Scheduling in Heterogeneous Cellular Networks with Mobility

Pooya Monajemi  
Department of Electrical Engineering  
University of California, Los Angeles  
Los Angeles, CA 90095, USA  
Email: pmonajemi@ucla.edu

Richa Sahasrabudhe  
Department of Electrical Engineering  
University of California, Los Angeles  
Los Angeles, CA 90095, USA  
Email: rsahasrabudhe@ucla.edu

John Villasenor  
Department of Electrical Engineering  
University of California, Los Angeles  
Los Angeles, CA 90095, USA  
Email: villa@ee.ucla.edu

Abstract—We examine the scheduling problem in the context of heterogeneous cellular networks in the presence of mobility. We demonstrate that the QoS for users in a cellular network can be optimized using an interference-aware approach that is most effective at low Doppler frequency scenarios or a channel-aware approach that is most effective at higher low Doppler frequency scenarios.

Among the most recent products introduced to the quickly evolving cellular market are devices called “femtocells”, also known as “Home Node B”s in some contexts. Femtocells act similarly to regular base stations (BSs) in a cellular network, with the main exception that they use internet connections within their deployment area to connect to the cellular network. Low cost and low power consumption make femtocells attractive alternatives to the deployment of larger BSs in areas where demand is low or the cost of new BS deployments is high.

The present paper focuses on the scheduling of data packets—specifically in the uplink—in a heterogeneous cellular network consisting of both macrocells and femtocells. The significance of this problem in the case of such networks is the limitation in the connectivity between the femtocells and the provider’s network. Such limitations can include high latencies and relatively low data rates, which make operations such as soft hand-offs and fast synchronization impossible to achieve. Hence, what is needed is a scheduling approach that utilizes minimal data overhead and has low latency requirements.

Scheduling in general has been widely studied in topics such as task assignment problems in operations research (see for example [1]), or scheduling of parallel applications in multiprocessor networks [2]. In [3] a taxonomy of various scheduling algorithms is presented. In the field of cellular packet data networks several studies have been performed. A fairness criterion introduced by [4] referred to as proportionally fair scheduling is a well known and widely deployed mechanism. In [5] it is shown that proportionally fair scheduling converges under various conditions to the solution of a log-sum optimization criterion. In [6], [7], and [8] generalized versions of the proportionally fair optimization criteria are considered.

The uplink scheduling algorithm presented here is based on simulated annealing, which was first proposed as an optimization method in [9]. Simulated annealing is an iterative optimization technique that has proven to be effective in solving general combinatorial optimization problems. In previous work, simulated annealing has been applied in a wide range of applications including scheduling in parallel systems [10], [11], [12].

In this work, we present a unique application of simulated annealing in the context of interference aware uplink packet scheduling in a cellular network. Results are presented that support its utility in the developed system framework, along with a foundation for the analysis of the optimization and necessary conditions for convergence of the proposed algorithm.

Our work demonstrates that, in scenarios with Doppler frequencies within the low to normal range, an interference aware scheduling such as the augmented simulated annealing can provide a better QoS experienced by disadvantaged users in a cellular network as compared to a channel aware scheduling, while maintaining minimal signaling overhead. By contrast, in high Doppler frequency scenarios, channel-aware scheduling becomes more effective than interference-aware scheduling. This is because the high mobility ensures that random statistical variations in instantaneous channel quality will create windows of opportunity for higher rate transmission. Exploiting these windows gives more throughput improvement than an approach based purely on interference monitoring.

The rest of the paper is organized as follows: in Section I the system framework is defined and the scheduling algorithm is proposed. In Section II we provide necessary conditions for the convergence of the proposed algorithm. In Section III simulation results are presented, and conclusions are presented in Section IV.

I. System Framework

We consider uplink packet scheduling in a time division multiple access cellular system in which there are $N$ BSs with $u_n$ users registered to each BS $n \in [1,N]$. We define block scheduling as a scheme in which scheduling is performed in pre-determined blocks consisting of $L$ time slots. Within each time slot one UE is scheduled. The UEs within the block schedule of a particular BS are not necessarily unique. It is further assumed that these scheduling blocks are repeated in time for the duration of the study.
We assume that a particular UE connects to a BS based on the strongest received pilot signal power. We further assume that all UEs are in fixed positions for the duration of the optimization, and the UE–BS associations are static. The UEs in this multicell system are characterized based on two traits: 1) their strength of connectivity to their own serving BS (strongly-connected or weakly-connected UEs, referred to as “strong” or “weak” UEs respectively); and 2) the levels of interference generated to neighboring BSs (“interfering” or “non-interfering” relative to a particular BS). The strength of connectivity is based on the link loss experienced between the UE and its associated BS, referred to as the weakness threshold. Similarly, the interference characterization is based on the ratio between the interferer’s linkloss to its associated BS and its linkloss to another BS, referred to as the interference threshold. A “collision” occurs when a weakly connected UE is scheduled concurrently with an interfering UE from a neighboring cell. Therefore, the block schedule of a particular BS is compared to those of all the remaining $N-1$ BSs on a per time slot basis to determine the occurrence of collisions.

We consider three scheduling schemes: round robin (RR) scheduling, block scheduling with simulated annealing (BS-SA), and block scheduling with augmented simulated annealing (BS-ASA). We proceed by first describing the simulated annealing algorithm, and then describing the BS-SA and BS-ASA algorithms as its derivatives in the context of uplink packet scheduling.

A. Simulated Annealing

Simulated annealing is a generic iterative optimization method where random mutations in a state variable occur at each iteration. The algorithm is summarized in the pseudo-code given in Fig. 1 [13].

At the end of the iteration, the simulated annealing algorithm compares the mutated state to the current state and probabilistically decides whether to transition to this state or remain in the previous state. If the cost of the mutated state is lower than that of the current state, then a transition to the mutated state occurs. Otherwise, the mutation is accepted with a probability given by $e^{-\Delta C / T}$ (referred to as the acceptance probability [13]), where $\Delta C$ is the difference in cost between the mutated state and the previous state and $T$ is the temperature, which is used as a control parameter. The mechanism of accepting cost-increasing mutations is implemented to avoid convergence to a local minimum.

B. Block Scheduling with Simulated Annealing (BS-SA)

Applying the simulated annealing algorithm to block scheduling, we derive the BS-SA scheme. The system state we consider in BS-SA is the combined block schedule for all $N$ BSs in the system. It is assumed that every BS uses block scheduling with the same block size of $L$, where $L \gg \max (u_1, \ldots, u_N)$, and the block schedules on the BSs are synchronized. The cost function considered is the total number of collisions per block.

The mutated state is one in which two time slots in the block schedule for every BS are randomly switched. A central entity (such as the radio network controller) receives information on the number of collisions each BS experiences after every block and assesses the difference in the collision rates between the mutated state and the current state. It will then compute the acceptance probability for the mutated state and send a message to each BS to determine whether the BS should accept the mutation or not.

Note that under this scheme the overhead signaling imposed on the BS for each scheduling block is minimal and includes only a collision measurement transmitted to the network and an accept/reject signal received back. Possible latencies in the arrival of the accept/reject notifications from the central network come at an insignificant cost: the BS can simply continue its current schedule into the next block for a few time slots until the arrival of the notification. The expected number of differences between the old and the new schedules in these time slots is small.

It is worthy of mention that one could also consider the application of simulated annealing in which only one switch in the entire system state is applied at each iteration. Such a scheme would significantly reduce the optimization speed and is prohibitively slow, and hence is not considered by the authors.

C. Block Scheduling with Augmented Simulated Annealing (BS-ASA)

While the collision information is utilized at the decision making stage of the BS-SA, this data is disregarded at the mutation stage where switches are performed between any two randomly chosen slots. Block scheduling with augmented simulated annealing (BS-ASA) is a modified version of the BS-SA in which the collision information is used in choosing the time slots that will be switched. Explicitly, the mutated state considered is a random switch between every time slot in the block schedule in which a single collision occurs with another randomly chosen time slot in the block. A faster convergence rate for this algorithm is expected and is observed in the simulation results presented in Section III.
II. CONVERGENCE ANALYSIS

In this section we establish an analytical framework related to the convergence of both the BS-SA and BS-ASA algorithms. Our attention will be focused on the homogeneous algorithm in which the temperature, $T$, is kept constant. The optimization process is modeled as a Markov chain where we define the state of the system to be the collection of all the individual block schedules (as mentioned previously in Section I):

$$i^k = \bigcup_{n=1}^{N} i^k_n,$$

where $i^k_n = [M^k_{n,1}, M^k_{n,2}, \ldots, M^k_{n,k}]$, $N$ is the number of BSs, $i^k_n$ is BS $n$’s block schedule at the $k$th iteration, and the UE scheduled at index $l$ of BS $n$’s block schedule at the $k$th iteration is denoted as $M^k_{n,l}$, and $l \in [1, L]$ . It is noted here that the state space $R$ is not the set of all permutations of the UEs in the state vector, but only the subset of those permutations in which the UE-BS associations have been kept the same. The resulting total state space size is then $(L!)^N$.

The transition probability matrix of the optimization process as described in [13] is:

$$P_{ij}(k) = \begin{cases} A_{ij}(k), & i \neq j, \\ 1 - \sum_{\ell \neq i} G_{ij} A_{ij}(k), & i = j, \end{cases}$$

where $G_{ij}$ is the probability of state $j$ being generated from state $i$, and $A_{ij}(k)$ is the probability of the generated state $j$ being accepted from state $i$. In the conventional simulated annealing algorithm, where all mutations are equally likely, $G_{ij}$ is given by:

$$G_{i,j} = \begin{cases} 0, & j \notin R_i, \\ \frac{1}{|R_i|}, & j \in R_i, \end{cases}$$

where $R_i$ is the set of all states accessible from state $i$ in one random mutation (the neighbor set). The admittance probability matrix as a function of the temperature is given by:

$$A_{ij}(k) = \begin{cases} 1, & C(j) < C(i), \\ e^{-\frac{C(j) - C(i)}{T(k)}}, & C(j) > C(i). \end{cases}$$

Convergence of this process to steady state will be conditioned on irreducibility and aperiodicity of the underlying Markov chain. This is explored in the following subsections.

A. Aperiodicity

To establish aperiodicity in a Markov chain it is sufficient and convenient to simply prove that there is a non-zero probability of staying in a state, i.e. $\exists i : P_{ii} > 0$. Formally we can prove the existence of such a state as follows.

**Proof:** Let the process begin in a globally optimal state, $i^1_{\text{opt}} \in R_{\text{opt}}$, where $R_{\text{opt}}$ is the set of all globally optimal states. After a single iteration of switches on an optimal state, a new state is generated. Such a state can be generated according to two cases. In the first case, a non-optimal state $i^2$ that is cost-increasing is considered. Then the original state $i^1_{\text{opt}}$ proves aperiodicity due to a non-zero probability of rejected admission for $P_{i^1_{\text{opt}} \rightarrow i^2} = e^{-\Delta C} < 1$ where $\Delta C = C(i^2) - C(i^1_{\text{opt}}) > 0$ and $T > 0$.

In the second case, the second state can be another optimal state $i^2_{\text{opt}} \in R_{\text{opt}}$. Then, assuming irreducibility and $R_{\text{opt}} \neq R$, after enough iterations we will arrive at a non-optimal state $i^k$, in which case the optimal state $i^k_{\text{opt}}$ proves aperiodicity.

B. Irreducibility

We now focus on conditions under which irreducibility can be achieved. Irreducibility of a Markov chain is guaranteed if $\forall i, j \in R, \exists k \in : P^k_{i,j} > 0$, i.e. there is a possible path between any two states in the chain. Obviously at $T = 0$ this condition is not met since no cost-increasing mutations are admitted therefore we study only the cases in which $T > 0$. The analysis is performed separately for the conventional simulated annealing and the augmented version presented in this paper.

1) Irreducibility for BS-SA: In the case of conventional simulated annealing, if we allow only one BS to mutate at each iteration then irreducibility is guaranteed: there will always be a set of mutations that connect any two arbitrary states. Such an algorithm is very slow in convergence, hence we consider the cases where each BS performs a mutation at each step. If a mutation is forced at each BS, irreducibility is not guaranteed. For a counter example consider the set of two BSs, BS1 and BS2, i.e. $N = 2$, each of which is connected to an independent set of three UEs indexed 1, 2, and 3, $u_1 = u_2 = 3$. Let the state vector be $[v_1; v_2]$, where $v_1$ and $v_2$ represent the block schedule for BS1 and BS2, respectively. Fig. 2 represents the state transition graph for a BS with a block schedule for $L = 3$. Suppose we start from an initial state of $[3,2,1; 3,2,1]$, and target an end state of $[1,2,3; 1,3,2]$. Here BS1 will need an odd number of transitions to arrive at its target, while BS2 will need an even number of such transitions and hence the target state will never be achieved simultaneously for both BSs. In order to overcome this obstacle we will need to allow for idle transitions at BSs, in which case each BS can perform the number of required mutations to arrive at the target and stay idle afterwards while others arrive at their target states.

2) Irreducibility for BS-ASA: The BS-SA algorithm will not provide for irreducibility in general. For example, consider a BS that has no interfering UEs connected to its neighboring BSs. In such a case no mutations will be performed on this BS schedule, since no collisions are detected. With respect to performance, this will be acceptable only if the UEs connected to this BS are not interferers to their neighbors. Where interferers exist, however, the schedule of the interferers is of concern to the network and needs to be flexible.

We consider two cases in which such irreducibility is guaranteed. The first case considers combining BS-SA and BS-ASA and allowing for random mutations performed independently of the collisions. Here, again, all mutations are...
possible and therefore there exists a path between any two states.

The second case is a model in which the interference is not perfectly known, and is only detected by the BS that receives the uplink data. Furthermore, detection of such interference is performed by the BS on all upcoming data, regardless of whether the UE from which the data is being received is weak or strong. In this case, once again, any UE can be detected as being under interference from the neighborhood, and can possibly be mutated in its schedule.

While in both cases mentioned above, irreducibility comes at the cost of slower convergence, in the second case the lack of perfect knowledge incurs the extra cost of instability at the global optimum: any false detections can bring the system out of the global optimum after it is reached.

III. SIMULATION

In this section we present the collision rates for the algorithms presented in Section I, followed by simulation results comparing the performance of BS-ASA to that of the proportionally fair scheduling.

A scenario featuring 14 BSs and 200 UEs is studied on a 4000 meter by 4000 meter 2D map. Constant, full buffer uplink traffic is considered for each UE. An ideal channel is assumed and a simplified path loss model is used. A block length of \( L = 200 \) time slots is used and a weak distance threshold of 600m is set along with an interference threshold of 1.3. Note that under a static scenario distance translates to a constant link loss. Simulation results are shown in Fig. 3 as collision rates per slot. A temperature of \( T = 0 \) is used with both BS-SA and BS-ASA. With RR showing the expected constant collision rate, we observe sluggish improvement in the collision rates under the BS-SA algorithm as compared to the BS-ASA. For both the BS-SA and BS-ASA algorithms a temperature of zero means that there is no guaranteed irreducibility, however the convergence to a zero collision rate is still possible. What is further observed is that disregarding the collision information when making the random switches in the block schedule is shown to be extremely costly to the performance of the optimization, to the point that it renders the BS-SA method impractical for implementation for cellular environments because of its extremely slow rate of convergence as observed in Fig. 3.

From the perspective of a network operator, an optimization algorithm must be fast enough to cope with the variations in the scenarios, such as hand offs and deep fades. Considering that such events occur in the order of magnitude of once every few seconds, the optimization must be likely to convergence to almost-optimal solutions in similar lengths of time with high probability. In our study, the BS-ASA optimization was reiterated numerous times for the same scenario described above. After 1000 independent runs the cumulative distribution function (CDF) for convergence to various collision rates was extracted as a function of the number of time slots and plotted in Fig. 4. The BS-ASA is shown capable of achieving a 10% collision rate with 90% probability after 4000 time slots, which proves to be a feasible performance criterion for cellular networks. Fig. 4 also illustrates the CDF for collision rates of achieving less than 5% and 0% collision rates serving as additional performance criteria.

In order to assess the performance of the BS-ASA under imperfect collision detection, the same simulation was performed using multiple pairs of false alarm and misdetection probabilities for weak UE’s, noted as \( P_{fa} \) and \( P_{md} \) respectively. Results, as shown in Fig. 5, show that the algorithm is far more vulnerable to false alarm rates than to misdetection rates, hence the collision detection scheme must be designed biased towards misdetections. This bias is due to the fact that the number of collisions is a small fraction of the total number of packets scheduled, and hence a small fraction of false alarms on the non-collision slots can have the same effect as a large fraction of misdetections on the collision slots.

In order to compare the performance between BS-ASA and PF scheduling schemes, it is notable that the proportionally fair algorithm as defined in [5] assumes perfect knowledge of both the channel state and the interference level at the time of transmission. While the former can be attained with a reasonable level of accuracy, the interference levels are difficult to predict in a network with highly dynamic behavior and low inter-BS coordination. Therefore we implement a channel-aware version of the PF algorithm to be compared against BS-ASA. Shown in Fig. 6 are plots comparing the performance of the two schemes under four different Doppler conditions. We observe that under the small Doppler frequencies where channel variation levels are low, the interference aware BS-ASA scheduling shows superior results, while in the highest Doppler frequency the channel variations become more important to take advantage of, giving the superiority to the PF scheduling.

IV. CONCLUSION

In this paper we propose the use of an augmented version of simulated annealing for scheduling in heterogeneous networks. An analytical framework is provided for the modeling of such
algorithms, along with the analysis of necessary conditions for convergence of the algorithms to optimum. Simulation results are provided to demonstrate the advantages of the BS-ASA to the conventional BS-SA, and establish feasibility for implementation of such algorithms in femtocell networks. Furthermore, results are provided showing the trade off between channel awareness and interference awareness in scheduling under different mobility conditions. Areas of interest for further investigation include convergence rate analysis, robustness studies against changes in BS-UE associations, and opportunities for combining the BS-ASA with other scheduling algorithms.

ACKNOWLEDGMENT

The authors would like to thank Dr. Shaunak Joshi for his supporting role in the project.

REFERENCES


