon^c$ for $k = b_{i\ell(j-1)}$
else if $m_{i\ell(j-1)} = D$ **then**
  $b_{i\ell(j)} = k$ w.p. $\frac{1}{|\mathcal{K}_i|}$ for all $k \in \mathcal{K}_i$
end if
Each user i updates his mood at T_{j+1}^- as follows:
if $(m_{i\ell(j-1)} = C)$ and $([b_{i\ell(j)}, u_{i\ell(j)}] = [b_{i\ell(j-1)}, u_{i\ell(j-1)}])$ **then**
  $m_{i\ell(j)} \leftarrow C$
else
  if $f(r_{i\ell(j)}, c_{i\ell(j)}) = 1$ **then**
    $m_{i\ell(j)} \leftarrow C$ w.p. ϵ^{1-u_i}
    $m_{i\ell(j)} \leftarrow D$ w.p. $1-\epsilon^{1-u_i}$
  else if $f(r_{i\ell(j)}, c_{i\ell(j)}) = 0$ **then**
    $m_{i\ell(j)} \leftarrow D$
  end if
end if

B. Convergence

In this section, we show that the fraction of time the system spends in states in which all the users' constraints are satisfied and the system utility is maximised can be made as close to 1 as desired by choosing an appropriate $\epsilon > 0$.

Let $\boldsymbol{z}_{i\ell(j)} = [b_{i\ell(j)}, u_{i\ell(j)}, m_{i\ell(j)}]$ be the state of player i at the end of the j^{th} period. Then, $\boldsymbol{z}_{\ell(j)} = [\boldsymbol{z}_{1\ell(j)} \cdots \boldsymbol{z}_{N\ell(j)}]$ is the state of the system at the end of the j^{th} period. Let $\mathcal{Z} = \{\boldsymbol{z}_\ell \ : \ \boldsymbol{b}_\ell \in \mathcal{B}\}$ denote the collection of all possible system states. $\mathcal{Z} \subset \mathcal{S} = \prod_{i\in\mathcal{N}} \mathcal{K}_i \times \mathcal{U}_i \times \mathcal{M}$, where $\mathcal{M} = \{C, D\}$. Note that $\mathcal{Z}$ is a strict subset of $\mathcal{S}$. Specifically, $\mathcal{Z}$ is that subset of $\mathcal{S}$ which contains all the states $[\boldsymbol{b}_\ell, \boldsymbol{u}_\ell, \boldsymbol{m}_\ell] \in \mathcal{S}$, excluding the states in which the utilities and associations are not aligned, i.e. $u_{i\ell} \neq u_i(\boldsymbol{b}_\ell)$ for some $i \in \mathcal{N}$, and also excluding the ones which have $m_{i\ell} = C$ when $f_i(r_{i\ell}, c_{i\ell}) = 0$.

Lemma 1: The system state process $\{\boldsymbol{z}_{\ell(j)}\}_{j=1,2,...}$ is a finite state, irreducible, and aperiodic Markov chain for every $\epsilon > 0$.

Proof: Note from Algorithm 1 that for each user i, at the instance T_j, we first decide an association. The association is solely decided by the mood of the user $m_{i\ell(j-1)}$. Once the association for all the users in decided, i.e. when $\boldsymbol{b}_{\ell(j)}$ is decided, then for every i utility $u_{i\ell(j)}$, throughput $r_{i\ell(j)}$ and cost $c_{i\ell(j)}$ is also decided. Thus, user i can decide his mood $m_{i,\ell(j)}$ based on the values of $f_i(r_{i\ell(j)}, c_{i\ell(j)})$ and $[b_{i\ell(j-1)}, u_{i\ell(j-1)}]$ and $[b_{i\ell(j)}, u_{i\ell(j)}]$. This shows that the state $\boldsymbol{z}_{\ell(j)} = [\boldsymbol{b}_{\ell(j)}, \boldsymbol{u}_{\ell(j)}, \boldsymbol{m}_{\ell(j)}]$ obtained at the end of j^{th} update interval is determined only by the state $\boldsymbol{z}_{\ell(j-1)} = [\boldsymbol{b}_{i\ell(j-1)}, \boldsymbol{u}_{i\ell(j-1)}, \boldsymbol{m}_{i\ell(j-1)}]$ observed at the end of $(j-1)^{\text{th}}$ update interval. Thus, the state evolution process is Markovian.

We know that the set $\mathcal{K}_i$ is finite. The set $\mathcal{U}_i$ is also finite, since for each unique element in $\mathcal{U}_i$, there exists at least one unique association which results in that utility. So, $|\mathcal{U}_i|$ is bounded by $|\mathcal{B}|$. Thus, $\mathcal{Z}$ is finite. The probability of transitioning from any state $\boldsymbol{z}_\ell = [\boldsymbol{b}_\ell, \boldsymbol{u}_\ell, \boldsymbol{m}_\ell] \in \mathcal{Z}$ to any other state $\boldsymbol{z}_{\ell'} = [\boldsymbol{b}_{\ell'}, \boldsymbol{u}_{\ell'}, \boldsymbol{m}_{\ell'}] \in \mathcal{Z}$ is greater than 0. So, every state $\boldsymbol{z}_\ell \in \mathcal{Z}$ communicates with every other state $\boldsymbol{z}_{\ell'} \in \mathcal{Z}$. Thus, the whole state space, $\mathcal{Z}$ is a single communicating class, implying that the process is irreducible. The probability of making a transition in a single step, from $\boldsymbol{z}_\ell \in \mathcal{Z}$, to $\boldsymbol{z}_\ell$ again, is also greater than 0, thus making the process aperiodic. ■

The process is irreducible, aperiodic, and has a finite state space. So, it is ergodic. Let P^ϵ be the transition probability matrix (TPM) for this process and μ^ϵ be its unique stationary distribution, since the process is ergodic. Now, let us define $\mathcal{C}^0$ as the subset of states of $\mathcal{Z}$ in which the mood of each player is content. By $\mathcal{D}^0$, we refer to the subset of states in which each user's mood is discontent.

Though Lemma 1 shows that the state evolution matrix P^ϵ is irreducible for every $\epsilon > 0$, same is not true for $\epsilon = 0$. Consider P^0, the transition probability matrix of a process on the finite state space $\mathcal{Z}$ with $\epsilon = 0$. For our proof, we will view P^ϵ as a regular perturbation of P^0. Let us first understand the topology of P^0. This will be useful in establishing the final result.

Lemma 2: The recurrence classes of the process P^0 are $\mathcal{D}^0$ and all singletons $z \in \mathcal{C}^0$.

Proof: For any two states $\boldsymbol{z}_{\ell_1}, \boldsymbol{z}_{\ell_2} \in \mathcal{D}^0$, the probability of transitioning from $\boldsymbol{z}_{\ell_1}$ to $\boldsymbol{z}_{\ell_2}$ is greater than 0 as each user chooses his association uniformly randomly, and his mood can not be updated to Content as $\epsilon = 0$. Thus, all states in $\mathcal{D}^0$ communicate with each other. Also, there is no possibility of leaving $\mathcal{D}^0$, making $\mathcal{D}^0$ a closed, aperiodic recurrence class of P^0.

Note that a user can be in Content mood only if his constraints are satisfied. Thus, for any state $\boldsymbol{z}_\ell \in \mathcal{C}^0$, the users will keep on choosing the same association with probability 1 as $\epsilon = 0$. Thus, each such state in $\mathcal{C}^0$ is an absorbing state.

Consider a state, $\boldsymbol{z}_\ell$, in which a proper subset of users $\mathcal{J} \subset \mathcal{N}$ are discontent and the associations and the utilities of the users are given by $\boldsymbol{b}_\ell$ and $\boldsymbol{u}_\ell$, respectively. Using interdependence, there exists a user $i \notin \mathcal{J}$ and a strategy $\boldsymbol{b'}_{\mathcal{J}\ell}$ such that $u_{i\ell} \neq u_i(\boldsymbol{b'}_{\mathcal{J}\ell}, \boldsymbol{b}_{\mathcal{J}\ell})$. The users in $\mathcal{J}$ will play this strategy $\boldsymbol{b'}_{\mathcal{J}\ell}$ at some time with probability 1. Thus, the user i will no longer be able to stay content. This argument can be repeated to show that finally all the users become discontent. Therefore, such states are transient and cannot belong to any recurrence class. ■

Now, we consider P^ϵ as a perturbed version of P^0. By perturbations, we mean the users experiment with some small probabilities, i.e., with high probabilities, the users make transitions as per P^0, but with small probabilities, some transitions happen which would not have happened in P^0. Let perturbations be characterized by a scalar $\delta > 0$ in a certain perturbed process. This scalar δ gives a sense of the size of the perturbations. We can see that the process P^ϵ is also a perturbed process, derived from P^0, with δ in this case being equal to ϵ. Using our knowledge about the process P^0, we want to deduce the behavior of the process P^ϵ. If somehow we are able to comment on the fraction of time the system, using the dynamics in P^ϵ, can spend in the optimum utility states, we will have an idea about the performance of the algorithm. For this, we use the concept of stochastically stable states [9].

Definition 2: The stochastically stable states of the process P^ϵ are defined as the states $\boldsymbol{z}_\ell \in \mathcal{Z}$ such that $\lim_{\epsilon\to 0} \mu^\epsilon_{\boldsymbol{z}_\ell} > 0$.

Remark: Note that P^0 does not necessarily have a unique stationary measure. Thus, though $\lim_{\epsilon\to 0} P^\epsilon = P^0$, $\lim_{\epsilon\to 0} \mu^\epsilon_{\boldsymbol{z}_\ell}$ depends on the relative rate at which various perturbed transitions approach 0.

From [10], we know that the time spent in the stochastically stable states can be made to go as close to 1 as desired, by a correct choice of ϵ. Formally,

Statement 1: Given any small $\alpha > 0$, there exists a number $\epsilon_\alpha > 0$ such that for any $0 < \epsilon \le \epsilon_\alpha$, the process will be in one of the stochastically stable states for at least $1-\alpha$ fraction of the times.

If we can show that the stochastically stable states of the process P^ϵ are in fact the states which maximise the system utility while satsifying the user constraints, we will be done. This is what we show in Theorem 1.

Theorem 1: A state of the process P^ϵ, $\boldsymbol{z}_\ell = [\boldsymbol{b}_\ell, \boldsymbol{u}_\ell, \boldsymbol{m}_\ell] \in \mathcal{Z}$, is stochastically stable if and only if:

- The configuration $\boldsymbol{b}_\ell$ maximises the system utility, while satisfying the constraints of all the users ($f_i(r_{i\ell}, c_{i\ell}) = 1 \ \forall i \in \mathcal{N}$)
- The mood of each agent is content, i.e. $m_i = C \ \forall i \in \mathcal{N}$

Proof: We know that P^ϵ is a perturbation of the process P^0. A regular perturbed process is a perturbed process which satisfies the following additional constraints, from [9]:

- P^ϵ is ergodic $\forall \epsilon \in (0, a], a > 0$
- $\lim_{\epsilon\to 0} P^\epsilon = P^0$
- $P^\epsilon_{\ell\ell'} > 0$ for some ϵ implies $\exists\ \varrho_{\ell\ell'} \ge 0$ such that, $0 < \lim_{\epsilon\to 0} \epsilon^{-\varrho_{\ell\ell'}} P^\epsilon_{\ell\ell'} < \infty$.

We call $\varrho_{\ell\ell'}$ as the *resistance* of the transition from the state $\boldsymbol{z}_\ell$ to the state $\boldsymbol{z}_{\ell'}$. Note that $\varrho_{\ell\ell'} = 0$ implies $P^0_{\ell\ell'} > 0$, i.e., the resistance of the $\boldsymbol{z}_\ell \to \boldsymbol{z}_{\ell'}$ transition is zero if the probability of reaching $\boldsymbol{z}_{\ell'}$ from $\boldsymbol{z}_\ell$ is positive in the unperturbed process. Taking examples from our dynamics, if the transition probability of a transition is ϵ^{1-u_i}, its resistance would be $1 - u_i$. On the other hand, if the probability were $1 - \epsilon^{1-u_i}$, then the resistance for the transition would be 0.

As we have already seen, the perturbed process defined by P^ϵ is ergodic. The second requirement, $\lim_{\epsilon\to 0} P^\epsilon_{\ell\ell'} = P^0_{\ell\ell'}\ \forall \boldsymbol{z}_\ell, \boldsymbol{z}_{\ell'} \in \mathcal{Z}$, holds, since P^0 is defined as such. Also, since all the transition probabilities are finite summations of non-negative exponents of ϵ, the resistances for the transitions will be non-negative. So, the process introduced earlier is a regular perturbed Markov process.

Let $\mathcal{Z}_1, \mathcal{Z}_2, ..., \mathcal{Z}_K$ be the recurrence classes of P^0. An $s-t$ path κ is defined as a sequence of states, $\boldsymbol{z}_1 \to \boldsymbol{z}_2 \to \cdots \to \boldsymbol{z}_M$, such that $\boldsymbol{z}_1 \in \mathcal{Z}_s$ and $\boldsymbol{z}_M \in \mathcal{Z}_t$. The resistance of this path is defined as:

$$\varrho(\kappa) = \varrho_{\boldsymbol{z}_1\boldsymbol{z}_2} + \varrho_{\boldsymbol{z}_2\boldsymbol{z}_3} + ... + \varrho_{\boldsymbol{z}_{M-1}\boldsymbol{z}_M}.$$

Let ρ_{st} be the minimum resistance amongst all $s-t$ paths.

Now let us define a directed graph $\tilde{\mathcal{G}}$, with $\mathcal{Z}_1, \mathcal{Z}_2, ..., \mathcal{Z}_K$ as vertices and for each pair of vertices $(\mathcal{Z}_i, \mathcal{Z}_j)$, there is a weighted edge from $\mathcal{Z}_i$ to $\mathcal{Z}_j$, with weight ρ_{ij}. A $j-$tree is a spanning subtree of $\tilde{\mathcal{G}}$ such that for every vertex $\mathcal{Z}_i \neq \mathcal{Z}_j$, there exists exactly one path from $\mathcal{Z}_i$ to $\mathcal{Z}_j$. For each j, find a $j-$tree of least resistance, and the resistance of this $j-$tree is called the stochastic potential, γ_j, of the class $\mathcal{Z}_j$. Let $\mathcal{Z}_1, \mathcal{Z}_2, ..., \mathcal{Z}_K$ be the recurrence classes of P^0. From [9], we get,

- $\lim_{\epsilon\to 0} \mu^\epsilon$ is a stationary distribution μ^0 of P^0
- $\mu^0_{\boldsymbol{z}_\ell} > 0 \iff \boldsymbol{z}_\ell \in \mathcal{Z}_j : \mathcal{Z}_j \in \arg\min_{\mathcal{Z}_k} \gamma_k$.

Thus, we know that the stochastically stable states of P^ϵ are the set of states which lie in the recurrence class(es) of P^0 with minimum stochastic potential. Now, we try to find the stochastic potential of the recurrence classes of P^0.

The resistance of the transition from $\boldsymbol{z}_\ell \in \mathcal{C}^0$ to $\boldsymbol{z}'_\ell \in \mathcal{D}^0$ is c, since change in association of one player can lead the system to a $\mathcal{D}^0$ state.

The resistance of the transition from $\boldsymbol{z}_\ell \in \mathcal{D}^0$ to $\boldsymbol{z}'_\ell \in \mathcal{C}^0$ is $\sum_{i\in\mathcal{N}}(1 - u_{i\ell'})$, since the transition for each user $i \in \mathcal{N}$ has a resistance $(1 - u_{i\ell'})$.

The resistance of the transition from $\boldsymbol{z}_\ell \in \mathcal{C}^0$ to $\boldsymbol{z}'_\ell \in \mathcal{C}^0$ is $\varrho_{\boldsymbol{z}_\ell\to\boldsymbol{z}_{\ell'}} \ge \min_{i\in\mathcal{N}}(1 - u_{i\ell'}) + c$, since, at least one player needs to change his action, which has a resistance c, and then accept a utility of $(1 - u_{i\ell'})$ to become content.

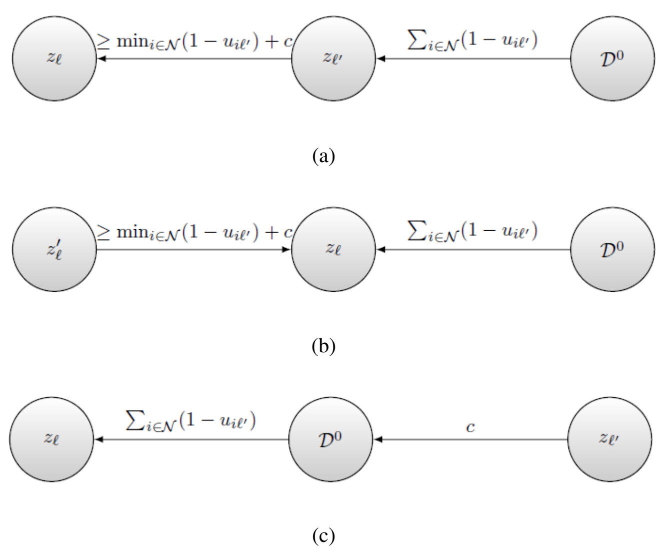
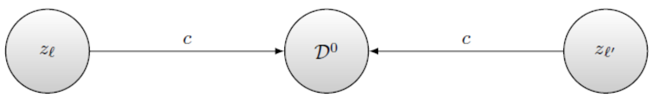


Fig. 2. The three $j-$trees rooted at $\boldsymbol{z}_\ell \in \mathcal{C}^0$. The tree in Fig. 2(c) has the minimum stochastic potential

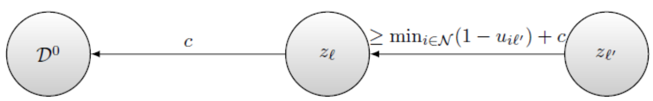
Lemma 3: The stochastic potential of any state $\boldsymbol{z}_\ell = [\boldsymbol{b}_\ell, \boldsymbol{u}_\ell, \boldsymbol{m}_\ell] \in \mathcal{C}^0$ is $\gamma(\boldsymbol{z}_\ell) = c(|\mathcal{C}^0| - 1) + \sum_{i\in\mathcal{N}}(1 - u_{i\ell})$, and this is less than the stochastic potential of the $\mathcal{D}^0$ class.

Proof: Let us start with a scenario in which there is only one state $\boldsymbol{z}_\ell \in \mathcal{C}^0$, i.e. $|\mathcal{C}^0| = 1$. In this case, there can only be a single $j-$tree rooted at D^0 and a single $j-$tree rooted at $\boldsymbol{z}_\ell \in \mathcal{C}^0$. Thus, the stochastic potential of $\boldsymbol{z}_\ell \in \mathcal{C}^0$ is $\sum_{i\in\mathcal{N}}(1-u_{i\ell})$. The stochastic potential of $\mathcal{D}^0$ is $> c$, and thus, greater than that of $\boldsymbol{z}_\ell \in \mathcal{C}^0$.

Now, suppose we have two states, $\boldsymbol{z}_\ell, \boldsymbol{z}_{\ell'} \in \mathcal{C}^0$, i.e. $|\mathcal{C}^0| = 2$. There are 3 possible $j-$trees rooted at $\boldsymbol{z}_\ell$, as can be seen in Fig. 2. There are 3 possible $j-$trees rooted at $\mathcal{D}^0$, of which two are similar - interchanging $\boldsymbol{z}_\ell$ and $\boldsymbol{z}_{\ell'}$ in one tree will get the other, with the resistances remaining the same. We have shown one of those two trees and also the third tree in Fig. 3. As we can see, the stochastic potentials of $\boldsymbol{z}_\ell, \boldsymbol{z}_{\ell'} \in \mathcal{C}^0$ satisfy the formula given in Lemma 3 and that the stochastic potential of the $\mathcal{D}^0$ state is more than the two $\mathcal{C}^0$ states' potentials, as per Lemma 3.



(a)



(b)

Fig. 3. The two different $j-$trees rooted at $\mathcal{D}^0$

Now, we consider the general case, when there are $M+1$ states in $\mathcal{C}^0$, $\boldsymbol{z}_{\ell 0}, \cdots, \boldsymbol{z}_{\ell M}$. In this case, the minimum weight $j-$tree rooted at $\boldsymbol{z}_{\ell 0}$ will be the one shown in Fig. 4. For the interested reader, a rigorous proof is provided in the following paragraphs.

Consider a tree T as described by Fig. 4. This tree has $(|\mathcal{C}^0|-1)$ edges of the kind $\boldsymbol{z}_{\ell'} \in \mathcal{C}^0 \to \mathcal{D}^0$, each having a resistance c. There is one edge of the form $\mathcal{D}^0 \to \boldsymbol{z}_\ell \in \mathcal{C}^0$, with a resistance of $\sum_{i\in\mathcal{N}}(1-u_{i\ell})$. Thus, the stochastic potential of a state $\boldsymbol{z}_\ell \in \mathcal{C}^0$ is upper bounded by $c(|\mathcal{C}^0|-1)+\sum_{i\in\mathcal{N}}(1-u_{i\ell})$.

Now, let us suppose that there exists another tree T' whose resistance is less than that of T, i.e., $R(T') < R(T)$. Since the tree is rooted at $\boldsymbol{z}_\ell \in \mathcal{C}^0$, there exists a path P from $\mathcal{D}^0$ to $\boldsymbol{z}_\ell$,

$P = \{\mathcal{D}^0 \to \boldsymbol{z}_{\ell 1} \to \boldsymbol{z}_{\ell 2} \to \cdots \to \boldsymbol{z}_{\ell K} \to \boldsymbol{z}_\ell\}$

Notice that the resistance of the path P is $R(P) \geq Kc + \sum_{i\in\mathcal{N}}(1-u_{i\ell})$, since, at some point during the set of transitions, each user i needs to accept a utility of $u_{i\ell}$. Also, during the last K transitions, at least one player will have to first change his/her action, which has a resistance of c, other than the resistance associated with accepting a certain utility.

Let's construct a new tree T'' which is derived from T'. We begin by removing the edges in P. The nodes which get removed through this edge removal are added to T'' by adding the following edges to T'': $\mathcal{D}^0 \to \boldsymbol{z}_\ell$, with a resistance of $\sum_{i\in\mathcal{N}}(1-u_{i\ell})$, and $\boldsymbol{z}_{\ell j} \to \mathcal{D}^0 \ \forall\ 1 \leq j \leq K$, with each of the K edges with a resistance of c. All other edges in T' are preserved in T''. Thus, the resistance of T'', $R(T'') \leq R(T')$.

We construct a new tree T''' from T'' as follows. For any edge in R'' of the form $\boldsymbol{z}_{\ell'} \to \boldsymbol{z}_{\ell''}$, which has a resistance $\geq c$, remove that edge and instead add an edge $\boldsymbol{z}_{\ell'} \to \mathcal{D}^0$, which has a resistance of c. Thus, $R(T''') \leq R(T'') \leq R(T')$. But, after all the transformations are done, what we have with us is R, i.e., $R''' = R$. But this contradicts our assumption that $R(T') < R(T)$. So, the minimum weight tree rooted at $\boldsymbol{z}_\ell$ is T and has a resistance of $c(|\mathcal{C}^0|-1) + \sum_{i\in\mathcal{N}}(1-u_{i\ell})$.

Now all we are left to prove is that the stochastic potential of $\mathcal{D}^0$ is greater than $c(|\mathcal{C}^0|-1)+\sum_{i\in\mathcal{N}}(1-u_{i\ell})$. Let us assume the contrary and call the tree rooted at $\mathcal{D}^0$ with the minimum weight as T_1. T_1 must have an edge of the form $\boldsymbol{z}_\ell \to \mathcal{D}^0$, with a resistance of c. Remove this edge and add an edge $\mathcal{D}^0 \to \boldsymbol{z}_\ell$, with a resistance of $\sum_{i\in\mathcal{N}}(1-u_{i\ell}) < N < c$. This gives a tree rooted at $\boldsymbol{z}_\ell$, T_1' with $R(T_1') < R(T_1)$. But we had assumed that $R(T_1) \leq R(T)$. Therefore, we get, $R(T_1') < R(T)$. This can't be possible, as we have proved earlier. Thus, our assumption was incorrect. So, $\mathcal{D}^0$ does not have the minimum stochastic potential. ∎

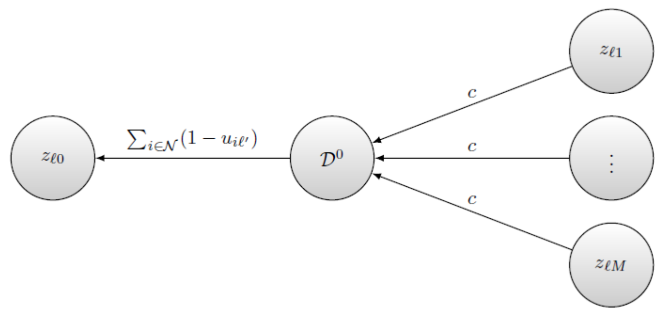


Fig. 4. The minimum weight $j-$tree of the state $\boldsymbol{z}_{\ell 0} \in \mathcal{C}^0$

Thus, from Lemma 3, we have that the stochastically stable states are contained in $\mathcal{C}^0$. We know that they have the minimum stochastic potential. Therefore, a state $\boldsymbol{z}_\ell = [\boldsymbol{b}_\ell, \boldsymbol{u}_\ell, \boldsymbol{m}_\ell] \in \mathcal{Z}$ is stochastically stable if and only if:

$$\boldsymbol{z}_\ell \in \arg\max_{\boldsymbol{z}_{\ell'} \in \mathcal{C}^0} \sum_{i\in\mathcal{N}} u_{i\ell'}$$

This completes the proof of Theorem 1. ∎

This shows that for any $\alpha > 0$, there exists an $\epsilon > 0$ such that the fraction of time for which the optimal association is chosen by the algorithm is greater than $1-\alpha$. As α becomes smaller and smaller, we also need to correspondingly reduce ϵ. Notice that as ϵ decreases, the algorithm becomes more cautious in accepting a new association and also in deviating from the accepted association. Thus, for a smaller value of ϵ, it takes longer for the algorithm to find an optimal association, but once it is found, the algorithm retains it for a much longer time as well.

C. Heuristic

We expect the average social utility of the system, U^{social}, to increase with decreasing ϵ, due to Statement 1. Let T^{sat} denote the time taken to reach for the first time, a state in which all the users' constraints are satisfied. Then, it is also expected that T^{sat} will increase with decreasing ϵ. Since, with decreasing ϵ, the probability that the users experiment decreases, i.e. they take more time to reach to any given configuration. In accordance with our expectations, we provide a heuristic to improve the performance of our algorithm. Instead of keeping ϵ fixed throughout, we can start with a high initial value of ϵ, $\epsilon_{\text{initial}}$, and then at each iteration, we can subsequently decrease ϵ by a small fraction until it reaches a final value of ϵ_{final}.

IV. Numerical Results

In this section, we evaluate the proposed algorithm using simulations.

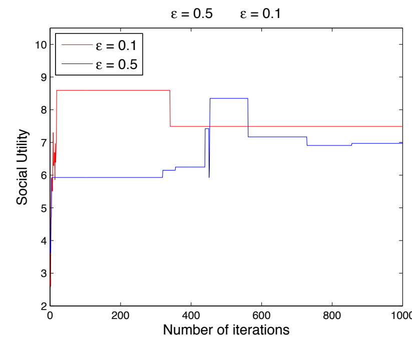
A. Network Model and Parameter Values

We simulate a network in which base stations and mobile users are positioned randomly on a 2-dimensional square region of side length 5 units. We consider topologies with ten users ($N = 10$) and five base stations ($K = 5$). We only consider path loss, and assume path loss exponent to be 2 (simulation results for other values of path loss exponent show the same trend and are omitted on account of the space constraints). So, if user i is connected to the base station k, then the transmission rate for i is assumed to be $r_{ik} = \log(1 + d_{ik}^{-2}/\sigma_{ik})$, where d_{ik} is the distance between i and the base station j and σ_{ik} is the noise power spectral density on the channel from i to k. We choose σ_{ik} uniformly at random from $[0.1, 1]$. We consider the base stations to implement time fairness, i.e., each base station provides equal amount of access time to each of the users that is connected to it. We also choose $T_j = 1$ for every j. Thus, if n users are associated to the k^{th} base station, then the throughput of user i associated to k in j^{th} update period is r_{ik}/n. We allow each user to associate with one of the three base stations that are located closest to him. For the constraints, we consider the minimum throughput requirement for each user. To generate the feasibility constraints, we first associate the users randomly to any of the possible base stations, and then choose the minimum required throughput to be the 50% of the throughput received by each player in this configuration. Also, we consider the problem of maximizing network throughput, i.e., the utility for a user is equal to its throughput. We take $c = 11$.

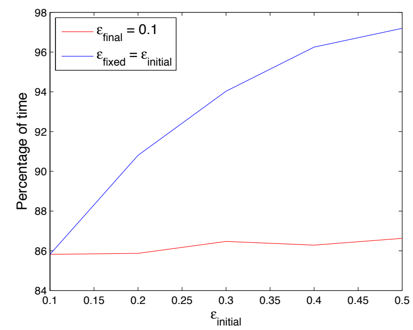
We simulate 1000 randomly generated topologies, in each of which our algorithm is run for 10^5 iterations. $\epsilon_{\text{initial}}$ is varied from 0.1 to 0.5 in all the topologies. In the implementation of the heuristic, the value of ϵ is decreased by 0.1% at each epoch and ϵ_{final} is taken to be 0.1.

B. Observations and Results

Fig. 5(a) shows a typical sample path of the algorithm for $\epsilon = 0.1$ and 0.5. The sample paths look broadly as a step function, where the flat portions are the states in which all the users are content (which also implies that their throughput requirements are satisfied). Note that for a larger value of ϵ, a state in which all the users are content is found sooner than for a smaller value of ϵ. However, for a larger value of ϵ, the system explores different states more aggressively and as a result the states change more often. This is a classical exploration versus exploitation trade-off. For a larger value of ϵ, the exploration dominates whereas when ϵ is smaller the system tends to respond comparatively slowly, but it maintains an optimal state for a much longer duration. This trade-off is further illustrated in Fig. 5(b) and 6(a). In Fig. 5(b), we observe that the fraction of time all the users' constraints are satisfied increases slowly for higher values of ϵ, whereas in the proposed heuristic, this fraction increases much more slowly, almost looking to be constant. Note that the heuristic starts with a higher value ϵ and then decreases it gradually. Same phenomenon can be seen in Fig. 6(a). Here we plot the number of iterations required for all the users to be satisfied for the first time simultaneously. Note that this value is smaller for a larger value of initial ϵ. The heuristic gives the benefit of a faster response in this case, although not as fast as the original algorithm. Finally, Fig. 6(b) plots the achieved system throughput as a function of ϵ. Note that the throughput is higher for the smaller values of ϵ. This justifies Statement 1. From these figures, we can see that while using the heuristic, the long term characteristics of the system depend almost only on the final value of ϵ, whereas the the initial response also depends on the initial value of ϵ, as we would expect.

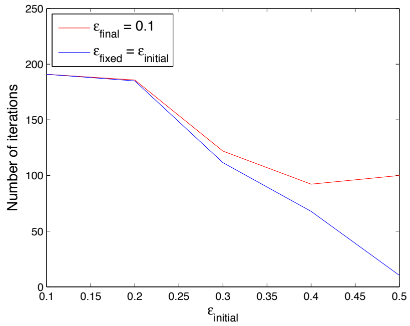


(a) Social utility vs. number for iterations for $\epsilon = 0.1$ and $\epsilon = 0.5$

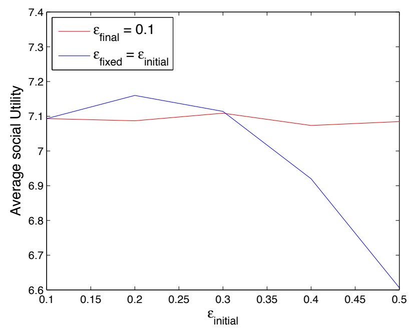


(b) Percentage of time all the users' constraints are satisfied, for varying values of $\epsilon_{\text{initial}}$

Fig. 5.



(a) Number of iterations required to reach a profile in which all the users' constraints are satisfied for the first time, for varying values of $\epsilon_{\text{initial}}$



(b) Average social utility, for varying values of $\epsilon_{\text{initial}}$

Fig. 6.

V. Related Work

A lot of research is being done to cater to the problem of finding an efficient user association scheme. [4] consider fairness and load balancing in wireless LANs but in a centralised manner. [5] also provide a centralised algorithm. Coucheney et al., in [1] provide a distributed scheme for association which requires base stations to send their users rewards, which help the users reach an α-fair profile [8]. [7] also provide a distributed scheme which requires the base stations to convey information to the users. In [6], Kauffmann et al. propose a distributed algorithm for channel association and user association, albeit again with user - base station communication. The difficulty with requiring the base stations to interact in some way with the users is that if they were to be implemented, every base station, independent of its technology, would be needed to follow these algorithms. This requirement is very difficult to be satisfied. Bejerano and Han, in [11], present schemes for optimal load balancing in IEEE 802.11 WLANs, by utilising cell breathing techniques. They only require that the access points have the ability of dynamically changing the transmission power of their beacons. But this technique is only applicable in the case of single technology networks, where only IEEE 802.11 access points are present.

VI. Conclusions And Future Work

In this paper we have introduced a user association algorithm which requires no interaction amongst the users and the base stations. We have proved the convergence of the algorithm in a stochastically stable sense. Through numerical experiments, we have also demonstrated that the algorithm can lead to efficient system usage, with each user getting throughputs meeting his or her desired throughput and cost requirements.

This leads to even more interesting problems, such as trying to rid the requirement of synchronicity in user updates; and to accommodate the presence of noise in the utilities observed by the users, and still contrive the algorithm in such a way that it leads to optimal configurations.

Acknowledgment

This work was supported by India-UK Advanced Technology Centre (IU-ATC) of Excellence in Next Generation Networks, Systems and Services.

References

[1] P. Coucheney, C. Touati, and B. Gaujal, "Fair and efficient user-network association algorithm for multi-technology wireless networks," in *INFOCOM 2009, IEEE*, pp. 2811–2815, IEEE, 2009.

[2] H. Gong and J. Kim, "Dynamic load balancing through association control of mobile users in WiFi networks," *Consumer Electronics, IEEE Transactions on*, vol. 54, no. 2, pp. 342–348, 2008.

[3] J. Marden, H. Young, and L. Pao, "Achieving pareto optimality through distributed learning," tech. rep., 2011.

[4] Y. Bejerano, S. Han, and L. Li, "Fairness and load balancing in wireless LANs using association control," in *Proceedings of the 10th annual international conference on Mobile computing and networking*, pp. 315–329, ACM, 2004.

[5] K. Leung and B. Kim, "Frequency assignment for ieee 802.11 wireless networks," in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 3, pp. 1422–1426, IEEE, 2003.

[6] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot, "Measurement-based self organization of interfering 802.11 wireless access networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1451–1459, IEEE, 2007.

[7] M. Haddad, S. Elayoubi, E. Altman, and Z. Altman, "A hybrid approach for radio resource management in heterogeneous cognitive networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 4, pp. 831–842, 2011.

[8] E. Altman, K. Avrachenkov, and A. Garnaev, "Generalized α-fair resource allocation in wireless networks," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pp. 2414–2419, IEEE, 2008.

[9] H. P. Young, "The evolution of conventions," *Econometrica*, vol. 61, no. 1, pp. 57–84, 1993.

[10] B. Pradelski and H. Young, "Learning efficient Nash equilibria in distributed systems," *Games and Economic Behavior*, 2012.

[11] Y. Bejerano and S. Han, "Cell breathing techniques for load balancing in wireless LANs," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 6, pp. 735–749, 2009.