Automated Self-Optimization in Heterogeneous Wireless Communications Networks

David Lynch, Michael Fenton, David Fagan, Stepan Kucera, *Senior Member, IEEE*,
Holger Claussen, *Senior Member, IEEE*, and Michael O'Neill

***Abstract*— Traditional single-tiered wireless communications networks cannot scale to satisfy exponentially rising demand. Operators are increasing capacity by densifying their existing macro cell deployments with co-channel small cells. However, cross-tier interference and load balancing issues present new optimization challenges in channel sharing heterogeneous networks (HetNets). One-size-fits-all heuristics for allocating resources are highly suboptimal, but designing *ad hoc* controllers requires significant human expertise and manual fine-tuning. In this paper, a unified, flexible, and fully automated approach for end-to-end optimization in multi-layer HetNets is presented. A hill climbing algorithm is developed for reconfiguring cells in real time in order to track dynamic traffic patterns. Schedulers for allocating spectrum to user equipment are automatically synthesized using grammar-based genetic programming. The proposed methods for configuring the HetNet and scheduling in the time–frequency domain can address *ad hoc* objective functions. Thus, the operator can flexibly tune the tradeoff between peak rates and fairness. Far cell edge downlink rates are increased by up to 250% compared with non-adaptive baselines. Alternatively, peak rates are increased by up to 340%. The experiments illustrate the utility and future potential of natural computing techniques in software-defined wireless communications networks.**

***Index Terms*— Heterogeneous networks, software defined networking, genetic programming, self-organizing networks.**

I. INTRODUCTION

MOBILE traffic has grown eighteen-fold over the past five years [1]. Traditional wireless infrastructure cannot scale to satisfy demand during the current era of exponential growth [2]. Densification, virtualization, and self-optimization are three key design principles for realizing the 5G vision of ultra-high performance mobile communications networks.

Densification refers to the deployment of Small Cells (SCs) alongside existing Macro Cells (MCs) in order to increase capacity by bringing the network closer to User Equipments (UEs: smartphones, tablets, etc.). Operators agree that densification is the only way to address the existence of fluid hotspots. Thus, the cellular paradigm has evolved from a homogeneous single-tiered model to a heterogeneous multi-tiered model. Heterogeneous Networks (HetNets) are spectrally efficient since both cell tiers can transmit across the same scarce and expensive spectrum (i.e. bandwidth) [3]. However, two major issues arise in dense channel sharing architectures [4]. Firstly, lower-powered SCs struggle to offload UEs from much stronger MCs. Secondly, UEs at cell edges experience severe interference from nearby co-channel cells. Inefficient offloading, coupled with increased interference due to higher cell density, results in frequent handovers and more dropped calls. This presents a major problem for operators because maintaining high customer satisfaction is crucial in the fiercely competitive wireless telecommunications industry.

Release 10 of the 3rd Generation Partnership Project (3GPP) [5] specified mechanisms for load balancing and interference mitigation in HetNets. However, manually designing controllers that implement these protocols is time consuming and labor intensive. One-size-fits-all algorithms are highly suboptimal because they cannot handle corner cases that arise in different deployment scenarios. In previous work, ad-hoc heuristics that address isolated sub-problems were proposed. Specialized controllers tend to produce contradictory control actions when combined. Operators will not permit trial and error optimization in expensive deployments, nor inexplicable decision-making by unpredictable multi-layer heuristics. We present a novel, fully automated, and unified approach to self-optimization in multi-layer HetNets that devises complementary settings across different layers. Automation replaces ad-hoc, costly, and inefficient design by human experts.

In this paper the NP-hard [6] problem that lies at the heart of Long Term Evolution (LTE) HetNets is addressed. Techniques for jointly optimizing UE-cell associations and scheduling UEs to receive data in the time-frequency domain are developed. This coupled optimization problem is challenging for three main reasons. Firstly, HetNets consist of multiple layers of parameters; the settings devised at a particular layer (e.g. SC powers) must be complementary with those at the other layers (e.g. MC muting frequencies). Secondly, updated schedules are required every few milliseconds. Thirdly, wireless networks are highly dynamic and uncertain environments. The contributions of this study are as follows:

- A unified and data-driven hill climbing algorithm for reconfiguring cells in real time as traffic patterns fluctuate. The hill climber discovers synergistic settings across multiple parameter layers of a HetNet. By contrast, less

---

Manuscript received July 25, 2017; revised February 21, 2018, October 10, 2018 and December 17, 2018; accepted December 23, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor G. Zussman. Date of publication January 10, 2019; date of current version February 14, 2019. This work was supported by Science Foundation Ireland under Grant 13/IA/1850. *(Corresponding author: David Lynch.)*

D. Lynch, D. Fagan, and M. O'Neill are with the Natural Computing Research and Applications Group, University College Dublin, Dublin, A94 XF34 Ireland (e-mail: david.lynch.1@ucdconnect.ie; david.fagan@ucd.ie; m.oneill@ucd.ie).

M. Fenton was with the Natural Computing Research and Applications Group, University College Dublin, Dublin, A94 XF34 Ireland. He is now with Corvil, Dublin, D01 K5C7 Ireland (e-mail: michael.fenton@ucd.ie).

S. Kucera and H. Claussen are with Nokia Bell Labs at Dublin, Dublin, D15 Y6NT Ireland (e-mail: stepan.kucera@nokia-bell-labs.com; holger.claussen@nokia-bell-labs.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TNET.2018.2890547

unified heuristics are prone to antagonistic decision-making, which manifests in unpredictable behavior.

- A technique from natural computing [7] called Grammar-based Genetic Programming (GP) [8], [9] is employed to automatically synthesize schedulers that allocate spectrum on a millisecond timescale. Automation is highly valued since costly manual fine tuning of human-designed schedulers is often infeasible. GP outperforms a state of the art benchmark because it tailors schedulers to the deployment scenario. Performance comparisons with less expressive linear models demonstrate the ability of GP to construct useful feature representations.
- A novel objective function that enables flexible and fine-grained control over fairness trade-offs is presented. Thus, the operator can prioritize cell edge performance over peak rates, or vice versa. Service differentiation is a core feature of the LTE standard and the ongoing 5G standardization [10].
- Semantic analysis of the hill climbing algorithm and evolved schedulers reveals the nuanced yet intuitive strategies that are discovered for allocating resources in this dynamic environment.

The emerging virtualization paradigm [11], enabled by software-defined networking [12] and abstracted hardware with reprogrammable control (e.g. Nokia's Airscale [13] technologies), will support an efficient implementation of the proposed techniques. Virtualization facilitates the real-time collection of measurement reports and the dissemination of optimization decisions.

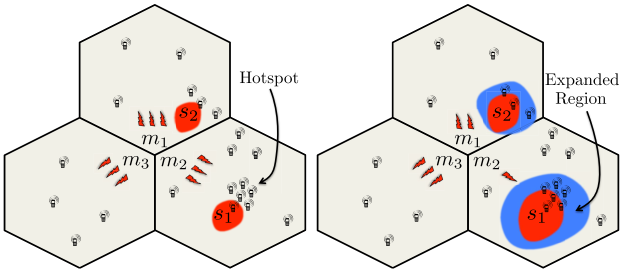
The paper is organized as follows. Section II formalizes the optimization problem. Previous work on optimizing HetNets is reviewed in Section III. Methods for optimizing the network configuration and scheduling are developed in Section IV. The simulation environment and experiments are described in Section V. Section VI demonstrates better than human-competitive performance across a range of network topologies and traffic scenarios. The paper concludes with directions for future work in Section VII.

II. PROBLEM DEFINITION

Downlink rates can be managed by configuring cells and intelligently scheduling UEs. This section outlines the mechanisms for load balancing, interference mitigation, and scheduling that were specified in Release 10 of the 3GPP standard [5].

A. Load Balancing and Interference Mitigation in HetNets

The SC tier is typically underutilized in HetNets because low-powered SCs struggle to offload UEs from stronger MCs. Figure 1a depicts this load balancing issue in a toy deployment with two SCs $s_1, s_2 \in \mathcal{S}$ and three MCs $m_1, m_2, m_3 \in \mathcal{M}$, where $\mathcal{S}$ and $\mathcal{M}$ denote the sets of SCs and MCs in a HetNet. Here, m_2 is congested because a hotspot is embedded within its coverage area. The underutilized SC s_1 could relieve congestion on m_2 by absorbing UEs from the adjacent hotspot. Now, s_1 can extend its coverage area by increasing its power P_{s_1}. However, s_1 may be unable to expand its footprint sufficiently before reaching maximum power.



(a) (b)

Fig. 1. Toy HetNet containing three MCs with two embedded SCs. (a) Unoptimized $\mathscr{C}$. (b) Optimized $\mathscr{C}$.

A range expansion mechanism has been proposed in LTE-advanced to enable more efficient offloading onto the SC tier [4], [5]. SCs $s \in \mathcal{S}$ can expand their footprint by broadcasting a cell selection bias (β_s) that UEs factor into their attachment decisions. Hence, UE u attaches to cell $c \in \mathcal{M} \cup \mathcal{S}$:

$$c = \arg\max_{k} \; (signal_{u,k} + \beta_k), \quad \forall\, k \in \mathcal{M} \cup \mathcal{S}, \tag{1}$$

where $\beta_k \geq 0$ [dB], $\forall k \in \mathcal{S}$ and $\beta_k = 0$ [dB], $\forall k \in \mathcal{M}$ since MCs do not implement range expansion. In Equation 1, $signal_{u,k}$ is the signal strength (in [dBm]) that UE u receives from cell k, and is given by:

$$signal_{u,k} \leftarrow P_k + g_{u,k}, \tag{2}$$

where P_k is the transmitting power of k (in [dBm]) and $g_{u,k}$ is channel gain (in [dB]) from k to u's location.

The blue annuli in the optimized HetNet of Figure 1b indicates the 'expanded regions' that form around s_1 and s_2 when they broadcast non-zero selection biases. Herein, UEs attach to the SCs despite receiving stronger signals from the overlaid MCs. Let $\mathcal{A}_c$ denote the set of UEs that attach to, and hence receive packets from, cell c. Range expansion reduces the load imbalance from $|\mathcal{A}_{m_2}| - |\mathcal{A}_{s_1}| = 9$ in Figure 1a, to just $|\mathcal{A}_{s_1}| - |\mathcal{A}_{m_2}| = 1$ in Figure 1b. Similarly, s_2 offloads three UEs from m_1 by using range expansion.

UEs in the expanded region at the edge of s_1 and s_2 will experience severe cross-tier interference from the overlaid MCs, which transmit at higher power across the same bandwidth. Channel conditions can be improved dramatically at SC edges by forcing interfering MCs to mute periodically. When a MC mutes we say that it executes an Almost Blank Subframe (ABS) [4]. Note that subframes are 1 [ms] intervals during which cells send packets to their attached UEs. There are 40 subframes in a full frame $\mathcal{F}_t$, where $t \in \mathbb{N}$ is a counter for the number of elapsed frames.

Table I displays the seven ABS patterns that a MC can execute during a frame. For instance, if the ABS ratio of m_2 from Figure 1b is set to $ABS^{ratio}_{m_2} = 5/40$, then m_2 executes the ABS pattern in the first row of Table I, such that, m_2 mutes in 35 out of 40 subframes and transmits every eighth subframe. In the toy HetNet, a low ABS ratio for m_1 and m_2 will reduce interference for UEs in the expanded regions of their embedded SCs. In contrast, m_3 should run a higher ABS ratio as there are no SCs within its coverage area. Note that SCs cannot mute so they are active in all subframes $f \in \mathcal{F}_t$ (as indicated by the last row of Table I).

TABLE I

ABS RATIOS AND PATTERNS FOR $m \in \mathcal{M}$ AND $s \in \mathcal{S}$

ABS_m^{ratio}	$ABS_{m,f}^{pattern}$				
$5/40$	00000001	00000001	00000001	00000001	00000001
$10/40$	00000011	00000011	00000011	00000011	00000011
$15/40$	00000111	00000111	00000111	00000111	00000111
$20/40$	00001111	00001111	00001111	00001111	00001111
$25/40$	00011111	00011111	00011111	00011111	00011111
$30/40$	00111111	00111111	00111111	00111111	00111111
$35/40$	01111111	01111111	01111111	01111111	01111111
ABS_s^{ratio}	$ABS_{s,f}^{pattern}$				
$40/40$	11111111	11111111	11111111	11111111	11111111

In summary, the powers and biases of SCs[1] can be modulated in order to balance load between cells, and severe interference at SC edges can be mitigated by periodically muting MCs. Hence, the 'configuration' $\mathscr{C}$ of a HetNet with $|\mathcal{S}|$ SCs and $|\mathcal{M}|$ MCs is described by the array:

$$\mathscr{C} = \left[P_1, \ldots, P_{|\mathcal{S}|}, \beta_1, \ldots, \beta_{|\mathcal{S}|}, ABS_1^{ratio}, \ldots, ABS_{|\mathcal{M}|}^{ratio}\right],$$

where $P_s \in$ [23.0 [dBm], 35.0 [dBm]], $\beta_s \in$ [0.0 [dB], 15.0 [dB]], and $ABS_m^{ratio} \in [5/40, \ldots, 35/40]$.

B. Computing Downlink Rates Using Shannon's Formula

The rate at which data is transmitted from cell c to a UE $u \in \mathcal{A}_c$ during subframe $f \in \mathcal{F}_t$ depends (logarithmically) on the signal to interference and noise ratio:

$$SINR_{u,f} = \frac{signal_{u,c,f} \times ABS_{c,f}^{pattern}}{noise + \sum\limits_{k \in \mathcal{M} \cup \mathcal{S} \backslash c} signal_{u,k,f} \times ABS_{k,f}^{pattern}}, \tag{3}$$

where $signal_{u,k,f}$ is the signal strength (converted to Watts here) that u receives from cell $k \in \mathcal{M} \cup \mathcal{S}$ during f, $ABS_{k,f}^{pattern}$ indicates whether k transmits or mutes during f, and $noise = 4 \times 10^{-16}$ [Watts] approximates the background electromagnetic noise. Shannon's formula [14] gives the instantaneous downlink rate (in [Mbps]) that u receives in f:

$$R_{u,f} = 20 \text{ [MHz]} \times sch_{u,f} \times Q_{u,f}, \tag{4}$$

where 20 [MHz] is the fixed bandwidth, $0.0 \leq sch_{u,f} \leq 1.0$ is the proportion of the bandwidth that u is scheduled to receive during f, and $Q_{u,f} = \log_2(1 + SINR_{u,f})$ is the 'channel quality' experienced by u in f. Shannon's formula gives an accurate approximation for LTE throughput in real HetNets [15]. Finally, the (average) downlink rate for u over frame $\mathcal{F}_t$ is given by

$$\overline{R}_u = |\mathcal{F}_t|^{-1} \sum_{f \in \mathcal{F}_t} R_{u,f}. \tag{5}$$

C. Optimizing the Network Configuration

The quality of service that a customer experiences depends on the downlink rate $\overline{R}_u$ received by their UE. Extremely low rates will result in significant packet losses, causing applications to become unresponsive. High-rate UEs near cell centers can usually afford to liberate resources for the sake of low-rate UEs at cell edges. In this paper, the terms 'cell edge' and 'cell center' refer loosely to regions of low and high $SINR$; they do not necessarily imply physical proximity to the cell.

A UE's received downlink rate (Equation 5) depends implicitly on the HetNet configuration $\mathscr{C}$ through Equations 1, 2, and 3, which determine the UE-cell association, received signals from serving and interfering cells, and the signal to interference and noise ratio respectively. Therefore, the quality of service experienced by customers can be managed by optimizing the elements of $\mathscr{C}$. Let $\mathcal{U}$ denote the set of UEs that have been active in the HetNet during 25 frames sampled at random since the HetNet was last reconfigured. Furthermore, let $f_{config}\left(\left\{\overline{R}_u | u \in \mathcal{U}\right\}, \gamma\right)$ be an objective function that integrates the downlink rates experienced by $u \in \mathcal{U}$ to a scalar fitness, where $\gamma \in \mathbb{R}$ is a tunable parameter. We develop a hill climbing algorithm for configuring cells by adapting the elements of $\mathscr{C}$ in real time, such that $f_{config}\left(\left\{\overline{R}_u | u \in \mathcal{U}\right\}, \gamma\right)$ is maximized for a given choice of γ. Along with a fully autonomous self-optimizing framework, one of the main contributions of this paper is a formulation for $f_{config}(\cdot, \gamma)$ that enables the operator to manage the trade-off between peak and cell edge rates by simply adjusting γ.

D. Scheduling in the Time-Frequency Domain

HetNets are reconfigured infrequently because traffic patterns change gradually, and frequent reconfigurations cause undesirable ping-pong handovers. By contrast, channel quality and demand fluctuate every few milliseconds. Downlink rates can be managed on this much shorter timescale by intelligently scheduling UEs in every 40 [ms] frame.

Shannon's formula (Equation 4) states that the instantaneous downlink rate $(R_{u,f})$ received by UE u attached to cell c in subframe f depends linearly on $sch_{u,f}$. Therefore, c can devise a 'schedule' for frame $\mathcal{F}_t$ by setting the values of $sch_{u,f}, \forall (u,f) \in \mathcal{A}_c \times \mathcal{F}_t$, such that a fitness metric $f_{sch}\left(\left\{\overline{R}_u | u \in \mathcal{A}_c\right\}, \gamma\right)$ is maximized. As in Section II-C, $f_{sch}(\cdot, \gamma)$ could capture the operator's desire for greater fairness, or alternatively, higher peak rates depending on the choice of γ. Then, the scheduling problem is described by the following non-linear program:

$$\begin{aligned}
&\text{maximize } f_{sch}\left(\left\{\overline{R}_u | u \in \mathcal{A}_c\right\}, \gamma\right),\\
&\text{such that (1) } \sum_{u \in \mathcal{A}_c} sch_{u,f} = 1.0, \quad \forall f \in \mathcal{F}_t, \text{ and}\\
&\qquad\qquad \text{(2) } 0.0 \leq sch_{u,f} \leq 1.0, \quad \forall (u,f) \in \mathcal{A}_c \times \mathcal{F}_t,
\end{aligned}$$

where constraint (1) ensures that the bandwidth is fully utilized in every subframe, and constraint (2) states that an individual UE receives between 0% and 100% of the bandwidth in a given subframe.

III. PREVIOUS WORK

Densification is a dominant theme in the evolution towards 5G [16] because co-channel HetNets are seen as the most cost-effective paradigm for scaling with future

---

[1] MCs use a constant power of 43.3 [dBm] and do not implement biases.

demand [2], [3], [5], [17]. Technical challenges associated with managing HetNets were reviewed in [4] and [18]. Aliu *et al.* [19] and Peng *et al.* [20] identified self-optimization as a key capability since frequent manual reconfiguration of dynamic HetNets is infeasible.

A. Configuring Multi-Layer HetNets

Okino *et al.* [21] confirmed that system capacity can be increased by range expansion and interference mitigation. However, the authors did not develop control algorithms for implementing these mechanisms. Vasudevan *et al.* derived an expression for adapting MC ABS ratios in response to time-varying traffic. Fairness was improved over a baseline that implemented fixed ABS ratios. Tall *et al.* [23] proposed self-organizing algorithms for adapting SC selection biases to balance load between MCs and SCs, and for setting MC ABS ratios for managing interference. Deb *et al.* [6] developed a more unified approach based on non-linear programming. Their algorithm solves the coupled problems of optimizing MC ABS ratios and UE-cell associations by setting SC biases. Deb *et al.* [6] show that configuring multi-layer HetNets is an NP-hard problem. Their algorithm relies on problem relaxations that result in suboptimal solutions. Our goal is to enable zero-touch control in HetNet optimization. Therefore, the heuristic proposed in this paper is validated with respect to a non-adaptive baseline that implements a static configuration.

B. Scheduling in HetNets

Simple scheduling strategies were utilized in the aforementioned studies. For example, SC attached UEs are split into two groups in [22]; UEs in a SC's expanded region are scheduled solely during protected ABSs, whereas UEs in the SC center are scheduled only during non-ABSs. Weber and Stanze [24] showed that this strategy is inferior to a "dynamic scheduler" that can assign UEs to both ABSs and non-ABSs on the basis of proportional fairness. Pang *et al.* [25] select the set of 'victim UEs' $u \in \mathcal{A}_s$ that are scheduled in ABSs using a dynamic programming method. All remaining UEs are scheduled solely in non-ABSs. Jiang and Lei [26] perform victim selection using a two player Nash bargaining game, where ABSs and non-ABSs compete for $u \in \mathcal{A}_s$. The victim selection step in [25] and [26] is interleaved with brute force search over all feasible ABS ratios. Thus, synchronous ABS patterns and schedules are jointly optimized. López-Pérez and Claussen [27] proposed a greedy algorithm for scheduling on SCs that will serve as our benchmark. Their algorithm attempts to equalize the downlink rates of the worst performing UEs scheduled in ABSs and non-ABSs.

C. Natural Computing in Wireless Communications Networks

Natural computing techniques [7] have been successfully applied in wireless communications networks. Ho *et al.* [28] and Hemberg *et al.* [29] evolved femtocell power control algorithms for coverage optimization using GP [30] and Grammatical Evolution [9] respectively. Grammar-based Genetic Programming [8] was employed to evolve scheduling heuristics in [31] and references therein. Fagan *et al.* [32] trained deep neural network schedulers using instructive feedback from a Genetic Algorithm. Previous works by Fenton *et al.* [33] also examined multi-layer optimization of HetNets. The SC power, SC bias, MC ABS ratio, and SC scheduling layers were addressed individually using Grammar-based GP. However, performance was constrained by limitations of the fitness function and inefficiencies due to antagonistic control actions. In this paper, limitations of previous work are overcome with:

1) novel parameterizable objective functions for better service differentiation in LTE HetNets,
2) a feature-rich grammatical representation, and
3) a unified hill-climbing algorithm for robust optimization of the network configuration.

In summary, most contributions in the literature address individual layers of a HetNet. In this paper a unified approach for joint multi-layer optimization is proposed. Hence, inefficiencies resulting from antagonistic decision-making by uncooperative controllers are mitigated. The methods described in the next section can be implemented in software, using Nokia's AirFrame data center solution [13] for online data collection, optimization, and control action dissemination.

IV. METHOD

This section presents an algorithm for optimizing the HetNet configuration $\mathscr{C}$. A flexible framework for automatically evolving schedulers using GP is then outlined, along with two baselines and a state of the art benchmark scheduler.

A. Optimizing the Network Configuration

Optimizing HetNets is challenging because measurement reports are noisy and incomplete, and the optimum settings at a particular cell depend on how other nearby cells are configured. For example, MC m_2 in the toy HetNet of Figure 1 should run a high ABS ratio since the embedded SC s_1 uses range expansion. A static baseline network configuration $\mathscr{C}_{\mathrm{BL}}$ is often implemented by operators in practice, where $P_s = 35.0$ [dBm], $\beta_s = 10.0$ [dB] for all SCs $s \in \mathcal{S}$, and $ABS_m^{ratio} = 20/40$ for MCs $m \in \mathcal{M}$. However, static settings are suboptimal since wireless traffic is highly dynamic. Figure 2 displays the proposed hill climbing (HC) algorithm for adapting $\mathscr{C}$.

The elements of $\mathscr{C}_{\mathrm{HC}}$ are initialized to $\mathscr{C}_{\mathrm{BL}}$ (line 3), and they are continuously adapted over time (line 4). Lines 11–23 describe how SC powers and biases are updated. For example, when updating the power of SC 2 (P_2), a Gaussian random variate is added to $\mathscr{C}_{\mathrm{HC}}[2]$ (line 14). The fitness of this new hypothesis for $\mathscr{C}_{\mathrm{HC}}$ is evaluated on line 16 as follows:

- Let $\mathcal{U}$ be the set of UEs that received data during 25 frames sampled since the HetNet was last reconfigured, and let $\mathcal{U}_{\mathcal{F}_t}$ be the set of UEs that were active in one such frame $\mathcal{F}_t$.
- Each $u \in \mathcal{U}_{\mathcal{F}_t}$ reports the channel gains $(g_{u,c})$ from their serving and three most strongly interfering cells — gains from all other cells cannot be measured, so they are assumed to be $-\infty$ [dB].

```
1: procedure OPTIMIZE $\mathscr{C}$(channel gain matrices : $G_{\mathcal{F}_t}, \gamma, p\_accept$)
2:     Initialize the training set $\mathcal{T} \leftarrow \emptyset$;
3:     Initialize $\mathscr{C}_{\mathrm{HC}} = \mathscr{C}_{\mathrm{BL}}$;
4:     for $iteration \in [0, 1, \ldots, \infty]$ do #iterate continuously
5:         if $(iteration \bmod 3) == 0$ then #update training set every 3 iterations
6:             $\mathcal{T} \leftarrow 25$ instances of $G_{\mathcal{F}_t}$ sampled since previous update;
7:             $f^{old}_{config} \leftarrow$ evaluate Equation 6 using channel gain matrices in $\mathcal{T}$;
8:             Push out the settings in $\mathscr{C}_{\mathrm{HC}}$ onto cells;
9:         $IDs \leftarrow shuffle\,([1, 2, \ldots, |\mathcal{M}| + 2 \times |\mathcal{S}|])$;
10:        for $i \in IDs$ do #update elements of $\mathscr{C}_{\mathrm{HC}}$ in an arbitrary order
11:            if $1 \le i \le 2 \times |\mathcal{S}|$ then #updating SC power or bias
12:                $\mathscr{C}^{old}_{\mathrm{HC}} \leftarrow \mathscr{C}_{\mathrm{HC}}\,[i]$;
13:                for $steps \in [1, 2, \ldots, 10]$ do #try ten new settings
14:                    $\mathscr{C}_{\mathrm{HC}}\,[i] \leftarrow \mathscr{C}_{\mathrm{HC}}\,[i] + \mathcal{N}(0, 2)$; #add Gaussian noise
15:                    clip $\mathscr{C}_{\mathrm{HC}}\,[i]$ so that it lies within the allowed range;
16:                    $f^{new}_{config} \leftarrow$ evaluate Equation 6 using $\mathcal{T}$;
17:                    if $f^{old}_{config} > f^{new}_{config}$ then #if the new setting is not better
18:                        if $random(0, 1) < p\_accept$ then
19:                            $f^{old}_{config} \leftarrow f^{new}_{config}$; #sometimes accept worse move
20:                        else
21:                            $\mathscr{C}_{\mathrm{HC}}\,[i] \leftarrow \mathscr{C}^{old}_{\mathrm{HC}}$; #retain the old setting
22:                    else #if the new setting is better then always accept it
23:                        $f^{old}_{config} \leftarrow f^{new}_{config}$; #keep new setting, and update fitness
24:            else if $2 \times |\mathcal{S}| \le i \le 2 \times |\mathcal{S}| + |\mathcal{M}|$ then #updating MC ABS ratios
25:                $\mathscr{C}^{old}_{\mathrm{HC}} \leftarrow \mathscr{C}_{\mathrm{HC}}\,[i]$;
26:                for $ABS^{ratio}_{m} \in [5/40, 10/40, \ldots, 35/40]$ do
27:                    $\mathscr{C}_{\mathrm{HC}}\,[i] \leftarrow ABS^{ratio}_{m}$; #evaluate all possible settings
28:                    $f^{new}_{config} \leftarrow$ evaluate Equation 6 using $\mathcal{T}$;
29:                    if $f^{old}_{config} > f^{new}_{config}$ then #if the new setting is not better
30:                        if $random(0, 1) < p\_accept$ then
31:                            $f^{old}_{config} \leftarrow f^{new}_{config}$; #sometimes accept worse move
32:                        else
33:                            $\mathscr{C}_{\mathrm{HC}}\,[i] \leftarrow \mathscr{C}^{old}_{\mathrm{HC}}$; #retain the old setting
34:                    else #if the new setting is better then always accept it
35:                        $f^{old}_{config} \leftarrow f^{new}_{config}$; #keep new setting, and update fitness
```

Fig. 2. Optimizing $\mathscr{C}$ by hill climbing.

- The channel gains from frame $\mathcal{F}_t$ are arranged in a 'channel gain matrix' $G_{\mathcal{F}_t}$ of dimensions $(|\mathcal{M}| + |\mathcal{S}|) \times |\mathcal{U}_{\mathcal{F}_t}|$. All 25 channel gain matrices form a training set $\mathcal{T}$.
- Downlink rates are computed for the UEs $u \in \mathcal{U}_{\mathcal{F}_t}$ in each frame $\mathcal{F}_t$ by evaluating Equations 1, 2, 3, 4, and 5 as described in Section II-C. These computations are carried out after cells have been configured using $\mathscr{C}_{\mathrm{HC}}$.
- Hence, Equation 6 is evaluated for all $u \in \mathcal{U}$ to give the fitness of $\mathscr{C}_{\mathrm{HC}}$ with the new setting for P_2:

$$f_{config}\left(\{\overline{R}_u | u \in \mathcal{U}\}, \gamma\right) \leftarrow \sum_{u'=1}^{|\mathcal{U}|} \left(\log_e\left(\overrightarrow{R}_{u'}\right)\right) \times e^{\frac{\gamma \times u'}{|\mathcal{U}|}}, \tag{6}$$

  where $\overrightarrow{R}$ is an ordered array storing the downlink rates $\forall u \in \mathcal{U}$, such that $\overrightarrow{R}_1 \ge \overrightarrow{R}_2 \ge \ldots \ge \overrightarrow{R}_{|\mathcal{U}|}$, and $\gamma \in \mathbb{R}$ is a parameter specified by the operator for controlling fairness trade-offs. Positive values of γ make Equation 6 sensitive to cell edge performance. Conversely, setting $\gamma < 0$ will encourage solutions that increase peak rates.

The new setting for P_2 is immediately accepted if it improves the fitness of $\mathscr{C}_{\mathrm{HC}}$ (line 23). However, worse moves are occasionally accepted in order to escape from local optima: if the new setting is worse than the previous setting it is accepted with probability p_accept (line 19), else the previous setting is retained (line 21). Ten hypothesis settings are evaluated before proceeding to adapt another element of $\mathscr{C}_{\mathrm{HC}}$. Similarly, lines 24–35 describe a brute force search over the seven allowed ABS ratios for MCs.

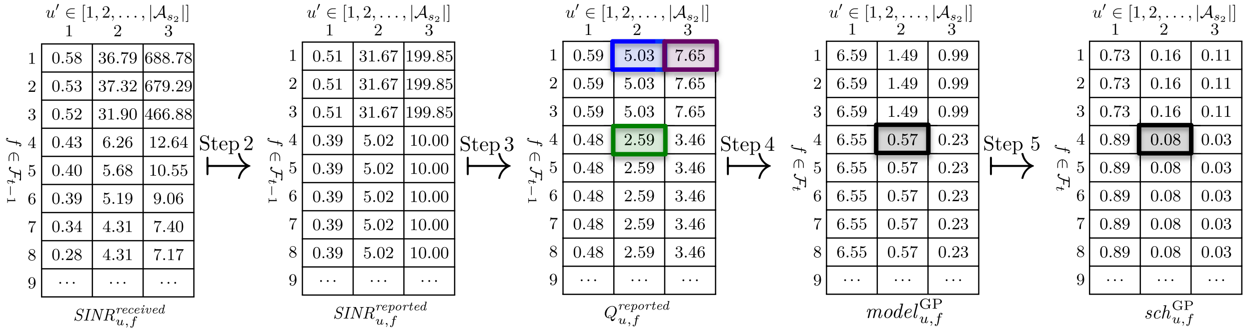
In a single iteration, $steps \times (2 \times |\mathcal{S}|)$ calls to Equation 6 (the fitness function) are needed to update all SC powers and biases (lines 10–23), where $steps = 10$ different settings are examined when updating a SC's power or bias. Similarly, $7 \times |\mathcal{M}|$ calls to Equation 6 are needed to update all MC ABS ratios (lines 24–35), where seven different settings are possible for each MC. Each call to Equation 6 requires downlink rates to be computed for $|\mathcal{U}|$ UEs in the 25 different frames represented by the current training set $\mathcal{T}$. The downlink rate for a single UE is given by Shannon's formula as described in Section II-B. The fitness evaluations can be parallelized so that optimization in real time is feasible. For instance, it takes roughly 90 seconds to complete 1 iteration on a 3.2 [GHz] machine with 4 cores, in a HetNet with 63 SCs and 21 MCs serving 2500 UEs. The algorithm requires only those data that are available for decision making in real HetNets, namely the channel gain matrices. It can be implemented on a centralized server, or in a distributed manner by hill climbing locally on cells. The training set is updated periodically (line 6), so that it reflects the most recent traffic conditions. $\mathscr{C}_{\mathrm{HC}}$ can be pushed out onto cells at any time. Frequent reconfigurations may be beneficial if traffic conditions change rapidly. Here, cells are reconfigured when the training set is updated (line 8).

B. Evolving Schedulers With Genetic Programming

This section describes how a schedule is produced for SC s_2 in the toy HetNet of Figure 1b. We first show how measurement reports are mapped to a schedule for s_2 using a toy model that was evolved with GP. The fitness function used to drive the evolutionary search is then described. Finally, the GP algorithm itself is presented, along with two baseline methods and the benchmark.

1) Generating Schedules: Consider some arbitrary frame $\mathcal{F}_t$ in which three UEs receive packets from SC s_2. Figure 3 illustrates how measurements of $SINR^{received}_{u,f}$ that are reported by UEs $u \in \mathcal{A}_{s_2}$ during frame $\mathcal{F}_{t-1}$ are mapped to the schedule that s_2 will execute in $\mathcal{F}_t$. The schedule displayed in the rightmost panel is computed as follows.

- **Step 1:** s_2 first determines the ABS ratio of its most strongly interfering MC $\left(m^{int}\right)$ — say m_1 for the sake of argument. Let $ABS^{ratio}_{m_1} = 25/40$ so that m_1 mutes in subframes 1–3, 9–11, 17–19, 25–27 and 33–35 (row 5 of Table I). Notice in Figure 3 that $SINR^{received}_{*,f}$ is larger in protected subframes during which m_1 mutes.
- **Step 2:** s_2 instructs its attached UEs to measure their average $SINR$ over subframes overlapping with the ABSs and non-ABSs of m_1. The UEs report back 'wideband' $SINR$'s that are rounded to the nearest decibel, and clipped within the range [−5.0 [dB], 23.0 [dB]]. Wideband reports are clipped because packets are dropped if the $SINR < -5.0$ [dB], and the UE's hardware cannot exploit values of $SINR > 23.0$ [dB].
- **Step 3:** s_2 transforms wideband $SINR$'s to wideband channel qualities: $Q^{reported}_{u,f} \triangleq \log_2\left(1 + SINR^{reported}_{u,f}\right)$.



	1	2	3
1	0.58	36.79	688.78
2	0.53	37.32	679.29
3	0.52	31.90	466.88
4	0.43	6.26	12.64
5	0.40	5.68	10.55
6	0.39	5.19	9.06
7	0.34	4.31	7.40
8	0.28	4.31	7.17
9	⋯	⋯	⋯

$SINR^{received}_{u,f}$

	1	2	3
1	0.51	31.67	199.85
2	0.51	31.67	199.85
3	0.51	31.67	199.85
4	0.39	5.02	10.00
5	0.39	5.02	10.00
6	0.39	5.02	10.00
7	0.39	5.02	10.00
8	0.39	5.02	10.00
9	⋯	⋯	⋯

$SINR^{reported}_{u,f}$

	1	2	3
1	0.59	5.03	7.65
2	0.59	5.03	7.65
3	0.59	5.03	7.65
4	0.48	2.59	3.46
5	0.48	2.59	3.46
6	0.48	2.59	3.46
7	0.48	2.59	3.46
8	0.48	2.59	3.46
9	⋯	⋯	⋯

$Q^{reported}_{u,f}$

	1	2	3
1	6.59	1.49	0.99
2	6.59	1.49	0.99
3	6.59	1.49	0.99
4	6.55	0.57	0.23
5	6.55	0.57	0.23
6	6.55	0.57	0.23
7	6.55	0.57	0.23
8	6.55	0.57	0.23
9	⋯	⋯	⋯

$model^{\mathrm{GP}}_{u,f}$

	1	2	3
1	0.73	0.16	0.11
2	0.73	0.16	0.11
3	0.73	0.16	0.11
4	0.89	0.08	0.03
5	0.89	0.08	0.03
6	0.89	0.08	0.03
7	0.89	0.08	0.03
8	0.89	0.08	0.03
9	⋯	⋯	⋯

$sch^{\mathrm{GP}}_{u,f}$

Fig. 3. SC s_2 from Figure 1b computes a schedule for frame $\mathcal{F}_t$ based on measurement reports from the previous frame $\mathcal{F}_{t-1}$; $sch^{\mathrm{GP}}_{u,f}$ specifies how the bandwidth will be allocated among the UEs attached to s_2 $\left(\text{i.e. } u \in \mathcal{A}_{s_2}\right)$ during each subframe f of the next frame $\mathcal{F}_t$. Rows and columns represent subframes and UEs respectively. UEs are sorted based on their average channel quality over $\mathcal{F}_{t-1}$ and then assigned dummy IDs $u' \in [1,2,\ldots,|\mathcal{A}_{s_2}|]$.

- **Step 4:** Statistical features are computed over the set of wideband channel quality reports. Hence, an evolved model maps these features to the schedule that s_2 will execute in frame $\mathcal{F}_t$. The first two columns of Table II define the features $T_1, T_2, \ldots, T_{13}$ that GP can utilize. Equation 7 displays a simple model evolved by GP that uses only three features. The process by which GP arrives at an expression like Equation 7 will be outlined in Section IV-B3. The model is executed $\forall(u,f) \in \mathcal{A}_{s_2} \times [1,\ldots,8]$ yielding $model^{\mathrm{GP}}_{u,f} \in \mathbb{R}$. For instance, the output for UE 2 in subframe 4 is given by

$$\begin{aligned} model^{\mathrm{GP}}_{2,4} &= (T_{10} \,\%\, T_4) \,\%\, (T_4 - T_1) \\ &= (7.65 \,\%\, 5.03) \,\%\, (5.03 - 2.59) \\ &\approx 0.57, \end{aligned} \tag{7}$$

where the operators are defined in Table II. The color coding in Equation 7 and Figure 3 highlight those channel quality reports that are involved in computing $model^{\mathrm{GP}}_{2,4}$. For instance, $T_4 = 5.03$ is the maximum channel quality reported by UE 2 over frame $\mathcal{F}_{t-1}$.

- **Step 5:** The model's outputs are scaled:

$$sch^{\mathrm{GP}}_{u,f} \leftarrow \left|model^{\mathrm{GP}}_{u,f}\right| \Big/ \sum_{u \in \mathcal{A}_{s_2}} \left|model^{\mathrm{GP}}_{u,f}\right|.$$

Scaling ensures that constraints (1) and (2) from the non-linear program in Section II-D are respected.

- **Step 6:** The partial schedule for subframes $f \in [1,\ldots,8]$ is duplicated fourfold to yield the complete schedule that s_2 will execute in $f \in [1,\ldots,40]$ of $\mathcal{F}_t$. It suffices to compute a partial schedule because the ABS patterns in Table I repeat in blocks of eight subframes.
- **Step 7:** Finally, s_2 uses the schedule to decide how much bandwidth UEs $u \in \mathcal{A}_{s_2}$ will receive in each subframe. In the LTE standard, a 20 [MHz] block of bandwidth is partitioned into 100 separate channels. The variable $sch^{\mathrm{GP}}_{u,f}$ is interpreted as the number of channels allocated to UE u in subframe f. Channels are then allocated to UEs on a first come first served basis. For example consider the schedule that is displayed in Figure 3. In subframe $f = 1$, the first 73 channels are allocated to UE 1, the next 16 channels are allocated to UE 2, and the remaining 11 channels are allocated to UE 3. Orthogonal frequency division multiplexing (OFDM) ensures that there is negligible intra-cell interference in LTE HetNets, since orthogonal channels are allocated to a cell's attached UEs. However, each channel is subject to interference from neighboring cells, if they re-use the same channel. Evolved schedulers allocate all 100 channels in every subframe. Therefore in any subframe f, a UE will experience interference on their allocated channels from all other SCs and MCs (if they are not executing an ABS during f) in the HetNet.

TABLE II

INTERPRETATION OF GRAMMAR ELEMENTS

Feature	Interpretation	Operator	Interpretation
T_1	$Q^{reported}_{u,f}$	$(x+y)$	$x+y$
T_2, T_3, T_4	avg, min, max $\left\{Q^{reported}_{u,*}\right\}$	$(x-y)$	$x-y$
T_5, T_6, T_7	avg, min, max $\left\{Q^{reported}_{*,f}\right\}$	$(x \times y)$	$x \times y$
T_8, T_9, T_{10}	avg, min, max $\left\{Q^{reported}_{*,*}\right\}$	$(x\%y)$	$\frac{x}{\sqrt{1+y^2}}$
T_{11}	$u' \in [1,2,\ldots,\|\mathcal{A}_s\|]$	$plog(x)$	$\log(1+\|x\|)$
T_{12}	$f \in [1,2,\ldots,8]$	$sine(x)$	$\sin(x)$
T_{13}	$ABS^{pattern}_{m_{int},f} \in \{0,1\}$	$psqrt(x)$	$\sqrt{\|x\|}$

A schedule for a MC $m \in \mathcal{M}$ is computed in a similar fashion. Now, T_{11} and T_{13} from Table II are interpreted as $u' \in [1,2,\ldots,|\mathcal{A}_m|]$ and $ABS^{pattern}_{m,f}$ respectively, and $sch^{\mathrm{GP}}_{u,f} \leftarrow 0.00$ during subframes in which m mutes.

The computations in Steps 3–7 are executed locally on a cell, based on the wideband reports collected from UEs. It typically takes under 1 [ms] to compute the schedule, which is much less than the 40 [ms] duration of a single frame.

2) Fitness Evaluation: A schedule's quality or 'fitness' is given by

$$f_{sch}\left(\left\{\overrightarrow{R}_u | u \in \mathcal{A}_c\right\}, \gamma\right) \leftarrow \sum_{u'=1}^{|\mathcal{A}_c|} \left(\log_e\left(\overrightarrow{R}_{u'}\right)\right) \times e^{\frac{\gamma \times u'}{|\mathcal{A}_c|}}, \tag{8}$$

where $\mathcal{A}_c$ is the set of UEs attached to cell c, $\overrightarrow{R}$ is an ordered array storing the downlink rates $\forall u \in \mathcal{A}_c$, such that $\overrightarrow{R}_1 \geq \overrightarrow{R}_2 \geq \ldots \geq \overrightarrow{R}_{|\mathcal{A}_c|}$, and $\gamma \in \mathbb{R}$ is a parameter specified by the operator for controlling fairness. The fitness

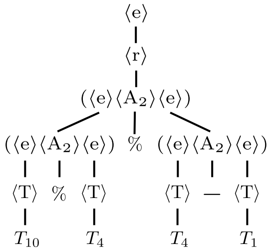
$$
\underset{Q_{u,f}^{received}}{
\begin{array}{c|c|c|c}
 & \multicolumn{3}{c}{u' \in [1,2,\ldots,|\mathcal{A}_{s_2}|]} \\
 & 1 & 2 & 3 \\
1 & 0.66 & 5.26 & 9.43 \\
2 & 0.62 & 5.26 & 9.41 \\
3 & 0.60 & 5.04 & 8.87 \\
4 & 0.52 & 2.86 & 3.77 \\
5 & 0.49 & 2.74 & 3.53 \\
6 & 0.47 & 2.63 & 3.33 \\
7 & 0.42 & 2.41 & 3.07 \\
8 & \boxed{0.36} & 2.41 & 3.03
\end{array}}
\;\times\;
\underset{sch_{u,f}^{\mathrm{GP}}}{
\begin{array}{c|c|c}
\multicolumn{3}{c}{u' \in [1,2,\ldots,|\mathcal{A}_{s_2}|]} \\
1 & 2 & 3 \\
0.73 & 0.16 & 0.11 \\
0.73 & 0.16 & 0.11 \\
0.73 & 0.16 & 0.11 \\
0.89 & 0.08 & 0.03 \\
0.89 & 0.08 & 0.03 \\
0.89 & 0.08 & 0.03 \\
0.89 & 0.08 & 0.03 \\
\boxed{0.89} & 0.08 & 0.03
\end{array}}
\times 20\ [\mathrm{MHz}] =
\underset{R_{u,f}^{\mathrm{GP}}\ [\mathrm{Mbps}]}{
\begin{array}{c|c|c}
\multicolumn{3}{c}{u' \in [1,2,\ldots,|\mathcal{A}_{s_2}|]} \\
1 & 2 & 3 \\
9.14 & 16.43 & 19.65 \\
8.58 & 16.49 & 19.62 \\
8.31 & 15.81 & 18.49 \\
8.84 & 4.20 & 2.25 \\
8.33 & 4.02 & 2.11 \\
7.99 & 3.86 & 1.99 \\
7.14 & 3.54 & 1.84 \\
\boxed{0.00} & 3.54 & 1.81
\end{array}}
$$

$f \in \mathcal{F}_t$

Fig. 4. The schedule for SC s_2 is evaluated by computing the downlink rates $R_{u,f}$ for $u \in \mathcal{A}_{s_2}$ using Shannon's formula, and hence calling Equation 8. The boxed cells indicate that UE 1 is scheduled in subframe 8 because the reported $SINR$ exceeds -5 [dB] (≈ 0.32). However, the $SINR$ actually received is less than -5 [dB] so packets are in fact dropped and $R_{1,8}^{\mathrm{GP}} = 0.00$.

$\langle \mathrm{e} \rangle\ ::= \langle \mathrm{r} \rangle \mid \langle \mathrm{r} \rangle \mid \langle \mathrm{r} \rangle \mid \langle \mathrm{T} \rangle$
$\langle \mathrm{r} \rangle\ ::= \langle \mathrm{A}_1 \rangle(\langle \mathrm{e} \rangle) \mid (\langle \mathrm{e} \rangle\langle \mathrm{A}_2 \rangle\langle \mathrm{e} \rangle)$
$\langle \mathrm{A}_1 \rangle ::= plog \mid sine \mid psqrt$
$\langle \mathrm{A}_2 \rangle ::= + \mid - \mid \times \mid \%$
$\langle \mathrm{T} \rangle\ ::= T_1 \mid T_2 \mid , \ldots, \mid T_{13} \mid \langle n \rangle \mid \langle n \rangle$
$\langle \mathrm{n} \rangle\ ::= -1.0 \mid -0.9 \mid , \ldots, \mid 1.0$

(a)



(b)

Fig. 5. The grammar (a) for evolving schedulers, and the derivation tree of a GP individual (b) representing the model: $(T_{10}\%T_4)\%(T_4 - T_1)$.

TABLE III

DOWNLINK RATES UNDER BL AND GP SCHEDULING

Method	$\overrightarrow{R}_1$ [Mbps]	$\overrightarrow{R}_2$ [Mbps]	$\overrightarrow{R}_3$ [Mbps]
BL	3.00	34.08	52.98
GP	7.29	8.49	8.47

functions described by Equations 6 and 8 are identical, except the summation is over $\mathcal{U}$ in the former and $\mathcal{A}_c$ in the latter.

For example, Figure 4 illustrates how Shannon's formula yields the downlink rates received by the UEs $u \in \mathcal{A}_{s_2}$ when the schedule from Figure 3 is executed by s_2. Similar calculations yield rates received under baseline scheduling. Table III shows how the baseline represents an unfair allocation of the bandwidth since $\overrightarrow{R}_3^{\mathrm{BL}} > \overrightarrow{R}_2^{\mathrm{BL}} \gg \overrightarrow{R}_1^{\mathrm{BL}}$. In contrast, GP improves fairness by granting cell edge UE 1 more bandwidth than cell center UEs 2 and 3. Hence, the downlink rate for UE 1 more than doubles relative to baseline and $\overrightarrow{R}_1^{\mathrm{GP}} \approx \overrightarrow{R}_2^{\mathrm{GP}} \approx \overrightarrow{R}_3^{\mathrm{GP}}$. Fairness ensures that all three UEs experience an acceptable quality of service.

The parameter γ controls the sensitivity of Equation 8 to the trade-off between cell edge and cell center performance. For instance, $f_{sch}^{\mathrm{GP}} = 304.1 > f_{sch}^{\mathrm{BL}} = 296.5$ when $\gamma = 4.0$, reflecting the fact that the model from Figure 5 produces a fairer schedule than the baseline. Positive values for γ make Equation 8 sensitive to cell edge performance. Conversely, $\gamma < 0$ encourages the evolution of models that increase peak rates at the expense of fairness.

3) The Genetic Programming Search Loop: A form of evolutionary computation [30] called Grammar-based Genetic Programming (GP) [8], [9], [34], [35] is employed to automatically construct schedulers. GP mimics natural evolution by evolving a population of individuals (i.e. schedulers) over a fixed number of generations. A Backus-Naur-Form grammar defines the space of solutions that can be explored by GP. For example, Figure 5b shows how the model $(T_{10} \,\%\, T_4) \,\%\, (T_4 - T_1)$ from Section IV-B1 is constructed by expanding the start symbol $\langle e \rangle$ via a sequence of randomly selected production rules from the grammar in Figure 5a. The derivation tree is decoded yielding a symbolic expression that can be evaluated in simulation. A GP run proceeds as follows:

- **Step 1:** A population of 1000 randomly generated individuals is initialized. The following steps are then iterated for 250 generations.
- **Step 2:** Each individual in the current population is assigned a fitness value by computing the average of Equation 8 over 250 training cases (i.e. matrix tuples $(Q^{reported}, Q^{received})$). Figure 4 shows how a schedule is evaluated on a single case. Hence, 990 individuals are selected for recombination using tournament selection.
- **Step 3:** In the recombination step, selected individuals are randomly grouped into 495 pairs. Subtree crossover is applied with a probability of 0.7 to each pair. A random subtree is identified in each parent, and then these subtrees are swapped. The two resulting children contain genetic material from both parents.
- **Step 4:** The 990 children resulting from crossover are all subjected to mutation. Subtree mutation replaces the subtree rooted at a randomly identified non-terminal in an individual's derivation tree, with a new randomly generated subtree.
- **Step 5:** All but the 10 fittest individuals (elites) in the current population are replaced with the 990 children resulting from crossover and mutation.
- **Step 6:** The fittest individual at generation 250 is returned, and the algorithm terminates.

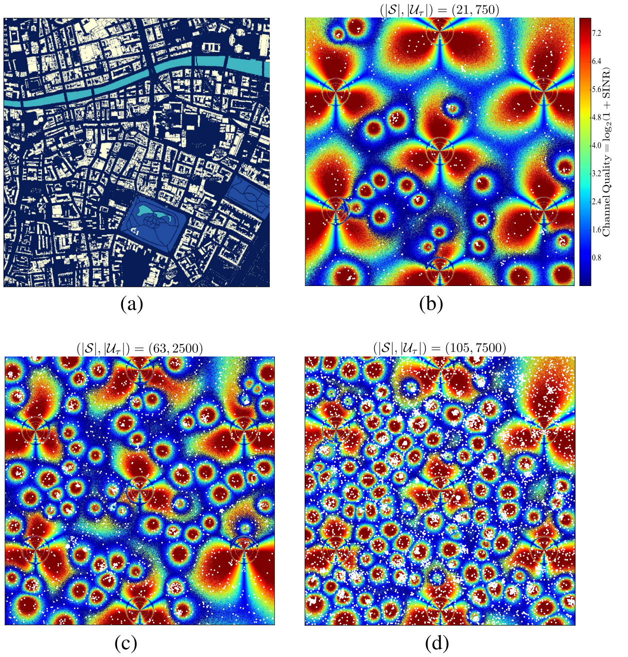
4) Logistic Model: It is hypothesized that GP constructs useful high-order feature representations. GP's ability to discover hierarchical structure is assessed by comparing it with a logistic model (LM) of the form:

$$model_{u,f}^{\mathrm{LM}} = \frac{1}{1 + e^{-(\alpha_1 \times T_1 + \alpha_2 \times T_2 +, \ldots, + \alpha_{13} \times T_{13})}},$$

where $\alpha_1, \alpha_2, \ldots, \alpha_{13} \in \mathbb{R}$ are optimized by the Covariance Matrix Adaptation Evolution Strategy [36].

5) Baseline: GP is also compared to a baseline (BL) round robin scheduler that splits the bandwidth evenly among the UEs attached to cell c ($u \in \mathcal{A}_c$), such that $sch_{u,f}^{\mathrm{BL}} = 1/|\mathcal{A}_c|$ in all subframes. Our interference model simulates channel fading in the time domain but not the frequency domain, precluding comparisons with a proportional fair scheduler. Industrial schedulers are complicated parametrized versions of proportional fairness with very complex decision-making. The simplification allows us to focus on the concept of using GP as a tool for automating the scheduler design process.

6) Benchmark Algorithm: Finally, evolved schedulers are benchmarked against a greedy algorithm adapted from [27]. The benchmark splits a SC's attached UEs into two queues overlapping and non-overlapping with the ABSs of the most



(a) (b)

(c) (d)

Fig. 6. Three different HetNet deployments were simulated in Dublin City. Each network contained 21 MCs (visible as large tri-sector lobes), but the number of SCs ($|\mathcal{S}|$) was varied. UEs are indicated by white dots. The number of UEs simulated in each frame ($|\mathcal{U}_\tau|$) was larger in the denser topologies. (a) Downtown dublin. (b) Sparse deployment. (c) Standard deployment. (d) Dense deployment.

strongly interfering MC of the SC. UEs are iteratively exchanged between queues in an attempt to equalize the downlink rates of the worst performers in each queue type. Intuitively, the benchmark sacrifices the best performing UEs by unscheduling them in muted subframes, so that cell edge UEs receive extra bandwidth during these quieter intervals. The algorithm proposed in [27] is a suitable benchmark since it captures the intuition underlying proportional fair scheduling.

V. EXPERIMENTS

Simulations were carried out in three different HetNet topologies in order to assess:

- adaptive hill climbing (see Figure 2) for optimizing the HetNet configuration $\mathscr{C}$ versus static baseline settings,
- performance gains from jointly optimizing $\mathscr{C}$, and schedules on MCs and SCs,
- schedulers evolved using GP versus strong baselines and a state of the art benchmark algorithm, and
- how the trade-off between peak rates and fairness can be controlled by adjusting γ.

A. Simulation Environment

Figure 6a displays the 3.24 square kilometer region of Dublin City Center in which three different topologies were simulated: a sparse HetNet with $|\mathcal{M}| = 21$ MCs and $|\mathcal{S}| = 21$ SCs serving 750 UEs per frame (Figure 6b), a standard topology with $(|\mathcal{M}|, |\mathcal{S}|) = (21, 63)$ serving 2500 UEs per frame (Figure 6c), and a dense topology with $(|\mathcal{M}|, |\mathcal{S}|) = (21, 105)$ serving 7500 UEs per frame (Figure 6d). SCs were placed at random locations on the map reflecting their ad-hoc installation in real networks near traffic hotspots. MCs were arranged on a regular hexagonal grid. Realistic channel gains were computed by modeling the distribution of buildings, streets, parks, and waterways. The parameters of the path loss model are available in the supplementary materials. Our goal is to validate the proposed zero-touch automation techniques using only the core features of LTE HetNets.

The traffic model was designed to simulate the two main properties of wireless demand in real HetNets. Firstly, UEs request data from unpredictable locations in any given frame. Secondly, hotspots materialize, move around, and dissipate over time. Periods of activity hereafter referred to as ‘snapshots’ were simulated. A snapshot consisted of 200 frames indexed by $\tau \in [1, 2, \ldots, 200]$ that were sampled over thirty minutes of activity in the HetNet. $|\mathcal{S}|$ hotspots were placed on the map at the beginning of a snapshot. Half of the hotspots were placed within 20 meters of randomly selected SCs, and half were placed at random locations. The physical size, load, and location of a hotspot HS was varied:

- The radius of HS was chosen randomly from the range [10 [m], 40 [m]] at $\tau = 1$ and $\tau = 200$, and it was varied linearly between these values during the thirty minute snapshot.
- $\mathcal{N}(\mu_{HS}(\tau), 2)$ UEs were dropped into hotspot HS in frame τ, where $\mathcal{N}$ is the normal distribution with mean $\mu_{HS}(\tau)$ and standard deviation $\sigma = 2$. In the sparse HetNet, μ_{HS} was chosen randomly from the range $[5, 15]$ at $\tau = 1$ and $\tau = 200$, and it varied linearly between these values during the snapshot. Similarly, μ_{HS} varied linearly between $[5, 30]$ and $[5, 65]$ over a snapshot in the standard and dense HetNets respectively. Thus, denser HetNets contained more congested hotspots.
- Finally, HS moved between 0 [m] and 50 [m] from its initial location in a random direction over a snapshot.

Hotspots were populated at the beginning of each frame. UEs were then dropped at random until 750, 2500 and 7500 existed on the map in the sparse, standard and dense HetNets respectively. UEs were not permitted within 100 [m] of the map’s edge where interference may be artificially low.

B. Evolutionary Learning of Schedulers

GP was employed to evolve tailored schedulers for MCs and SCs in the deployment scenarios displayed in Figure 6. Runs using $\gamma = 4$ and $\gamma = -4$ were executed in each scenario to show how fairness and peak rates are improved across a range of HetNet topologies and traffic scenarios. Schedulers were evolved for $\gamma \in [-4, -3, \ldots, 4]$ in the standard deployment in order to illustrate how fairness trade-offs are controlled. A scheduler was evolved for a given scenario as follows:

- **Step 1:** A value for γ was specified, and $\mathscr{C}_{\text{HC}}$ was optimized for a large number of different snapshots.
- **Step 2:** A training set for GP was generated by sampling training cases from randomly selected cells in the optimized HetNets. A single training case was the matrix tuple $\left(Q^{reported}, Q^{received}\right)$ from a SC (if evolving a SC scheduler), or from a MC (if evolving a MC scheduler). Training cases were extracted from optimized HetNets

because evolved schedulers would be executed online in pre-configured HetNets.

- **Step 3:** GP is a stochastic metaheuristic, so any single run may converge to a poor local optimum. Therefore, thirty independent runs of GP were executed. Individuals were evaluated by computing the average of Equation 8 over all 250 training cases in the training set. Recall section IV-B2 which described how an individual was evaluated on a single training case.
- **Step 4:** The best model from the final generation of all 30 runs was saved. These highly fit solutions were then evaluated on an unseen test set consisting of 250 test cases in order to identify the single best model overall.

The logistic models (LMs) presented in Section IV-B4 were trained by using CMA-ES [36] to optimize $\alpha_1, \alpha_2, \ldots, \alpha_{13}$. Identical training and validation sets were used in the runs with GP and CMA-ES. The experimental parameters for all evolutionary runs are available in the supplementary materials.

VI. RESULTS AND DISCUSSION

This section first illustrates how the hill climbing algorithm outperforms static baseline settings by adapting a HetNet's configuration ($\mathscr{C}$) with respect to time-varying traffic patterns. Evolved schedulers are benchmarked against a state of the art algorithm [27], a strong logistic model, and a round robin baseline scheduling method. The ability to control fairness by tuning γ in Equations 6 and 8 is then assessed. Finally, a semantic analysis reveals the intuitive strategies that GP discovers for allocating bandwidth in the time-frequency domain.

A. Optimizing the Network Configuration

The hill climbing algorithm (Figure 2) was used to optimize $\mathscr{C}$ in the sparse, standard and dense topologies. Pilot experiments indicated that the best results are achieved when the probability of accepting worse moves (p_accept) is set to zero. That is, greedy hill climbing gives the best performance. Figure 7 displays the convergence of average fitness over 30 scenarios for the cases $\gamma = 4$ (solid lines, maximizing fairness) and $\gamma = -4$ (dashed lines, maximizing peak rates). At each parameter update, fitness is given by the ratio of Equation 6 evaluated for the optimized configuration $\mathscr{C}_{\mathrm{HC}}$, and the static baseline settings $\mathscr{C}_{\mathrm{BL}}$ (red dashed line).

The abscissa indicates the iteration number on line 4 of the algorithm in Figure 2, such that time increases from left to right. An initial training set is formed by sampling 25 channel gain matrices during iterations -3 to 0 (green shaded region). The algorithm begins optimizing $\mathscr{C}_{\mathrm{HC}}$ at iteration 0. Performance rapidly improves compared to baseline during iterations 0–3 as the SC powers and biases, and MC ABS ratios are optimized. The training set is replaced at every third iteration with 25 new channel gain matrices, which are sampled during the preceding three iterations. Thus, $\mathscr{C}_{\mathrm{HC}}$ is continuously adapted with respect to the most recent traffic conditions. Sharp drops in fitness are visible when the training set is updated, but $\mathscr{C}_{\mathrm{HC}}$ quickly adapts to the new reports.

The red shaded region in Figure 7 is a warm up period during which the algorithm stabilizes. Iterations 12–24 in gray shading are reserved as a test period in which the scheduling techniques described in Section IV are assessed. Schedulers are evaluated by executing them in 100 frames ($\tau \in [100, 101, \ldots, 200]$) that are sampled uniformly over the test period. Hence, downlink rates are computed for UEs that receive data in the optimized and unoptimized HetNets, under the various scheduling regimes. The following sections analyze cumulative distribution functions of received downlink rates in the various experimental set-ups.

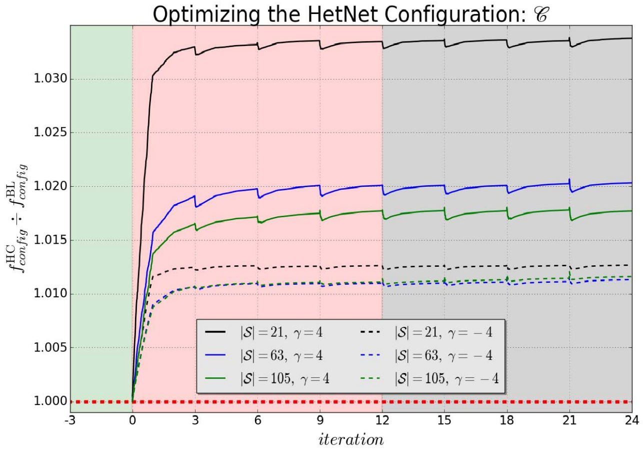


Fig. 7. Average fitness over 30 different scenarios in three different HetNet topologies for the cases $\gamma = 4$ (solid lines) and $\gamma = -4$ (dashed lines).

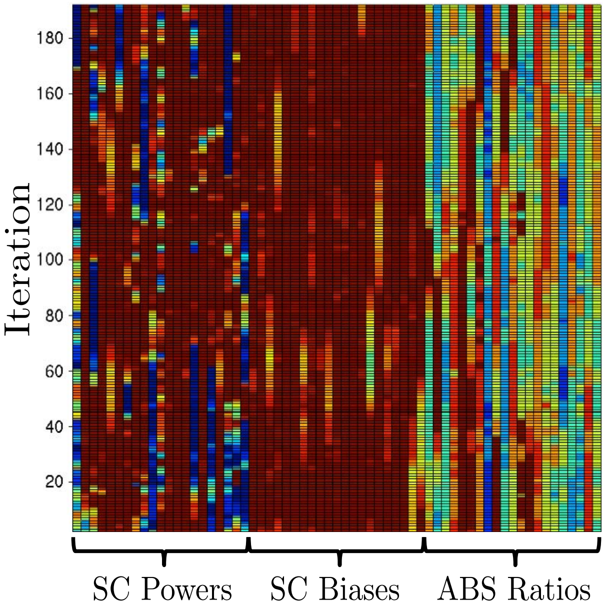
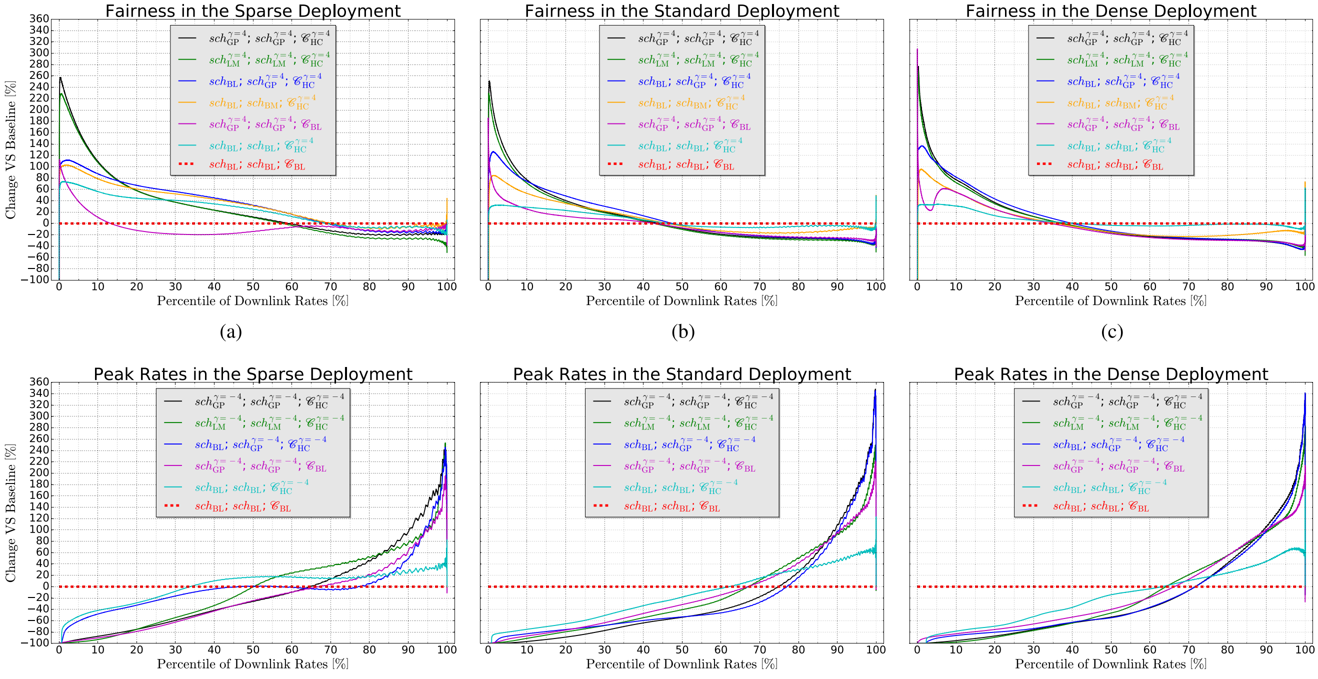


Fig. 8. Adaptation of $\mathscr{C}_{\mathrm{HC}}$ over 192 iterations in the sparse HetNet when maximizing fairness ($\gamma = 4$). Three large hotspots containing 150 UEs move smoothly across the map, and 550 UEs are dropped randomly in every frame.

Figure 8 visualizes how SC powers and biases, and the MC ABS ratios are adapted during an extended snapshot (192 iterations instead of 24) in the sparse deployment (Figure 6b). Columns correspond to the elements of $\mathscr{C}_{\mathrm{HC}}$. For example, the first column indicates the power of SC 1 – deep red implies P_1 is at its maximum value of 35 [dBm], and deep blue implies P_1 takes the minimum allowed value of 23 [dBm]. Time increases vertically along the ordinate. The red hue in columns 22–42 confirms that SCs implement positive cell


Fairness in the Sparse Deployment

Fairness in the Standard Deployment

Fairness in the Dense Deployment

Peak Rates in the Sparse Deployment

Peak Rates in the Standard Deployment

Peak Rates in the Dense Deployment

Fig. 9. Optimizing the different HetNet layers in a sparse, standard, and dense deployment scenario. The legends indicate the scheduling method employed on MCs and SCs respectively, and the method used to configure the HetNet. Fairness (i.e. cell edge throughput) is optimized in plots (a)–(c) by setting $\gamma = 4$ in the fitness functions, whereas peak rates are optimized in plots (d)–(f) by setting $\gamma = -4$. GP schedulers (black CDF) achieve higher cell edge and peak rates than logistic models (green CDF). Cell edge rates are significantly improved by scheduling both MC and SC attached UEs (black VS blue CDFs).

TABLE IV

RANGE EXPANSION AND INTERFERENCE MITIGATION

	$\lvert\mathcal{S}\rvert = 21$	$\lvert\mathcal{S}\rvert = 63$	$\lvert\mathcal{S}\rvert = 105$
Mean of SC Biases [dB]	13.44	12.90	11.09
Mode of MC ABS ratios	30/40	20/40	30/40

selection biases. The light palette in columns 43–63 implies that MCs mute periodically, by implementing ABS ratios less than $40/40$. Therefore, both range expansion and interference mitigation are employed to improve fairness. All settings vary throughout the snapshot as hotspots move around on the map. SC powers are particularly variable. SCs are sometimes powered down to 23 [dBm] in order to reduce intra-tier and cross-tier interference to nearby loaded SCs and MCs respectively.

Table IV displays the central tendencies of SC biases and MC ABS ratios in the sparse, standard and dense deployments. Selection biases are larger in less densified HetNets because there are fewer SCs to offload UEs from the MC tier.

B. Multi-Layer Optimization

This section isolates the contribution to overall performance from optimizing individual layers of a HetNet. The baseline (BL), evolved (GP), logistic (LM) and benchmark (BM) scheduling techniques are compared. Simulations were carried out in the sparse, standard and dense topologies from Figure 6. We restrict our attention to the cases when $\gamma = 4$ (maximizing fairness) and $\gamma = -4$ (maximizing peak rates). Schedulers were executed in the HetNets after cells had been configured using either the optimized settings $(\mathscr{C}_{\text{HC}})$, or baseline settings $(\mathscr{C}_{\text{BL}})$. Hence, downlink rates $\left(\overline{R}_u\right)$ were computed for all UEs that were active in 100 frames sampled over the test periods (gray region in Figure 7) of 30 different snapshots.

1) Cumulative Distribution Functions (CDFs): Figures 9a, 9b and 9c display CDFs of the downlink rates for various set-ups relative to baseline (red dashed line), in the sparse, standard and dense scenarios respectively, when fairness is prioritized. The percentage increase of downlink rates versus baseline are plotted (ordinate) for all percentiles of $\overline{R}_u$ (abscissa). The baseline implements baseline scheduling on MCs and SCs (denoted by sch_{BL} in the legend) and the static baseline configuration $(\mathscr{C}_{\text{BL}})$.

Cell edge rates (LHS of the plot) are improved relative to baseline in all scenarios by optimizing only the network configuration (cyan). Intelligent scheduling on SCs augments the effect of optimizing the HetNet configuration (blue), and further gains are observed when schedules on MCs are also optimized (black). Joint optimization of the HetNet configuration, and schedules on MCs and SCs, boosts far cell edge rates by over 255%, 250% and 260% in the sparse, standard and dense HetNets respectively. Figures 9d, 9e and 9f display CDFs for the experiments using $\gamma = -4$. Here, peak rates (RHS of the plot) are increased by over 240%, 300% and 340% in the sparse, standard and dense scenarios under joint optimization of schedules and $\mathscr{C}_{\text{HC}}$. Notice that peak rates are increased by sacrificing cell edge performance, and vice versa.

In all cases, intelligent scheduling in optimized HetNets (black) dramatically outperforms; 1) baseline scheduling in optimized HetNets (cyan), and 2) intelligent scheduling in unoptimized HetNets (magenta). This result underscores the need to optimize both the HetNet configuration and schedules. Schedulers evolved using GP (black) outperform the logistic

TABLE V

1st PERCENTILE OF DOWNLINK RATES [Mbps]

$sch_{*}^{\gamma}, \forall m \in \mathcal{M}; sch_{*}^{\gamma}, \forall s \in \mathcal{S}; \mathscr{C}_{*}^{\gamma}$	Figure 9a	Figure 9b	Figure 9c
$sch_{\mathrm{GP}}^{\gamma=4}; sch_{\mathrm{GP}}^{\gamma=4}; \mathscr{C}_{\mathrm{HC}}^{\gamma=4}$	0.83 ± 0.10	0.53 ± 0.06	0.26 ± 0.04
$sch_{\mathrm{LM}}^{\gamma=4}; sch_{\mathrm{LM}}^{\gamma=4}; \mathscr{C}_{\mathrm{HC}}^{\gamma=4}$	0.78 ± 0.09	0.49 ± 0.05	0.24 ± 0.04
$sch_{\mathrm{BL}}; sch_{\mathrm{GP}}^{\gamma=4}; \mathscr{C}_{\mathrm{HC}}^{\gamma=4}$	0.49 ± 0.05	0.37 ± 0.04	0.18 ± 0.03
$sch_{\mathrm{BL}}; sch_{\mathrm{BM}}; \mathscr{C}_{\mathrm{HC}}^{\gamma=4}$	0.47 ± 0.04	0.30 ± 0.03	0.15 ± 0.02
$sch_{\mathrm{GP}}^{\gamma=4}; sch_{\mathrm{GP}}^{\gamma=4}; \mathscr{C}_{\mathrm{BL}}$	0.49 ± 0.09	0.32 ± 0.06	0.13 ± 0.03
$sch_{\mathrm{BL}}; sch_{\mathrm{BL}}; \mathscr{C}_{\mathrm{HC}}^{\gamma=4}$	0.41 ± 0.03	0.22 ± 0.02	0.10 ± 0.01
$sch_{\mathrm{BL}}; sch_{\mathrm{BL}}; \mathscr{C}_{\mathrm{BL}}$	0.24 ± 0.03	0.17 ± 0.02	0.08 ± 0.01

TABLE VI

99th PERCENTILE OF DOWNLINK RATES [Mbps]

$sch_{*}^{\gamma}, \forall m \in \mathcal{M}; sch_{*}^{\gamma}, \forall s \in \mathcal{S}; \mathscr{C}_{*}^{\gamma}$	Figure 9d	Figure 9e	Figure 9f
$sch_{\mathrm{GP}}^{\gamma=-4}; sch_{\mathrm{GP}}^{\gamma=-4}; \mathscr{C}_{\mathrm{HC}}^{\gamma=-4}$	62.60 ± 4.36	42.13 ± 2.07	21.18 ± 0.75
$sch_{\mathrm{LM}}^{\gamma=-4}; sch_{\mathrm{LM}}^{\gamma=-4}; \mathscr{C}_{\mathrm{HC}}^{\gamma=-4}$	60.18 ± 4.18	35.78 ± 1.51	18.81 ± 0.79
$sch_{\mathrm{BL}}; sch_{\mathrm{GP}}^{\gamma=-4}; \mathscr{C}_{\mathrm{HC}}^{\gamma=-4}$	59.16 ± 4.88	41.33 ± 2.05	21.00 ± 0.86
$sch_{\mathrm{GP}}^{\gamma=-4}; sch_{\mathrm{GP}}^{\gamma=-4}; \mathscr{C}_{\mathrm{BL}}$	50.69 ± 4.14	31.73 ± 0.96	15.67 ± 0.69
$sch_{\mathrm{BL}}; sch_{\mathrm{BL}}; \mathscr{C}_{\mathrm{HC}}^{\gamma=-4}$	26.87 ± 2.53	19.26 ± 0.79	10.00 ± 0.27
$sch_{\mathrm{BL}}; sch_{\mathrm{BL}}; \mathscr{C}_{\mathrm{BL}}$	19.44 ± 1.39	11.80 ± 0.74	6.11 ± 0.40

models (green) in all scenarios. Kolmogorov-Smirnov tests confirm that the black and green CDFs are significantly different at $\alpha = 0.01$. Therefore, GP builds useful high-order representations from minimal domain knowledge. Importantly, GP (black) also outperforms the BM (gold) by achieving higher cell edge rates in Figures 9a, 9b and 9c. The gains are larger in denser HetNets containing more SCs.

2) Statistical Tests: Tables V and VI display average (over 30 snapshots) percentiles of the downlink rates when cell edge and peak rates are maximized respectively. One-way analyses of variance (ANOVAs) across the columns of both tables indicate statistically significant differences between the groups means ($p = 0.000$ for all tests with $N = 30$, and 209 & 179 total degrees of freedom). Tukey's post-hoc tests, at a significance level of $\alpha = 0.05$, suggest that 1st percentile (in Table V) and 99th percentile (in Table VI) rates are statistically significantly increased over baseline (red) by optimizing only the HetNet configuration (cyan),[2] or only the schedules (magenta), or both (black).

In summary, the proposed methods scale well with cell density and traffic load. Downlink rates are significantly improved relative to a baseline that is used in practice. In fact, cell edge and peak rates are increased by over 240% in all simulated deployment scenarios by setting $\gamma = 4$ and $\gamma = -4$ respectively. GP outperforms less expressive LMs by tapping structure in the training data that cannot be captured by a linear combination of the features. Finally, GP is superior to the benchmark in terms of both performance and flexibility. Unlike the benchmark, GP generates tailored schedulers that address the operator's ad-hoc objectives for service differentiation.

C. Benchmarking the Hill Climbing Algorithm

The hill climbing algorithm for optimizing the HetNet configuration is benchmarked against self-organizing network algorithms (SONs) adapted from Tall *et al.* [23]. The benchmark consists of two components: a load balancing SON, and a SON for setting Macro Cell ABS ratios. The load balancing SON iteratively updates SC biases in order to manage congestion on MCs. For example, consider a MC m with an embedded SC s. Let $\hat{\rho}_m$ be the number of UEs currently attached to m, and let $\hat{\rho}_s^{expanded}$ denote the load in the expanded region of s. The bias of s is updated as follows:

$$\beta_s^{\text{new}} = \beta_s^{\text{old}} + \epsilon \times \left(\hat{\rho}_m - \hat{\rho}_s^{expanded}\right), \qquad (9)$$

where $\epsilon = 0.01$ controls the update size. Intuitively, β_s increases if there are more UEs attached to m (i.e. the MC is relatively congested), so that UEs are offloaded onto s. Load balancing is a scalable heuristic for increasing capacity because it reduces congestion on overloaded cells.

The ABS ratio of Macro Cell m is given by:

$$ABS_m^{ratio} = \frac{\hat{\rho}_m}{\hat{\rho}_m + \left(2 \times \hat{\rho}_m^{expanded}\right)}, \qquad (10)$$

where $\hat{\rho}_m$ is the average load on m, and $\hat{\rho}_m^{expanded}$ is the average number of UEs within the footprint of m, that are also in the expanded regions of embedded SCs. Intuitively, m is muted frequently (executes a lower ABS ratio) if there are a relatively large number of vulnerable UEs in SC expanded regions. The SONs described by Equations 9 and 10 are executed continuously on SCs and MCs respectively.

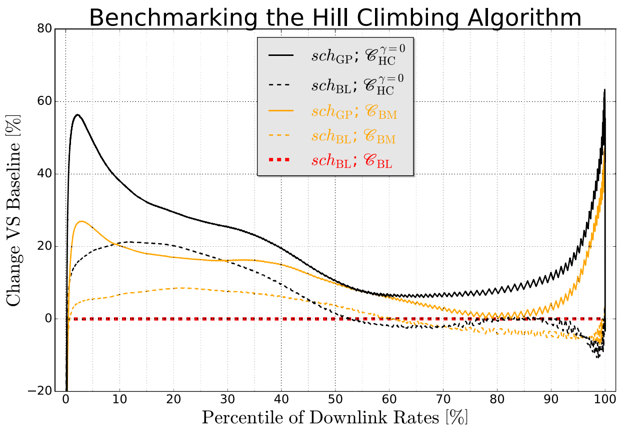


Fig. 10. The hill climbing algorithm is compared with benchmark SONs adapted from [23]. Downlink rates are computed in the standard scenario. The hill climber is executed using $\gamma = 0$ in the fitness function, since the benchmark is designed to maximize a proportionally fair objective.

Figure 10 displays the CDF plots that are achieved by the hill climbing algorithm (HC) and benchmark SONs (BM). Both techniques are afforded the same computational budget. The hill climbing algorithm (dashed black) and benchmark (dashed gold) achieve higher cell edge downlink rates than the non-adaptive baseline (dashed red), when baseline scheduling is employed. Downlink rates are further increased if schedules are optimized using evolved models (solid lines). The hill climbing algorithm outperforms the benchmark because it optimizes SC powers, in addition to SC biases and MC muting patterns. Unlike less flexible manually designed algorithms, the proposed heuristic can easily

---

[2] Except when maximizing fairness in the dense scenario where $p = 0.09$.

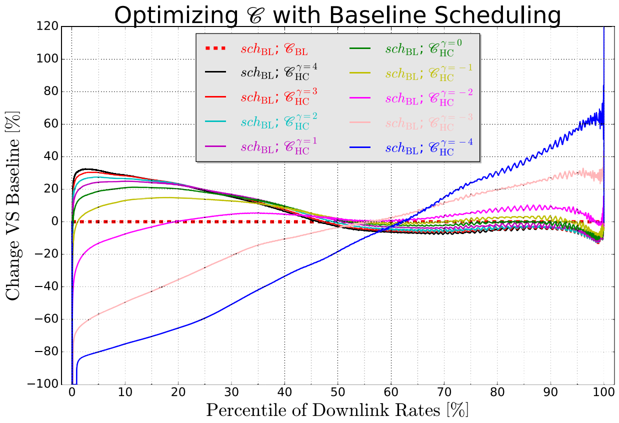


Fig. 11. Managing fairness by optimizing $\mathscr{C}_{\mathrm{HC}}$ but not schedules. The legend indicates the scheduling method that was employed on both MCs and SCs, and the method used to configure the HetNet.

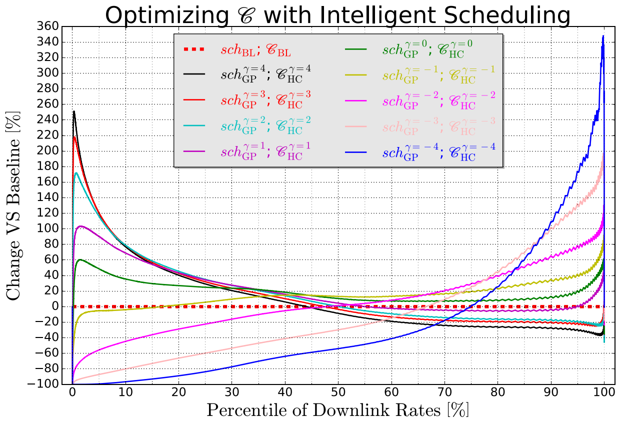


Fig. 12. Managing fairness by optimizing both $\mathscr{C}_{\mathrm{HC}}$ and schedules. The legend indicates the scheduling method that was employed on both MCs and SCs, and the method used to configure the HetNet.

incorporate additional parameter layers. For example, mobility could be managed by including a layer of time-to-trigger parameters.

D. Tuning Performance

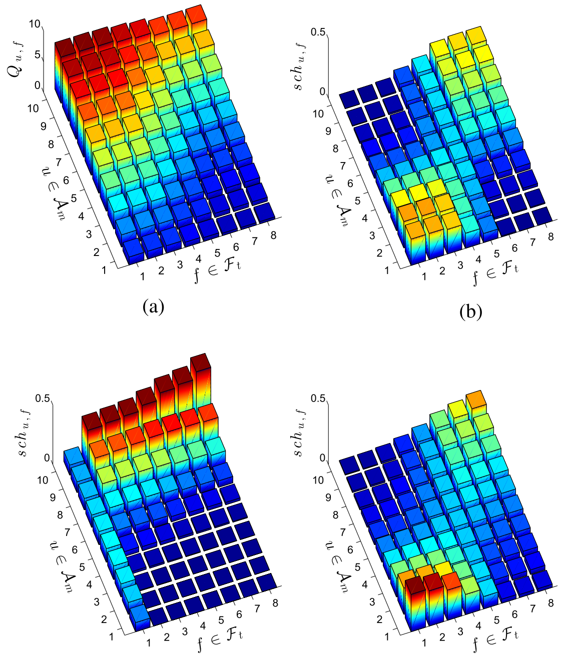
This section explores how fairness trade-offs are managed in the proposed framework. We first study the case where the HetNet configuration is optimized, but schedules are not optimized (baseline schedules are executed on MCs and SCs). The additional performance gains from intelligent scheduling is assessed in the second subsection. Results are displayed for simulations in the standard topology (Figure 6c).

1) Optimizing the HetNet Configuration With Baseline Scheduling: Figure 11 displays CDFs of the downlink rates for $\mathscr{C}_{\mathrm{HC}}$ versus $\mathscr{C}_{\mathrm{BL}}$, where each curve corresponds to a different setting for γ in Equation 6. As in the previous section, downlink rates $\left(\overline{R}_u\right)$ were computed for all UEs in 100 frames sampled over the test periods (gray region in Figure 7) of 30 different snapshots.

Fine grained control over fairness can be exercised by optimizing the HetNet configuration for different choices of γ. Setting $\gamma = 4$ improves cell edge performance by over 30% (black CDF). Conversely, setting $\gamma < 0$ lifts peak rates at the expense of fairness. Consider the blue CDF corresponding to the experiments with $\gamma = -4$. Here, cell edge rates are sacrificed so that peak rates increase by 60% versus baseline.

2) Optimizing the HetNet Configuration and Schedules: Figure 12 displays the CDFs realized after the HetNet is configured by optimizing $\mathscr{C}_{\mathrm{HC}}$ for $\gamma \in [-4, -3, \ldots, 4]$, and UEs are scheduled using models evolved by GP for the same values of γ in Equation 8. Far cell edge rates are increased by approximately 250% with $\gamma = 4$. Peak rates are increased by over 340% with $\gamma = -4$. Therefore, scheduling has a dramatic impact on performance in well configured HetNets.

Operators value the ability to control the trade-off between cell edge and peak rates. High peak rates are often advertised in order to attract customers, but cell edge performance must be properly managed to maintain an acceptable quality of service. The proposed techniques clearly enable flexible service differentiation.



(a)

(b)

(c)

(d)

Fig. 13. Expected channel qualities and schedules for SCs with ten attached UEs. Each barchart is the average of 500 instances sampled from cells in the standard scenarios after cells had been configured. UEs are ordered with respect to received channel quality. For instance, UE 1 experiences the lowest average channel quality of all $u \in \mathcal{A}_s$, whereas UE 10 is closer to the cell center. Only 8 out of 40 subframes are displayed for clarity. (a) Channel Quality on SCs. (b) BM on SCs. (c) GP on SCs with $\gamma = -4$. (d) GP on SCs with $\gamma = 4$.

E. Semantics

This section presents a semantic analysis of the benchmark (BM) and evolved (GP) schedulers. Figure 13a displays a three dimensional barchart indicating expected channel qualities $Q_{u,f}$ for UEs that attach to SCs serving exactly ten UEs. $Q_{u,f}$ is higher in the first few subframes wherein more interfering MCs mute (recall Table I).

Section VI-D showed how larger values of γ encourage better fairness, while setting $\gamma < 0$ increases peak rates. Figure 13c displays the strategy discovered by a scheduler evolved using $\gamma = -4$. Peak rates are increased by awarding high channel quality cell center UEs 7–10 most of the bandwidth.

Figure 13d illustrates how the strategy changes for $\gamma = 4$. Better fairness is achieved by heavily sacrificing cell center UEs during subframes 1–4. Thus, cell edge UEs are granted extra bandwidth when their channel quality is highest. Cell edge UEs are largely unscheduled during subframes 6–8, when their channel quality is lowest due to high MC interference. Hence, cell center UEs are compensated for their sacrifice of low-interference airtime during subframes 1–4.

Comparing Figures 13d and 13b we see that GP rediscovers the BM's essential strategy. Both methods implement a 'Robin Hood' policy whereby the best-off UEs are sacrificed in quieter subframes for the sake of fairness. Section VI-B established that GP outperforms the BM. GP's principle advantage rests in its ability to automatically discover a diverse range of tailored strategies.

VII. FUTURE WORK AND CONCLUSIONS

In this paper a unified and fully automated approach for managing multi-layer software-defined HetNets was proposed. The fundamental problems of optimizing UE-cell associations, interference mitigation, and scheduling in the time-frequency domain were addressed. The hill climbing algorithm for optimizing SC powers and selection biases, and MC muting patterns, increases 1st (cell edge) percentile downlink rates by up to 71% compared to a non-adaptive baseline. This result illustrates the importance of continually adapting cells with respect to time-varying traffic patterns in highly dynamic wireless communications networks. Schedulers evolved using GP boost cell edge rates by at most 104% in unoptimized HetNets. However, gains of up to 246% are achieved when evolved schedulers intelligently allocate bandwidth in well-configured HetNets. Therefore, the HetNet configuration and schedules should be optimized jointly.

Evolved schedulers outperform a state of the art benchmark because they are automatically tailored to the deployment scenario. Interestingly, GP also outperforms less expressive logistic models. Therefore, GP automatically synthesizes useful representations from minimal domain knowledge. Semantic analysis revealed that GP discovered nuanced yet intuitive strategies for allocating bandwidth. Rigid one-size-fits-all algorithms, such as the benchmark, cannot express the range of behaviors that are accessible to GP.

One of the main contributions in this paper is a flexible framework for fine-grained service differentiation. A novel parameterized fitness function enables the operator to tune the trade-off between cell edge and peak performance. When the latter is prioritized, 99th percentile rates are increased by up to 257%. The ability to accommodate bespoke objectives will be key to monetizing the next generation of 5G HetNets.

Future work could investigate multi-channel scheduling, whereby UEs simultaneously receive packets across LTE, Wi-Fi and millimeter wave links. Zero-touch automation will be essential in more complex multi-connectivity 5G architectures. Finally, the main contribution of this paper is a proof of concept that genetic programming can automatically generate better than human-competitive solutions for NP-hard problems in software-defined HetNets. In future work the approach could be leveraged to optimize 5G technologies like dynamic TDD [37].

ACKNOWLEDGMENTS

The authors are grateful to the reviewers and the NCRA team for their insights.

REFERENCES

[1] "At-A-Glance: The Internet of things," Cisco, San Jose, CA, USA, Tech. Rep., 2016. [Online]. Available: https://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf

[2] Qualcomm "The 1000x data challenge," Qualcomm, San Diego, CA, USA, Tech. Rep., 2012. [Online]. Available: https://www.qualcomm.com/documents/1000x-mobile-data-challenge

[3] A. Damnjanovic *et al.*, "A survey on 3GPP heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 10–21, Jun. 2011.

[4] D. Lopez-Perez, I. Guvenc, G. de la Roche, M. Kountouris, T. Q. S. Quek, and J. Zhang, "Enhanced intercell interference coordination challenges in heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 22–30, Jun. 2011.

[5] *Overview of 3GPP Release 10*, document V0.2.1, Dec. 2014. [Online]. Available: http://www.3gpp.org/specifications/releases/70-release-10

[6] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, "Algorithms for enhanced inter-cell interference coordination (eICIC) in LTE HetNets," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 137–150, Feb. 2014.

[7] A. Brabazon, M. O'Neill, and S. McGarraghy, *Natural Computing Algorithms*. Berlin, Germany: Springer-Verlag, 2015.

[8] R. McKay, N. X. Hoai, P. A. Whigham, S. Yin, and M. O'Neill, "Grammar-based genetic programming—A survey," *Genetic Program. Evolvable Mach.*, vol. 11, nos. 3–4, pp. 365–396, Sep. 2010.

[9] M. O'Neill and C. Ryan, *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Norwell, MA, USA: Kluwer, 2003.

[10] R. Agrawal, A. Bedekar, R. J. La, and V. Subramanian, "Class and channel condition based weighted proportional fair scheduler," *Teletraffic Sci. Eng.*, vol. 4, pp. 553–567, Feb. 2001.

[11] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, 1st Quart., 2015.

[12] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[13] "AirScale radio access," Nokia, Espoo, Finland, Tech. Rep., 2017. [Online]. Available: https://networks.nokia.com/products/airscale-radio-access

[14] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, no. 1, pp. 10–21, Jan. 1949.

[15] P. Mogensen *et al.*, "LTE capacity compared to the Shannon bound," in *Proc. IEEE 65th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2007, pp. 1234–1238.

[16] N. Bhushan *et al.*, "Network densification: The dominant theme for wireless evolution into 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 82–89, Feb. 2014.

[17] H. Claussen, L. T. W. Ho, and L. G. Samuel, "Financial analysis of a pico-cellular home network deployment," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 5604–5609.

[18] I. Hwang, B. Song, and S. S. Soliman, "A holistic view on hyper-dense heterogeneous and small cell networks," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 20–27, Jun. 2013.

[19] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 336–361, 1st Quart., 2013.

[20] M. Peng, D. Liang, Y. Wei, J. Li, and H.-H. Chen, "Self-configuration and self-optimization in LTE-advanced heterogeneous networks," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 36–45, May 2013.

[21] K. Okino, T. Nakayama, C. Yamazaki, H. Sato, and Y. Kusano, "Pico cell range expansion with interference mitigation toward LTE-Advanced heterogeneous networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2011, pp. 1–5.

[22] S. Vasudevan, R. N. Pupala, and K. Sivanesan, "Dynamic eICIC—A proactive strategy for improving spectral efficiencies of heterogeneous LTE cellular networks by leveraging user mobility and traffic dynamics," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 4956–4969, Oct. 2013.

[23] A. Tall, Z. Altman, and E. Altman, "Self organizing strategies for enhanced ICIC (eICIC)," in *Proc. 12th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2014, pp. 318–325.

[24] A. Weber and O. Stanze, "Scheduling strategies for HetNets using eICIC," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 6787–6791.

[25] J. Pang *et al.*, "Optimized time-domain resource partitioning for enhanced inter-cell interference coordination in heterogeneous networks," in *Proc. Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 1613–1617.

[26] L. Jiang and M. Lei, "Resource allocation for eICIC scheme in heterogeneous networks," in *Proc. IEEE 23rd Int. Symp. Pers. Indoor Mobile Radio Commun.-(PIMRC)*, Sep. 2012, pp. 448–453.

[27] D. López-Pérez and H. Claussen, "Duty cycles and load balancing in HetNets with eICIC almost blank subframes," in *Proc. IEEE 24th Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC Workshops)*, Sep. 2013, pp. 173–178.

[28] L. T. W. Ho, I. Ashraf, and H. Claussen, "Evolving femtocell coverage optimization algorithms using genetic programming," in *Proc. IEEE 20th Int. Symp. Pers. Indoor Mobile Radio Commun.*, Sep. 2009, pp. 2132–2136.

[29] E. Hemberg, L. Ho, M. O'Neill, and H. Claussen, "A comparison of grammatical genetic programming grammars for controlling femtocell network coverage," *Genetic Program. Evolvable Mach.*, vol. 14, no. 1, pp. 65–93, Mar. 2013.

[30] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, vol. 1. Hoboken, NJ, USA: Wiley, 2006.

[31] D. Lynch, M. Fenton, S. Kucera, H. Claussen, and M. O'Neill, "Ensemble techniques for scheduling in heterogeneous wireless communications networks," in *Operations Research Proceedings*. Cham, Switzerland: Springer, 2016.

[32] D. Fagan *et al.*, "Deep learning through evolution: A hybrid approach to scheduling in a dynamic environment," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 775–782.

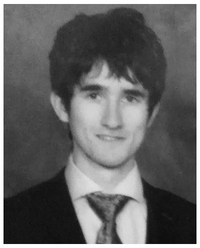
[33] M. Fenton, D. Lynch, S. Lynch, H. Claussen, and M. O'Neill, "Multilayer optimization of heterogeneous networks using grammatical genetic programming," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2938–2950, Sep. 2017.

[34] P. A. Whigham, "Grammatically-based genetic programming," in *Proc. Workshop Genetic Program. From Theory Real-World Appl.*, Jul. 1995, vol. 16, no. 3, pp. 33–41.

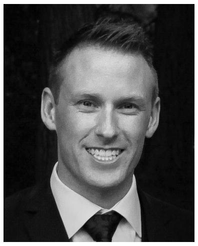
[35] M. Fenton *et al.* (2017). "PonyGE2: Grammatical evolution in python." [Online]. Available: https://arxiv.org/abs/1703.08535

[36] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1996, pp. 312–317.

[37] M. Ding, D. López-Pérez, R. Xue, A. V. Vasilakos, and W. Chen, "On dynamic time-division-duplex transmissions for small-cell networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 8933–8951, Nov. 2016.



David Lynch received the B.Sc. degree in physics in astronomy and space science from University College Dublin (UCD) and the M.Sc. degree in business analytics from the UCD Michael Smurfit Graduate Business School, where he is currently pursuing the Ph.D. degree in computer science with the Natural Computing Research and Applications Group. His research focus is on the applications of natural computing techniques in software-defined wireless communications networks.



Michael Fenton received the M.E. degree and the Ph.D. degree in engineering from the School of Civil Engineering, University College Dublin, Dublin, Ireland, in 2010 and 2015, respectively. He is currently a Data Scientist with Corvil, where he is utilizing network data to drive user behavior analytics via machine learning techniques to identify security threats inside corporate network environments. His main research interests include network security, machine learning and its applications, and problem solving.

David Fagan received the B.Sc. degree in computer science and the Ph.D. degree from the School of Computer Science, University College Dublin (UCD), Dublin, Ireland, in 2009 and 2013, respectively, where he is currently an Assistant Professor in management information systems with the UCD Michael Smurfit Graduate Business School. He is a member of the Natural Computing Research and Applications Group, UCD. His research interests include evolutionary computing, neural networks, and the general fields of AI and machine learning.

Stepan Kucera (S'14–M'06–SM'13) is currently a Senior Research Scientist with the Small Cells Research Department, Nokia Bell Labs at Dublin, Ireland, where he serves as a principal investigator and a technical lead for multiple projects aiming to create novel disruptive technologies for wireless networking. His expertise lies mainly in the area of wireless and IP networking technology, including both proprietary solutions and 3GPP/IEEE/IETF standards.

He has filed over 40 patents and has published over 30 book chapters, transactions, and conference papers in peer-reviewed IEEE venues. He was a co-recipient of four professional awards (the Irish Laboratory of the Year Award, the Irish Commercial Laboratory of the Year Award in 2014, the Best Student Paper Award at the IEEE Vehicular Technology Conference Fall in 2006, and the IEEE Kansai Student Researcher Encouragement Award in 2007). He was nominated for the IEICE Young Researcher Encouragement Award in 2009 and the Ericsson Young Scientist Award in 2010.

He received the M.Sc. degree from Czech Technical University, Prague, Czech Republic, in 2003, and the Ph.D. degree in informatics from the Graduate School of Informatics, Kyoto University, Kyoto, Japan, in 2008. From 2008 to 2011, he was a Research Scientist with the New Generation Wireless Communications Research Center, Keihanna Research Laboratories, NICT, Japan. He actively serves on technical boards of major IEEE journals and conferences.

Holger Claussen (SM'10) is currently the Leader of the Small Cells Research Department, Nokia Bell Labs, Ireland and USA, where he and his team are innovating in all areas related to future evolution, deployment, and operation of small cell networks to enable exponential growth in mobile data traffic. His research in this domain has been commercialized in Nokia's small cell product portfolio and continues to have significant impact.

He was the Head of the Autonomous Networks and Systems Research Department, Nokia Bell Labs at Dublin, Ireland, where he has been directing research in the area of self-managing networks to enable the first large-scale femtocell deployments since 2009. He received the 2014 World Technology Award in the individual category communications technologies for innovative work of the greatest likely long-term significance.

He received the Ph.D. degree in signal processing for digital communications from the University of Edinburgh, U.K., in 2004. He joined Nokia Bell Labs in 2004, where he began his research in the areas of network optimization, cellular architectures, and improving energy efficiency of networks. He has authored more than 100 publications and holds 120 filed patent applications. He is a fellow of the World Technology Network and a member of the IET.

Michael O'Neill is currently the ICON Chair of Business Analytics with the UCD Michael Smurfit Graduate Business School and also the Founding Director of the UCD Natural Computing Research and Applications Group. He has served in senior leadership roles, including the Associate Dean–the Director of the UCD Michael Smurfit Graduate Business School, the Vice-Principal of the UCD College of Business (Research, Innovation, and Impact), and the Director of the UCD's Institute for Interdisciplinary Research. He has published in excess of 300 peer-reviewed publications including four books.