

Evolutionary Learning of Scheduling Heuristics for Heterogeneous Wireless Communications Networks

David Lynch^{*}
Natural Computing Research
& Applications Group
School of Business
University College Dublin

Michael Fenton
Natural Computing Research
and Applications Group
School of Business
University College Dublin

Stepan Kucera
Bell Laboratories
Nokia-Ireland

Holger Claussen
Bell Laboratories
Nokia-Ireland

Michael O'Neill
Natural Computing Research
and Applications Group
School of Business
University College Dublin

ABSTRACT

Network operators are struggling to cope with exponentially increasing demand. Capacity can be increased by densifying existing Macro Cell deployments with Small Cells. The resulting two-tiered architecture is known as a Heterogeneous Network or 'HetNet'. Significant inter-tier interference in channel sharing HetNets is managed by resource interleaving in the time domain. A key task in this regard is scheduling User Equipment to receive data at Small Cells. Grammar-based Genetic Programming (GBGP) is employed to evolve models that map measurement reports to schedules on a millisecond timescale. Two different fitness functions based on evaluative and instructive feedback are compared. The former expresses an industry standard utility of downlink rates. Instructive feedback is obtained by computing highly optimised schedules offline using a Genetic Algorithm, which then act as target semantics for evolving models. This paper also compares two schemes for mapping the GBGP parse trees to Boolean schedules. Simulations show that the proposed system outperforms a state of the art benchmark and is within 17% of the estimated theoretical optimum. The impressive performance of GBGP illustrates an opportunity for the further use of evolutionary techniques in software-defined wireless communications networks.

Keywords

Heterogeneous Networks, Scheduling, Grammar-based Genetic Programming, Evaluative and Instructive Feedback.

^{*}Email: david.lynch.1@ucdconnect.ie

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '16, July 20-24, 2016, Denver, CO, USA

© 2016 ACM. ISBN 978-1-4503-4206-3/16/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2908812.2908903>

1. INTRODUCTION

Wireless traffic is growing exponentially. Cisco Systems, Inc. forecast a sixfold rise in demand by 2020 due to the proliferation of smart devices and the shift from voice to multimedia traffic [3]. Operators such as Vodafone Group, plc must increase the capacity of their networks in order to keep pace with this trend.

In traditional single-tiered networks, data is transmitted to User Equipments via high-powered antennas called Macro Cells (MCs) which are distributed on a hexagonal grid pattern. Note that any network-connected device, such as a smartphone or tablet, is denoted a 'User Equipment', or 'UE' for short. MCs are susceptible to high load which occurs when too many UEs request data simultaneously. Small Cells (SCs) have been proposed as a means of offloading UEs from strained MCs [2]. SCs are low-powered antennas which absorb UEs from the MC tier in concentrated traffic hotspots. When operating jointly, SCs and MCs constitute a Heterogeneous Network (HetNet).

MCs and SCs share the same bandwidth in co-channel HetNets. Spectral efficiency is valued by operators because bandwidth is expensive and scarce [1]. However, SCs are subject to significant interference from stronger MCs in co-channel mode. Interference is mitigated in HetNets by resource interleaving in the time domain which is partitioned into 1 ms intervals called subframes. Cells can transmit packets to their attached UEs during a subframe. Now, SCs can schedule different subsets of their attached UEs to receive data in successive subframes. Section 2.1 will describe how capacity is increased by intelligent scheduling.

Operators currently implement greedy heuristics based on proportionally fair scheduling. Tailoring these highly sub-optimal methods to corner cases demands significant human effort. Evolutionary heuristics facilitate the automatic discovery of robust schedulers for arbitrary network configurations. Furthermore, the solutions can be implemented at negligible cost as a software upgrade on existing hardware.

This paper compares two different fitness functions and two methods for generating Boolean schedules from the real-valued outputs of parse trees. We employed Grammar-based

Genetic Programming (GBGP) [21] with fitness functions based on evaluative and instructive feedback, as defined in [24]. The mappers enforced strong and weak constraints on how much airtime UEs received from their serving cell. Runs with evaluative feedback and strongly constrained mapping converged to better solutions earlier. Simulations revealed that GBGP significantly outperforms a benchmark scheme from the wireless networks literature [19].

This paper is organised as follows. Section 2 formalises the problem. Section 3 locates our contribution in the literature. Our simulation environment is described in Section 4. The methods and experiments are outlined in Section 5, with results and discussion following in Section 6. The paper concludes with directions for future work in Section 7.

2. PROBLEM SPECIFICATION

HetNets are spectrally efficient because both cell tiers transmit on the same bandwidth. However, two factors limit capacity when low-powered SCs operate in co-channel mode alongside stronger MCs. Firstly, SCs struggle to offload UEs from the high-powered MC tier. Secondly, UEs at the edges of SCs experience severe interference from neighbouring MCs.

The 3rd Generation Partnership Project–Long Term Evolution framework introduced the notion of Cell Range Expansion to achieve more efficient offloading [4]. To this end, SCs broadcast a Cell Selection Bias (β) such that $\beta_s \geq 0, \forall s \in \mathcal{S}$, the set of all SCs. MCs are typically overloaded due to their high transmit power, so $\beta_m := 0, \forall m \in \mathcal{M}$, the set of all MCs. Each UE u attaches to cell k :

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

where $Signal_{u,c}$, the signal from cell c as perceived by u , depends on the path losses from environmental obstacles, cell gain, and shadow fading. Underutilised SCs can either increase their power or bias to absorb additional UEs.

Cell Range Expansion aggravates the interference issue at SC edges. Consider a UE u that attaches to $s \in \mathcal{S}$ because $\beta_s > 0$ but would otherwise attach to some $m \in \mathcal{M}$. Then, by definition, the interference from m will exceed the signal from s . Therefore, u will experience a Signal to Interference and Noise Ratio ($SINR_u$) less than unity, resulting in significant packet losses. The UE is said to reside in the ‘expanded region’ of s . Note that $SINR_u$ is defined as the signal u receives from its serving MC or SC, divided by the interfering signals from all other cells plus background noise.

SC edge interference is mitigated by forcing MCs to mute in selected subframes. When a MC mutes we say that it executes an Almost Black Subframe (ABS) because it broadcasts only minimal control signals. SC attached UEs experience greatly reduced interference when nearby MCs undergo an ABS. MCs can execute conjunctions of the eight base ABS patterns displayed in Table 1, where ‘1’ indicates that the MC transmits and ‘0’ implies an ABS. Note that a full frame spans $|\mathcal{F}| = 8$ subframes or 8ms of network time¹. This is a convenient interval over which measurement reports can be collected to assess the network state. See [6] for a detailed description of the enhanced Inter-Cell Interference Coordination (eICIC) mechanism.

¹For the sake of clarity and WLOG it is assumed that $|\mathcal{F}| = 8$. Schedules are simply duplicated fivefold if $|\mathcal{F}| = 40$.

Subframe	1	2	3	4	5	6	7	8
Pattern 1	0	1	1	1	1	1	1	1
Pattern 2	1	0	1	1	1	1	1	1
Pattern 3	1	1	0	1	1	1	1	1
Pattern 4	1	1	1	0	1	1	1	1
Pattern 5	1	1	1	1	0	1	1	1
Pattern 6	1	1	1	1	1	0	1	1
Pattern 7	1	1	1	1	1	1	0	1
Pattern 8	1	1	1	1	1	1	1	0

Table 1: Base ABS Patterns.

In summary, inter-tier load is balanced via the Cell Range Expansion mechanism. Prohibitive interference at SC edges is then mitigated by MC resource interleaving in the time domain. Heuristics for optimising SC powers and biases and setting MC ABS patterns were outlined in [11]. Herein, we restrict our attention the problem of scheduling SC attached UEs, where large performance gains are expected [19, 8].

2.1 Scheduling in HetNets

Consider the toy network with two MCs and one SC depicted in Figure 1. UE2 and UE7 are located in the SC centre. However, UE4 resides in the expanded region (as suggested by gold shading) and is subject to severe interference from MC2. Let MC1 execute ABS Pattern 1 and let MC2 implement the conjunction of Patterns 1 and 2 from Table 1. Hence, MC1 mutes in subframe {1} and MC2 mutes in subframes {1, 2}.

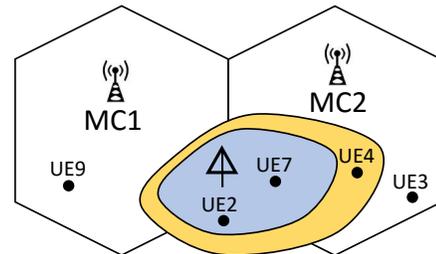


Figure 1: Toy HetNet.

Both MCs mute in subframe {1} so that UE2, UE7 and UE4 enjoy a high $SINR$ from their serving SC. By also muting in subframe {2}, MC2 grants extra protected airtime to UE4 in the SC’s expanded region. In the remaining subframes {3...8} both MCs transmit resulting in high interference (and low $SINR$) for the SC attached UEs. Shannon’s formula [23] gives the downlink rate ($R_{u,f}$) that will be experienced by UE u in subframe f via,

$$R_{u,f} = \frac{B}{N_f} \times Q_{u,f}, \quad (2)$$

where $B = 20$ MHz is the bandwidth, N_f is the number of UEs scheduled in f (congestion) and $Q_{u,f} = \log_2(1 + SINR_{u,f})$ is the channel quality experienced by u in f . $R_{u,f}$ quantifies the rate at which data can be transferred from the network to u in f .

Now, the reciprocal term ($1/N_f$) in Equation 2 implies that $R_{u,f}$ drops linearly with each additional UE that shares the bandwidth (B) in f . Therefore, it is generally suboptimal for SC s to transmit data to all its attached UEs (the set \mathcal{A}_s) in every subframe. On the other hand, if some $u \in \mathcal{A}_s$ is unscheduled $\forall f \in \mathcal{F}$, then it will receive no data. An intel-

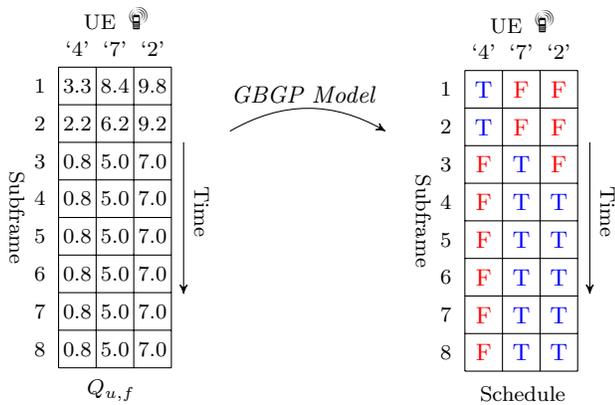


Figure 2: Mapping measurement reports to SC schedules.

ligent SC scheduling heuristic should negotiate the trade-off between airtime and per-subframe congestion such that a utility of UE downlink rates is maximised.²

Figure 2 illustrates the scheduling problem which can be stated as follows. Measurement reports are collected from all UEs in the network after each frame. Based on these reports, SCs estimate the channel qualities ($Q_{u,f}$) that will be experienced by their attached UEs ($\in \mathcal{A}_s$) over the subsequent frame. Each SC composes a schedule by mapping features over the set $\{Q_{u,f} | u \in \mathcal{A}_s, f \in \mathcal{F}\}$ to a Boolean matrix. The rightmost panel of Figure 2 depicts a feasible schedule for the SC of our toy network in Figure 1. For instance, this schedule states that UE4 will receive data from the SC in subframes $\{1, 2\}$ but not in $\{3 \dots 8\}$. We instrument GBGP to learn a mapping from statistics on the set $\{Q_{u,f}\}$ at SCs to schedules.

The performance of a scheduler is given by the sum log of average downlink rates:

$$\text{sum-log-rates} := \sum_{u \in \{SC \text{ UEs}\}} \ln R_u^{\text{avg}}, \quad (3)$$

where,

$$R_u^{\text{avg}} = \frac{1}{|\mathcal{F}|} \sum_{f=1}^{|\mathcal{F}|} R_{u,f},$$

is the average downlink rate for u over $|\mathcal{F}| = 8$ subframes. Notice that $R_{u,f}$ and hence the overall utility depends on the schedule through the $1/N_f$ term in Shannon’s formula (2). Equation 3 expresses the standard proportionally fair fitness metric for evaluating HetNet controllers [8, 17, 22, 25]. Operators strive for fairness because dropped calls or slow data speeds are unacceptable from a customer satisfaction standpoint.

3. PREVIOUS WORK

Scheduling problems arise in domains of operations research ranging from rostering [10] and job shop scheduling [18], to air traffic control [12]. The feasible solution space is usually explored directly via enumerative heuristics. However, executable rules that can compute solutions on the fly are often motivated by practical constraints.

²All UEs attached to $m \in \mathcal{M}$ are simply scheduled in every non-ABS subframe that m executes.

Jakovocić and Marasović (2012) identified the limitations of enumerative and search-based techniques when execution time presents as a constraint [16]. They manually designed meta-algorithms tailored to specific job-shop scheduling environments. Evolved priority functions operate within these meta-algorithms. Thus, domain knowledge informs the solution structure but Genetic Programming automatically uncovers the complex mapping from statistical features to schedules. Branke et al. (2015) emphasize the importance of carefully tailoring meta-algorithms for the application at hand [5]. However, they acknowledge the tedious nature of designing problem-specific heuristics and recognise the value of automated approaches.

In [20], the authors showed that robust human-competitive SC schedulers could be evolved with GBGP. However, our literature review has uncovered no other papers addressing scheduling in HetNets under the evolutionary paradigm. Such methods are motivated here because they yield robust solutions in dynamic environments [27, 9]. Ho and Claussen (2009) used Genetic Programming to optimise the coverage of femtocell deployments in enterprise environments [15]. Femtocells are SCs with a range of several meters. Their study represented a proof of concept that it is possible to automatically evolve controllers for wireless networks. Hemberg et al. (2011-13) used Grammatical Evolution to evolve symbolic expressions for femtocell coverage optimisation [13, 14]. The best solutions outperformed human designed heuristics on two of the three objectives.

Weber and Stanze (2012) manually designed a ‘strict’ and ‘dynamic’ SC scheduler [26]. The former sacrifices all cell-centre UEs during protected or ABS subframes. Thus, interfered UEs at cell edges receive extra bandwidth when their channel quality is highest. Cell-edge UEs are unscheduled during non-ABS subframes, thereby liberating bandwidth for cell-centre UEs. The dynamic scheduler allows edge UEs to receive data during both ABS and non-ABS subframes. Experiments showed that the dynamic scheduler achieves a better trade-off between cell-edge rates and spectral efficiency. We will see that GBGP automatically discovers the same core strategy as this human designed heuristic.

Pang et al. (2012) proposed a scheduling method based on dynamic programming [22]. Jiang and Lei (2012) modelled the scheduling problem as a two player Nash bargaining game in which protected (ABS) and normal (non-ABS) subframes compete for UEs [17]. Finally, Deb et al. (2014) formulated the problem as a non-linear programming instance [8]. Simulation revealed that cell edge UEs gain significantly, but, the algorithm requires information that is not readily deduced from measurement reports [19].

GBGP can automatically discover good solutions, even for corner cases, when guided by an appropriate fitness function. Sutton and Barto (1998) distinguish between evaluative and instructive feedback [24]. The former evaluates a solution without indicating how closely it emulates the optimum. Instructive feedback provides the search heuristic with the correct actions needed to solve a task. Fitness functions based on both types of feedback are described in Section 5.2.

López-Peréz and Claussen (2013) proposed the benchmark scheme which works as follows [19]. Each SC records the *SINRs* received by its attached UEs in subframes overlapping and non-overlapping with the nearest MC’s muted subframes. UEs are placed into either an overlapping queue

or a non-overlapping queue based on their average $SINR$ over $f \in \mathcal{F}$. The UEs with lowest average $SINR$ are identified in each queue. The SC computes target queue sizes which would equalise the rates of those worst performing UEs. One UE is transferred from the source to destination queue (subject to constraints) and the target sizes are recalculated. The above steps are iterated until the desired queue sizes stabilise. UEs are finally scheduled according to their queue type. This heuristic improved the 5th percentile downlink rates of SC attached UEs by 55% compared to a baseline method. The paper demonstrated that considerable gains can be achieved by intelligent scheduling. We adopt López-Peréz and Claussen (2013) as a benchmark.

4. SIMULATION ENVIRONMENT

A HetNet serving a 3.61 km² region of Dublin city centre is simulated. 21 MCs provide blanket coverage on a hexagonal grid, with 30 SCs deployed in traffic hotspots, which materialise at random locations. Cells locations are fixed but UEs are randomly distributed in each frame. A total of 1250 UEs are simulated giving an average density of 60 UEs per MC sector. 30 hotspots are placed on the map, such that they form around a SC with probability 0.9 and at a random location with probability 0.1. Each hotspot contains between 5 and 20 UEs. UEs that are not assigned to hotspots materialise at random locations on the map.

MC powers are fixed at 37 dBm and they do not use bias such that $\beta_m := 0, \forall m \in \mathcal{M}$. SC powers and biases are set at the beginning of a frame using an evolved heuristic [11]. Each UE then computes the signals received from all cells $c \in \mathcal{M} \cup \mathcal{S}$ in the network via,

$$Signal_{u,c} = P_c^{\text{TX}} + G[c, x, y], \quad (4)$$

where P_c^{TX} is the transmitting power of c in decibel milliwatts and $G[c, x, y]$ is the gain from c to u 's location (x, y) in decibels. G is computed by modelling the distributions of buildings, waterways and open spaces as they appear on a Google Maps³ image of the physical terrain.

Each UE determines its serving cell via Equation 1. Next, ABS patterns are set $\forall m \in \mathcal{M}$ using a heuristic described in [11]. In order to achieve a feasible pattern, each MC combines base patterns P1 to P8 of Table 1. Two constraints are respected; firstly, each MC can neither mute nor transmit $\forall f \in \mathcal{F}$, so the minimum duty cycle is 1/8 and the maximum is 7/8, secondly, muted subframes are ‘front-loaded’. For example, base patterns P1, P2 and P3 are combined if the desired duty cycle is 3/8. This second constraint reduces channel quality variance between adjacent subframes [11]. Finally, $SINR_{u,f}$ is computed $\forall (u, f) \in \{\text{All UEs}\} \times \mathcal{F}$.

The simulation was used to generate the training data so that individuals could be evaluated in standalone code. This approach reduced the runtime by two orders of magnitude. A training set consisting of 200 cases and a validation set with 100 cases were saved in MAT-files. A case is simply the $SINR_{u,f}$ values $\forall u \in \mathcal{A}_s$ over a single frame for SC s . These data are sufficient to construct the terminal set for GBGP and to evaluate the fitness of a candidate scheduler (see Section 5.2.1). Training cases were saved from seven different frames to encourage solutions that generalise well.

³<https://www.google.ie/maps/@53.3450749,-6.2697249,15.3z>

5. METHOD

This section describes how schedules are constructed using features extracted from channel quality reports. Two methods for interpreting the real-valued output of the parse trees are reviewed. Finally, two different learning regimes based on evaluative and instructive feedback are discussed.

5.1 Construction of Schedules

5.1.1 Terminal Set

Schedulers were evolved as binary classifiers on the domain $(u, f) \in \mathcal{A}_s \times \mathcal{F}$, where \mathcal{A}_s is the set of UEs attached to $s \in \mathcal{S}$. A tree is evaluated for all (u, f) pairs in order to construct the schedule for s . Twenty hand engineered features are provided to the tree in each execution, they are constructed from measurement reports as follows.

SC s has knowledge of the channel qualities $Q_{u,f}$ that will be experienced $\forall u \in \mathcal{A}_s$ over $f \in \{1..8\}$ of the coming frame. Shannon’s formula, which was presented in Section 2.1, states that the downlink rate for u in f is proportional to $Q_{u,f}$. Therefore, since we seek to maximise a utility of downlink rates, $Q_{u,f}$ is a sensible feature for inclusion in the terminal set. Statistics on $\{Q_{*,f}\}$ are admitted to contextualise u with respect to other UEs sharing f . Similarly, the channel qualities for u across all subframes are relevant when deciding whether to schedule u in a particular subframe. As such, statistics on the set $\{Q_{u,*}\}$ are included. More global cell-wide information is extracted from the set $\{Q_{*,*}\}$. Now, $Q_{u,f} \leq 1$ implies that the channel quality will be too low for u to receive packets in f . The number of times (if any) u experiences dropped calls is given by $|\{Q_{u,f} | Q_{u,f} \leq 1\}|$. If a dropped call occurs for u in f then $\llbracket Q_{u,f} \leq 1 \rrbracket$ evaluates to True, where $\llbracket \dots \rrbracket$ denotes the Iverson bracket. Finally, $u \in \{1..|\mathcal{A}_s|\}$ and $f \in \{1..8\}$ indicate the UE and subframe being considered in the current execution.

In sum, the terminal set consists of the following features:

- $Q_{u,f}$,
- $\left\{ \text{statistics}(\{Q_{*,f}\}, \{Q_{u,*}\}, \{Q_{*,*}\}) \right\}$
- $|\{Q_{u,f} | Q_{u,f} \leq 1\}|, \llbracket Q_{u,f} \leq 1 \rrbracket, u \in \mathbb{Z}, f \in \mathbb{Z}$
- $-1.0, -0.9, \dots, +0.9, +1.0,$

where, ‘statistics’ is an operator that returns the maximum, minimum, mean, 25th and 75th percentiles of its argument.

5.1.2 Mapping Schemes

The parse trees will return a real-valued number when evaluated on the features for UE u in subframe f . This signal must be interpreted as a Boolean decision specifying whether u will be scheduled to receive data from the SC in f or not. Two different mapping schemes were considered. Panel 2 of Figure 3 shows the decisions made by a ‘Threshold Mapper’ whereby, if $\text{Tree}(\text{features})_{u,f} > 0$ then $\text{schedule}_{u,f} \leftarrow \text{True}$ else $\text{schedule}_{u,f} \leftarrow \text{False}$. Notice that UE2 will not receive any data because $\text{Tree}(\text{features})_{2,f} \leq 0, \forall f \in \mathcal{F}$. Threshold Mapping was used to good effect in [20], however, we will see that it gives rise to solutions that ‘play it safe’ at the expense of performance.

Panel 3 of Figure 3 illustrates an alternative mapper which sets the largest four cells to True in each column. That is, each UE will receive data in exactly 4 subframes out of

$|\mathcal{F}| = 8$. Exploratory experiments suggested that an ‘Air-time Ratio’ of 4/8 gave the best performance. GBGP need only learn how best to interleave the T’s and F’s so that congestion is optimally managed. Section 6 outlines how better solutions emerge earlier in runs when the ‘Constrained Mapper’ is adopted.

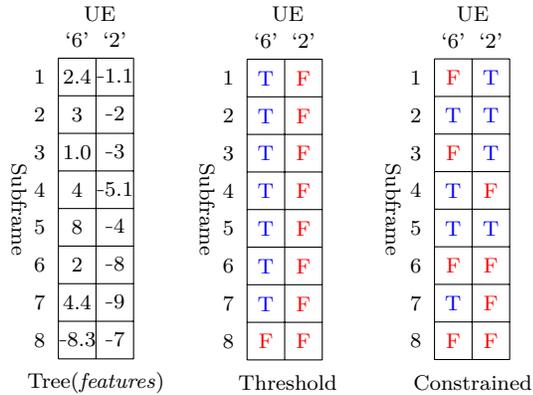


Figure 3: Mapping schemes.

5.2 Fitness Functions

5.2.1 Evaluative Feedback

Equation 2 implies that $R_{u,f}$ depends on the congestion in subframe f (through $1/N_f$) and the known channel quality $Q_{u,f}$. Recall that $R_{u,f}$ denotes the downlink rate or the rate at which data is delivered from the SC to u in f . Now, since N_f can be inferred from the schedule, it follows that $R_{u,f}$ can be determined $\forall(u, f) \in \mathcal{A}_s \times \mathcal{F}$. Equation 3 then yields contribution to the *sum-log-rates* from SC s . Aggregating over all SCs in the training set gives the overall fitness of the scheduler.

GBGP is tasked with maximising Equation 2. The model composes schedules for all training cases and then it receives a single scalar reward. Thus, the fitness function guides a form of reinforcement learning based on evaluative feedback.

5.2.2 Instructive Feedback

A new schedule is required every 8 ms which renders enumerative heuristics impractical for real-time optimisation. Nonetheless, a Genetic Algorithm (GA) can be executed offline to generate highly optimised schedules for all SCs in the training set. Perhaps intelligent strategies would be tapped by GBGP if models were trained to approximate the GA-derived schedules. Let the fitness of a model be defined as the classification accuracy it achieves over all (u, f) pairs. Thus, the fitness function guides a form of supervised learning based on instructive feedback. We hypothesised that higher *sum-log-rates* would be realised by models trained to emulate the GA, which works as follows.

The *sum-log-rates* for a schedule can be computed based on $Q_{u,f}, \forall(u, f) \in \mathcal{A}_s \times \mathcal{F}$. These data are provided to the GA for each SC in the training set. A GA individual encodes a SC schedule as an integer valued array of length $8 * |\mathcal{A}_s|$. Each codon can assume a value from the range [1...8]. Fitness is computed by reshaping the array as an $8 * |\mathcal{A}_s|$ matrix. The largest four cells in each column are set to True and the smallest four cells are set to False. Finally,

Equations 2 and 3 yield the *sum-log-rates*. The schedule that realises the highest *sum-log-rates* is extracted after 75 generations and stored for later use in the GBGP runs. Table 2 displays the GA’s tuned evolutionary parameters.

5.3 Experiments

Experiments were carried out to compare the Threshold and Constrained Mapping schemes for models trained on both evaluative and instructive feedback. Figure 4 presents the grammar used in Backus-Naur Form. Four non-linear transforms were admitted including $\text{sign}(x)$ which outputs 1 if $x \geq 0$, else -1 . The logarithm and square root functions were protected via $\ln(1 + |x|)$ and $\sqrt{|x|}$. Protected division, $x\%y$, returns x if y is zero, else x/y .

```

<E> ::= <R> | <R> | <R> | <terminal>
<R> ::= <arith><E>,<E> | <arith><E>,<E> |
        <arith><E>,<E> | <arith><E>,<E> |
        <NL><E> | <NL><E>
<arith> ::= + | - | * | %
<NL> ::= sin | ln | sqrt | sign
<terminal> ::= a feature from the terminal set

```

Figure 4: Grammar Definition.

The experimental settings for GBGP are displayed in Table 2. Each experiment consisted of 50 independent runs. The space of derivation trees was explored using subtree crossover and subtree mutation. A global derivation tree depth limit of 16 was imposed. The best model was selected based on its performance on a validation set consisting of 100 cells. A run with *population size* = 1000 and *generations* = 500 took 6.5 hours on one core of a 3.2 GHz machine. The experiments were designed to find the best combination of mapper and feedback type for maximising the *sum-log-rates*.

	GBGP	GA
Pop Size	1000	1500
Generations	200	75
Initialisation	Sensible	Random
Initial Max. Depth	6	N/A
Global Max. Depth	16	N/A
Selection	Fair Tournament	Tournament
Tournament Size	1% of pop size	4
Elitism	1%	1%
Crossover	Subtree	Two Point
Mutation	Subtree	Bit flip
Crossover Prob (p)	$p = 0.7$	$p = 1$
Mutation Prob (p)	Once per ind ^x	$p = 0.05$ per codon

Table 2: Evolutionary Parameters

The same network configuration with 30 SCs and 21 MCs was used to generate training and testing data. A case is simply the channel qualities received by UEs attached to a SC. The training set consisted of 200 cases sampled from 7 different frames. UEs were randomly re-distributed in each frame. Model selection was performed using a validation set of 100 cases sampled from 4 frames. A separate test set consisting of 3000 cases was saved from 100 frames.

6. RESULTS AND DISCUSSION

6.1 Training and Validation

Figures 5 (a) and (b) display the mean best-of-generation fitness from 50 independent runs for each experimental setup. Shaded regions enclose 95% confidence intervals about

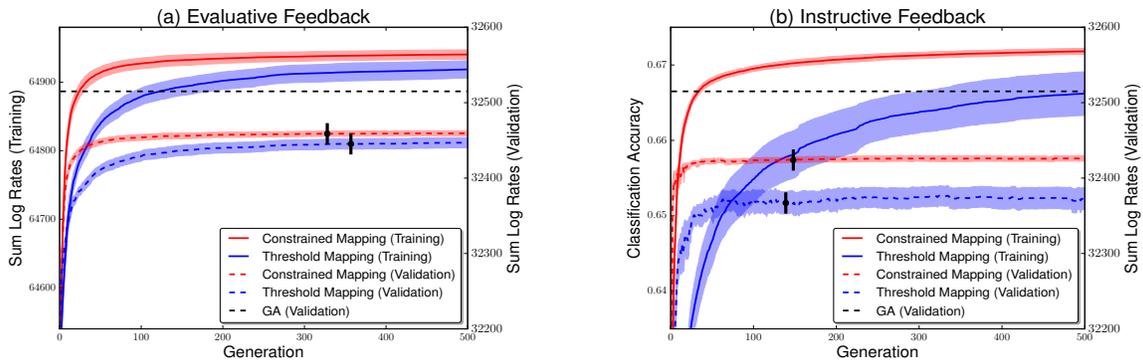


Figure 5: Mean training and validation fitness over 50 runs with 95% confidence intervals.

Quantity	Evaluative		Instructive	
	Threshold	Constrained	Threshold	Constrained
$\Delta sum - log - rates$	69.0 ± 7.2	75.1 ± 4.3	48.2 ± 14.3	61.3 ± 1.9
$\Delta 5^{th} \% - tile$ [Mbps]	0.159 ± 0.006	0.153 ± 0.008	0.146 ± 0.018	0.147 ± 0.006
$\Delta 50^{th} \% - tile$ [Mbps]	0.072 ± 0.026	0.102 ± 0.017	0.019 ± 0.051	0.111 ± 0.007

Table 3: Mean test performance of the 50 best-of-run models.

the means. Solid lines represent the average best fitness on training data. Dashed lines show the average best *sum-log-rates* of individuals on the validation set.

Figure 5 (a) illustrates how runs progress under evaluative feedback. In the Evaluative-Constrained set-up, runs converge to significantly higher *sum-log-rates* on the training and validation sets⁴. Furthermore, better models are evolved earlier than in the Evaluative-Threshold experiment. This suggests that forcing an Airtime Ratio of 4/8 subframes simplifies the task for GBGP. Model selection is performed by extracting the fittest individual on validation data. On average, the best models appear at generations 328 and 357 (black bars) under Constrained and Threshold Mapping respectively. Mild overfitting ensues around generation 400 when the average best fitness on the validation set converges. In conclusion, Figure 5 (a) implies that trees should be mapped subject to constraints on the schedules when maximising the *sum-log-rates* explicitly through an evaluative fitness function.

Figure 5 (b) summarises the progress of runs guided by instructive feedback. Exploratory experiments showed that instructive models trained with an Airtime Ratio of 4/8 broke on unseen cases. Instead, for best performance, UEs are scheduled for 6/8 subframes on the validation and test cases. GBGP models achieve a classification accuracy of 1.0 if they perfectly reproduce the GA-derived schedules. Runs using Constrained Mapping yield models which achieve an accuracy of 0.67 on average. Runs using Threshold Mapping exhibit large variance and significantly worse fitness. Nonetheless, for both set-ups, the validation *sum-log-rates* increase rapidly within the first 50 generations. Therefore, GBGP can learn intelligent strategies from the GA through instructive feedback. Increases in classification accuracy after generation 50 do not engender higher *sum-log-rates*, which plateau abruptly for both experiments. This suggests that the GA fits highly specialised schedules which generalised GBGP models cannot adequately capture.

⁴All statistical tests are two-sample t-tests with $\alpha = 0.05$.

6.2 Test Performance

The 50 best-of-run models were assessed on unseen test data for each experiment. Table 3 reports statistics on the *sum-log-rates* and two standard performance metrics [7]. The 5th %-tile of downlink rates is a measure of the quality of service experienced by highly interfered UEs at cell edges. Operators seek to maximise the 5th %-tile rates, possibly by sacrificing more privileged cell-centre UEs. The 50th %-tile rates is the median downlink rate received by all UEs. Test performance is reported relative to a baseline scheduling strategy whereby each UE is scheduled in every subframe. The baseline represents a naive greedy strategy. Table 4 displays the baseline statistics on the test set.

<i>sum-log-rates</i>	5 th %-tile [Mbps]	50 th %-tile [Mbps]
18367.9	0.364	2.64

Table 4: Mean baseline statistics over 100 test frames.

Table 3 reveals that the Evaluative-Constrained runs increase the *sum-log-rates* over baseline more than the three other set-ups. This combination also negotiates the best trade-off between the 5th %-tile and median rates. The closest competitor is the Instructive-Constrained set-up which exhibits similar percentile rates and lower model variance. However, this combination yields significantly lower *sum-log-rates*, which is the accepted metric for comparing the performance of HetNet controllers. Finally, runs that use Threshold Mapping strike a poorer compromise between the percentile rates and they achieve lower *sum-log-rates* than Evaluative-Constrained runs.

	Evaluative	Instructive
Constrained	150.6 ± 38.2	157.5 ± 51.5
Threshold	175.2 ± 47.0	185.1 ± 51.2

Table 5: Average number of nodes in the 50 best parse trees.

Table 5 displays the average size of the 50 best-of-run parse trees. A one-way ANOVA implies that Threshold

Quantity	Evaluative		Instructive		Benchmark	GA
	Threshold	Constrained	Threshold	Constrained		
$\Delta_{sum-log-rates}$	79.0 ± 6.0	80.3 ± 4.9	68.0 ± 10.3	65.5 ± 5.8	27.2 ± 12.1	96.3 ± 5.0
$\Delta_{5^{th}\% - tile}$ [Mbps]	0.161 ± 0.044	0.163 ± 0.042	0.150 ± 0.040	0.155 ± 0.040	0.109 ± 0.037	0.181 ± 0.045
$\Delta_{50^{th}\% - tile}$ [Mbps]	0.081 ± 0.044	0.101 ± 0.046	0.036 ± 0.064	0.108 ± 0.045	-0.090 ± 0.063	0.131 ± 0.044

Table 6: Comparison of methods on the test set over 100 frames.

Mapping yields significantly larger parse trees under both types of feedback. This suggests that fitter yet less complex models are evolved under Constrained Mapping.

Table 6 compares the best models from each set-up with the GA and benchmark method [19], which was described in Section 3. The best model uses Constrained Mapping and was evolved with evaluative feedback. It achieves significantly higher *sum-log-rates* than the best Instructive-Constrained model, although the differences between 5th %-tile and median rates are not significant. Threshold Mapping yields models with significantly lower median rates. The benchmark significantly under-performs GBGP on all fitness metrics. Notice that the benchmark trades off median rates for the 5th %-tile rates. GBGP manages to boost downlink rates for the cell-edge UEs without sacrificing the median rates, which are in fact increased versus baseline. That evolved models outperform the benchmark with respect to the 5th %-tile rates is impressive, given that the benchmark was specifically designed to maximise this metric. GBGP is competitive with the GA on all metrics. We conclude that GBGP is a powerful heuristic for automatically devising SC schedulers.

The GA computes a schedule for a single SC in about 20 seconds. However, the network updates every 40 ms, rendering the GA impractical for online optimisation. The benchmark runs in Python code within 1 ms. GBGP models execute in under 0.1 ms because they exploit efficient matrix operations. Section 7 motivates an ensemble type approach whereby N models propose hypothesis schedules for a SC. N can be large if each model executes quickly.

The heat maps in Figure 6 aggregate the decisions made by the best evolved schedulers, benchmark and the GA on 100 SCs with exactly ten attached UEs. Each column repre-

sents a UE and rows represent subframes. UEs are arranged with respect to their received channel quality, from low to high. Hence, the first few columns represent cell-edge UEs and the final columns correspond to cell-centre UEs. Deep red in square (u, f) indicates that u is scheduled in f for most of the 100 cases.

6.3 Visualising the Semantics

The heat maps reveal that all methods implement the following basic strategy. Edge UEs are scheduled very often in the first subframe wherein centre UEs are sacrificed. Therefore, the SC reserves most of the bandwidth for low channel quality UEs in the most protected subframe (more MCs mute in $f = 1$ than any other subframe). Edge UEs are infrequently scheduled during less protected periods when their achievable rates are low (subframes 3-8). This allows centre UEs to compensate for the premium airtime that they conceded initially. In this way, the SC fairly balances resources between all attached UEs over the frame.

There is clearly a close correspondence between the benchmark and three of the best evolved models. However, the Evaluative-Constrained model admits a distinctive tessellated mosaic. These semantics are replicated in most of the 50 runs for this set-up. Therefore, the combination of evaluative feedback and Constrained Mapping supports the evolution of intricate strategies that could not be provisioned by human designers. Most cells in GA’s heat map take on a shallow colour. This confirms that the GA fits highly specialised schedules which may be difficult for generalising models to replicate.

7. FUTURE WORK AND CONCLUSIONS

Evolved schedulers significantly outperform a benchmark

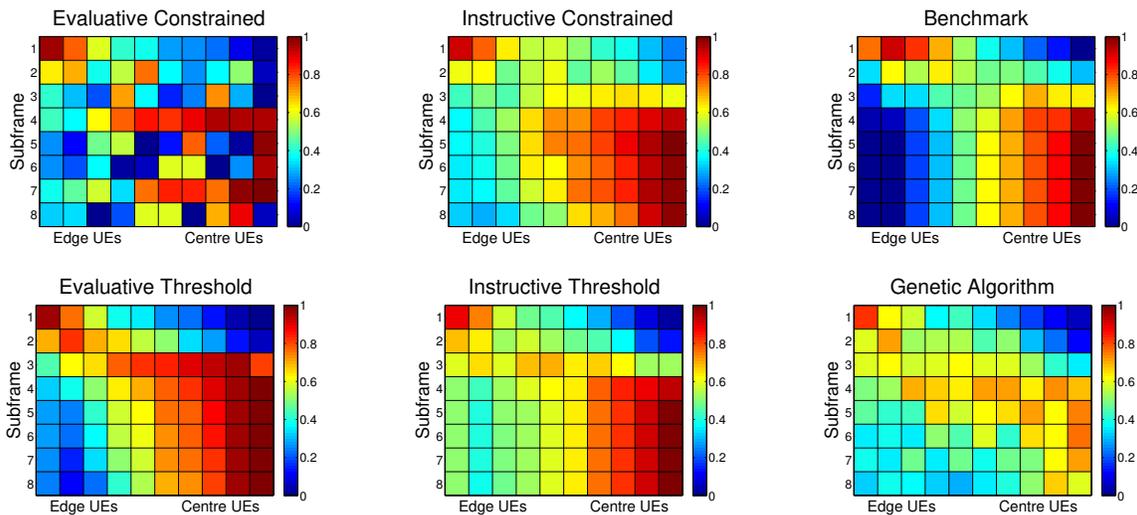


Figure 6: Visualising the semantics.

scheme from the literature on standard performance metrics. The results illustrate that GBGP is a powerful framework for automatically discovering effective HetNet controllers. Solutions are large black box expressions but their semantics are easily visualised by aggregating their decisions in heat maps. This perspective reveals a "rob from the rich and give to the poor" resource interleaving strategy.

Future work should investigate an ensemble approach in which several independent models generate hypothesis schedules for a SC. Pilot experiments indicate that an ensemble outperforms a single model by a large margin. Stateful models, such as recurrent neural networks, may better leverage instructive feedback by capturing the dependencies between successive scheduling decisions. Finally, the terminal set was constructed by hand in this work based on domain knowledge. A follow-up study could investigate the use of unsupervised techniques for feature synthesis.

Acknowledgements

This research is based upon works supported by the Science Foundation Ireland under grant 13/IA/1850.

8. REFERENCES

- [1] ComReg reveals 4G auction results – €450m instant windfall for Irish Govt. siliconrepublic, 2012. Article.
- [2] Small Cell Solutions. Alcatel-Lucent, 2015. Report.
- [3] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015-2020. Cisco, 2016.
- [4] 3GPP. December 2014. Specification (online).
- [5] J. Branke, S. Nguyen, C. Pickardt, and M. Zhang. Automated design of production scheduling heuristics: A review. *Evolutionary Computation, IEEE Transactions on*, 20(1):110–124, Feb. 2016.
- [6] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi. A survey on 3GPP heterogeneous networks. *Wireless Communications, IEEE*, 18(3):10–21, 2011.
- [7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.
- [8] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour. Algorithms for enhanced inter-cell interference coordination (eICIC) in LTE HetNets. *IEEE/ACM Transactions on Networking (TON)*, 22(1):137–150, 2014.
- [9] I. Dempsey, M. O’Neill, and A. Brabazon. *Foundations in grammatical evolution for dynamic environments*, volume 194. Springer, 2009.
- [10] A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, 153(1):3–27, 2004.
- [11] M. Fenton, D. Lynch, S. Kucera, H. Claussen, and M. O’Neill. Evolving coverage optimisation functions for heterogeneous networks using grammatical genetic programming. In *Applications of Evolutionary Computation*, pages 219–234. Springer, 2016.
- [12] J. V. Hansen. Genetic search methods in air traffic control. *Computers & Operations Research*, 31(3):445–459, 2004.
- [13] E. Hemberg, L. Ho, M. O’Neill, and H. Claussen. A symbolic regression approach to manage femtocell coverage using grammatical genetic programming. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, pages 639–646. ACM, 2011.
- [14] E. Hemberg, L. Ho, M. O’Neill, and H. Claussen. A comparison of grammatical genetic programming grammars for controlling femtocell network coverage. *Genetic Programming and Evolvable Machines*, 14(1):65–93, 2013.
- [15] L. T. Ho, I. Ashraf, and H. Claussen. Evolving Femtocell Coverage Optimization Algorithms using Genetic Programming. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 2132–2136. IEEE, 2009.
- [16] D. Jakobović and K. Marasović. Evolving priority scheduling heuristics with genetic programming. *Applied Soft Computing*, 12(9):2781–2789, 2012.
- [17] L. Jiang and M. Lei. Resource Allocation for eICIC Scheme in Heterogeneous Networks. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pages 448–453. IEEE, 2012.
- [18] A. Jones, L. C. Rabelo, and A. T. Sharawi. Survey of job shop scheduling techniques. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 1999.
- [19] D. López-Pérez and H. Claussen. Duty Cycles and Load Balancing in HetNets with eICIC Almost Blank Subframes. In *Personal, Indoor and Mobile Radio Communications, 2013 IEEE 24th International Symposium on*, pages 173–178. IEEE, 2013.
- [20] D. Lynch, M. Fenton, S. Kucera, H. Claussen, and M. O’Neill. Scheduling in Heterogeneous Networks using Grammar-based Genetic Programming. In *Genetic Programming*, pages 83–98. Springer, 2016.
- [21] R. I. McKay, N. X. Hoai, P. A. Whigham, Y. Shan, and M. O’Neill. Grammar-based Genetic Programming: a survey. *Genetic Programming and Evolvable Machines*, 11(3-4):365–396, 2010.
- [22] J. Pang, J. Wang, D. Wang, G. Shen, Q. Jiang, and J. Liu. Optimized Time-Domain Resource Partitioning for Enhanced Inter-Cell Interference Coordination in Heterogeneous Networks. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 1613–1617. IEEE, 2012.
- [23] C. E. Shannon. Communication In The Presence Of Noise. *Proceedings of the IRE*, 37(1):10–21, 1949.
- [24] R. Sutton and A. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [25] A. Tall, Z. Altman, and E. Altman. Self organizing strategies for enhanced ICIC (eICIC). In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on*, pages 318–325. IEEE, 2014.
- [26] A. Weber and O. Stanze. Scheduling strategies for HetNets using eICIC. In *Communications (ICC), 2012 IEEE International Conference on*, pages 6787–6791. IEEE, 2012.
- [27] S. Yang, Y.-S. Ong, and Y. Jin. *Evolutionary computation in dynamic and uncertain environments*. Springer Science & Business Media, 2007.