

On Optimization Techniques for Energy Efficient Operation of Cellular Networks

Dissertation

submitted in partial fulfillment of the requirements

for the degree of

Master of Technology
in Communication Engineering

by

Karunakaran Kumar

Roll No: 123079005

under the guidance of

Prof. Abhay Karandikar

and

Prof. Prasanna Chaporkar



Department of Electrical Engineering,
Indian Institute of Technology Bombay,
Mumbai - 400076.

June, 2015

Dedicated to every one who made this possible.

Acknowledgments

I am extremely grateful to Prof. Abhay Karandikar and Prof. Prasanna Chaporkar for giving me an opportunity to pursue my education here at IIT Bombay, and for all the guidance they have provided me, and especially for their immense patience in spite of me certainly having tested their patience (more than just once) :-)
Thank you very much.

I would also like to thank all my lab members and friends for all the support and for making my life here at IIT memorable.

I would like to thank all the faculty members at IITB. I had an amazing learning experience in the past three years.

Thank you all,

Karunakaran Kumar

IIT Bombay

June 10, 2015

Declaration

I declare that this written submission represents my ideas in my own words and wherever others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/ data/ fact/ source in my submission. I understand that any violation of the above will be cause for disciplinary action by IIT Bombay and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: June 9, 2015

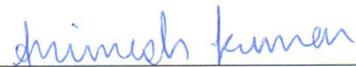

Karunakaran Kumar

Dissertation Approval

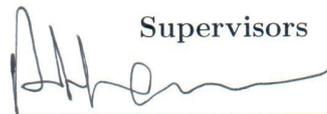
This dissertation entitled **On Optimization Techniques for Energy Efficient Operation of Cellular Networks** by Karunakaran Kumar (Roll No. 123079005) is approved for the degree of **Master of Technology in Communication Engineering**.

Examiners




Drimesh Kumar

Supervisors





Chairperson



Date: 09/06/2015

Place: Mumbai

Abstract

Energy consumption of cellular wireless networks is a growing concern. Even though it accounts for only a small fraction of the total worldwide energy consumption, the rate of growth of these networks is tremendous. The cost of operating cellular base-stations (BSs) constitutes a significant chunk of the total operational cost (OPEX) of the operator. All the BSs in the network remain fully even though the demand for services varies throughout the day, with extremely little demand during the night. An equivalent way of seeing this is that there is excess capacity during off peak hours which can be traded in for energy savings. In this work, we propose a general framework for approaching this problem. We re-configure the network at different times of the day based only the capacity that would be required for that period of time. However, it is known that determining such an optimal network configuration is very difficult computationally. We present two approaches – 1) centralized optimization based on simulated annealing and 2) using game-theoretic tools to distribute the computational task among the nodes. We propose a distributed optimization algorithm based on existing works for this optimization. We also validate the results on simulations performed on a real-world network using data obtained from a leading telecom operator in India.

Contents

Acknowledgments	iii
Abstract	ix
List of Figures	xiii
1 Introduction	1
1.1 Traffic Dependent BS Operation	2
1.2 Energy Saving Techniques	3
1.2.1 Challenge	5
1.3 Related Work	5
1.4 Organization of Report	7
2 Centralized Optimization Approach	9
2.1 Prediction and Network Simulator	9
2.2 Markov Chain Monte Carlo Optimization	13
2.2.1 Metropolis Hastings Algorithm	14
2.3 An Example Implementation	15
3 Distributed Optimization Approach	19
3.1 Game Model	20
3.1.1 Application: Welfare of a Nation	22
3.2 Spatial Adaptive Play	22

3.3	Concurrent SAP (C-SAP)	24
3.3.1	C-SAP Scheme-1	26
3.3.2	C-SAP Scheme-2	27
3.4	Application to Energy Optimization	28
3.4.1	An Implementation Framework	28
3.4.2	Practical Implementation	33
3.5	Rate of Convergence Simulations	35
3.5.1	Estimating Mixing Times	36
3.5.2	Results	37
3.6	Chapter Appendices	38
3.6.1	Counter example	38
3.6.2	Equivalence of Wonderful Life and Local Altruistic Utilites .	41
3.6.3	Proofs of theorems	42
4	Conclusions	47
	Bibliography	51
	Publications	52

List of Figures

1.1	Power consumption of BS	2
1.2	Network energy consumption	3
1.3	Cell Zooming	4
2.1	Framework for the centralized optimization approach	10
2.2	Conservative estimation of voice traffic	11
2.3	Network information as obtained from operator	15
2.4	Network simulator	16
3.1	1-hop and 2-hop C-SAP	25
3.2	Interaction graph	29
3.3	Signal quality constraint for reducing ping-pong	31
3.4	Channel allocation constraint	32
3.5	Local interaction for few QoS constraints	33
3.6	Block diagram of the framework developed in this chapter for energy minimization.	35
3.7	Mixing time simulation setup	36
3.8	Convergence rate results	38
3.9	Convergence rate results	39
3.10	Counter example for 1-hop C-SAP	40

Chapter 1

Introduction

In the recent past, there has been a surge of interest in green communications and networking. This is especially more significant for the case of mobile cellular networks. Energy consumption of cellular networks directly translates to carbon emissions and operational cost (OpEx) to network operators. The electricity bill alone accounts for a major chunk of this OpEx cost [1]. Moreover, the demand for mobile services has been doubling every 4-5 years, due to the proliferation of smart phones and handheld devices [2]. In order to meet this increase in demand while maintaining service quality, operators must improve their network infrastructure. This results in additional (capital) expenditure and an increase in OpEx. However, in developing countries like India, the Average Revenue per User (ARPU) is extremely low and thus it affects the profitability of operators. This in turn affects penetration of services as well.

Therefore, in order to sustain and improve network services, there is a need to reduce OpEx, especially energy consumption. If we consider the energy consumption of a cellular network infrastructure, it is estimated that around 60% of it is due to the Radio Access Network (RAN) i.e, the Base Stations (BSs). A lot of study has been carried out to identify and devise ways to reduce energy consumption of the RAN like – efficient power amplifiers, energy harvesting, natural

cooling, heterogeneous deployment, cell-breathing, traffic dependent BS on/off [3]. In this thesis, we focus on the latter i.e., traffic dependent BS operation.

1.1 Traffic Dependent BS Operation

Among other things, a typical network design involves determining Base Station (BS) locations and allocation of resources like transmit (Tx) power and channel allocation. Network design is carried out so as to meet certain Quality of Service (QoS) metrics such as network coverage and Call Blocking Rate (CBR). The network design is usually over-provisioned and more BS and resources are often allotted to cater to the peak demand. However, the traffic experienced by a BS would not be the same throughout the day. In fact we observed that for most of the time, BSs are under utilized – voice traffic at a BS is below 90% of the peak for about 33% of the time.

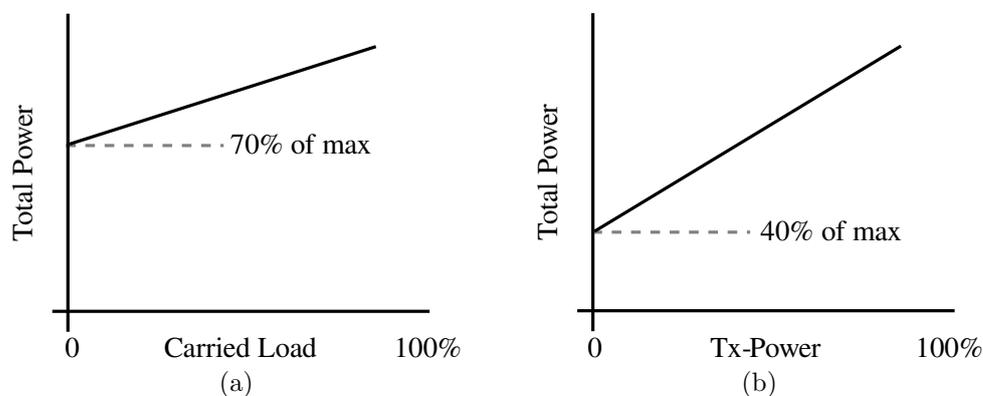


Figure 1.1: Power Consumption of BS. (a) depicts the total power consumed with respect to carried load for a GSM 900 sector. Adapted from [4]. (b) is with respect to transmit power of a macro BS. Adapted from [5]

It has also been observed that the energy consumed by a BS is *fairly* independent of the carried traffic [6] and thus it consumes a similar amount of energy even when it is underutilized as depicted in Figure 1.1a. BS power consumption is approximately linear in carried load, with an offset when Tx-power is zero [4]. The offset is due to air conditioning, among other things. However, the slope of this is

less compared to the offset. Thus, BSs are inherently non energy-proportional.

We thus observe that the energy consumption of the entire network does not scale with the carried traffic as depicted in Figure 1.2a. In order to reduce energy consumption, we may “re-configure” the network during times of low traffic such that the consumed energy scales with the traffic as depicted in 1.2b. For example, when the traffic at a BS is low, it can be switched-off while its neighbouring BSs cater to the users in the affected region. A number of such well known techniques may be employed to achieve network energy scaling. We discuss a few of these briefly.

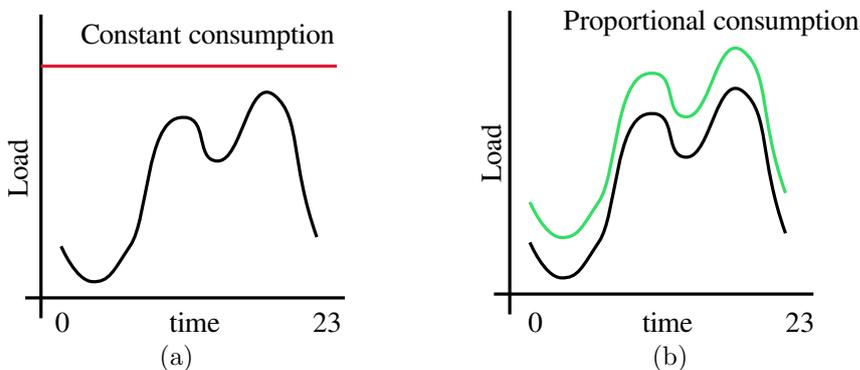


Figure 1.2: Network energy consumption of existing networks is not energy proportional as depicted in (a). It is desirable to achieve energy proportionality using these non-energy proportional devices. Adapted from [7]

1.2 Energy Saving Techniques

In the current literature, a number of techniques to reduce energy consumption of cellular networks have been discussed [8]. We briefly discuss a few of these, which are relevant to this work.

BS on/off As the name suggests, in this technique, unused or low utilization BSs are switched off to save energy. As such, it may result in coverage holes being formed. Therefore this technique is relevant in heterogeneous networks

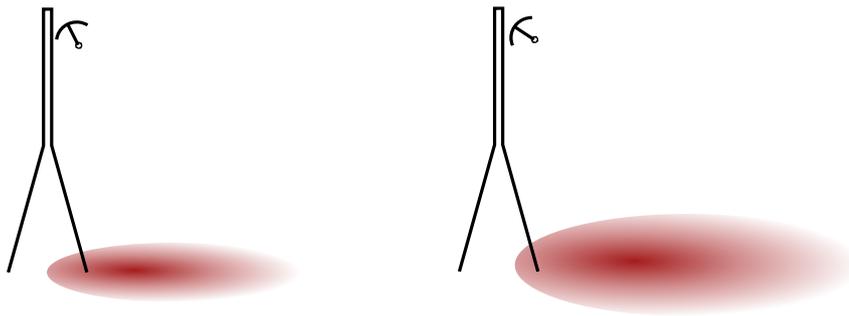


Figure 1.3: Cell Zooming by antenna tilt/Tx-power adjustment. Adapted from [9]

wherein macro BSs provide umbrella coverage and pico/micro BSs may be switched off during periods of low activity.

Cell Breathing/Zooming This technique may be viewed as an extension of BS on/off. The coverage of a BS can be varied by changing the transmit power and/or tilt of the antennas. Therefore during times of low network utilization, a few BSs (probably even macro ones) may be switched off while the others provide service by ‘zooming out’.

Inter RAN Sharing When a BS is switched off, the resulting coverage holes may also be served by BSs of *other* service providers operating the same region, for a certain fee. In this approach one can imagine the collective RAN of all the operators as a single network and all redundant BSs can be switched off. This is a very lucrative technique and can result in very high energy savings. However there are additional practical concerns such as cost sharing between the operators and regulatory issues.

Service Dimming The cost of providing a certain service is dependent on its type as well. For example, providing Third Generation (3G) services is more energy consuming than plain Second Generation (2G) services. However, the former generates a higher revenue but, during low activity periods the revenue would be less. Thus we may ‘switch off’ certain services at certain

BSs to save energy.

These energy saving techniques may be employed, however it must be ensured that the QoS guarantees mandated must not be violated. It turns out that obtaining the optimal network ‘state of operation’ while maintaining these QoS constraints is quite a challenge as we discuss in the following.

1.2.1 Challenge

Consider the BS on/off problem, with N number of BSs in the network. Now which of these are to be switched ‘off’ and which must remain ‘on’ while meeting QoS guarantees? In the worst case, there are 2^N possibilities which must be checked in order to obtain the optimal. Thus brute force is not a viable option, however we cannot do any *better*! The authors of [10] have considered a simple network model for BS on/off and showed that it is NP Hard. Similar results have been shown in [11] as well. If we consider cell zooming as well, energy consumption of a BS increases with increase in transmit power and it is not straightforward to determine which BSs must be switched off and which would compensate for the coverage holes formed. In this work, we address this problem of determining the optimal way in which to operate the network subject to QoS guarantees. We present a review of existing literature in the following section.

1.3 Related Work

The earlier works such as [12, 13, 14] among others have investigated potential savings of BS on/off mechanisms. These authors have demonstrated that a significant savings (upto 25%-30%) of energy consumption is possible. Further more, with multi-operator sharing even more savings can be obtained.

These works have focused on demonstrating the feasibility and potential for savings. The subsequent works have approached the problem of energy efficient

operation as an optimization problem with the objective of minimizing energy consumption of BSs, subject to certain QoS constraints. Several papers consider different network models and optimization techniques.

The authors of [15] for example, uses integer programming method for optimization. In [16], the authors consider optimal base station location problem which is much similar to BS on/off problem and also employs integer programming tools. These methods however are centralized in nature and is difficult to scale them to large networks.

In [10], the authors consider a network model for the BS on/off problem and demonstrated that the problem is NP-hard. Subsequently, many papers have proposed polynomial time heuristics for optimization. In [17], the set of BSs is partitioned into a few clusters of BSs, and during times of low load, only BS per cluster is switched on. In [18], the authors proposed a greedy heuristic to concentrate traffic on a few active BSs while switching off the rest. Similar greedy heuristics are considered in [19] and numerical simulations performed demonstrate the effectiveness of the proposed algorithms. The greedy heuristic has been justified with the observation that the energy consumption as a function of set of active BSs is *approximately* sub-modular in nature. The authors of [11], proposed algorithms based on the observation that BS on/off can be posed as a minimal set cover problem, building on their previous work [20]. They also prove that the approximation ratio of the proposed algorithm is a constant. [10] also obtains similar results. However, the algorithms proposed are centralized in nature and are restricted to BS on/off alone.

Cell breathing may be seen as an extension of BS on/off, to include transmit power, height and tilt of antennas as controllable parameters as well. In [9, 21, 22], the authors propose distributed algorithms for cell breathing based on the idea to concentrate load on a few BSs alone, so as to switch off the rest. However, one important thing to consider for cell breathing (and BS on/off) is that channel

allocations must be changed along with changes in antenna configuration in order to avoid inter-cell interference [23].

In contrast, a recent paper [24] formulates the BS on/off problem as a game played by BSs and proposes an asymptotically converging algorithm based on Spatial Adaptive Play (SAP). However, this approach is more general and has been applied to other problems involving wireless networks [25, 26]. In this work, we show the generality of this approach which can be used for cell zooming and joint channel allocation problem.

Most of the works so far suffer from issues – mainly oversimplified modelling assumptions – as was pointed out in a recent survey [27]. The framework presented in this thesis addresses several of these issues and also guarantees asymptotic convergence to the global optimal. Also, most of the works consider online heuristics based algorithms for optimization. Although there are pros and cons with online as well as offline algorithms [27], in this thesis we consider offline optimization due the following reasons – 1) Traffic behaviour being periodic in nature, we expect that for a given time of the day, the optimal network state will be the same for consecutive days. That is, there is no need to re-solve the same optimization problem over and over. 2) Using offline algorithms give us the additional flexibility for resource allocation e.g., cell breathing, channel reallocation, service dimming. 3) The number of switching operations will be few. 4) The algorithm presented in this thesis exhibits asymptotic convergence, and is thus preferable to be used in an offline manner. 5) It aids the service provider to have a global picture and control of the network and to monitor network health.

1.4 Organization of Report

The rest of the thesis is organized as follows. In Chapter 2, we present a centralized optimization approach to the energy minimization problem using Markov Chain

Monte Carlo (MCMC) methods. In Chapter 3, we present an alternative approach for optimization using game theory. This approach lends itself to distributed implementation, which is beneficial for large networks. Finally, conclusions and future directions are provided Chapter 4.

Chapter 2

Centralized Optimization

Approach

Conceptual outline of the proposed offline approach is presented in Figure 2.1. Relevant network information such as BS location, number of sectors, orientation of sectorial antennas, Tx-power and channel allocations are obtained from the network operator. Apart from this, historic data related to traffic is also obtained. This consists of hourly average Erlang load (for voice traffic) experienced by BSs for a period of a few months. Using this historic data, a prediction of hourly load for the next day is obtained. For these predicted loads, the energy optimal network operation state is obtained, subject to certain QoS constraints. We elaborate this in the following section.

2.1 Prediction and Network Simulator

Consider the set of BSs in a given area of interest. We refer to each sector of a BS as a node. Let $\mathcal{N} = \{1, 2, \dots, N\}$ denote the set of all nodes in the network. We assume that every node has some configurable parameters like, Tx-power, height/tilt of the antenna, and channels allocated. A particular configuration of

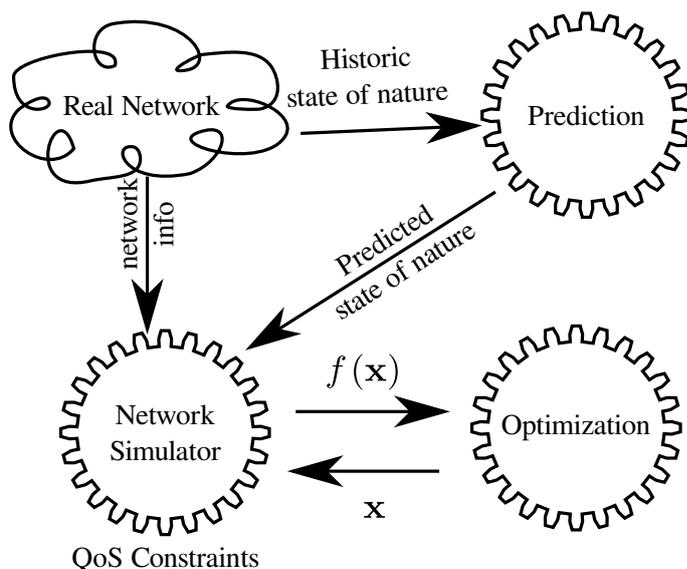


Figure 2.1: Framework for the centralized optimization approach

these parameters is referred to as a state of that particular node. That is, for a node j , x_j denotes a particular state from \mathcal{S}_j , the set of all possible states. We however, restrict \mathcal{S}_j to be a finite set. Therefore, all continuous valued parameters like Tx-power, are considered to be varied only in discrete steps. The vector $\mathbf{x} := (x_1, x_2, \dots, x_N)$ then denotes a state of the entire network which is an element of the set $\mathcal{S} := \mathcal{S}_1 \times \mathcal{S}_2 \times \dots \times \mathcal{S}_N$, the collection of all possible states of the network.

We impose the condition that the network state can be changed only at specific instants of time.¹ That is, we divide the entire day into ‘time frames’ of duration say, one hour each. At the beginning of each time frame, a particular network state is chosen and is kept the same throughout the duration of that frame. We assume that the way in which we operate the network in a time frame does not have any affect on another time frame. When frame duration is large, like one hour, this is a rather mild assumption. Therefore, we restrict attention to a fixed frame and determine the optimal network state, for that duration.

Since we cannot change states within a frame, we must ‘foresee’ the traffic

¹This assumption is not too restrictive. Service providers in fact prefer network re-configurations to be minimal and carried out only at specific hours.

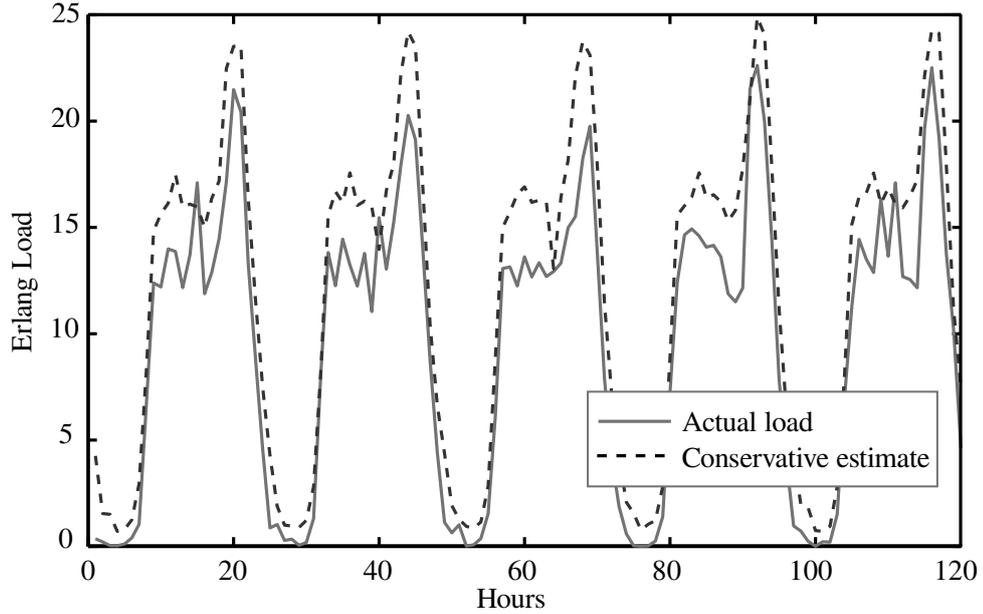


Figure 2.2: Hourly conservative estimates of Erlang load at a certain BS, obtained through Auto Regressive Moving Average modelling (adapted from [29]). The actual load will be lower than this estimate with a probability of 0.97.

conditions that the network would experience for the duration of that frame, and account for it while we seek the optimal state. Consider for example the case of voice traffic. We propose the use of methods suggested in [28, 29] to obtain conservative predictions of voice traffic (\hat{w}_j) at a node j for the given hour. In [29], the historic Erlang load as obtained from the operator has been modelled using Auto Regressive Moving Average (ARMA) method. Confidence intervals are obtained as shown in Figure 2.2. We observe that with a large probability, the actual load w_j at the node will be less than this 97-percentile conservative estimate \hat{w}_j . We then impose a QoS constraint that, for these conservative predictions \hat{w}_j , and a given network state \mathbf{x} , the call block rate $CBR_j(\mathbf{x}, \hat{w}_j)$ at node j does not exceed a certain threshold. We can follow a similar procedure for data traffic throughput requirements as well.

In general, QoS metrics² at node j would be of the form $q_j(\mathbf{x}, \omega_j)$, where ω_j

²QoS metrics like call drop ratio, lowest signal quality at serving Mobile Unit (MU), handover drop ratio, handover ping-pong and average packet delay are computed at each node. These would depend on some state of nature and the state of network.

is some quantity which is beyond our control – a *state of nature*, if we may. We obtain appropriate conservative predictions $\hat{\omega}_j$ for these ω_j (from the prediction block in Figure 2.1) and reduce $q_j(\cdot)$ to a function of \mathbf{x} alone. Then, for a given state \mathbf{x} , these QoS metrics are computed by the network simulator block depicted in Figure 2.1. Let $Q_j(\mathbf{x})$ denote whether or not any QoS violation at node j would happen. That is, $Q_j(\mathbf{x})$ is set to 1, whenever any of the QoS metric $q_j(\mathbf{x})$ is unacceptable, otherwise it is set to 0.

Given the constraints as described, the objective is to minimize the total energy (cost) consumed. Let $c_j(\mathbf{x})$ denote the cost incurred by node j , when the network state is \mathbf{x} (over the duration of the frame). The optimization problem then is,

$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{S}} \sum_{j \in \mathcal{N}} c_j(\mathbf{x}), \\ \text{s.t. } Q_j(\mathbf{x}) = 0 \quad \forall j \in \mathcal{N}. \end{aligned} \tag{2.1}$$

In order to convert this constrained problem into an unconstrained one, we do the following. Associated with every node j , we construct a ‘welfare function’ $f_j(\cdot)$ as follows:

$$f_j(\mathbf{x}) := \begin{cases} -c_j(\mathbf{x}), & \text{if } Q_j(\mathbf{x}) = 0, \\ -\Delta, & \text{otherwise,} \end{cases}$$

where Δ is a large positive number (penalty). That is, function $f_j(\cdot)$ captures the cost incurred by j so long as QoS constraints are not violated, otherwise a large penalty is awarded. With sufficiently large Δ (for e.g., the largest possible energy consumed by the network i.e., $\Delta = \sum_j \max_{\mathbf{x}} c_j(\mathbf{x})$), the optimization problem (2.1) is equivalent to the following unconstrained problem:

$$\max_{\mathbf{x} \in \mathcal{S}} \sum_{j \in \mathcal{N}} f_j(\mathbf{x}). \tag{2.2}$$

Thus it is enough to maximize the social welfare function, $f(\mathbf{x}) := \sum_{j \in \mathcal{N}} f_j(\mathbf{x})$,

over the set \mathcal{S} . The optimization block depicted in Figure 2.1 performs this optimization by using Markov Chain Monte Carlo (MCMC) based algorithms which is described next. A concrete implementation example is provided in Section 2.3

2.2 Markov Chain Monte Carlo Optimization

Markov Chain Monte Carlo (MCMC) is a general method to generate variates of large dimensional random variables. It has many applications, one of which is optimization (see [30] for more details). Suppose we are given a function $f(\mathbf{x})$ which needs to be maximized. For the sake of simplicity, suppose that the domain \mathcal{S} of $f(\cdot)$ is finite. Now, consider a random variable Z_β whose distribution is as given by,

$$\Pr(Z_\beta = \mathbf{x}) = \frac{e^{\beta f(\mathbf{x})}}{\sum_{\mathbf{x}' \in \mathcal{S}} e^{\beta f(\mathbf{x}')}}, \quad (2.3)$$

known as the Gibbs distribution. It can be observed that for large β , the distribution of Z is biased towards the global optimal \mathbf{x}^* of $f(\cdot)$. That is,

$$\lim_{\beta \rightarrow \infty} \Pr(Z_\beta = \mathbf{x}^*) = 1, \quad (\text{assuming a unique global optimum}).$$

Therefore if we can generate variates of Z , then with a large probability, the sample generated will be the global optimum. Note that Z is a very large dimensional random variable and it is difficult to generate variates by conventional methods. Moreover, the distribution of Z is not known completely as the denominator of (2.3) cannot be computed.

In such a case, variates of Z are obtained by constructing a Markov chain, whose unique stationary distribution is as given by (2.3). There are many well known methods to do so, the simplest and the first developed one being Metropolis-Hastings (MH) algorithm, which is described next.

2.2.1 Metropolis Hastings Algorithm

The MH algorithm is given in Algorithm 1. The intuition behind this algorithm is as follows. Suppose we are at a state \mathbf{x} . We then pick a (uniformly) random state \mathbf{x}' from the set \mathcal{S} . We now consider the question should this state be accepted or not. If the new function value $f(\mathbf{x}')$ is better (higher) than our current function value $f(\mathbf{x})$, then we readily accept this state. Otherwise, the new state is accepted with a certain ‘risk’.

Algorithm 1 MH algorithm

Require: $f(\cdot)$, \mathcal{S} , β **Ensure:** \mathbf{x} is global optimizer of $f(\cdot)$, after a large number of iterations, with high probability.

```
1: loop
2:    $\mathbf{x}' \leftarrow$  uniformly random state from  $\mathcal{S}$ 
3:   if  $f(\mathbf{x}') > f(\mathbf{x})$  then
4:      $\mathbf{x} \leftarrow \mathbf{x}'$ 
5:   else
6:      $\mathbf{x} \leftarrow \mathbf{x}'$  with probability  $e^{\beta(f(\mathbf{x}')-f(\mathbf{x}))}$ 
```

The stationary distribution of the ensuing Markov chain is as given by (2.3), and therefore after a large number of iterations and for sufficiently large β , we obtain the global optimal with a large probability.

With reference to Figure 2.1, the flow is as follows. First conservative predictions for the states of nature are obtained and the optimizer is started with some starting state. Then, the optimizer proposes a random state \mathbf{x}' . The network simulator then simulates the performance of this proposed state \mathbf{x}' . The value $f(\mathbf{x}')$ is computed taking into consideration the QoS metrics of interest. The proposed state is then accepted or rejected as discussed above and this whole process is repeated for a large number of iterations.



Figure 2.3: Network information as obtained from operator

2.3 An Example Implementation

As a concrete example for the proposed model above, we present an implementation, for the BS on/off problem. We have obtained data consisting of BS position and antenna orientations of 66 BSs (as depicted in 2.3) in an urban region (aprox. $6\text{km}\times 6\text{km}$) of the city of Lucknow in the state of Uttar Pradesh of North India, from the network of one of the largest service providers in India. Additionally, the actual voice Erlang load experienced by these BSs for a particular day at 3 AM has been obtained. The simulation parameters are summarized in Table 2.1. For the purpose of determining received signal strength and simulating ‘static’ users, we generate a uniform spatial grid of points spaced 10m apart from each other. Grid points are associated to the node with the highest received signal strength, if it is greater than -95dBm (see Figure 2.4a). Otherwise, the grid point is considered to be out of coverage. Traffic generated by a grid point is modelled as follows. We first obtain user–node associations when all the nodes are switched-on, then divide the actual Erlang load \hat{w}_j (as obtained from the data collected) at node j ,

equally to all grid points associated to it. This divided load is the traffic generated by a grid point (see Figure 2.4b).

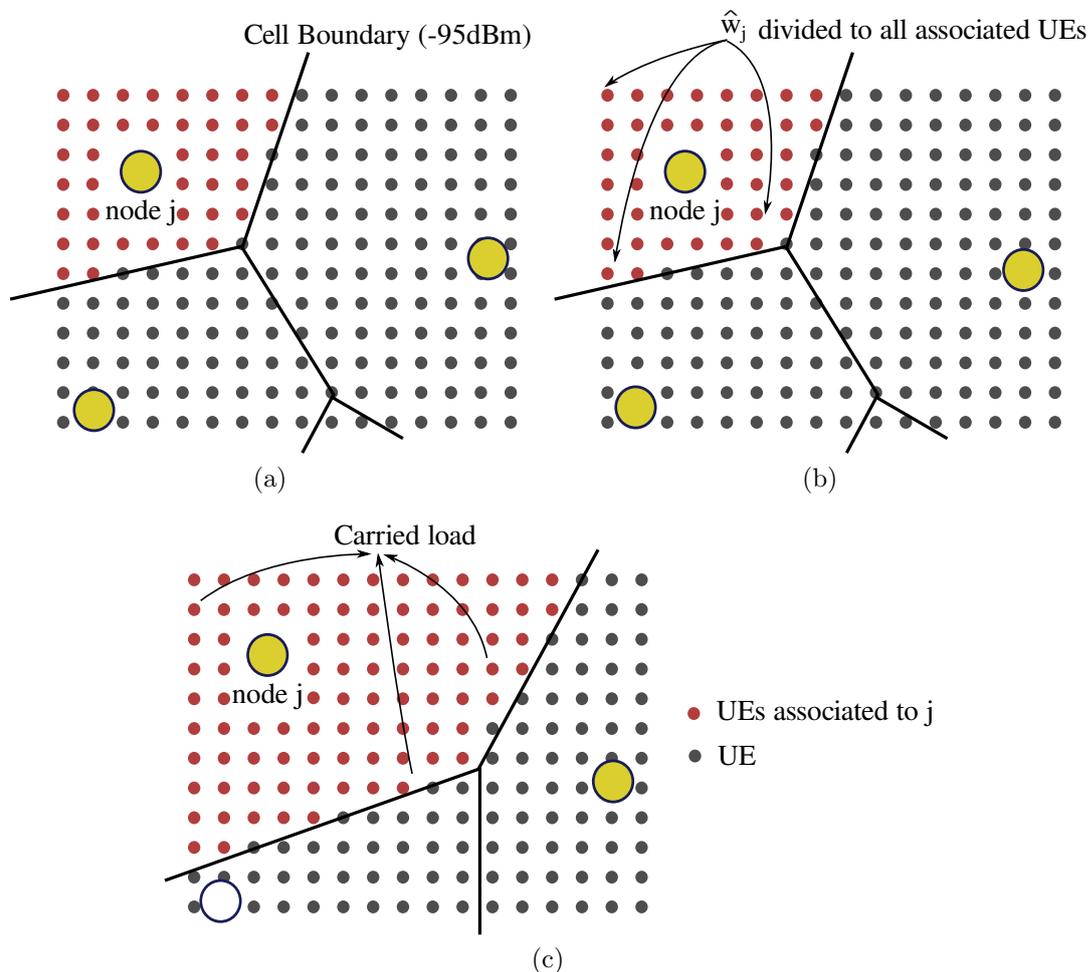


Figure 2.4: Network simulator. (a) and (b) are for the state – all nodes switched-on. We obtain the traffic generated by UEs by dividing \hat{w}_j . For some other state, the carried traffic is got by accumulating the traffic from UEs.

Now, for a different state \mathbf{x} , we re-compute user–node associations, and say that the traffic-load node j is ‘expected’ to carry is computed by accumulating the traffic from all the individual grid points associated to node j in this new state (see Figure 2.4c). We then, compute the CBR at node j for this new traffic-load. A QoS violation is said to occur at node j if either, (a) CBR computed is greater than 0.01, or (b) any of the grid points which could *potentially* have been served by node j , is out of coverage.

Table 2.1: Simulator parameters

Number of BSs	66
Nodes per BS	3
States of node	‘off’/‘on’
Transmit power	10W if ‘on’, else 0W
Propagation model	free space loss model, exponent = 4.5
Radiation pattern	$\cos(\theta), \theta \in [-\pi/2, \pi/2]$
Receive signal threshold	-95dBm
User-node association	node with highest Rx power
Num. TDMA voice channels per node	$7 \times (3\text{TRX cards}) = 21$ channels

The cost of operating a BS is modelled as follows – 0.4 units due to air conditioning, which is consumed whenever at least one of its 3 sectors (children nodes) are switched-on, in addition to 0.2 units per each switched-on child node. The welfare function of node j then is modelled as:

$$f_j(\mathbf{x}) = \begin{cases} -\text{cost of parent BS}/3, & \text{if } Q_j(\mathbf{x}) = 0, \\ -50, & \text{otherwise.} \end{cases}$$

Thus, given \mathbf{x} , the individual welfares are computed, followed by the social welfare $f(\mathbf{x})$. The value of $f(\mathbf{x})$ is given to the MH-algorithm as depicted in Figure 2.1.

This BS on/off problem as discussed was implemented in MATLAB. Theoretically, MCMC algorithms must run for an infinite duration while *slowly* increasing β to infinity as well. However, practically, it is not possible to do so. Thus, the algorithm has to be terminated after a finite duration. Further in practice, the whole algorithm is typically run again several times over and the best output among them is taken as the final output. Repeated runs of the optimization was performed with $\beta = 10$. The best outcome from these runs was that out of the considered 198 nodes, only 93 were required to be switched-on. This implies a significant reduction in energy consumption. The result might be a bit over liberal, as the Tx-power of all nodes was assumed to be 10 Watt (due to lack of sufficient information). In such a highly dense region, this would not be the case.

To conclude this chapter, we point out the pros and cons of such centralized simulation based optimization. One advantage is that we can model the network to arbitrary precision and we are not restricted to simplistic models used in literature. Though this does come at a price, as the number of nodes increases, simulations may take up a lot of time. The main disadvantage is the centralized nature. Using distributed algorithms, we can break up the simulations and optimizations into smaller parts. In the next chapter, we discuss one such approach for distributed computation.

Chapter 3

Distributed Optimization

Approach

In this chapter we take an alternative approach for optimization. As mentioned in previous chapter, MCMC based optimization though offers the flexibility to model a network with arbitrary constraints, is centralized in nature. Simulating a large network may cause difficulties. Thus, we look to ways for distributed computation. Game theoretic tools can be employed to obtain distributed algorithms to achieve a common objective. This comes under the broad topic called Utility Design [31], wherein there are a certain number of self-interested agents who look to always maximize their own ‘utility,’ and a designer who intends to achieve a certain goal. The task of the designer is to design the rules of interaction among the agents and designing utilities of the players such that their self-interested behaviour leads to the desired goal of the designer. In our case, as the designer our objective is to minimize the total energy consumption of the network and we model the nodes as being self interested agents and the task is to design their utilities in order to meet our objective.

Since the approach presented in this chapter can be applied in many situations, we present and discuss an abstract game model first. In Section 3.4, we

demonstrate the applicability of this model for the case of energy minimization.

3.1 Game Model

Consider a game played by players from a finite set \mathcal{N} , where each player has a finite action set \mathcal{S}_j . Associated with each player is a welfare function $f_j(\mathbf{x})$, which depends on the action chosen by not only player j , but on other player's actions as well. It is of interest to maximise the social welfare $f(\cdot)$, which is the sum total of the welfare of all the players in the game.

If we let the utility of each player to be the same as the social welfare function $f(\cdot)$, then selfish behaviour to maximise this will lead to the desired performance. That is, this is a purely cooperative game. But, we will have problem implementing this if the number of players is large.

However, if the action of a player affects only a small group of players, then we can do better. For each player j , let the set $\mathcal{N}_j \subseteq \mathcal{N}$ be the set of players whose welfares are affected by player j 's actions. Set \mathcal{N}_j is called the set of neighbours of j and we implicitly have $j \in \mathcal{N}_j$. That is, there exists a directed graph (called as interaction graph) with edges going from player j to the players in $\mathcal{N}_j - \{j\}$ and changes to state of player j does not have any effect on non-neighbouring players. Formally,

Definition 1 (Local interaction). A set of finite valued functions $\{f_j(\cdot) \forall j\}$ is said to satisfy local interaction property if, $\exists \mathcal{N}_j \subseteq \mathcal{N}$ (called neighbours of j) such that, $\forall k \notin \mathcal{N}_j$ and $x_j, x'_j \in \mathcal{S}_j$,

$$f_k(\mathbf{x}) = f_k(x_j, \mathbf{x}_{-j}) = f_k(x'_j, \mathbf{x}_{-j}). \quad (3.1)$$

Now, if we let the utility $u_j(\mathbf{x})$ of each player to be the sum of welfares of all

the players in it's group i.e.,

$$u_j(\mathbf{x}) := \sum_{k \in \mathcal{N}_j} f_k(\mathbf{x}), \quad (3.2)$$

then, the resulting game is a potential game, with the potential function to be the social welfare function itself¹.

Theorem 1. *The local altruistic game is a potential game, with the social welfare function as its potential function.*

Potential games are characterized by the existence of a function $\phi(\mathbf{x})$ (called potential function) with the following property.

$$u_j(x_j, \mathbf{x}_{-j}) - u_j(x'_j, \mathbf{x}_{-j}) = \phi(x_j, \mathbf{x}_{-j}) - \phi(x'_j, \mathbf{x}_{-j}), \quad (3.3)$$

with the standard notation, $\mathbf{x} \equiv (\mathbf{x}_j, \mathbf{x}_{-j})$. Therefore an action of player j which improves its utility, also improves the potential function. It so happens that the potential function is the same as the social welfare function. Thus, self-interested behaviour of agents tend to achieve the goal of the designer. We elaborate on this in the next section. In order to show the generality of this game model, we next present a sample application. We apply this model for our problem of energy optimization in Section 3.4.

Since the game is a potential game, the global optimizer of $f(\mathbf{x})$ is a Nash equilibrium. It also has Finite Improvement Path (FIP) property and thus best-response dynamics converge to a Nash equilibrium, since the domain of $f(\cdot)$ is a finite set. However there may be multiple inefficient Nash equilibria corresponding to the local maximas of $f(\mathbf{x})$. In this work, we consider SAP dynamics to seek the global optimal which is considered in the next section.

¹There are other ways to design player utilities so as to have this *alignment* such as Identical Interest Utilities (IIU) and Wonderful Life Utilities (WLU) [32]. IIU is impractical in most cases and we show that WLU and LAU are equivalent as far as this work is concerned in Section 3.6.2

3.1.1 Application: Welfare of a Nation

We digress briefly to show the general nature of this game model and show a sample application. Consider a nation, where each person is a player. Suppose that by welfare of a person, we mean her “happiness quotient”. In general, the action of a common person (*aam-aadmi*) affects the happiness of her immediate family members and friends only. So, if each person chooses an action which makes her family (herself included) happy, then the entire nation will become happy. However, this is not entirely accurate, because actions of say the prime minister of the nation affects the welfare of the entire population. So, practically implementing this may not be possible.

3.2 Spatial Adaptive Play

The spatial adaptive play algorithm first introduced by Peyton Young [33], can be applied to any (finite) potential game to determine the global optimum of its potential function. We describe the SAP algorithm:

Algorithm 2 SAP algorithm

Require: \mathcal{N} , $\{f_j(\mathbf{x})\}$, β **Ensure:** \mathbf{x} is global optimizer of $f(\cdot)$, after a large number of iterations, with high probability.

- 1: **loop**
 - 2: $j \leftarrow$ random player from \mathcal{N}
 - 3: **for all** $x \in \mathcal{S}_j$ **do**
 - 4: Obtain $f_k(x, \mathbf{x}_{-j}) \forall k \in \mathcal{N}_j$
 - 5: Compute $u_j(x, \mathbf{x}_{-j}) \leftarrow \sum_k f_k(x, \mathbf{x}_{-j})$
 - 6: $P(x) \leftarrow e^{\beta u_j(x, \mathbf{x}_{-j})}$
 - 7: Convert $P(\cdot)$ to a valid pmf
 - 8: $x_j \leftarrow$ random sample from the pmf $P(\cdot)$
-

For notational convenience, we represent the algorithm as an infinite loop, and the variables used change ‘in-place’. This algorithm is like perturbed best-

response dynamics. In each iteration of the algorithm, a node j is picked at random and its state x_j is updated. The other nodes maintain their previous state \mathbf{x}_{-j} . The new state of j is obtained as follows. For each possible state $x \in \mathcal{S}_j$, obtain corresponding $f_k(x, \mathbf{x}_{-j})$ from all its neighbours $k \in \mathcal{N}_j$. Compute the corresponding utilities $u_j(\cdot)$ according to (3.2) and subsequently compute $P(x)$ as (3.4). Then, x_j is a random sample from the following distribution over \mathcal{S}_j ,

$$P(x) = \frac{e^{\beta u_j(x, \mathbf{x}_{-j})}}{\sum_{x'} e^{\beta u_j(x', \mathbf{x}_{-j})}}. \quad (3.4)$$

Here, $\beta > 0$ is a learning rate parameter. For small values of β , the nodes take a higher risk and pick non optimal actions, while for large β , only optimal actions are taken. Thus, β determines the tradeoff between exploration and exploitation. We note that with $\beta = \infty$, the algorithm is equivalent to the best response dynamics. For finite β however, the randomness in picking an action ensures that the algorithm does not get stuck at a local optimum, akin to simulated annealing. This algorithm, in principle can be used to obtain the global optimizer of the potential function with arbitrarily high probability.

Theorem 2. *SAP Algorithm defines a Markov chain on the state space \mathcal{S} with a unique stationary distribution given by the Gibbs distribution,*

$$\boldsymbol{\pi}_\beta(\mathbf{x}) = \frac{e^{\beta f(\mathbf{x})}}{\sum_{\mathbf{x}' \in \mathcal{S}} e^{\beta f(\mathbf{x}')}}. \quad (3.5)$$

The proof of the theorem is given in [25, 26] and it follows along the lines of that given in [33]. The boundedness of the welfare functions is needed to ensure that all the states form a single communicating class. Thus, SAP is similar to simulated annealing [30], a centralized algorithm which may also be used for optimization. Note that,

$$\lim_{\beta \rightarrow \infty} \boldsymbol{\pi}_\beta(\mathbf{x}) = 0, \text{ if } \mathbf{x} \text{ is not a global optimal.} \quad (3.6)$$

As mentioned earlier, SAP is applicable to any potential game, if the agents know their respective utility functions. As such, it does not exploit the local interaction property. In the next section, we present the Concurrent-SAP algorithm which exploits local interaction property as a means to improve the convergence rate of the algorithm. Also note that until this point, we do not require the interaction graph to be undirected. For the C-SAP algorithm we require the interaction graph to be undirected.

3.3 Concurrent SAP (C-SAP)

In [25] a modification to SAP has been proposed – nodes which are more than one-hop away from each other (i.e., non-neighbouring nodes) can update concurrently and independently (hence the name concurrent). For example, nodes 2, 5, 6, and 10 in Figure 3.1 can update simultaneously within the same iteration. However, for this version of the C-SAP, the stationary distribution *may not* be the same as the Gibbs distribution (3.5). We construct a counter example to demonstrate this, which is presented in Section 3.6.1. Moreover, for the counter example presented, in the limit $\beta \rightarrow \infty$, the global optimal is picked with a probability of zero. Thus optimality of this version of C-SAP is not guaranteed. In order to rectify this, we propose that nodes which are more than *two*-hops away from each other can update concurrently. For example, with reference to Figure 3.1, nodes 1, 4 and 9 can update in the same iteration. We show that this modification is sufficient to ensure that the stationary distribution is the same as the Gibbs distribution. Moreover, it can result in a faster convergence compared to SAP, where only one node is updated at a time. We introduce some more notations before presenting the modified C-SAP algorithm.

Definition 2 (Selection Scheme). A selection scheme consists of $(\mathcal{C}, \mathbf{p})$, where the set $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_M\}$ denotes a collection of sets $\mathcal{C}_k \subset \mathcal{N}$ such that,

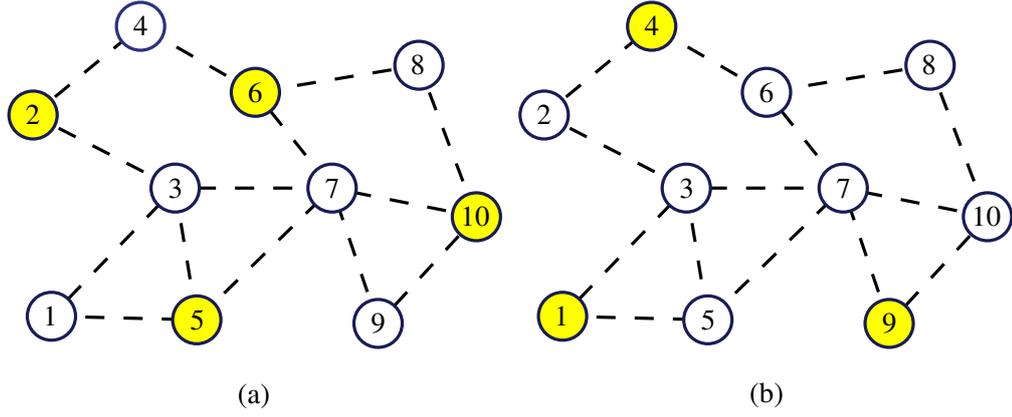


Figure 3.1: Illustration of the difference between 1-hop C-SAP as proposed in [25] and 2-hop C-SAP proposed in this work. Nodes 2, 5, 6 and 10 as shown in (a) are at least than one-hop away from each other. Nodes 1, 4, and 9 shown in (b) are at least than 2-hops away from each other.

1. For all distinct $i, j \in \mathcal{C}_k$, we have $i \notin \mathcal{E}\mathcal{N}_j$.

$\mathcal{E}\mathcal{N}_j$ denotes the set of two hop neighbours of j .

We call such a set \mathcal{C}_k as a two-hop independent set.

2. $\forall j \in \mathcal{N}$, there exists a k such that, $j \in \mathcal{C}_k$. That is, every node is present in at least one of the classes.
3. $\mathbf{p}_k > 0$ denotes the probability of picking the set \mathcal{C}_k .

Now, for a given selection scheme, the modified C-SAP algorithm is as follows.

Algorithm 3 C-SAP algorithm

Require: \mathcal{N} , $\{f_j(\mathbf{x})\}$, \mathcal{C} , β

Ensure: \mathbf{x} is global optimizer of $f(\cdot)$, after a long time, with high probability.

- 1: **loop**
 - 2: $\mathcal{C}_k \leftarrow$ random class from \mathcal{C} according to \mathbf{p}
 - 3: **for all** $j \in \mathcal{C}_k$ **do**
 - 4: Update node j as in steps 3–8 of SAP algorithm.
-

Here, in each iteration, we do the following. Select a random class \mathcal{C}_k from \mathcal{C} according to \mathbf{p} . Then, all nodes in \mathcal{C}_k independently update their states as in the SAP algorithm.

Theorem 3. For a fixed selection scheme $(\mathcal{C}, \mathbf{p})$, the C-SAP algorithm defines a Markov chain on the state space \mathcal{S} , with a unique stationary distribution as given in (3.5).

See Section 3.6.3 for proof. It can be seen that SAP corresponds to a particular selection scheme – $\mathcal{C}_k = \{k\}$ and constant vector \mathbf{p} . There are infinitely many possible selection schemes, of which we next describe two selection schemes which, intuitively might have better convergence rate than that of the original SAP.

3.3.1 C-SAP Scheme-1

In this scheme, we build a set of nodes which would update, for a given iteration of the C-SAP algorithm as follows. We start with empty sets \mathcal{C}_k and \mathcal{B} . The set \mathcal{B} contains the extended neighbours \mathcal{EN}_j of all nodes selected in \mathcal{C}_k . Repeat the following till we exit the loop. Pick a random node $j \in \mathcal{N}$. If $j \in \mathcal{B}$ then the node selection is done and we exit. Else, we select j and add all its extended neighbours to set \mathcal{B} . The pseudocode is as follows.

Algorithm 4 Scheme-1

Require: $\mathcal{N}, \{\mathcal{N}_j\}$

Ensure: \mathcal{C}_k is a two-hop independent set.

- 1: $\mathcal{C}_k \leftarrow \emptyset, \mathcal{B} \leftarrow \emptyset$
 - 2: **loop**
 - 3: $j \leftarrow$ random node from \mathcal{N}
 - 4: **if** $j \in \mathcal{B}$ **then**
 - 5: **break**
 - 6: **else**
 - 7: $\mathcal{C}_k \leftarrow \mathcal{C}_k \cup \{j\}$
 - 8: $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{EN}_j$
-

Note that this algorithm induces a valid selection scheme – consisting of *all* possible 2-hop independent subsets of \mathcal{N} , each of which has a strictly positive probability of being picked. We show that this scheme converges at least as fast

as the original SAP scheme.

Theorem 4. *If the ϵ -mixing time of SAP is m , then the ϵ -mixing time of Scheme-1 is at most m .*

The proof is presented in Section 3.6.3.

3.3.2 C-SAP Scheme-2

In this scheme, \mathcal{C} is the ‘best’ possible partition of the set \mathcal{N} , i.e., the partition of the lowest cardinality while satisfying the conditions of a selection scheme. \mathbf{p} is a constant vector. We present our motivation for considering Scheme-2. Obtaining this partition \mathcal{C} is a two-hop colouring problem on the interaction graph. That is, assign colours to the nodes such no two nodes which are at most two-hops away from each other have the same colour. This is the standard colouring problem on the two-hop interaction graph constructed from the interaction graph by adding additional edges from a node to all its two hop neighbours. It is well known that the number of distinct colours needed is bounded from above by $\gamma + 1$, where γ is the degree of the graph. In reality, we expect that the degree of the two-hop interaction will saturate after sufficiently large \mathcal{N} .

There are well known existing distributed algorithms for the graph colouring problem which can take at most $\gamma + 1$ number of colours. Therefore, once the nodes in the network obtain their respective colours, implementing C-SAP on this scheme is a more practical than Scheme-1 which repeatedly tries to construct \mathcal{C}_k in each iteration.

So far in this chapter, we described a local interaction game model and discussed algorithms to obtain the globally efficient Nash equilibrium for this type of game. In the next section, we will apply this game model for the problem of energy minimization of cellular networks.

3.4 Application to Energy Optimization

As mentioned earlier, this game model can be applied to any problem which exhibits the local interaction property. For a certain network models and QoS constraints, the energy minimization problem exhibits local interaction property. The network model presented in Section 2.3 is one such example. In fact we can strengthen this model to incorporate additional features as well which we discuss in this section.

Consider a cellular network. If we are to change the state of a particular node say j , then it is intuitively appealing to say that nodes which are spatially far away from j are completely unaffected i.e., their welfare functions are not dependent on j 's actions. However, how “far” apart should the nodes be in order for this to hold? If the size of set \mathcal{N}_j is large, then it becomes impractical to apply the methods discussed earlier.

In the following, we present a network model with a definition of neighbours and discuss local interaction properties for some of the QoS metrics.

3.4.1 An Implementation Framework

Consider a set of nodes (sectors of BS). Each node has the following re-configurable parameters – 1) Transmit power 2) Height/tilt of antenna 3) Frequency channels allocated.

User Equipment (UE) are assumed to be static and spread throughout the area of interest with a separation of say, 10m from each other as with the network model presented in Section 2.3. We consider that users only generate voice traffic, however data traffic can be incorporated with models similar to that presented in [34]. UEs are assumed to be independent of each other and generate voice calls according to Poisson point process. The arrival rates of different UEs can be different. This models spatial traffic profile. Call hold duration is assumed to be

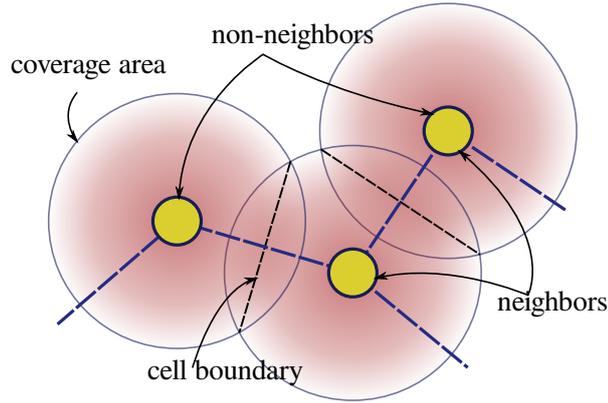


Figure 3.2: Interaction graph based on Definition 2. Nodes which can *potentially* overlap with each other are considered neighbours. We add an undirected edge from j to all its neighbours \mathcal{N}_j . Note that the neighbours of node j is *not* dependent on its current state. It is a fixed set.

distributed according to exponential distribution with rate μ . UE-node association is as follows.

Assumption (User–node association). UEs are always associated to the node with the highest received signal strength (RSS).

Propagation model is free-space path loss model with some arbitrary path-loss exponent (say 4.5). A UE is considered out of coverage if the RSS from the associated node is less than some threshold (say -95dBm). Therefore, the coverage of node is defined as the area within which the RSS is greater than this threshold. Shadowing can also be modelled with minor modifications. Neighbours to a node are defined as follows.

Definition 3 (Neighbours). A node k is said to be a neighbour of node j if there exists *some* states $x_k \in \mathcal{S}_k$ and $x_j \in \mathcal{S}_j$, such that their coverage areas overlap. Moreover, it is assumed that j is a neighbour of itself, i.e., $j \in \mathcal{N}_j$.

Note that by this definition, $k \in \mathcal{N}_j \Rightarrow j \in \mathcal{N}_k$ and the interaction graph reduces to an undirected graph (refer to Figure 3.2). Also note that the siblings of j also belong to \mathcal{N}_j . For this particular definition of neighbours, we discuss the

local interaction property of welfare functions as defined in Chapter 2 i.e.,

$$f_j(\mathbf{x}) := \begin{cases} -c_j(\mathbf{x}), & \text{if } Q_j(\mathbf{x}) = 0, \\ -\Delta, & \text{otherwise,} \end{cases}$$

where $c_j(\mathbf{x})$ captures the cost of operation of node j and $Q_j(\mathbf{x})$ captures the validity of all QoS metrics of interest at node j . The cost of j is modelled as before i.e., one third of cost of parent BS, where cost of parent BS is as given in Section 2.3. Note that $c_j(\cdot)$ thus satisfies the local interaction property. It only remains to consider $Q_j(\mathbf{x})$. In the following, we discuss about different QoS metrics which will hold.

Signal Quality

A UE is said to be out of coverage or not served, if the RSS at its location is less than a particular threshold, for example -95dBm. For a given state \mathbf{x} , a signal quality violation at node j is said to happen if there exists an out-of-coverage UE under the given state, which could *potentially* have been served by j . That is, suppose for the given state \mathbf{x} , a particular UE is out of coverage. But if j were to take a state x_j , then it would not be out-of-coverage. In other words, a signal quality violation at j occurs if there exists some UE within node j 's *purview*, which is not served. Note that this constraint satisfies the local interaction property, i.e., only the neighbours of j can affect this constraint.

Signal Quality for Handoffs

In order to minimize ping-pong effect during handoffs, the following constraint is generally imposed. The coverage of node j slightly extends over that of its neighbours as depicted in Figure 3.3. Therefore, j must provide a signal strength of at least -95dBm up to a distance of h from the cell boundary as well. This constraint also satisfies the local interaction property.

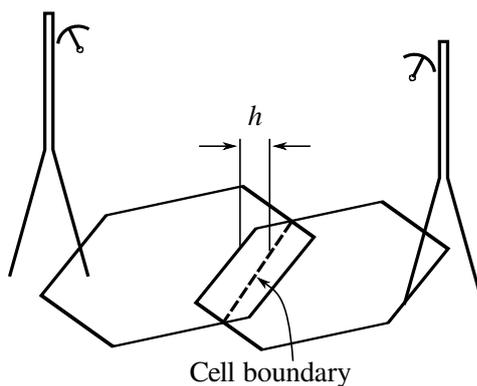


Figure 3.3: Signal quality constraint for reducing ping-pong

Call Block Ratio

For a switched-on BS/node, the call block ratio must not exceed a certain threshold say 0.01%. The accumulated arrival rate at j is computed by adding the rates of all UEs within its coverage. The CBR at node j can be computed by the standard Erlang-B formula, given the channels allocated to node j . Note that this also satisfies the local interaction property. To see this, consider the nodes as shown in Figure 3.5a. Node 3 is not a neighbor of node 1. Suppose now that node 3 is switched off, then the area covered by node 2 would be higher, and so would the traffic carried by it (thus, CDR_2 will increase). However, the traffic carried by node 1 would be unaffected (so, CDR_1 is unaffected). This is due to the above assumption that a UE in region A as shown in Figure 3.5b will be always served by node 2, and there is no ‘propagation’ of the effect of switching-off node 3.

Channel Allocation Constraint

For a given state \mathbf{x} , channel allocations of switched-on nodes cannot be arbitrary. Specifically, channels allocated among *active* neighbours must be orthogonal. Therefore, for a given state \mathbf{x} , a channel violation at node j is said to occur if the same channel is allocated to some other node whose coverage overlaps with that of j . Note the local dependence of this constraint as well.

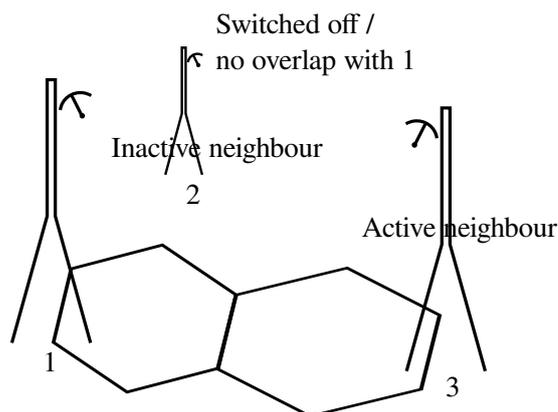


Figure 3.4: Channel allocation constraint. For the given state, nodes 1 and 3 must not have common channels.

A Constraint which Does NOT Satisfy Local Interaction

Handoff Drop Ratio (HDR) does *not* satisfy the local interaction property straight-away for the current definition of neighbours. Consider the network in Figure 3.5c, with node 3 switched-off initially. Now if node 3 is switched-on, then UEs moving from region B to region A will experience a higher HDR (i.e., HDR_1 increases). This is because with node 3 switched-on, there is a higher traffic handed over to node 2 which in turn is handed off to node 1. Thus the handoff arrival rate to node 1 increases and so does the handoff drop rate. Thus the HDR at node 1 is affected by changes to node 3 which is not a neighbour. However, with additional assumptions, this can be made to satisfy the local interaction property. For example, if we make the assumption that any call does not get handed over more than once, then the property holds. We may think of this assumption as follows – a call originating in a particular cell resides within that cell for a certain time. If this residence is sufficiently large compared to the call holding time then we may make such an assumption. However, additional work is needed with regards to mobility.

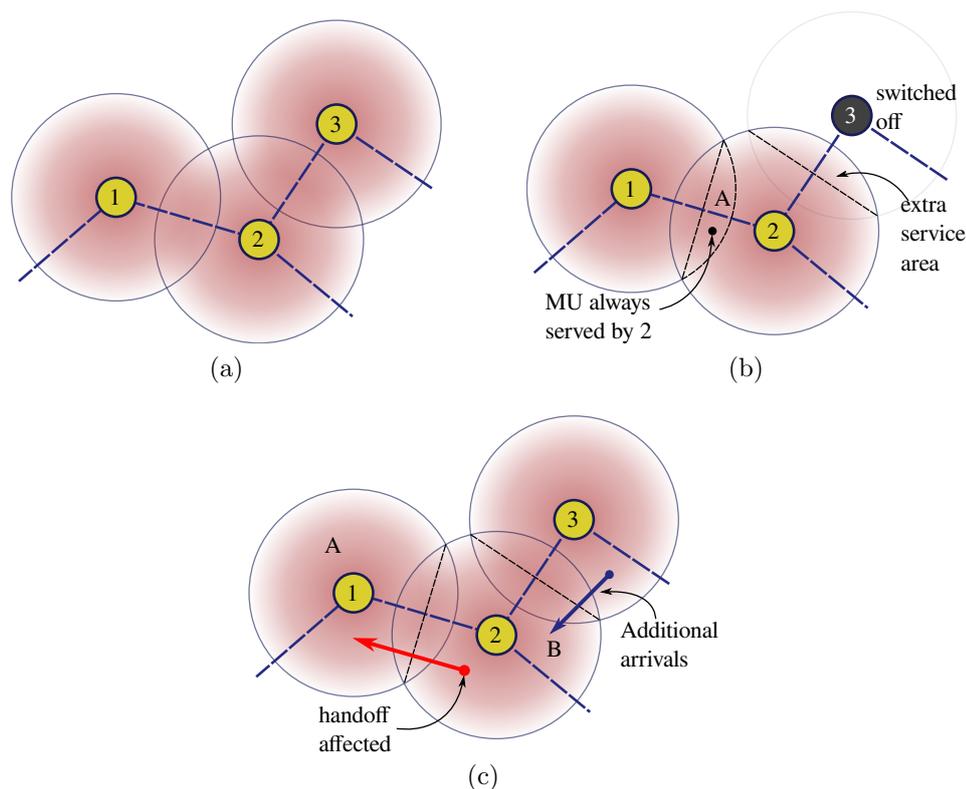


Figure 3.5: Illustration of local behaviour of certain QoS metrics. (a) is a three node network, where nodes 1 and 3 are non neighbours. If node 3 is switched off, as shown in (b) then the call drop ratio at node 1 is unaffected. However, for handover drop ratio this property is not valid (c).

3.4.2 Practical Implementation

We briefly describe the practical implementation of the framework proposed. In [35], a general architecture for green self organizing networks has been proposed, wherein nodes measure and profile traffic conditions and co-operatively operate the network. The framework presented in this chapter can be implemented on this architecture. That is, we envisage a situation as shown in Figure 3.6, where each node constantly ‘learns’ its welfare $f_j(\cdot)$ as a function of states of neighbours. This is done as follows. A node monitors and profiles all the ‘states of nature’ of interest like, traffic load and mobility, similar to what was presented in Chapter 2 for voice traffic. Based on these predictions, the simulator then simulates the *local* network behaviour for different possible states. We thus obtain relevant QoS metrics and

cost for a given local state. Based on these predictions, the nodes simulate the *local* network behaviour for different possible *local* network states. The relevant QoS metrics are estimated and thus welfare functions are obtained. Then, the social welfare is maximized using C-SAP algorithms as presented in this chapter. Moreover, the operation of network in a particular time frame does not influence the others. Thus, we can use a single run of the C-SAP algorithm to solve for the optimal states for all the time frames beforehand – i.e., in an offline manner. During runtime, an online anomaly detection mechanism is used to monitor real time network performance for the operating state. If the network performance is abnormally bad – for example due to an unexpected surge in traffic – then the network reverts back to the default state with all nodes fully powered.

This is then given to the optimization subsystem which implements C-SAP as presented in this chapter. This entire process happens offline. During runtime the, Anomaly Detection subsystem monitors and handles unexpected network behaviour.

Either of the algorithms presented in this chapter can be used for optimization. The SAP algorithm has the advantage of being completely distributed – we can have a poisson clock at each node and upon time-expiry, that particular node updates. For implementing C-SAP algorithms on the other hand, we need to have synchronized nodes – in order to maintain slots, however they converge faster than SAP.

CSAP Scheme-1 for example could be implemented as follows – in every odd slot, the nodes decide among themselves who have to update and in the next slot, those selected nodes will update, while the rest of them hold their states. However, this obviously has additional communication overhead to decide which nodes must update. CSAP Scheme-2 on the other hand gets away without this overhead. It has the advantage that we can pre-compute the best partition (either centralized or distributed) and later on selection of nodes to update will not require any

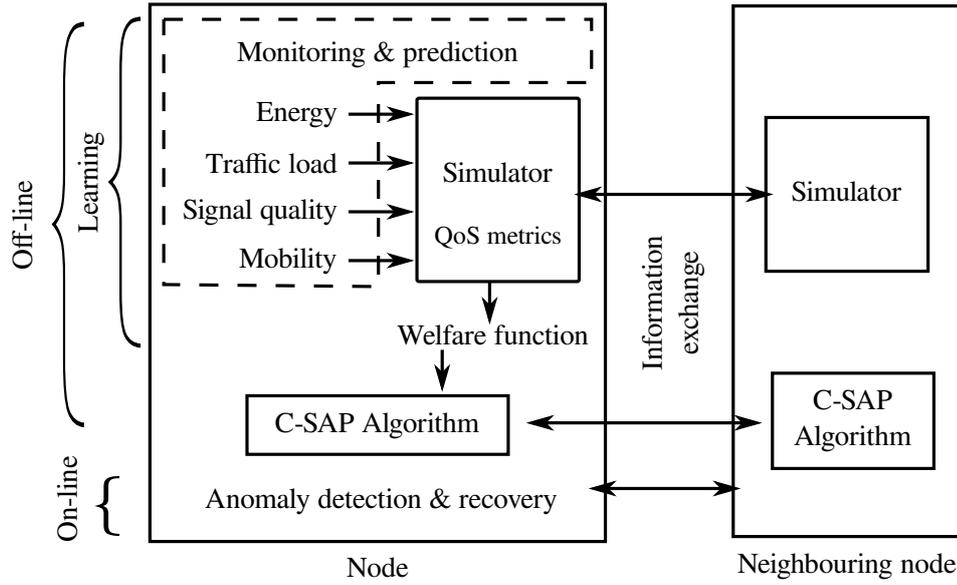


Figure 3.6: Block diagram of the framework developed in this chapter for energy minimization.

communication overhead.

3.5 Rate of Convergence Simulations

In this section, we compare the convergence rates of the SAP, CSAP Scheme-1 and CSAP Scheme-2 algorithms. All of them have the same stationary distribution (with β fixed). Therefore, to compare them, we look at how fast they converge to this distribution through simulations. Figure 3.7 illustrates the setup used in our simulations. For each algorithm, we estimate upper bounds for ϵ -mixing times, by determining coupling times [36] of two instances of that algorithm. We generate two instances of a network simulator that provide welfare function values $f_j(\cdot)$, as needed by the algorithms. The network simulator described in Chapter 2 was used for the simulations. We now explain the procedure used to estimate mixing times.

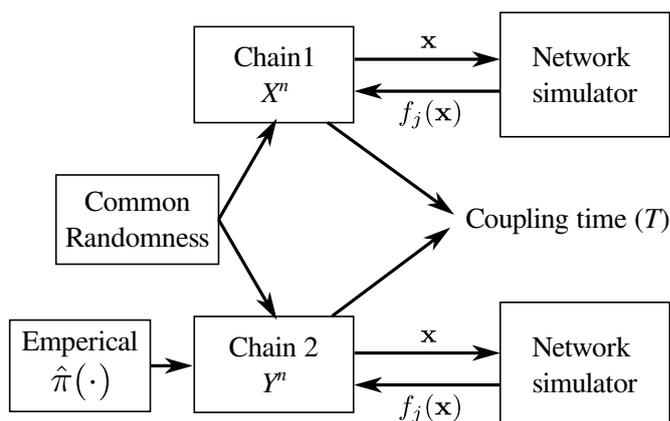


Figure 3.7: Block diagram of the simulation setup. Here X^n and Y^n are Markov chains which denote two instances of the algorithm under consideration. The starting state of Y^n is according to $\hat{\pi}$ which is empirically obtained. The starting state of X^n is – all nodes in state 1. X and Y are coupled by a common randomness. The time when both the chains meet is called coupling time T , a random variable, whose distribution is to be estimated. The network simulators evaluate node welfares $f_j(\cdot)$ for a given state \mathbf{x} .

3.5.1 Estimating Mixing Times

An irreducible, aperiodic discrete time Markov chain $\{X^n, n \geq 0\}$ has a unique stationary distribution π , and regardless of the initial state X^0 , the distribution of X^n (μ_n^X) tends to π as n increases, and the total variation distance $\|\mu_n^X - \pi\|_{tv}$ gives a ‘measure’ of how far from stationarity the chain is at time n . In order to obtain an upper bound for this distance, we generate another Markov chain $\{Y^n, n \geq 0\}$, *coupled* with the chain X^n . The usual coupling technique for Markov chains is to have a common random-number generator that determines transitions of both the chains. Let the initial distribution of $Y^n = \pi$, and let the random variable T denote the time when both the chains meet (i.e., $T = \inf_n \{X^n = Y^n\}$). Then, following [36],

$$\|\mu_n^X - \pi\|_{tv} \leq \Pr(X^n \neq Y^n) = \Pr(T > n).$$

Accordingly, we obtain the distribution of T , through simulations, in order to bound the distance to stationarity. However, the difficulty is that the initial distribution of Y^n must be π , which cannot be computed. Thus, we first estimate

π using the empirical frequency of states as follows:

$$\hat{\pi}(\mathbf{x}) = \frac{1}{W} \sum_{n=0}^W I_{\{X^n=\mathbf{x}\}},$$

where I is the indicator random variable and W is set to 10^5 . After computing the empirical frequencies, before hand, we generate a random starting state for Y^n according to the distribution $\hat{\pi}$. The initial state of chain X^n is set to – all nodes switched-on. We then estimate the probability distribution of T by generating 100 variates of it, which we present in the next subsection.

3.5.2 Results

Figures 3.8a and 3.8b compare the complementary-cumulative distribution of coupling times T of the three algorithms, for the case $\beta = 1$ and $\beta = 4$ respectively (and 10Watt Tx-power). As expected, C-SAP versions outperform the original SAP algorithm, and Scheme-2 does better than Scheme-1 ².

In order to better appreciate the improvements of C-SAP schemes, we must compare the mixing times for a much larger network. However, due to the difficulty in simulating a larger network, we instead use the same set of nodes but reduce the transmit power of nodes to 2.5Watt and repeat the same procedure. Figures 3.9a and 3.9b compare the coupling times of the algorithms for $\beta = 1$ and $\beta = 4$ respectively. Note that the gap between SAP and C-SAP schemes is larger in this scenario.

²For Scheme-2, we require the *best* partition (colouring) of nodes. This is a well known NP hard problem. The partition we used was obtained by framing it as an optimization problem and solving by simulated annealing for 10^6 iterations. The partition thus obtained cannot be claimed as *the* best partition, however it may be so with a large probability.

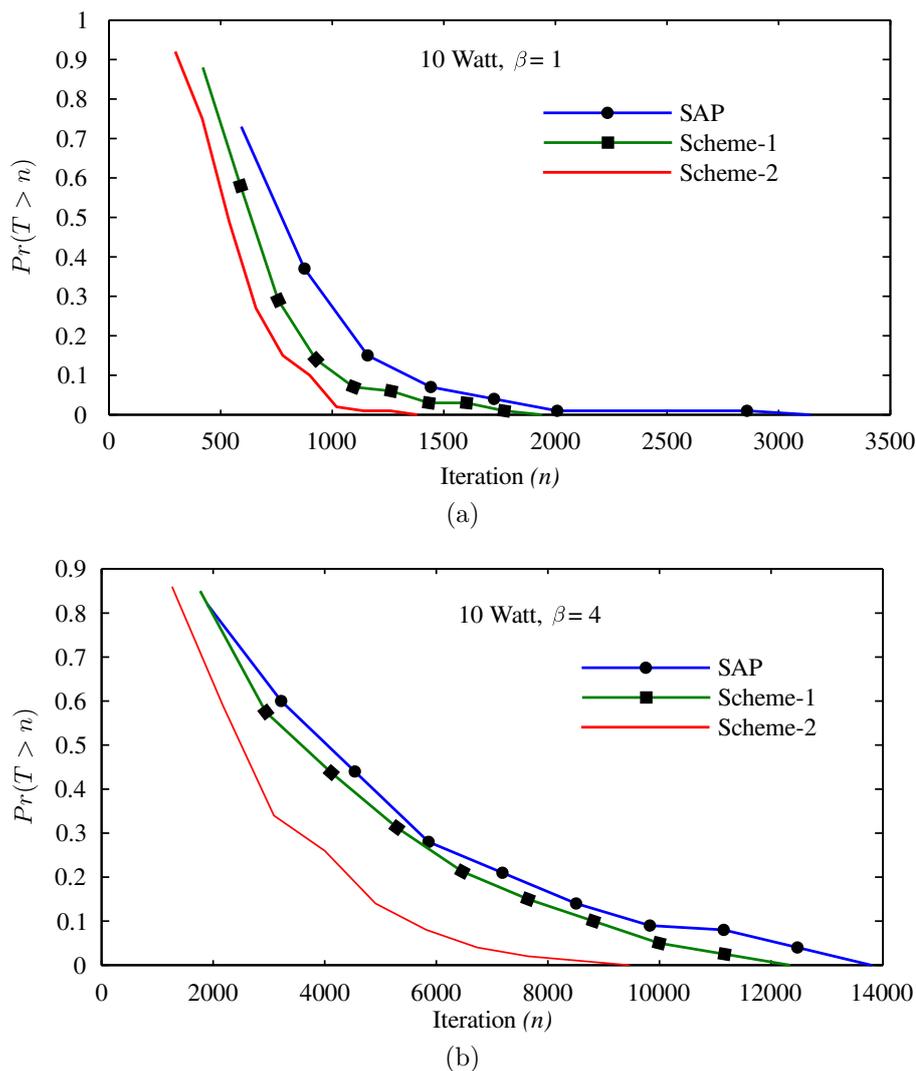


Figure 3.8: Convergence rate results. Plot of complementary cumulative distribution of T , the coupling time. These plots bound the ϵ -mixing times of the algorithms, i.e., $\|\mu_n^X - \pi\|_{tv} \leq \Pr(T > n)$. (a) and (b) correspond to the case $\beta = 1$ and $\beta = 4$ respectively for 10Watt Tx-power.

3.6 Chapter Appendices

3.6.1 Counter example

In this Section, we provide a counter example to demonstrate that the one-hop away C-SAP as proposed in [25], may not pick the global optimal state. Consider the three node network given in Figure 3.10a. Each node can take either of two states, i.e., $S_j = \{0, 1\}$. The individual welfare functions of the nodes are given in Figure 3.10a. We can verify that the welfare functions satisfy the local interaction

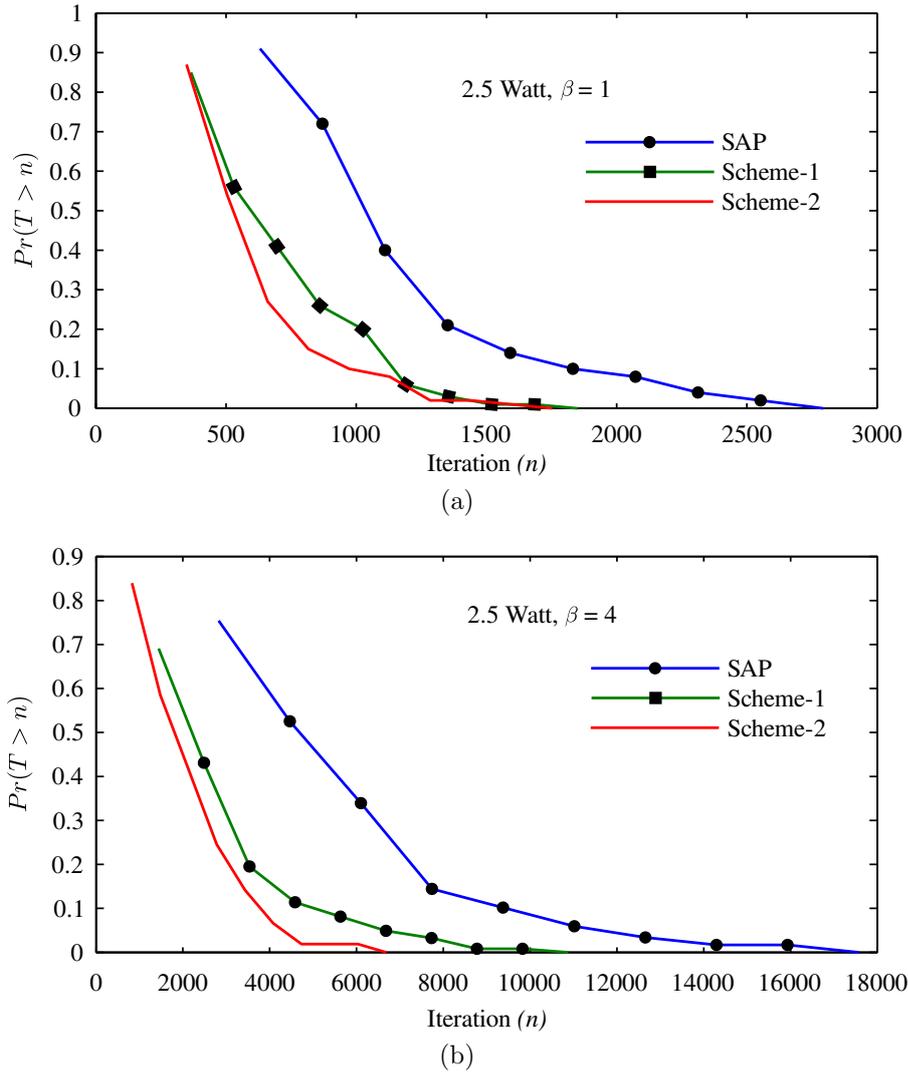


Figure 3.9: Convergence rate results, for 2.5 Watt Tx-power. (a) and (b) are for the case $\beta = 1$ and $\beta = 4$ respectively. The improvement is profound for the case of 2.5Watt Tx-power.

property over the interaction graph shown. The utilities of the nodes are given by (3.2). We use the ‘one-hop away’ selection scheme consisting of $\mathcal{C}' = \{\{1, 3\}, \{2\}\}$ and $\mathbf{p} = [0.5, 0.5]$. That is, with probability a 0.5, either of $\{1, 3\}$ or $\{2\}$ are selected for updating their states as with the C-SAP algorithm. The stationary distribution of this algorithm is denoted by $\boldsymbol{\pi}'_{\beta}$. The actual stationary distribution which we expect the algorithm to have is $\boldsymbol{\pi}_{\beta}$ given by (3.5). Figure 3.10b depicts the total variation distance $\|\boldsymbol{\pi}_{\beta} - \boldsymbol{\pi}'_{\beta}\|_{tv}$ between the two distributions as a function of β . Thus, the distributions are completely different. Moreover, for large β , the

distance is 1. This implies that the π' picks a global optima with a probability that goes to 0 as can be seen from Figure 3.10c. Thus, the optimality of one-hop away C-SAP algorithm is not guaranteed. MATLAB code for this example is given in [37]

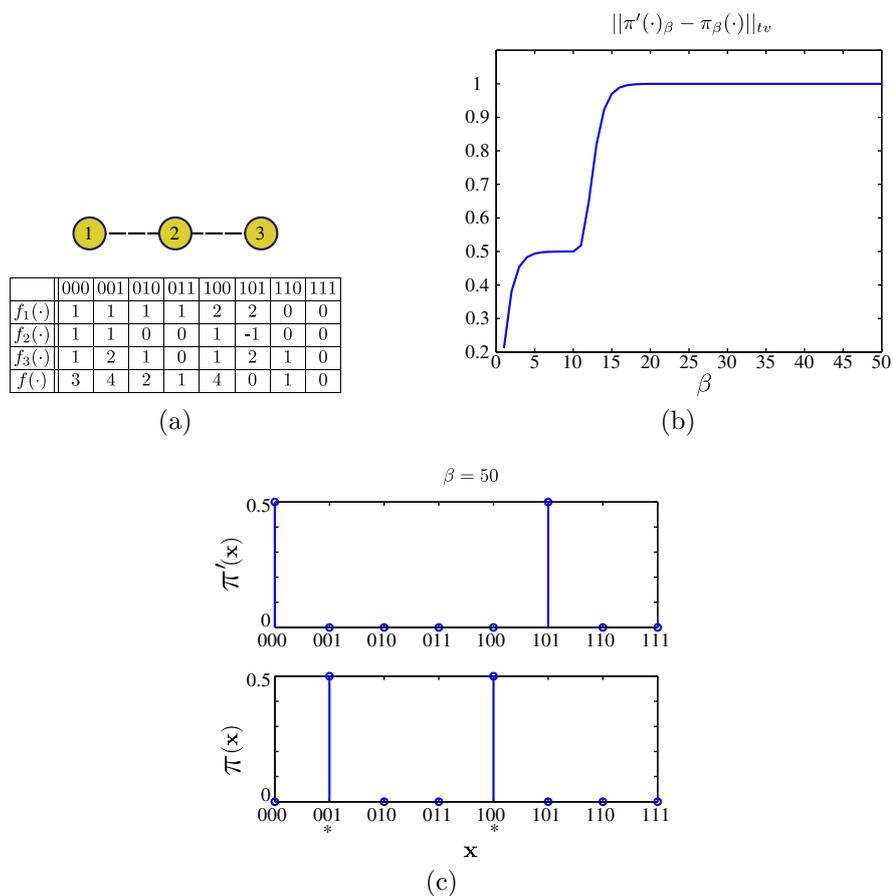


Figure 3.10: (a) is the interaction graph consisting of three nodes. Each node can take either of two actions 0 or 1. The corresponding welfare functions are as given. (b) is the plot of the difference between the stationary distribution so obtained via one-hop C-SAP and the Gibbs distribution. Note that for large β , a global optimal is picked with a probability of zero as depicted in (c).

3.6.2 Equivalence of Wonderful Life and Local Altruistic Utilites

An alternate way to define utilities so as to have proper alignment is to use Wonderful Life Utilities [38, 31],

$$u'_j(\mathbf{x}) = \phi(x_j, \mathbf{x}_{-j}) - \phi(x_j^0, \mathbf{x}_{-j}), \quad (3.7)$$

where $\phi(\cdot)$ is the social welfare function and $x_j^0 \in \mathcal{S}_j$ is some ‘clamped’ action of j . That is, the individual utilites determine how much marginal welfare do they add to the system by being in state x_j as opposed to what it would have been if it were x_j^0 . In the case when we have local interaction structure and when the social welfare is a sum as in our case, it simplifies to the following.

$$u'_j(\mathbf{x}) = \sum_{k \in \mathcal{N}_j} f_j(x_j, \mathbf{x}_{-j}) - f_j(x_j^0, \mathbf{x}_{-j}) \quad (3.8)$$

$$= u_j(x_j, \mathbf{x}_{-j}) - u_j(x_j^0, \mathbf{x}_{-j}), \quad (3.9)$$

where $u_j(\cdot)$ is the LAU of node j . It can be seen that as far as best-response dynamics and SAP are concerned, both the utilities are equivalent. To see this, consider the following. Let \mathbf{x} be the current state and consider the best response of j to the current state for WLU.

$$x = \operatorname{argmax}_{x_j \in \mathcal{S}_j} u'_j(\mathbf{x}) = \operatorname{argmax}_{x_j \in \mathcal{S}_j} u_j(x_j, \mathbf{x}_{-j}) - u_j(x_j^0, \mathbf{x}_{-j}) \quad (3.10)$$

$$= \operatorname{argmax}_{x_j \in \mathcal{S}_j} u_j(x_j, \mathbf{x}_{-j}), \quad (3.11)$$

which is the same as that for LAU. Similarly, for the case of SAP, the probability distribution from which j takes a random sample is the same as for WLU.

$$P(x) = \frac{e^{\beta u'_j(\mathbf{x})}}{\sum_{x' \in \mathcal{S}_j} e^{\beta u'_j(x', \mathbf{x}_{-j})}} \quad (3.12)$$

$$= \frac{e^{\beta u_j(x_j, \mathbf{x}_{-j}) - u_j(x_j^0, \mathbf{x}_{-j})}}{\sum_{x' \in \mathcal{S}_j} e^{\beta u_j(x', \mathbf{x}_{-j}) - u_j(x_j^0, \mathbf{x}_{-j})}} \quad (3.13)$$

$$= \frac{e^{\beta u_j(x_j, \mathbf{x}_{-j})}}{\sum_{x' \in \mathcal{S}_j} e^{\beta u_j(x', \mathbf{x}_{-j})}}, \quad (3.14)$$

which is the same as for WLU.

3.6.3 Proofs of theorems

Proof of Theorem 3: First note that since the welfare function of a node only depends on the state of its neighbors, the utility function of a node depends only the state of its extended neighbors (i.e., the two hop neighbors, denoted \mathcal{EN}_j). That is, by (3.1) and (3.2) we have,

$$u_j(\mathbf{x}) = u_j(\mathbf{x}_k, \mathbf{x}_{-k}) = u_j(\mathbf{x}'_k, \mathbf{x}_{-k}), \text{ if } k \notin \mathcal{EN}_j. \quad (3.15)$$

This proof also follows along the lines of the proof of Theorem 6.1 as given in [33]. We show that $\boldsymbol{\pi}$ satisfies the detailed balance condition, i.e., $\forall \mathbf{x}, \mathbf{y} \in \mathcal{S}$,

$$\boldsymbol{\pi}(\mathbf{x})P_{\mathbf{x}\mathbf{y}} = \boldsymbol{\pi}(\mathbf{y})P_{\mathbf{y}\mathbf{x}} \quad (3.16)$$

We can easily see that the algorithm describes the evolution of a markov chain. The chain is irreducible because every node gets to update with a non-zero probability and it is aperiodic because self-transition can happen with non-zero probability. Therefore, it has a unique stationary distribution $\boldsymbol{\pi}$, which is the solution of (3.16), which we now proceed to show.

When $\mathbf{x} = \mathbf{y}$, then (3.16) holds trivially. Therefore, let $\mathbf{x} \neq \mathbf{y}$.

Let $\mathcal{I}_{\mathbf{x}\mathbf{y}}$ denote the set of indices where \mathbf{x} and \mathbf{y} differ. Then the one-step transition $\mathbf{x} \rightarrow \mathbf{y}$ is feasible iff $\exists \mathcal{C}_k \ni \mathcal{I}_{\mathbf{x}\mathbf{y}} \subseteq \mathcal{C}_k$. Let $\mathcal{A}_{\mathbf{x}\mathbf{y}} = \{k : \mathcal{I}_{\mathbf{x}\mathbf{y}} \subseteq \mathcal{C}_k\}$ denote all such selections \mathcal{C}_k s. Therefore the transition $\mathbf{x} \rightarrow \mathbf{y}$ can happen only via either of the selections in the set $\mathcal{A}_{\mathbf{x}\mathbf{y}}$. Also note that

$$\mathcal{A}_{\mathbf{x}\mathbf{y}} = \mathcal{A}_{\mathbf{y}\mathbf{x}}. \quad (3.17)$$

When $\mathbf{x} \rightarrow \mathbf{y}$ is not feasible, then $\mathcal{A}_{\mathbf{x}\mathbf{y}} = \emptyset$, and $P_{\mathbf{x}\mathbf{y}} = P_{\mathbf{y}\mathbf{x}} = 0$, and (3.16) holds.

Let $\mathbf{x} \rightarrow \mathbf{y}$ be feasible. Then,

$$P_{\mathbf{x}\mathbf{y}} = \sum_{k \in \mathcal{A}_{\mathbf{x}\mathbf{y}}} \mathbf{p}(k) \Pr\{\mathbf{x} \rightarrow \mathbf{y} \mid \mathcal{C}_k \text{ was selected}\}, \quad (3.18)$$

where,

$$\Pr\{\mathbf{x} \rightarrow \mathbf{y} \mid \mathcal{C}_k\} = \prod_{j \in \mathcal{C}_k} \Pr\{j \text{ chooses action } y_j \mid \mathbf{x}\}, \quad (3.19)$$

where,

$$\Pr\{j \text{ chooses } y_j \mid \mathbf{x}\} = \frac{e^{\beta u_j(y_j, \mathbf{x}_{-j})}}{\sum_{z \in \mathcal{S}_j} e^{u_j(z, \mathbf{x}_{-j})}}, \quad (3.20)$$

$$= \frac{e^{\beta u_j(y_j, \mathbf{x}_{-j})}}{\sum_{z \in \mathcal{S}_j} e^{u_j(z, \mathbf{y}_{-j})}}. \quad (3.21)$$

where (3.21) follows from (3.20) due to (3.15). Multiplying (3.19) by $\boldsymbol{\pi}(\mathbf{x})$ and substituting (3.21) in it, we get,

$$\boldsymbol{\pi}(\mathbf{x}) \Pr\{\mathbf{x} \rightarrow \mathbf{y} \mid \mathcal{C}_k\} = \frac{1}{D_1 D_2} e^{\beta\{f(\mathbf{x}) + \sum_{j \in \mathcal{C}_k} u_j(y_j, \mathbf{x}_{-j})\}}, \quad (3.22)$$

$$= \frac{1}{D_1 D_2} e^{\beta\{\sum_{j \in \mathcal{C}_k} u_j(x_j, \mathbf{y}_{-j}) + f(\mathbf{y})\}}, \quad (3.23)$$

$$= \boldsymbol{\pi}(\mathbf{y}) \Pr\{\mathbf{y} \rightarrow \mathbf{x} \mid \mathcal{C}_k\} \quad (3.24)$$

where D_1 is the denominator of (3.5) and D_2 is the product of denominators from

(3.21),

To see how (3.23) follows from (3.22), consider the argument of the exponent in (3.22) and for the sake of clarity of presentation, let $\mathcal{C}_k = \{1, 2, 3\}$, and $\mathbf{x} = (x_1, x_2, x_3, \cdot)$ and $\mathbf{y} = (y_1, y_2, y_3, \cdot)$, where the ‘ \cdot ’ represents the state of the rest of the nodes, then we have,

$$\begin{aligned}
 & f(x_1, x_2, x_3, \cdot) + u_1(y_1, x_2, x_3, \cdot) + u_2(x_1, y_2, x_3, \cdot) + u_3(x_1, x_2, y_3, \cdot) \\
 &= \underline{f(x_1, x_2, x_3, \cdot) + u_1(y_1, x_2, x_3, \cdot)} + u_2(y_1, y_2, x_3, \cdot) + u_3(y_1, y_2, y_3, \cdot) \quad (3.25) \\
 &= u_1(x_1, x_2, x_3, \cdot) + \underline{f(y_1, x_2, x_3, \cdot) + u_2(y_1, y_2, x_3, \cdot)} + u_3(y_1, y_2, y_3, \cdot) \\
 &= u_1(x_1, x_2, x_3, \cdot) + u_2(y_1, x_2, x_3, \cdot) + \underline{f(y_1, y_2, x_3, \cdot) + u_3(y_1, y_2, y_3, \cdot)} \\
 &= u_1(x_1, x_2, x_3, \cdot) + u_2(y_1, x_2, x_3, \cdot) + u_3(y_1, y_2, x_3, \cdot) f(y_1, y_2, y_3, \cdot) \\
 &= u_1(x_1, y_2, y_3, \cdot) + u_2(y_1, x_2, y_3, \cdot) + u_3(y_1, y_2, x_3, \cdot) + f(y_1, y_2, y_3, \cdot) \quad (3.26)
 \end{aligned}$$

where we have used (3.15) to obtain (3.25) and (3.26); and applied (3.3) to the underlined terms. Finally using (3.24), (3.18) and (3.17), the equation (3.16) holds. ■

Proof of Theorem 4: We show this via a coupling argument. Let $\{X^n, n \geq 0\}$ and $\{Y^n, n \geq 0\}$ denote the Markov chains corresponding to SAP and C-SAP respectively, such that $X^0 = Y^0$. We form a coupling (\hat{X}^n, \hat{Y}^n) where \hat{Y}^n is a sampled version of \hat{X}^n and have the same marginals as X^n and Y^n respectively.

Let $\{j^n, n \geq 1\}$ denote an iid sequence of random variables according to $\text{unif}(|\mathcal{N}|)$. Using the same sequence, construct \mathcal{C}_k^1 as given in the selection scheme above. At some point, we would exit from the repeat loop and let L_1 denote the cardinality of the set \mathcal{C}_k^1 so obtained. Note that $L_1 \geq 1$. We use the sequence $\{j^n, 1 \leq n \leq L_1\}$ to update the nodes in the chain \hat{X}^n and the same set of nodes update simultaneously to obtain \hat{Y}^1 . We force the node updates to be the same

for both the chains, therefore $\hat{Y}^1 = \hat{X}^{L_1}$. That is, we use common underlying random number generators for the nodes during updates for both the chains.

We repeat the same process above, starting afresh with another random draw of $\{j^n\}$ to obtain L_2 and $\hat{Y}^2 = \hat{X}^{L_1+L_2}$. Continuing in this fashion, we observe that the chains \hat{X}^n and \hat{Y}^n adhere to their respective marginals and we have,

$$\hat{Y}^n = \hat{X}^{\hat{T}_n}, \quad \forall n \geq 1,$$

where $\hat{T}_n = \sum_{1 \leq l \leq n} L_l$.

Now, for any uncoupled chains X^n and Y^n starting in the same state, the total variation distance between the random variables Y^n and X^{T_n} is,

$$\|Y^n - X^{T_n}\|_{tv} \leq Pr\{\hat{Y}^n \neq \hat{X}^{T_n}\} = 0$$

where the first inequality is a standard result in coupling theory (see [36]). Therefore,

$$Y^n \stackrel{d}{=} X^{T_n}. \quad (3.27)$$

Let m denote some ϵ -mixing time of $\{X^n\}$. The random variable T_m takes values from the finite set $\{m, m+1, \dots\}$ with probabilities $[\alpha_m, \alpha_{m+1}, \dots]$. Let μ_i^X denote the distribution of X^i given the starting state X^0 . Then,

$$X^{T_m} \sim \alpha_m \mu_m^X + \alpha_{m+1} \mu_{m+1}^X + \dots \text{ (finite terms)} \quad (3.28)$$

Since m is the mixing time, we have

$$\|\mu_i^X - \boldsymbol{\pi}\|_{tv} < \epsilon, \quad \forall i \geq m. \quad (3.29)$$

$$\therefore \|\mu_{T_m}^X - \boldsymbol{\pi}\|_{tv} < \epsilon \quad (3.30)$$

$$\Rightarrow \|\mu_m^Y - \boldsymbol{\pi}\|_{tv} < \epsilon. \quad (3.31)$$

where, (3.30) follows from (3.29) and the fact that total-variation is a distance measure and is thus a convex function. Use (3.27) in (3.30) to obtain (3.31). ■

Chapter 4

Conclusions

In work paper, we have formulated a general framework for energy optimal load-dependent operation of cellular networks. We look at both centralized MCMC based and distributed optimization. We use game theoretic tools to break this into a number of smaller problems which are easier to solve. The nodes thus solve these smaller problems and by doing so they inherently solve the original problem. We use SAP dynamics to ensure that the global optima is achieved. We then propose a family algorithms based on SAP, to improve the convergence rate. We considered two members of this family – Scheme-1 and Scheme-2. We proved that Scheme-1 converges faster than the original SAP. We then validate the improvement of rate of convergence of both the schemes via simulations using data from a real-world network. Further work is needed to investigate to convergence rates of various members of the family. Another future direction is to investigate how to handle situations where local interaction property does not hold exactly but its effect diminishes after a few hops, like for the case of handoff-drop-ratio. Other game theoretic models such as auction mechanisms and trial and error based games may also be potentially considered.

Bibliography

- [1] E. Oh, B. Krishnamachari, X. Liu, and Z. Niu, "Toward dynamic energy-efficient operation of cellular network infrastructure," *Communications Magazine, IEEE*, vol. 49, no. 6, pp. 56–61, 2011.
- [2] G. Fettweis and E. Zimmermann, "ICT energy consumption – trends and challenges," in *Proceedings of the 11th International Symposium on Wireless Personal Multimedia Communications*, vol. 2, no. 4, 2008, p. 6.
- [3] Z. Hasan, H. Boostanimehr, and V. Bhargava, "Green cellular networks: A survey, some research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, pp. 524–540, 2011.
- [4] J. Lorincz, T. Garma, and G. Petrovic, "Measurements and modelling of base station power consumption under real traffic loads," *Sensors*, vol. 12, no. 4, p. 4281, 2012. [Online]. Available: <http://www.mdpi.com/1424-8220/12/4/4281>
- [5] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. Imran, D. Sabella, M. Gonzalez, O. Blume, and A. Fehske, "How much energy is needed to run a wireless network?" *IEEE Wireless Communications*, vol. 18, no. 5, pp. 40–49, 2011.
- [6] G. Auer, V. Giannini, C. Desset, and et al., "How much energy is needed to run a wireless network?" *Wireless Communications, IEEE*, vol. 18, no. 5, pp. 40–49, October 2011.
- [7] Z. Niu, "TANGO: Traffic-aware network planning and green operation," *IEEE Wirel. Commun.*, vol. 18, no. 5, pp. 25–29, 2011.
- [8] Z. Hasan, H. Boostanimehr, and V. Bhargava, "Green cellular networks: A survey, some research issues and challenges," *Communications Surveys Tutorials, IEEE*, vol. 13, no. 4, pp. 524–540, Apr. 2011.

- [9] Z. Niu, Y. Wu, J. Gong, and Z. Yang, "Cell zooming for cost-efficient green cellular networks," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 74–79, 2010.
- [10] W. T. Wong, Y. J. Yu, and A. C. Pang, "Decentralized Energy-efficient BS operation for Green cellular networks," in *GLOBECOM - IEEE Glob. Telecommun. Conf.*, 2012, pp. 2636–2641.
- [11] Y. Yang, L. Chen, W. Dong, and W. Wang, "Active Base Station Set Optimization for Minimal Energy Consumption in Green Cellular Networks," *IEEE Trans. Veh. Technol.*, vol. 9545, no. c, pp. 1–1, 2014. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6996018>
- [12] E. Oh, B. Krishnamachari, X. Liu, and Z. Niu, "Toward dynamic energy-efficient operation of cellular network infrastructure," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 56–61, 2011.
- [13] E. Oh and B. Krishnamachari, "Energy savings through dynamic base station switching in cellular wireless access networks," in *GLOBECOM - IEEE Glob. Telecommun. Conf.*, Dec. 2010, pp. 1–5.
- [14] M. Ajmone Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "Optimal Energy Savings in Cellular Access Networks," *2009 IEEE Int. Conf. Commun. Work.*, pp. 1–5, 2009. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5208045>
- [15] E. Pollakis, R. L. G. Cavalcante, and S. Stanczak, "Base station selection for energy efficient network operation with the majorization-minimization algorithm," in *International Workshop on Signal Processing Advances in Wireless Communications*, 2012, pp. 219–223.
- [16] P. González-Brevis, J. Gondzio, Y. Fan, H. V. Poor, J. Thompson, I. Krikidis, and P.-J. Chung, "Base station location optimization for minimal energy consumption in wireless networks," in *Vehicular Technology Conference*, vol. 73. IEEE, 2011, pp. 1–5.
- [17] C. Peng, S.-B. Lee, S. Lu, H. Luo, and H. Li, "Traffic-driven power saving in operational 3G cellular networks," in *International Conference on Mobile Computing and Networking*, 2011, pp. 121–132.
- [18] S. Zhou, J. Gong, Z. Yang, Z. Niu, P. Yang, and D. Corporation, "Green mobile access network with dynamic base station energy saving," *ACM MobiCom*, vol. 9, no. 262, pp. 10–12, 2009.

- [19] K. Son, H. Kim, Y. Yi, and B. Krishnamachari, "Base station operation and user association mechanisms for energy-delay tradeoffs in green cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1525–1536, 2011.
- [20] Y. Yang, L. Chen, and W. Wang, "A novel energy saving scheme based on base stations dynamic configuration in green cellular networks," *IEEE Veh. Technol. Conf.*, vol. 1, 2013.
- [21] L. a. Suarez, L. Nuaymi, and J. M. Bonnin, "Energy performance of a distributed BS based green cell breathing algorithm," *Proc. Int. Symp. Wirel. Commun. Syst.*, pp. 341–345, 2012.
- [22] M. F. Hossain, K. S. Munasinghe, and A. Jamalipour, "An eco inspired energy efficient access network architecture for next generation cellular systems," pp. 769–784, 2011.
- [23] H. Sharma, V. Vaid, P. Chaporkar, and G. S. Kasbekar, "QoS aware optimal base station ON/OFF policy and frequency planning," Indian Institute of Technology Bombay, Tech. Rep., 2015. [Online]. Available: tictet.iitb.ac.in/documents/powerfreq.pdf
- [24] J. Zheng, Y. Cai, X. Chen, R. Li, and H. Zhang, "A Game-Theoretic Approach for Optimal Base Station Sleeping in Green Cellular Networks," pp. 1–6, 2014.
- [25] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y.-D. Yao, "Opportunistic spectrum access in cognitive radio networks: Global optimization using local interaction games," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 2, pp. 180–194, 2012.
- [26] C. Singh and C. S. Chen, "Distributed downlink resource allocation in cellular networks through spatial adaptive play," in *International Teletraffic Congress (ITC)*, 2013, pp. 1–9.
- [27] J. Wu, Y. Zhang, M. Zukerman, and E. K.-n. Yung, "Energy-Efficient Base Stations Sleep Mode Techniques in Green Cellular Networks : A Survey," no. 9380044, pp. 1–25.
- [28] R. Cavalcante, S. Stanczak, A. Eisenblätter, T. Kürner, D. Montvila, R. Nowak, and S. Stefanski, "Voice and traffic prediction in GSM and UMTS cells with real network measurements," European

- Cooperation in Science and Technology, Tech. Rep., 2013. [Online]. Available: http://www.ic1004.org/uploads/Meetings/VII%20MC%20-%20Ilmenau/Workshop/Papers/Cavalcante_1.pdf
- [29] K. Kumar, A. Gupta, R. Shah, A. Karandikar, and P. Chaporkar, “On analyzing Indian cellular traffic for energy efficient network operation,” in *IEEE Nation Conference on Communication*, 2015.
- [30] P. Bremaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and Queues*. Springer, 1999.
- [31] J. R. Marden, G. Arslan, and J. S. Shamma, “Cooperative control and potential games,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1393–1407, 2009.
- [32] G. Arslan, J. R. Marden, and J. S. Shamma, “Autonomous vehicle-target assignment: A game-theoretical formulation,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 584–596, 2007.
- [33] H. Young, *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions*. Princeton University Press, 2001.
- [34] J. Zheng, Y. Cai, X. Chen, R. Li, and H. Zhang, “A game-theoretic approach for optimal base station sleeping in green cellular networks,” in *International Conference on Wireless Communications and Signal Processing*, 2014, pp. 1–6.
- [35] O. Blume, H. Eckhardt, S. Klein, E. Kuehn, and W. M. Wajda, “Energy savings in mobile networks based on adaptation to traffic statistics,” *Bell Labs Technical Journal*, vol. 15, no. 2, pp. 77–94, 2010.
- [36] D. Levin, Y. Peres, and E. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Soc.
- [37] [Online]. Available: ticet.iitb.ac.in/documents/example.zip
- [38] K. Tumer and D. Wolpert, “A survey of collectives,” in *Collectives and the design of complex systems*. Springer, 2004, pp. 1–42.

Publications

Part of this work was published in National Conference on Communication 2015, titled “On Analyzing Indian Cellular Traffic for Energy Efficient Network Operation,” co-authored by Archit Gupta, Rushabh Shah, Prof. Abhay Karandikar and Prof. Prasanna Chaporkar.