Configuring Dynamic Heterogeneous Wireless Communications Networks using a Customised Genetic Algorithm

David Lynch[1], Michael Fenton[1], Stepan Kucera[2],
Holger Claussen[2], and Michael O'Neill[1]

[1]Natural Computing Research and Applications Group, UCD
[2]Bell Laboratories, Nokia-Dublin
Email: david.lynch.1@ucdconnect.ie, michaelfenton1@gmail.com,
{stepan.kucera, holger.claussen}@nokia-bell-labs.com, m.oneill@ucd.ie

Abstract. Wireless traffic is surging due to the prevalence of smart devices, rising demand for multimedia content and the advent of the "Internet of Things". Network operators are deploying Small Cells alongside existing Macro Cells in order to satisfy demand during this era of exponential growth. Such Heterogeneous Networks (HetNets) are highly spectrally efficient because both cell tiers transmit using the same scarce and expensive bandwidth. However, load balancing and cross-tier interference issues constrain cell-edge rates in co-channel operation. Capacity can be increased by intelligently configuring Small Cell powers and biases, and the muting cycles of Macro Cells. This paper presents a customised Genetic Algorithm (GA) for reconfiguring HetNets. The GA converges within minutes so tailored settings can be pushed to cells in real time. The proposed GA lifts cell-edge (2.5^{th} percentile) rates by 32% over a non-adaptive baseline that is used in practice. HetNets are highly dynamic environments. However, customers tend to cluster in hotspots which arise at predictable locations over the course of a typical day. An explicit memory of previously evolved solutions is maintained and used to seed fresh runs. System level simulations show that the 2.5^{th} percentile rates are boosted to 36% over baseline when prior knowledge is utilised.

1 Introduction

Conventional wireless communications networks are served by high-powered and long-range antennas called Macro Cells (MCs). However, MC deployments are struggling to satisfy demand during an era of exponentially rising mobile traffic [1]. Small Cells (SCs) have been proposed as a scalable and cost-effective technology for boosting the capacity of MC deployments. SCs are lower-powered antennas that can be deployed in traffic hotspots to supplement the existing MC tier. When operating jointly, SCs and MCs constitute a so-called Heterogeneous Network or 'HetNet'. Network operators like Verizon Communications Inc. and AT&T Inc. are aggressively densifying with SCs. Densification is economically sensible because both cell tiers transmit across the same bandwidth, which is scarce and sells for billions of Euro at auction.

The term ‘User Equipment’ (UE) refers to any device (e.g. a smartphone) that attaches to a cell. Now, UEs at cell edges experience low channel quality (and hence low downlink rates) due to severe cross-tier interference from MCs and other nearby SCs. Low rates may result in packet losses and poor customer satisfaction. UEs that are closer to cell centres can liberate resources without noticing a degraded quality of service. This paper proposes a customised Genetic Algorithm (GA) for configuring the cells of a HetNet so that resources are fairly distributed among all UEs.

Wireless networks are highly dynamic environments, since channel conditions fluctuate dramatically on a millisecond timescale [23], and demand from a single device varies constantly [13]. However, it is not feasible to reconfigure cells more frequently than every ten minutes or so. Insensitive manipulation of the network configuration will cause unwanted ping-pong handovers resulting in latency or dropped calls. Nonetheless, gradual changes in the traffic pattern, such as hotspots moving around or dissipating, can be tracked by an adaptive evolutionary algorithm. This paper develops a customised GA that converges on approximately the same timescale as these changes occur.

Fresh runs are seeded with individuals from an explicit memory that stores previously evolved solutions. It is hypothesised that the GA will converge to better solutions more quickly by incorporating knowledge from prior scenarios [17,3,8]. Hence, the proposed approach tailors the network configuration to the current transient conditions, but by also tapping longer term trends.

The paper is organised as follows. Section 2 introduces two paradigms for managing HetNets before formalising the optimisation problem. Previous work is reviewed in Section 3. Section 4 describes a customised GA that is adapted for this instance of a dynamic environment. The simulation environment and experiments are described in Section 5. A discussion of the results follows in Section 6. The paper concludes with directions for future work in Section 7.

2 Problem Definition

Two factors limit the capacity of channel sharing HetNets. Firstly, the SC tier is typically underutilised because low-powered SCs struggle to offload UEs from stronger MCs. Secondly, poor channel conditions manifest at the edges of SCs due to severe cross-tier interference. A standards body called the 3$^{\text{rd}}$ Generation Partnership Project (3GPP)[1] has proposed mechanisms for load balancing and interference mitigation in HetNets. This section specifies the optimisation problems that arise in HetNets which implement the 3GPP standard.

2.1 Load Balancing

Figure 1.1 depicts a single SC $s' \in \mathcal{S}$ embedded within a MC sector $m' \in \mathcal{M}$, where $\mathcal{S}$ and $\mathcal{M}$ denote the sets of SCs and MCs respectively. Twelve UEs attach

[1] 3GPP (December 2010, http://www.3gpp.org/)

to m' so that $|\mathcal{A}_{m'}| = 12$, where $\mathcal{A}_c$ is the set of UEs attached to cell $c \in \mathcal{S} \cup \mathcal{M}$. However, only $|\mathcal{A}_{s'}| = 3$ UEs attach to s', so m' is congested relative to s'. High congestion on any cell c is undesirable because the limited bandwidth must be shared among $\mathcal{A}_c$. Range expansion can be employed in this context to offload more UEs onto s'.

UE u attaches to, and hence receives data from, cell $k \in \mathcal{S} \cup \mathcal{M}$:

$$k := \arg\max_c (Signal_{u,c} + \beta_c) = \arg\max_c (g_{u,c} + P_c + \beta_c), \; \forall c \in \mathcal{M} \cup \mathcal{S}, \qquad (1)$$

where, the signal strength $Signal_{u,c}$ experienced by u from c is given by adding the gain $g_{u,c}$ [dB] from c to u, to the transmitting power P_c [dBm] of c. The variable β_c is the Cell Selection Bias used by c, where $\beta_s \geq 0$ [dB], $\forall s \in \mathcal{S}$ and $\beta_m := 0$ [dB], $\forall m \in \mathcal{M}$ (since MCs are rarely underutilised). Hence, the underutilised SC s' of Figure 1.1 could absorb UEs from the adjacent hotspot by increasing $P_{s'}$, or artificially, by broadcasting a positive Cell Selection Bias $\beta_{s'}$. Figure 1.2 illustrates the expanded region that forms at the edge of s' when $\beta_{s'} > 0$. UEs in the expanded region attach to s' despite receiving a stronger signal from m'. Range expansion reduces the load imbalance from $|\mathcal{A}_{m'}| - |\mathcal{A}_{s'}| = (12 - 3) = 9$ in Figure 1.1, to just $|\mathcal{A}_{s'}| - |\mathcal{A}_{m'}| = (9 - 6) = 3$ in Figure 1.2.

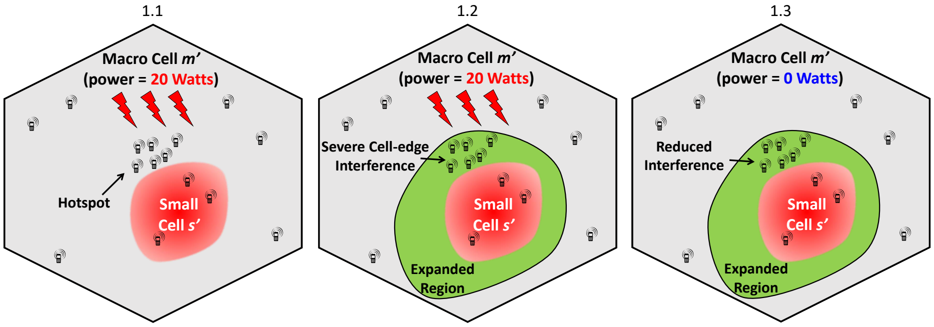

Fig. 1: Toy network containing one MC with an embedded SC.

2.2 Time Domain Interference Mitigation

Channel conditions can be very poor in the expanded regions of SCs due to severe interference from MCs (which transmit using the same bandwidth). An enhanced Inter-cell Interference Coordination (eICIC) paradigm has been proposed to mitigate cell-edge interference [15]. HetNets implementing eICIC mute MCs periodically so that SCs can transmit with dramatically reduced interference. On the other hand, $u \in \mathcal{A}_m$ receive no data when m mutes. For instance, in Figure 1, m' should be muted regularly enough to protect $u \in \mathcal{A}_{s'}$, but m' should transmit sufficiently often to provide an acceptable quality of service for $u \in \mathcal{A}_{m'}$.

Both MCs and SCs send individual packets during 1 millisecond intervals called subframes. There are 40 subframes in a 'frame' $\mathcal{F}_t$, where $t \in \mathbb{N}$ is the timestep. At the end of each frame, UEs report measurements of the channel conditions that they experienced in the preceding 40 milliseconds. When a MC mutes for some $f \in \mathcal{F}_t$ we say that it executes an 'Almost Blank Subframe' (ABS) during f. Table 1 displays the seven ABS patterns that a MC can use. The feasible ABS patterns for a MC m are specified by the ABS ratio for m, or $ABS_{r,m}$. For instance, if $ABS_{r,m} = 4$ then m will mute in 20 subframes out of 40. Furthermore, $ABS_{r,m} = 4$ implies that m will adopt the muting pattern in row 4 of Table 1, where '0' in position f indicates an ABS (muted) subframe and '1' implies that the MC transmits during f. Lower ABS ratios suggest more aggressive muting.

ABS Ratio	Subframe (f)				
	1...				...40
$ABS_{r,m} = 1$	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1
$ABS_{r,m} = 2$	0 0 0 0 0 0 1 1	0 0 0 0 0 0 1 1	0 0 0 0 0 0 1 1	0 0 0 0 0 0 1 1	0 0 0 0 0 0 1 1
$ABS_{r,m} = 3$	0 0 0 0 0 1 1 1	0 0 0 0 0 1 1 1	0 0 0 0 0 1 1 1	0 0 0 0 0 1 1 1	0 0 0 0 0 1 1 1
$ABS_{r,m} = 4$	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1
$ABS_{r,m} = 5$	0 0 0 1 1 1 1 1	0 0 0 1 1 1 1 1	0 0 0 1 1 1 1 1	0 0 0 1 1 1 1 1	0 0 0 1 1 1 1 1
$ABS_{r,m} = 6$	0 0 1 1 1 1 1 1	0 0 1 1 1 1 1 1	0 0 1 1 1 1 1 1	0 0 1 1 1 1 1 1	0 0 1 1 1 1 1 1
$ABS_{r,m} = 7$	0 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1

Table 1: Possible ABS Patterns for $\mathcal{F}_t$ (time increases left to right).

In summary, cell range expansion enables efficient offloading from the MC tier onto SCs. Prohibitive interference at SC edges is then mitigated by periodically muting MCs in the time domain. The notions of range expansion and eICIC are formalised in [15].

2.3 Multi-layer Optimisation of HetNets

Let $\mathcal{C}$ denote the 'configuration' of a HetNet with $|\mathcal{M}|$ MCs and $|\mathcal{S}|$ SCs:

$$\mathcal{C} \leftarrow \left[ABS_{r,1}, \dots, ABS_{r,|\mathcal{M}|}, P_1, \dots, P_{|\mathcal{S}|}, \beta_1, \dots, \beta_{|\mathcal{S}|}\right], \tag{2}$$

where, $ABS_{r,m}$ is an integer from $[1, \dots, 7]$, P_s takes a real value in the interval $[23.0 \text{ [dBm]}, 35.0 \text{ [dBm]}]$, and β_s is a real value in $[0.0 \text{ [dB]}, 15.0 \text{ [dB]}]$. Recall that MCs do not use bias (so $\beta_m := 0.0$ [dB]) and $P_m := 43.3$ [dBm] is invariant $\forall m \in \mathcal{M}$ during non-ABS subframes, but $P_m = 0.0$ [dBm] when m mutes. This paper presents a customised GA for optimising $\mathcal{C}$.

A solution $\mathcal{C}$ is evaluated with a fairness-based utility of downlink rates. Consider UE u attached to cell $k \in \mathcal{S} \cup \mathcal{M}$. Denote the channel quality experienced by u in subframe f of frame $\mathcal{F}_t$ by $Q_{u,f} = \log_2(1 + \mathit{SINR}_{u,f})$, where $\mathit{SINR}_{u,f}$ is the signal to interference and noise ratio experienced by u during f:

$$\mathit{SINR}_{u,f} = \frac{\mathit{Signal}_{u,k,f} \text{ [Watts]}}{\sum\limits_{c \in \mathcal{S} \cup \mathcal{M} \setminus k} \mathit{Signal}_{u,c,f} \text{ [Watts]} + \mathit{Noise}(= 4 \times 10^{-16}) \text{ [Watts]}}. \tag{3}$$

Shannon's formula [20] then gives the (instantaneous) downlink rate for u in f,

$$R_{u,f} = \frac{20\,\mathrm{MHz}}{|\mathcal{A}_c| - \nu_{c,f}} \times Q_{u,f}, \tag{4}$$

where, 20 MHz is the fixed bandwidth, and $|\mathcal{A}_c| - \nu_{c,f}$ is the total number of UEs that receive data from u's serving cell in f. The term $\nu_{c,f}$ is the number of UEs attached to c that cannot be scheduled in f because $Q_{u,f}$ is too low ($\nu_{c,f} = |\{u|u \in \mathcal{A}_c, SINR_{u,f} \leq -5.0 \text{ [dB]}\}|$). Intuitively, $R_{u,f}$ is high if channel conditions are favourable for u in f and if u shares the bandwidth with few UEs. The arithmetic mean of $R_{u,f}$ over $\mathcal{F}_t$ yields the 'downlink rate' ($\overline{R}_u$) for u over that frame. Notice that $\overline{R}_u$ depends on $\mathcal{C}$ through $\mathcal{A}_c$, $\nu_{c,f}$ and $Q_{u,f}$, each of which depend on the state of all cells in the network (wireless signals do not recognise cell boundaries).

Finally, the fitness of $\mathcal{C}$ over frame $\mathcal{F}_t$ is adapted from the industry standard sum-log-rates metric for evaluating HetNet control algorithms [7]:

$$\mathit{fitness}_{\mathcal{C}}^{\mathcal{F}_t} \leftarrow \sum_{u \in \mathcal{U}} \left(\log_e \overline{R}_u\right), \tag{5}$$

where, $\mathcal{U}$ is the set of users receiving data from the HetNet during $\mathcal{F}_t$. The logarithm is sensitive to changes in the lowest values of $\overline{R}_u$. Thus, cell configurations $\mathcal{C}$ that increase cell-edge rates will receive high fitness.

Our goal is to discover synergistic settings at the SC power and bias, and MC ABS layers of HetNets with a customised GA. This problem is non-trivial due to inter-layer and intra-layer coupling. For instance, if the SC of Figure 1 uses a large bias then the MC should mute often to protect interfered UEs in its expanded region. Furthermore, the GA must converge quickly in order to keep pace with rapidly changing traffic patterns.

3 Previous Work

3.1 Heterogeneous Network Optimisation

Techniques for eICIC are described at the conceptual level from release 10 of 3GPP, but no algorithms are specified. As such, network operators have the freedom to interpret and implement these concepts as they see fit. The bulk of the literature on HetNet optimisation focuses on improving the effectiveness of eICIC. Self Organising Network (SON) algorithms have been proposed to modulate small cell powers and biases [9,10] and eICIC parameters such as MC ABS ratios [11]. SONs have also been designed to increase energy efficiency [22], minimise inter-cell interference [7,16,18], and maximise fairness-based utilities [21]. The authors in [2] presented a detailed survey of SONs in the context of LTE.

Our goal is to maximise fairness. Auxiliary objectives, such as improving energy efficiency, are ignored because customer satisfaction takes priority in this highly competitive industry. Deb et al. (2014) tackled the single-objective problem using non-linear programming [7]. The authors proved that optimising $\mathcal{C}$ is NP-hard, even in a network with one MC and multiple SCs. Fairness was dramatically improved by optimising β_s, $\forall s \in \mathcal{S}$, and $ABS_{r,m}$, $\forall m \in \mathcal{M}$. However, their algorithm did not modulate SC powers and it required inputs that may be difficult to obtain in real networks [14]. Our framework requires only channel gain data that are reported by UEs.

López-Pérez and Claussen (2013) proposed a method for tuning $ABS_{r,m}$ so that performance targets can be specified by the operator $\forall u \in \mathcal{A}_m$. In their approach, the bias and ABS layers were configured independently. However, HetNet layers are coupled such that, for instance, optimal MC ABS ratios depend on the size of SC expanded regions (i.e. SC biases). The framework developed in this paper jointly optimises the power, bias and ABS layers of a HetNet. Furthermore, the proposed algorithm is easily implemented on a centralised server and it requires inputs that are readily available in real networks.

3.2 Genetic Algorithms

GAs have enjoyed recent success in challenging real-world applications. For instance, Deb and Myburgh (2016) developed a customised GA to solve a billion variable problem [6]. Their findings are ground-breaking for two reasons:

1. firstly, the billion variable barrier is breached for the first time on a real-world constrained optimisation problem, and,
2. secondly, the proposed approach outperformed two commercial optimisers (`glpk` and `CPLEX`) with respect to solution quality and convergence speed. Indeed, the GA discovers near optimal solutions in a fraction of the time.

The authors concur with the critiques levelled by Deb and Myburgh (2016) regarding the use of customised heuristics. Their paper argues that the components of evolutionary heuristics should be viewed as a toolbox. Researchers should carefully select the appropriate tools for the problem at hand and avoid blindly applying canonical settings and operators that were developed for unrelated applications or toy problems. The GA in [6] was tailored for the real world problem at hand. It employed customised initialisation, re-combination, and mutation operators, which guaranteed constraint satisfiability.

An extensive literature describes the applications of GAs in dynamic environments [4,8,17]. Of the approaches that incorporate memory, Branke [4] identifies two main paradigms based on the notions of explicit and implicit memory. The former stores genetic material from previously fit individuals in a memory cache. The latter attempts to capture prior knowledge using degenerate genetic material. GAs that rely on implicit memory struggle if the environment alternates between many states [4]. On the other hand, explicit memory is beneficial when similar conditions recur frequently, as in wireless networks. The next section presents a customised GA that leverages explicit memory (see also [3,12,19]).

4 Customised Genetic Algorithm

4.1 Encoding and Mapping

Recall Equation 2 which expressed the configuration $\mathcal{C}$ of a HetNet as a mixed type array storing the MC ABS ratios, SC powers and SC biases. Let $\mathcal{I}$ (for $\mathcal{I}$ndividual) encode the network configuration $\mathcal{C}$. The first $|\mathcal{M}|$ elements of $\mathcal{I}$ are integers which encode $ABS_{r,m}$ for $m = 1, \ldots, |\mathcal{M}|$, the next $|\mathcal{S}|$ elements are real values encoding P_s for $s = 1, \ldots, |\mathcal{S}|$, and the remaining real-valued elements encode β_s for $s = 1, \ldots, |\mathcal{S}|$. $\mathcal{I}$ must first be mapped to $\mathcal{C}$ before it can be evaluated.

Consider again the toy network of Figure 1 with a single MC m' containing the embedded SC s'. Let $\mathcal{I} = [11, -0.2, 4.5]$ encode the network configuration $\mathcal{C} = [ABS_{r,m'}, P_{s'}, \beta_{s'}]$. $\mathcal{I}$ is mapped to $\mathcal{C}$ as follows. The ABS ratio for m' is given by,

$$\mathcal{C}[0] = ABS_{r,m'} = ABS\_ratios\,[\mathcal{I}[0]\ \%\,12]\,,$$

where, $ABS\_ratios \coloneqq [1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2]$ and % is the modulo operator. Since $\mathcal{I}[0] = 11$ it follows that $\mathcal{C}[0] = ABS\_ratios\,[11\%12] = ABS\_ratios\,[11] = 2$. This implementation preserves locality in the genotype to phenotype mapping and respects the constraints on $ABS_{r,m'}$. Furthermore, $\mathcal{I}[0]$ can take any integer value and still satisfy the constraints. There is a bias against aggressive MC muting ($ABS_{r,m'} = 1$) and high activity ($ABS_{r,m'} = 7$).

$\mathcal{I}[1]$ encodes the power of SC s', which is given by,

$$\mathcal{C}[1] = P_{s'} = 23.0 + Sigmoid(\mathcal{I}[1]) \times (35.0 - 23.0),$$

where, the sigmoid function $Sigmoid(x) = 1/1 + e^{-x}$ returns a value between 0 and 1, and where the minimum and maximum powers for s' are, respectively, 23.0 dBm and 35.0 dBm. The sigmoid facilitates fine grained exploitative mutations, but occasionally it allows for large explorative steps. Similarly,

$$\mathcal{C}[2] = \beta_{s'} = 0.0 + Sigmoid(\mathcal{I}[2]) \times (15.0 - 0.0),$$

where, the minimum and maximum biases for s' are, respectively, 0.0 dB and 15.0 dB. Notice that $\mathcal{I}[1]$ and $\mathcal{I}[2]$ can adopt any real value without violating the constraints.

In summary, the individual $\mathcal{I} = [11, -0.2, 4.5]$ is mapped to the network configuration $\mathcal{C} = [ABS_{r,m'}, P_{s'}, \beta_{s'}] \approx [2, 28.40, 14.84]$.

4.2 Search Operators

Each pair of selected parents are recombined using uniform crossover to yield two children. Elements in the parent strings are swapped with a probability $p_{cross} = 0.5$ per locus, so that approximately half of the genetic material is interchanged between parents.

Each child undergoes Gaussian mutation applied with a probability $p_{mut} = 0.05$ per element. Consider the individual $\mathcal{I}$ from Section 4.1. Assume for the sake of exposition that each element is mutated. Then,

$$\begin{aligned}
\mathcal{I}[0] &\leftarrow \mathcal{I}[0] + \mathfrak{R} \times \left(\left| round\left(\mathcal{N}\left(0, \sigma^2_{ABS}\right)\right) \right| + 1 \right), \\
\mathcal{I}[1] &\leftarrow \mathcal{I}[1] + \mathcal{N}\left(0, \sigma^2_{P,\beta}\right), \\
\mathcal{I}[2] &\leftarrow \mathcal{I}[2] + \mathcal{N}\left(0, \sigma^2_{P,\beta}\right),
\end{aligned}$$

where, $\mathfrak{R}$ is a random variable drawn from the set $\{-1, +1\}$, $|x| = \sqrt{x^2}$, $round(x)$ maps x to the nearest integer, and $\mathcal{N}$ is the normal distribution parametrised with zero mean and standard deviation σ_{ABS} or $\sigma_{P,\beta}$. The parameters σ_{ABS} and $\sigma_{P,\beta}$ control the severity of mutations applied to elements encoding ABS ratios, and respectively, power and bias settings. A parameter sweep on hold out data suggested that $\sigma_{ABS} = 1.0$ and $\sigma_{P,\beta} = 0.5$ gives good performance. The search parameters were not tuned exhaustively in this proof of concept study.

4.3 Fitness assignment

Wireless signals experience some path loss gain as they propagate from a transmitting cell to a UE. Individuals are evaluated using simulated reports of these channel gains, the realistic analogues of which would be available in a physical deployment. Let $g^{\mathcal{F}_t}_{u,c}$ denote the channel gain (in [dB]) experienced by UE u from cell c during frame $\mathcal{F}_t$. These data are collected from all $|\mathcal{U}|$ UEs in the network and arranged in an $(|\mathcal{M}| + |\mathcal{S}|) \times |\mathcal{U}|$ matrix $G^{\mathcal{F}_t}$. Realistic values of $g^{\mathcal{F}_t}_{u,c}$, $\forall c \in \mathcal{S} \cup \mathcal{M}$, are computed by modelling the distribution buildings, waterways, streets and open spaces. The path loss model is described by the authors in [5].

The fitness of a solution $\mathcal{C}$ is calculated using $G^{\mathcal{F}_t}$ as follows. First, the signals received by UE u from each cell c are computed via,

$$Signal_{u,c} = g^{\mathcal{F}_t}_{u,c} + P_c,$$

where, P_c is the transmitting power of c (in [dBm]) as specified by $\mathcal{C}$. Then, the UE attachments are determined using Equation 1, where again, all elements of β_c are read from $\mathcal{C}$. Next, $SINR_{u,f}$ is computed for each UE $\forall f \in \mathcal{F}_t$ using Equation 3. Finally, Shannon's formula (Equation 4) is invoked to calculate the downlink rates, and Equation 5 is called to yield $fitness^{\mathcal{F}_t}_{\mathcal{C}}$– the fitness of individual $\mathcal{C}$ based on the channel gains reported by $u \in \mathcal{U}$ during frame $\mathcal{F}_t$.

Computing $fitness^{\mathcal{F}_t}_{\mathcal{C}}$ is computationally expensive, yet new solutions are required every ten minutes or so. In order to achieve this short run time, individuals are evaluated against only five[2] channel gain matrices $G^{\mathcal{F}_t}$, from five randomly sampled recent frames. The fitness assigned to individual $\mathcal{C}$ is given by the average of $fitness^{\mathcal{F}_t}_{\mathcal{C}}$ computed over the five $G^{\mathcal{F}_t}$ that currently constitute the training set. A moving window approach is adopted whereby the training set is replenished after every five generations with the most recent reports.

[2] If $pop\ size = 1000$ and $gens = 100$ and runs are executed on a machine with 50 cores operating at 2.66 GHz

4.4 Pseudocode

Algorithm 1 differs from the canonical GA in two respects. Firstly, the initial population is seeded from an explicit memory storing previously evolved solutions (line 2). Of the *pop size* individuals in the initial population, a total of $\lfloor pop\ size \times (mix/100.0) \rfloor$ are drawn from memory and the rest are randomly initialised, where $mix \in [0.0, 100.0]$. It is hypothesized that integrating knowledge from past scenarios should be beneficial since hotspots materialise at predictable locations during rush hour periods. Section 6 experimentally validates the use of explicit memory for this instance of a dynamic environment. Secondly, the training set is updated periodically during the run as described in Section 4.3.

Algorithm 1 Genetic Algorithm(Parameters in Table 2)

```
 1: procedure OPTIMISE $\mathcal{C}$
 2:     Seed $mix$ [%] of the initial population $\mathcal{P}$ with individuals from memory;
 3:     Sample channel gain matrices $G^{\mathcal{F}_t}$ from 5 recent frames to form training set $\mathcal{T}$;
 4:     for $gen = 1$ to $\#gens$ do
 5:         if $gen \,\%\, 5 == 0$ then
 6:             $\mathcal{T} \leftarrow$ update $\mathcal{T}$ with five more recent samples of $G^{\mathcal{F}_t}$;
 7:         Evaluate each individual in $\mathcal{P}$ against $\mathcal{T}$ as described in Section 4.3;
 8:         Select $\frac{pop\ size - \#elites}{2}$ pairs of parents by tournament selection (size=5);
 9:         Recombine each pair of selected parents using uniform crossover;
10:         Mutate both children from each crossover event;
11:         Evaluate the children against $\mathcal{T}$;
12:         Replace all but the $\#elites$ best individuals in $\mathcal{P}$ with the children;
    return the fittest individual on the final $\mathcal{T}$;
```

Table 2 displays the evolutionary parameters. The constants were tuned via a parameter sweep. The choice of $pop\ size = 1000$ and $\#gens = 100$ negotiated a satisfactory tradeoff between running time and performance for this dynamic problem.

Parameter	Value
pop size	1000
#gens	100
Initialisation	Random with *mix* [%] seeded from memory
Selection	Tournament
Tournament Size	5 (= 0.5% of *pop size*)
#elites	10 (= 1% of *pop size*)
Mutation and Crossover	Section 4.2
Crossover Probability	Each pair of selected parents
Mutation Probability	Each child is mutated

Table 2: Evolutionary Parameters.

5 Experiments

A HetNet with $|\mathcal{M}| = 21$ MCs and $|\mathcal{S}| = 63$ SCs (an average 3 per MC sector) was simulated in a $3.61\,\text{km}^2$ region of Dublin City Centre. Tri-sector MC towers were distributed on a hexagonal grid pattern. SCs were placed at random locations to mimic their ad-hoc deployment where hotspots tend to arise.

5.1 Traffic Model

The GA was trained on 100 samples of $G^{\mathcal{F}_t}$ collected over the first ten minutes of 'scenarios' lasting twenty minutes. The generalisation of evolved solutions was assessed on 100 samples of $G^{\mathcal{F}_t}$ from the final ten minutes of each scenario.

Sixty hotspots were simulated in a given scenario. Half of the hotspots were placed within 10 to 20 metres of (randomly selected) SCs, and half were placed at arbitrary locations. The number of UEs in hotspot HS during a sampled frame $\mathcal{F}_t$ was drawn from $\mathcal{N}(\mu_{HS}(t), \sigma_{HS}(t))$. Parameters $\mu_{HS}(t)$ and $\sigma_{HS}(t)$ denote, respectively, the mean and standard deviation of the number of UEs in HS at timestep t. The physical size $r_{HS}(t)$ of HS were also varied over the 20 simulated minutes from $t = 0$ to $t = 200$. Table 3 displays the minimum and maximum values that the parameters of the traffic model can take. Parameters μ_{HS}, σ_{HS} and r_{HS} were initialised to random values in their allowed ranges at timestep $t = 0$. These values were adjusted linearly[3] until the end of the train/test period at $t = 200$. The term δ_{HS} controlled how much a parameter changed as a fraction of its allowed range; it was selected uniformly from $[-0.25, 0.25]$ for each HS.

	μ_{HS}	σ_{HS}	r_{HS} [m]	δ_{HS}
min	5.0	0.5	5.0	−0.25
max	20.0	2.5	20.0	0.25

Table 3: Parameters of the traffic model.

Hotspots were populated first and then the remaining UEs were distributed randomly until $|\mathcal{U}| = 1200$ existed on the map. This 'full buffer' model simulates the dynamic properties of ephemeral hotspots and the stochastic character of wireless traffic.

5.2 Experimental Setup

The explicit memory was initialised by executing the GA with random initial populations on 1000 different scenarios. Training and testing channel gains matrices $G^{\mathcal{F}_t}$ were then saved from 100 unique scenarios, distinct from those used to initialise the memory. The next section analyses convergence on training data and assesses the generalisation of evolved solutions on test data. Finally, a case study illustrates how complementary settings are evolved at the power, bias and ABS layers.

[3] This traffic model is adopted since localisation errors are tens of meters in real networks. Hence, the properties of hotspots must be estimated.

6 Results and Discussion

A comprehensive statistical validation of all components of the proposed framework is computationally infeasible within the scope of this proof of concept study. Instead, the memory is initialised just once as described in Section 5.2. The reliability of the framework is then assessed by running the GA 100 times (with and without memory) on a single scenario, using different random seeds for each run. However, performance may be sensitive to the peculiar distribution of hotspots in the scenario selected for these runs. The GA's ability to generalise across many different scenarios is assessed by executing a single run for 100 distinct scenarios (in experiments with and without memory).

6.1 Training

Reliability of the GA on a Single Scenario: Figure 2 plots the average best-of-generation fitness on a single scenario (computed over 100 runs), for the baseline method and the GA with various choices of *mix* [%]. The very thin shaded 95% confidence intervals enclosing the means implies that the GA reliably produces fit solutions.

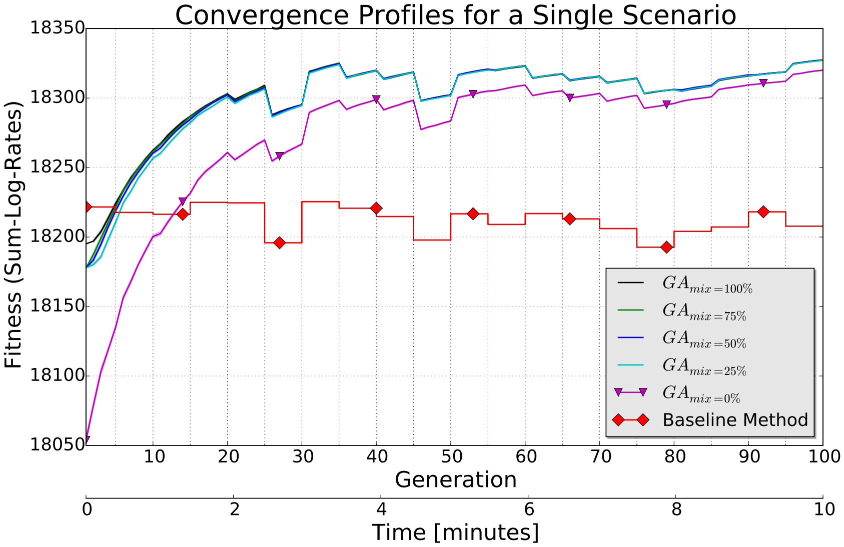

Fig. 2: The GA reliably discovers highly fit solutions.

The baseline implements constant settings of $P_s = 35$ [dBm] and $\beta_s = 10$ [dB] , $\forall s \in \mathcal{S}$, and $ABS_{r,m} = 0.5, \forall m \in \mathcal{M}$. This baseline is currently used by network operators in practice. Notice that the baseline fitness jumps every time the training set is refreshed.

Interestingly, the GA's performance improves dramatically when some of the initial population is seeded from memory. Seeded runs converge faster and they yield statistically significantly better end-of-run solutions than unseeded runs ($GA_{mix=0\%}$). Furthermore, seeded runs surpass the baseline method earlier at around generation five. This result highlights the benefit of incorporating knowledge from similar frequently encountered scenarios. Convergence profiles are displayed for experiments with $mix = 0\%,\ 25\%,\ 50\%,\ 75\%$ and 100%. Convergence is similar and stable for all values of $mix \geq 25\%$. The best models are evolved with $mix = 50\%$.

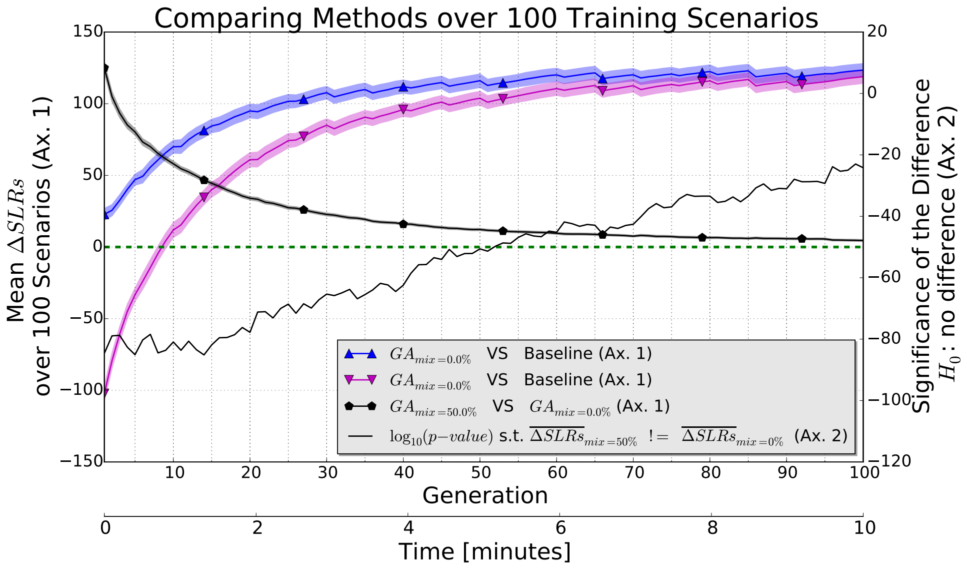

Fig. 3: The green dotted line on Ax. 1 indicates the zero level: no difference exists in the compared quantities. The black line on Ax. 2 indicates the logarithm of the p-values from paired t-tests with $H_0 : \overline{\Delta SLRs}_{mix=0\%} = \overline{\Delta SLRs}_{mix=50\%}$.

Performance across Multiple Scenarios: Ax. 1 in Figure 3 indicates the mean difference in sum-log-rates between various methods versus baseline, computed over 100 scenarios, with 95% confidence intervals included. The pattern from Figure 2 is echoed. That is, seeded runs converge to better solutions faster (blue line) than runs with random initial populations (magenta line). Seeded runs outperform the baseline after just 3 generations, compared to 13 generations for unseeded runs. The black curve on Ax. 1 indicates the difference between the blue and magenta lines. It never crosses the zero level of Ax. 1 (dotted green line). Thus, the seeded GA achieves higher mean sum-log-rates across the 100 scenarios throughout. Furthermore, the differences at each generation are significant (at $\alpha = 0.05$) based on two-sample paired t-tests (black line on Ax. 2).

6.2 Benchmarking on Test Data

The proposed GA was benchmarked against an algorithm adapted from [14]. The benchmark works by first hill climbing in the SC power and bias spaces, and then using the method from [14] to establish MC ABS ratios (the lone parameter α is tuned on validation data). Initially, cells are configured using the baseline settings, such that, $P_s := 35$ [dBm] and $\beta_s := 10$ [dB], $\forall s \in \mathcal{S}$, and $ABS_{r,m} := 4$, $\forall m \in \mathcal{M}$. Then the following steps are iterated fivefold:

- P_1 (the power of SC 1) is incremented from 23 [dBm] to 35 [dBm] in steps of 1 [dBm]. Equation 5 is evaluated at each step, and hence P_1 is set to the value that maximises the sum-log-rates. The process is repeated for P_2 (after P_1 has been updated), and so on until all SC powers have been updated.
- Similarly, SC biases are updated by incrementing β_s, $\forall s \in \mathcal{S}$, from 0 [dB] to 15 [dB] in steps of 1 [dB] - selecting the setting at each SC that maximises Equation 5.
- Finally, the rule from [14] is executed to update $ABS_{r,m}$, $\forall m \in \mathcal{M}$.

Note that the training set is refreshed after every five iterations of the preceding steps, as described in Section 4.3.

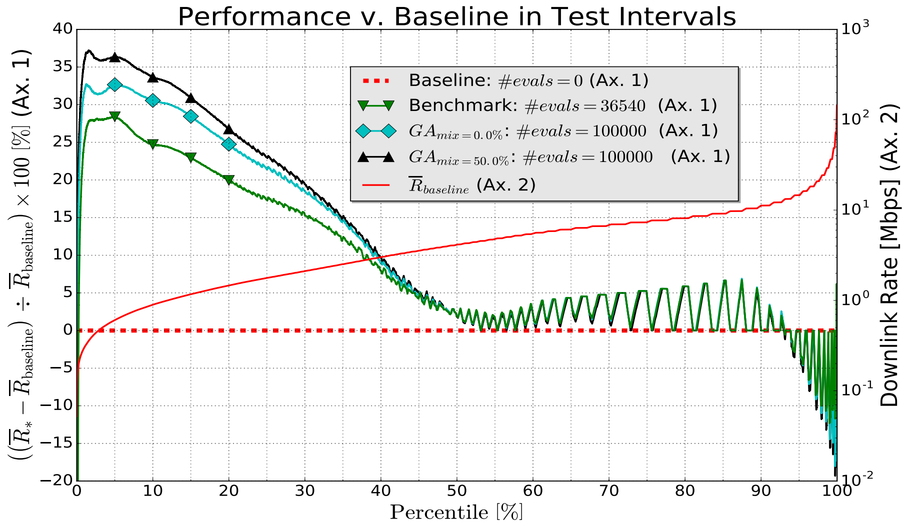

Fig. 4: Evolved settings boost cell-edge rates VS a static baseline by over 36% and outperform the benchmark scheme. The number of function evaluations required are indicated.

Figure 4 compares percentiles of the downlink rates received by $\#scenarios \times \#frames \times \#UEs = 25 \times 100 \times 1200$ UEs simulated during the test intervals of

25 different scenarios. Ax. 1 represents the percentage change of downlink rates versus baseline (dotted red line). Ax. 2 plots the downlink rates in [Mbps] for the baseline (solid red line).

The unseeded GA (cyan line) increases cell-edge rates (LHS of the plot) by around 32% versus baseline. These gains are achieved by sacrificing the highest-rate cell-centre UEs (RHS of the plot) by only $\approx 20\%$. Thus, edge UEs experience greatly improved quality of service at negligible cost to their cell-centre counterparts. Seeding boosts the 2.5^{th} percentile performance by a further 4% (black line). Clearly, the GA dramatically outperforms the benchmark scheme (green line) described above.

Two-sample Kolmogorov–Smirnov tests indicate that the distributions of downlink rates for all methods are significantly ($\mathrm{p-values} \ll 0.05$) different to that of the baseline. The following distributions are also mutually significantly different: $GA_{mix=50\%}$ v. $GA_{mix=0\%}$, and $GA_{mix=50\%}$ v. Benchmark.

6.3 Multi-layer Interactions

Figure 5 depicts a $0.05\,\mathrm{km}^2$ region of the simulated network, where an umbrella MC with an embedded SC is visible. The GA was executed for; scenario 1 (LHS) where a hotspot containing 30 UEs is adjacent to SC 1, and scenario 2 (RHS) where the hotspot is beyond the reach of SC 1.

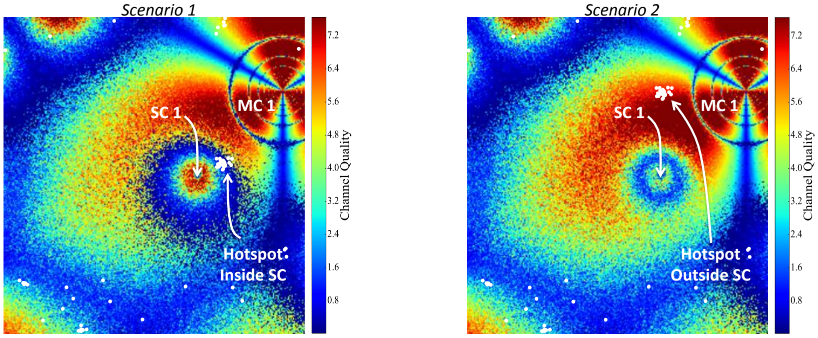

Fig. 5: Complementary settings are evolved at each layer.

The GA converged on $P_1 \leftarrow 33.0$ [dBm], $\beta_1 \leftarrow 11.6$ [dB] and $ABS_{r,1} \leftarrow 1$ for scenario 1. Thus, the SC employs a large bias to absorb UEs from the adjacent hotspot. SC 1 also transmits at near maximum power so the absorbed UEs receive high channel quality. Finally, MC 1 mutes for 35 out of 40 subframes to mitigate cross-tier interference in the expanded region of SC 1.

In contrast, $P_1 \leftarrow 23.2$ [dBm], $\beta_1 \leftarrow 1.8$ [dB] and $ABS_{r,1} \leftarrow 6$ for scenario 2. Here, SC 1 cannot offload the distant hotspot from the MC tier so it uses a small bias. In addition, P_1 is reduced to mitigate small-to-macro interference. MC 1 transmits often to satisfy its attached UEs. However, $ABS_{r,1} \leftarrow 6 < 7$ to protect distant SCs. Thus, the optimal settings at a particular cell depend on conditions at remote cells. In summary, the GA evolves synergistic settings at the ABS, power and bias layers in this region of the HetNet.

7 Future Work and Conclusions

HetNets must be reconfigured every few minutes to keep pace with changing traffic patterns. Such a timescale is on par with the running time of an evolutionary algorithm. This paper presented a customised GA for configuring multi-layer HetNets in real time. Cell-edge capacity was significantly increased over industry standard baselines in system level simulations. The experiments revealed that fitter solutions are evolved more rapidly when runs are seeded from an explicit memory. Smarter mechanisms for adapting the memory over time could be developed in a follow-up study.

A state of the art benchmark was significantly surpassed by the GA. The inferior performance of the greedy benchmark is unsurprising in this domain given that HetNet reconfiguration is an NP-hard problem. However, the benchmark significantly outperformed baseline settings that are currently used in practice. This result motivates future work to further explore the tradeoff between performance and running time of greedy versus non-greedy methods. Finally, this paper addressed a real-world dynamic optimisation problem that could serve as a test bed for designing novel adaptive heuristics in dynamic environments.

Acknowledgements

This research is based upon works supported by the Science Foundation Ireland under grant 13/IA/1850. The authors are grateful to the reviewers and Dr. Miguel Nicolau for their helpful comments.

References

1. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019. Cisco (2015), White Paper (online)
2. Aliu, O.G., Imran, A., Imran, M.A., Evans, B.: A survey of self organisation in future cellular networks. IEEE Communications Surveys Tutorials 15(1), 336–361 (First 2013)
3. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: In Congress on Evolutionary Computation CEC99. Citeseer (1999)
4. Branke, J.: Evolutionary optimization in dynamic environments, vol. 3. Springer Science & Business Media (2012)
5. Claussen, H., Ho, L.: Multi-carrier cell structures with angular offset. In: Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on. pp. 1179–1184. IEEE (2012)
6. Deb, K., Myburgh, C.: Breaking the billion-variable barrier in real-world optimization using a customized evolutionary algorithm. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference. pp. 653–660. ACM (2016)

7. Deb, S., Monogioudis, P., Miernik, J., Seymour, J.P.: Algorithms for enhanced Inter-cell Interference Coordination (eICIC) in LTE HetNets. IEEE/ACM Transactions on Networking (TON) 22(1), 137–150 (2014)
8. Dempsey, I., O'Neill, M., Brabazon, A.: Foundations in grammatical evolution for dynamic environments, vol. 194. Springer (2009)
9. Fenton, M., Lynch, D., Kucera, S., Claussen, H., O'Neill, M.: Evolving Coverage Optimisation Functions for Heterogeneous Networks using Grammatical Genetic Programming. In: Proceedings of the 19th International Conference on the Applications of Evolutionary Computation, EvoCOMNET 2016. Springer
10. Fenton, M., Lynch, D., Kucera, S., Claussen, H., O'Neill, M.: Load Balancing in Heterogeneous Networks using an Evolutionary Algorithm. In: Evolutionary Computation (CEC), 2015 IEEE Congress on. pp. 70–76. IEEE (2015)
11. Hämäläinen, S., Sanneck, H., Sartori, C.: LTE self-organising networks (SON): network management automation for operational efficiency. John Wiley & Sons (2012)
12. Karaman, A., Uyar, Ş., Eryiğit, G.: The memory indexing evolutionary algorithm for dynamic environments. In: Workshops on Applications of Evolutionary Computation. pp. 563–573. Springer (2005)
13. Liang, Y.: Real-time vbr video traffic prediction for dynamic bandwidth allocation. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 34(1), 32–47 (2004)
14. López-Pérez, D., Claussen, H.: Duty Cycles and Load Balancing in HetNets with eICIC Almost Blank Subframes. In: Personal, Indoor and Mobile Radio Communications (PIMRC Workshops), 2013 IEEE 24th International Symposium on. pp. 173–178. IEEE (2013)
15. Lopez-Perez, D., Guvenc, I., De la Roche, G., Kountouris, M., Quek, T.Q., Zhang, J.: Enhanced Intercell Interference Coordination Challenges in Heterogeneous Networks. IEEE Wireless Communications 18(3), 22–30 (2011)
16. Madan, R., Borran, J., Sampath, A., Bhushan, N., Khandekar, A., Ji, T.: Cell association and interference coordination in heterogeneous lte-a cellular networks. IEEE Journal on Selected Areas in Communications 28(9), 1479–1489 (December 2010)
17. Morrison, R.W.: Designing evolutionary algorithms for dynamic environments. Springer Science & Business Media (2013)
18. Peng, M., Liang, D., Wei, Y., Li, J., Chen, H.H.: Self-configuration and self-optimization in LTE-advanced heterogeneous networks. IEEE Communications Magazine 51(5), 36–45 (2013)
19. Ramsey, C.L., Grefenstette, J.J.: Case-based initialization of genetic algorithms. In: ICGA. pp. 84–91. Citeseer (1993)
20. Shannon, C.E.: Communication In The Presence Of Noise. Proceedings of the IRE 37(1), 10–21 (1949)
21. Tall, A., Altman, Z., Altman, E.: Self organizing strategies for enhanced ICIC (eICIC). In: Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on. pp. 318–325. IEEE (2014)
22. Tang, J., So, D.K., Alsusa, E., Hamdi, K.A., Shojaeifard, A.: Resource allocation for energy efficiency optimization in heterogeneous networks. IEEE Journal on Selected Areas in Communications 33(10), 2104–2117 (2015)
23. Winstein, K., Sivaraman, A., Balakrishnan, H.: Stochastic forecasts achieve high throughput and low delay over cellular networks. In: Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13). pp. 459–471 (2013)