Towards Automation & Augmentation of the Design of Schedulers for Cellular Communications Networks

Michael Fenton michaelfenton1@gmail.com
Data Science & Machine Learning Group, Corvil Ltd., Dublin, Ireland

David Lynch david.lynch.1@ucdconnect.ie
Natural Computing Research and Applications Group, School of Business, University College Dublin, Ireland

David Fagan david.fagan@ucd.ie
Natural Computing Research and Applications Group, School of Business, University College Dublin, Ireland

Stepan Kucera stepan.kucera@nokia-bell-labs.com
Bell Laboratories, Nokia, Dublin, Ireland

Holger Claussen holger.claussen@nokia-bell-labs.com
Bell Laboratories, Nokia, Dublin, Ireland

Michael O'Neill m.oneill@ucd.ie
Natural Computing Research and Applications Group, School of Business, University College Dublin, Ireland

Abstract

Evolutionary Computation is used to automatically evolve small cell schedulers on a realistic simulation of a 4G-LTE heterogeneous cellular network. Evolved schedulers are then further augmented by human design to improve robustness. Extensive analysis of evolved solutions and their performance across a wide range of metrics reveals evolution has uncovered a new human-competitive scheduling technique which generalises well across cells of varying sizes. Furthermore, evolved methods are shown to conform to accepted scheduling frameworks without the evolutionary process being explicitly told the form of the desired solution. Evolved solutions are shown to outperform a human-engineered state-of-the-art benchmark by up to 50%. Finally, the approach is shown to be flexible in that tailored algorithms can be evolved for specific scenarios and corner cases, allowing network operators to create unique algorithms for different deployments, and to postpone the need for costly hardware upgrades.

1 Introduction

Wireless communications networks are a global trillion dollar industry. The GSM Association reports the mobile industry comprised 4.4% of global GDP in 2016, amounting to $3.3 trillion (GSMA, 2017). In order to remain relevant in a vast and increasingly competitive market, network operators value any performance improvements that yield an edge over competitors. Globally, network operators are forecast to spend upwards of $1.4 trillion upgrading their systems through to 2020 (GSMA, 2017). As such, small performance improvements can scale to deliver significant cost savings in such a large domain.

Until recently, the main focus for the optimisation of wireless communications networks has observed conflicting goals of maximising coverage and network performance whilst minimising power consumption (Hemberg et al., 2011; Tang et al., 2015). However, with the exponential increase in mobile traffic (Cisco, 2015) arising from both rapid growth in the mobile devices market and the onset of the internet of things[1], this focus has shifted to pure capacity maximisation as network operators struggle to meet demand (Bian and Rao, 2014).

Three main avenues are available for network operators to address the capacity problem. The first is to increase bandwidth, which amounts to an often prohibitive financial cost. The second approach is to increase the signal to interference and noise ratio. This can be managed through intelligent network configuration and providing windows in time where interference is reduced. And the third approach, which is the focus of this study, is to optimise the number of devices/users sharing the bandwidth through intelligent scheduling in the time domain.

As part of capacity maximisation problem faced by network operators, it is now common for these operators to densify their networks through the deployment of small cells (Bian and Rao, 2014). Effectively, existing high-powered Macro Cell (MC) deployments are supplemented by lower-powered Small Cells (SCs) in a Heterogeneous Network, or HetNet. These SCs can be deployed *Ad hoc* within the operational range of the MC in order to offload User Equipments[2] (UEs) from the MC tier. As bandwidth is scarce and expensive, MCs and SCs typically operate on a co-channel deployment, using the same bandwidth.

Optimisation of HetNets can occur on a number of fronts, including SC transmit power optimisation and packet transmission scheduling in the time domain. Intelligent timeframe scheduling at the SC level is attractive to network operators as it represents a relatively cheap software solution, and does not require re-configuration of the network. As such, it is the focus of this study.

Previous works by the authors have examined timeframe scheduling at the SC level (Lynch et al., 2016b; Lynch et al., 2016a; Lynch et al., 2017; Fenton et al., 2017a). However, detailed global network statistics and measurement reports were available to evolution which allowed for precise control of all aspects of the network. This provides an unrealistic level of data granularity with which control decisions can be made in a real-world environment.

In this study we bring evolutionary computation closer to producing solutions which can be deployed in real networks. Real-world network deployments are extremely limited in the quality/granularity of measurement reports. Not only are reports highly constrained and limited, but reported data is quantized and averaged from its true form. Furthermore, such inaccurate reporting can have a significantly detrimental effect on end-user performance, to the extent where data transmissions can be permanently dropped if actual end-user signal differs too greatly from reported signal (3GPP, 2014). This information paucity adds an extra layer of complexity to the problem, and as such presents a far greater challenge to optimisation methods.

In this paper we set out to ascertain:

i whether it is possible for the evolutionary process to successfully produce viable

[1]Cisco estimates the internet of things will consist of 50 billion devices connected to the Internet by 2020, with the total number of connected devices doubling year-on-year (Cisco, 2016).

[2]Any network-connected devices, such as smartphones, tablets, laptop computers, etc.

solutions given sparse and inaccurate information about the true state of the network,

ii how easily and successfully these solutions can be augmented by human experts, and

iii whether these evolved and augmented solutions can out-perform a state-of-the-art human-designed benchmark across a range of scenarios.

We report the successful application of evolutionary computation, in particular a grammar-based form of Genetic Programming (McKay et al., 2010), to this pressing real-world communications network problem, which achieves beyond human-competitive performance, significantly outperforming human-designed state-of-the-art solutions reported in the communications networks literature. An additional advantage of the adopted encoding leaves the evolved solutions transparent to the network engineers, making them amenable to human understanding and augmentation. We demonstrate how an in-depth examination of both the evolved solutions and their semantic performance can yield an intuitive understanding of how human-competitiveness has been achieved.

The remainder of this paper is structured as follows. Section 2 details the problem specifics, while Section 3 overviews HetNet optimization under the current industry standards and describes grammatical GP. The simulation environment is described in Section 4, with Section 5 introducing the experiments. The results are examined in great detail in Section 6, including a breakdown and simplification of the best evolved solution itself in Sections 6.1 and 6.2. An extensive analysis of the performance of the solution is given in Section 7, while Section 8 examines the ability of the method to evolve solutions for different congestion scenarios. The paper closes with concluding remarks and suggested future directions in Section 9.

2 Problem Definition

Since the transmit power of small cells (SCs) is far lower than that of macro cells (MCs)[3], SCs typically tend to be underutilised as UEs greedily attach to the strongest serving cell. To increase the use of the SC tier, provision has been made under the 3rd Generation Partnership Project - Long Term Evolution (3GPP-LTE) framework (3GPP, 2014) for a Range Expansion Bias (REB) mechanism. REB artificially increases the observed transmit power of a SC, tricking UEs into attaching to a SC with a weaker signal in deference to their stronger serving MC for the global good of the network. Each cell i broadcasts a non-negative constant $\beta_i \in \mathbb{R}_{\geq 0}$ as its REB. A UE u will therefore attach to a cell k in accordance with Eq. 1:

$$k = argmax_i(S_{ui} + \beta_i), \tag{1}$$

where $S_{ui} \in \mathbb{R}_{\geq 0}$ is the received signal strength from cell i to UE u, $\beta_m = 0, \forall m \in \mathcal{M}$, the set of all MCs, and $\beta_s \geq 0, \forall s \in \mathcal{S}$, the set of all SCs. Once cell attachment is set, data must be scheduled by the hosting cells for transmission to attached UEs.

2.1 Scheduling of Data Transmissions

Cells transmit packets of data to attached UEs on a millisecond timescale known as subframes. A single subframe $f = 1$ ms duration, and a full frame $\mathcal{F}$ is comprised of 40

[3] SCs typically transmit at 3.16 W, MCs typically transmit at 21.6 W.

subframes. A full frame of data transmissions (40 discrete transmission periods) must be scheduled by the hosting cell in advance of their transmission, i.e. each cell must decide when to transmit data to whom over the course of the next 40 ms[4]. This means that each cell has exactly 40 ms to decide the optimum schedule for the ensuing 40 ms.

Once attached to a cell, the potential performance of a UE can be broadly known from its ratio of received signal strength from its serving cell to the combined strengths of all other interfering signals from all other cells in the network (plus some background noise), known as the Signal to Interference and Noise Ratio ($SINR$). The $SINR$ for a UE u attached to a cell i in a subframe f is given by:

$$SINR_{ui,f} = \frac{S_{ui,f}}{\sum_{n \in \mathcal{M} \cup \mathcal{S}} S_{un,f} - S_{ui,f} + \text{Noise}} \text{ [dB]}. \tag{2}$$

A higher $SINR$ allows data to be transmitted with less interference, resulting in a stronger connection and faster data transfer rates. However, $SINR$ values can only be changed either by re-configuring the network (e.g. changing cell powers) or if the UE moves to a new location with less interference (e.g. closer to the serving cell).

2.2 Almost Blank Subframes and Range Expansion Bias

The $SINR$ defined in Eq. 2 is defined on a per-subframe basis as the received signal strength from any given cell in a network can vary across the full frame $\mathcal{F}$ due to the effects of interference mitigation schemes such as Almost Blank Subframes. Any SC s implementing a non-zero REB β_s will experience high interference from neighboring higher-powered MCs at its cell edges. The additional area leveraged by the SC as a result of its non-zero REB is known as the "expanded region". From Eq. 1, it can be seen that any SC-attached UE within the cell's expanded region will experience greater signal strength in the form of interference from its strongest serving MC than from its hosting SC. It can therefore be appreciated from Eq. 2 that the $SINR$ of those UEs in the expanded region of SCs will be less than unity (i.e. they receive more interference from neighboring cells than signal from their serving cell).

As SC-attached UEs at the cell edge suffer from significant interference from the higher-powered MC tier, an enhanced Inter Cell Interference Co-ordination (eICIC) system known as Almost Blank Subframes (ABSs) is employed at the MC level under the 3GPP-LTE system (3GPP, 2014). ABSs are individual subframes during which a MC mutes all transmissions (save for some minimal necessary control signals) in order to allow nearby SCs to transmit to cell-edge UEs during periods of minimal interference[5]. MCs can implement any combination of 8 distinct patterns, shown in Table 1. Each MC in the network can implement its own unique ABS pattern (asynchronous patterns), or a global ABS pattern can be dictated to all MCs in the network (e.g. the average pattern requested by all MCs in the network; synchronous patterns).

The basic eICIC HetNet concept is illustrated in Fig. 1 with a toy HetNet showing one MC, one SC, and fifteen UEs (Fig. 1a). The SC has been placed within the operational range of the MC so as to serve a nearby hotspot of 7 UEs, and thus alleviate congestion from the MC tier. Since the transmission power of the SC is much less than that of the MC, a positive REB is used by the SC to offload UEs in the hotspot from the MC tier (Fig. 1b). However, severe cross-tier interference is experienced by UEs in the expanded region of the SC due to the higher-powered MC. Data is transmitted to these

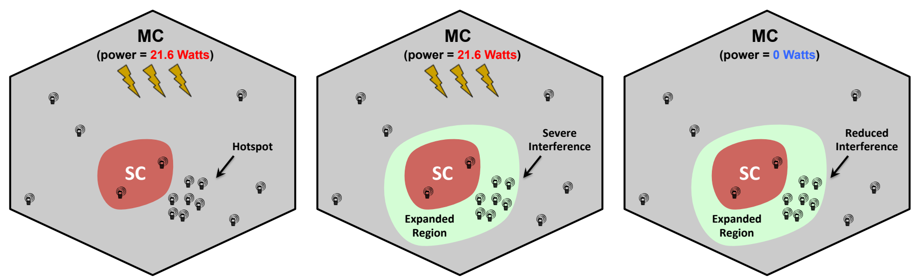
[4]Note that any number of UEs can be scheduled to receive data during a single subframe.

[5]Note that MC-attached UEs can not receive any data transmissions during ABSs.

Table 1: All 8 possible ABS patterns, shown across the full frame of 40 subframes. 1 indicates MC transmission during that subframe, and 0 indicates an ABS.

Subframe	1 – 8	9 – 16	17 – 24	25 – 32	33 – 40
ABS pattern 1	01111111	01111111	01111111	01111111	01111111
ABS pattern 2	10111111	10111111	10111111	10111111	10111111
ABS pattern 3	11011111	11011111	11011111	11011111	11011111
ABS pattern 4	11101111	11101111	11101111	11101111	11101111
ABS pattern 5	11110111	11110111	11110111	11110111	11110111
ABS pattern 6	11111011	11111011	11111011	11111011	11111011
ABS pattern 7	11111101	11111101	11111101	11111101	11111101
ABS pattern 8	11111110	11111110	11111110	11111110	11111110

UEs during ABSs (Fig. 1c), whereby the MC mutes and interference in the expanded region is dramatically reduced.



(a) SC is placed near a hotspot, within operational range of the MC.

(b) Hotspot is captured using REB, but severe interference is observed in expanded region.

(c) Inter-tier interference is reduced during ABS when MCs mute transmissions.

Figure 1: Toy heterogeneous network with one MC, one SC, and fifteen UEs. The SC has been placed near a hotspot of seven UEs, who are offloaded from the MC tier by the use of a positive REB from the SC. Interference is mitigated through the use of ABSs, during which data can be transmitted to UEs in the expanded region.

2.3 Downlink Rates

The ultimate downlink rate for a UE u attached to a cell i during subframe f is described by Shannon's equation for the transmission of wireless data in the presence of noise (Shannon, 1949):

$$R_{ui,f} = \frac{B}{N_{i,f}} \times \log_2(1 + SINR_{ui,f}) \text{ [bits/sec]} \tag{3}$$

where B is the available bandwidth (e.g. 20 MHz), and $N_{i,f} \in \mathbb{Z}_{\geq 0}$ is the total number of UEs sharing that bandwidth during subframe f[6]. From Eq. 3 it can be seen that there are three main approaches for increasing the downlink rate of a UE u in subframe f:

a. increase the bandwidth (B),

[6]This assumes a round-robin bandwidth scheduler. Other schemes exist, e.g. proportional fairness (Motorola, 2006).

b. increase the $SINR$ of UE u in subframe f, or

c. decrease the number of UEs ($N_{i,f}$) sharing that same bandwidth.

Bandwidth is limited and expensive; in 2012 the Irish Commission for Communications Regulation (ComReg) auctioned off 140 MHz of bandwidth across the 800 MHz, 900 MHz, and 1800 MHz frequency bands to four network carriers for a total price of €854.68 million (ComReg, 2012). $SINR$ values can only be changed by either moving the UE to a different location (not under the control of the network operator) or by re-configuring the network (either by changing MC ABS patterns or SC power or REB levels (Fenton et al., 2017a)). As such, the greatest resource (excluding bandwidth scheduling) available to network operators seeking to improve UE data throughput is therefore to change the numbers of UEs scheduled during each subframe (i.e. managing per-subframe congestion through intelligent timeframe scheduling).

A simple baseline scheduling technique is to schedule all attached UEs during every subframe $f \in \mathcal{F}$ (MC-attached UEs are thus scheduled). This will usually guarantee all UEs get some degree of data transmission, but this greedy strategy will maximise per-subframe congestion leading to unfair average downlink rates. Network operators typically seek to maximise per-cell throughput with respect to fairness. This is traditionally achieved through maximising the sum of the log of the downlink rates of all UEs in the network, commonly known as Sum-Log-Rates, or SLR (Andrews et al., 2014). The use of a logarithm in this function ensures downlink rate changes for lower-rate UEs are given a higher weighting than changes for better performing higher-rate UEs. In terms of timeframe scheduling, this fairness means that those highest-$SINR$ UEs (i.e. those with the strongest signal strength) can be sacrificed (i.e. scheduled for less available subframes) in order to minimise congestion and thus increase the downlink rates of lower-$SINR$ UEs.

2.4 UE Measurement Statistics

Networks are configured and schedules across the full frame set based on measurement reports from UEs to their serving cells. The serving cell configures the UE to report any of a range of desired statistics, in the form of channel gains, average downlink rates, channel quality indicator (CQI), and average $SINR$ values. Full cell channel gains and average downlink rates provide the best level of detail for network reports, but take an order of seconds for each UE to collate and as such are not feasible for use in scheduling applications (these values were used in all previous publications by the authors (Fenton et al., 2015; Fenton et al., 2017a; Lynch et al., 2016a; Lynch et al., 2016b; Lynch et al., 2017)). CQI and $SINR$ values on the other hand can be reported instantaneously but are far less detailed. For time-critical tasks such as scheduling across the full frame in real-world deployments[7], either CQI values or $SINR$ values must be used. Of these, $SINR$ reports provide a clearer view of the performance of the UE.

In a real-world scenario, UEs are only configured to report two $SINR$ values:

i the estimated average $SINR$ performance during ABSs, and

ii the estimated average $SINR$ performance during non-ABSs.

For SC-attached UEs, the hosting SC dictates the ABS pattern to the UE based on the strongest interfering MC for that SC. Furthermore, reported averaged $SINR$ values

[7]Measurement reports must be collated and schedules must be set in less than 40 ms.

are typically quantized to 1 or 2 dB, meaning reported $SINR$ values can differ significantly from values actually received by UEs.

UEs typically experience significant packet losses under conditions of low $SINR$. The out-of-sync threshold Q_{out} is defined as the lower $SINR$ limit for which data can be transmitted without severe packet losses (ETSI, 2016). Any UE with an $SINR$ equal to or lower than this threshold limit should therefore not be scheduled due to the risk of dropped transmissions.

Since there exists a discrepancy between the reported (estimated, averaged, quantized) $SINR$ and the actual $SINR$ experienced by UEs, failed transmissions can occur for UEs at the lower end of the $SINR$ scale. Furthermore, the use of asynchronous ABS patterns compounds this matter, as the ABS patterns dictated to a UE by their hosting SC for the purposes of measuring average ABS $SINR$ and average non-ABS $SINR$ may differ from the *actual* ABS patterns experienced by the UE in the field.

When a transmission fails, the dropped data is rescheduled for the next available free subframe in which the UE's *estimated* $SINR > Q_{out}$ (a minimum of 4 subframes after the original transmission due to finite processing time). A free subframe is defined as any subframe in which that UE is *not* scheduled, but can be *permitted* to be scheduled (due to their estimated $SINR > Q_{out}$). If there are no free subframes, then the retransmission is scheduled for the next subframe in which a transmission is already scheduled (thus shifting subsequent data transmissions down the transmission queue for that UE). If there are no more subframes in which data can be transmitted, then the data is permanently lost.

3 Background

3.1 Optimisation of Heterogeneous Networks

The vast majority of optimisation literature surrounding LTE Heterogeneous Networks addresses eICIC techniques. Large gains in network performance can be made with the use of Self-Organising Networks (SONs) (Hämäläinen et al., 2012), covering optimal cell power control, SC bias control, and cell handovers, among others. The literature ranges from improving energy efficiency to limit excessive power usage (Tang et al., 2015), to minimisation of inter-cell interference through automatic re-configuration (Peng et al., 2013; Madan et al., 2010; Deb et al., 2014). A more in-depth survey of the field of SONs in LTE can be found in (Aliu et al., 2013).

Madan et al. (Madan et al., 2010) provided a number of different algorithms for varying optimisation targets, with the aim of maximising downlink rates with respect to fairness in indoor HetNet/Femtocell deployments. They formulated two classes of problem: semi-static interference management, where optimisation occurs on the rate of 100s of milliseconds, and fast-dynamic interference management, where optimisation happens on a per-subframe (1ms) timescale. As with (Lynch et al., 2016b; Lynch et al., 2016a; Lynch et al., 2017; Fenton et al., 2017a), their models assumed perfect instantaneous knowledge about UE channel gains; they reasoned that in low-mobility indoor environments channel gains change far slower than with outdoor deployments.

Siomina & Yuan (Siomina and Yuan, 2012) applied an iterative two-stage approach to optimise SC range expansion bias in order to maximise Jain's fairness index (Jain et al., 1984). Their first step employed a statistical Design Of Experiments (DOE) approach to identify the most important factors for optimisation. Next, they used a regression-style analysis to evaluate the next set of factors for identification by the first stage DOE. By optimising based on cell load, Jain's fairness index is guaranteed to be concave and thus contain no local optima for continuous load variables.

3.2 Scheduling in Heterogeneous Networks

The bulk of literature on scheduling techniques for HetNets describes human-designed algorithms. The common scheduling strategy is to place the worst-performing UEs in the best available subframes, while reserving those subframes with the highest interference for the best-performing UEs.

Jiang and Lei (Jiang and Lei, 2012) developed an algorithm which separates SC-attached UEs into two distinct queues: those to be scheduled during protected ABS-overlapping subframes, and those to be scheduled during high-interference non-ABS subframes. They noted that higher UE numbers scheduled during ABS-overlapping subframes will require a more aggressive ABS ratio from hosting MCs. Consequently, they proposed a scheduling scheme that takes into account both the number of ABS subframes and those UEs to be scheduled during respective subframes. They formulated the problem as a two-player Nash Bargaining Solution game, with resources of ABS and non-ABS subframes competing for UEs. The ultimate goal of the game is to maximise the downlink rates of both ABS and non-ABS UEs.

Weber and Stanze (Weber and Stanze, 2012) examined two scheduling techniques for SC-attached UEs: strict and dynamic. Their strict scheduler schedules cell edge UEs during ABSs and cell center UEs during non-ABSs (similarly to Jiang and Lei), while their dynamic scheduler which assigns resources purely based on a proportional fairness metric. Both approaches rely on the use of proportional fairness bandwidth scheduling (Motorola, 2006). While their strict approach breaks down during low load conditions, the dynamic scheduler can allow cell-edge UEs to be scheduled during both protected ABS and high-interference non-ABS overlapping subframes, potentially improving their performance.

Similarly to Jiang and Lei, López-Peréz and Claussen (López-Pérez and Claussen, 2013) also divide all SC attached UEs into either ABS or non-ABS overlapping subframe queues. The difference between the two methods lies in *which* UEs are placed in either queue. López-Peréz and Claussen divide UEs such that the downlink rates of the two worst-performing UEs in each queue (i.e. the worst-performing UE in the ABS-overlapping queue and the worst-performing UE in the non-ABS-overlapping queue) are equalised. Their approach details an iterative algorithm, continually adding or removing UEs from one queue to another until convergence is achieved. Unlike Weber and Stanze, López-Peréz and Claussen's algorithm implicitly addresses corner cases such as low load conditions, providing good performance in all scenarios.

3.3 EC Applied to Heterogeneous Networks

EC methods have been successfully applied to the optimisation of cellular networks. Hemberg et al. employed Grammatical Evolution (GE) (O'Neill and Ryan, 2003) to evolve coverage optimisation algorithms for indoor homogeneous femtocell deployments in a number of works (Hemberg et al., 2011; Hemberg et al., 2013). In (Hemberg et al., 2011), their use of a hybrid of NSGA-II with Tabu search allowed them to both maximise coverage while minimising power consumption, improving on a scenario with static power levels. In (Hemberg et al., 2013), they compared both regression and conditional grammar designs for algorithmic femtocell controllers, noting that regression-based grammars required more fitness evaluations whereas the conditional-based grammars required more domain knowledge. Notably, in all publications they reported that the evolutionary methods overfitted to the simulation model by exploiting its assumptions.

Previous work by the authors has used GE and grammar-based Genetic Programming to optimise three components of a heterogeneous network: SC power and bias levels, MC ABS patterns, and a simplified version of the SC scheduling problem (Fenton et al., 2017a). Experiments compared the sequential optimisation of all three components of the network to the simultaneous optimisation of all three, but ultimately found that the fitness functions and grammatical representations used were inadequately designed. This led to the design of the grammars and fitness functions used in this study.

Further works examined the sole optimisation of scheduling for SCs in large-scale urban heterogeneous deployments (Lynch et al., 2016a; Lynch et al., 2016b; Lynch et al., 2017). However, the simulation environment for these previous methods (including (Fenton et al., 2017a)) bounded achievable performance under a complete information model on UE measurement reports, comprised of highly accurate complete channel gain matrix information (similar to (Madan et al., 2010)). As detailed in Section 2.4, detailed network reports such as the complete channel gain matrix take on the order of seconds for UEs to report in a real-world environment, and as such can not be used for fast-paced outdoor on-line scheduling applications. Furthermore, since instantaneous channel gain matrix reports are perfectly accurate, no data is dropped as no UEs are scheduled erroneously. As such, solutions evolved under these prior schemes may struggle to work in real deployments. This has motivated the work presented in this paper, where real-world limitations are imposed on the simulation model.

4 Simulation Setup

The simulation covers a 3.61 km^2 area of Dublin city center, as shown in Fig. 2. 21 MCs are arranged on a hexagonal grid, with 30 SCs placed in random locations befitting their operator-defined placement. Channel gains are calculated using background noise, path losses & shadow fading (i.e. signal decay), and environmental losses (e.g. buildings, trees). Full simulation parameters are given in Table 2.

Table 2: Simulation Parameters.

Parameters	Value
Scenario	
Indoor/outdoor map	Dublin (central eNodeB at WGS84 N 53.340494 and W 6.264374)
MC BS placement	7 eNodeB with 3 sectors each (hexagonal grid)
SC BS placement	Uniformly randomly distributed
Inter-MC BS distance	800 m
Scenario resolution	2 m
Transmit power	$P_{\mathrm{tx},n} = 21.6\,\mathrm{W}$ (MC), $3.16\,\mathrm{W}$ (SC)
Noise density	$-174\,\mathrm{dBm/Hz}$
SC REB	7 dB
Channel	
Bandwidth	20 MHz (1 LTE carrier with 10 LPCs of size $L = 8$
NLOS path-loss	$G_{\mathrm{Pn}} = -21.5 - 39\log_{10}(d)$ (MC) (3GPP E-UTRA, 2010)
	$G_{\mathrm{Pn}} = -30.5 - 36.7\log_{10}(d)$ (SC) (3GPP E-UTRA, 2010)
LOS path-loss	$G_{\mathrm{Pl}} = -34.02 - 22\log_{10}(d)$ (3GPP E-UTRA, 2010)
Shadow fading (SF)	6 dB std dev. (3GPP E-UTRA, 2012)
SF correlation	$R = \mathrm{e}^{-1/20d}$, 50% inter-site
Environment loss	$G_{\mathrm{E},n} = -20$ dB if indoor, 0 dB if outdoor
UE Measurement Reports	
SINR report range	$[-5 : 1 : 23]$ dB
Out-of-Sync Threshold	-5 dB

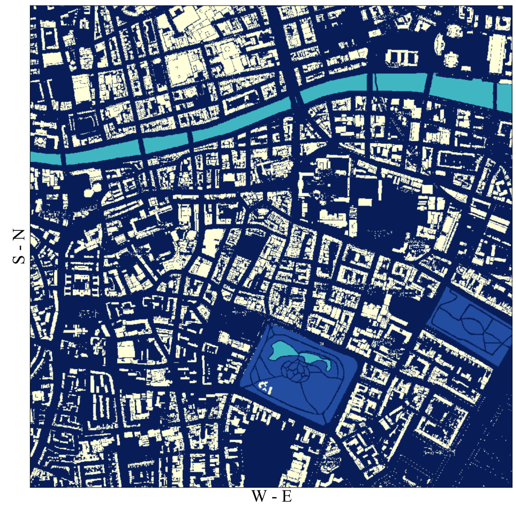


Figure 2: Simulation area.

4.1 UE Distributions

We simulate static UEs in a "full buffer" traffic model. This is analogous to a statistically inferred model of network distributions, whereby measurement statistics on UEs are recorded over a period of time in order to generate a probability distribution of UE placements, which is then used to generate typical distributions of UEs. In accordance with the literature (3GPP, 2014; López-Pérez and Claussen, 2013), UEs are uniformly randomly distributed throughout the network area at an average density of 60 UEs per MC sector, giving 1260 UEs in total. Between 20-40 UE hotspots are placed on the map, with each hotspot having a 90% probability of being located beside a SC. Hotspots range in size from 5 to 25 UEs, with a maximum radius of 24m per hotspot. The number of UEs attached to a SC has a Gaussian distribution, with a mean UE attachment number of 20.76 and a standard deviation of 6.8. Multiple "snapshots" of 40 ms of network run-time (i.e. a full frame $\mathcal{F}$) are taken in order to sample variations across UE distributions. Each such snapshot is analogous to a single data point.

4.2 ABS Setup

In this study, synchronous MC ABS ratios are set according to the rule proposed by the authors in (López-Pérez and Claussen, 2013). MC ABS patterns are front-loaded, such that an MC running an ABS ratio (ABSr) of 2/8 will implement patterns 1 & 2, an ABS ratio of 3/8 will implement the first three patterns, etc. The minimum ABSr that can be implemented by any MC is 1/8 (meaning no MC can transmit permanently), and the maximum ABSr is set at 7/8 (meaning no MC can be entirely muted). This ensures maximal synchronicity of ABS patterns across the entire network, while also guaranteeing at least one subframe in which SCs receive no interference from MCs.

Since ABS patterns are front-loaded and a static "full buffer" model is assumed, network conditions do not change across a full frame $\mathcal{F}$ of 40 subframes. Thus, UE $SINR$ values will be repeated every 8 subframes. Knowing this, schedules for the first 8 subframes can be repeated 4-fold in order to complete the scheduling process for the full frame of 40 subframes. Furthermore, it is possible for the cells to infer from failed transmissions. If a transmission fails in subframe n, then it follows that it will fail again in subframes $n + 8$, $n + 16$, $n + 24$, and $n + 32$ during a full frame. The cell can therefore adjust future scheduled transmissions accordingly, thus minimising failed transmissions and lost data. The out-of-sync threshold Q_{out} is set at -5 dB.

Since the evolved scheduling controllers schedule individual UEs on a per-subframe basis, it is possible that for SCs with low attachment numbers (e.g. < 6 UEs) no UEs could be scheduled for particular subframes. In such a case, all attached UEs are scheduled during any subframes $f \in \mathcal{F}$ where no UEs are scheduled for any SC.

5 Experiments

In earlier work schedulers were successfully evolved for simulated LTE network scenarios with complete & noiseless network state data (Lynch et al., 2016a; Lynch et al., 2016b; Lynch et al., 2017). In this study we move to the real-world environment when necessary information about the true state of the network is both severely limited and somewhat incorrect (true values are quantized, averaged, and given lower and upper bounds, as described in Section 2.4). If evolution is proven to be successful in such a situation, we aim to:

1. examine the best-of-run evolved solution and try to uncover its modus operandi,
2. try to augment the best-of-run evolved solution to improve generalisation, and
3. analyse the performance of the best-of-run evolved solution and to compare it against a state-of-the-art benchmark (López-Pérez and Claussen, 2013).

Finally, we aim to explore the flexibility of the approach for automatic function generation on varied scenarios of both low and high congestion network simulations.

Grammatical Evolution (O'Neill and Ryan, 2003), a form of grammar-based Genetic Programming (McKay et al., 2010), is used in this application via the PonyGE2 implementation (Fenton et al., 2017b). Evolutionary parameters are described in Table 3. While a full-scale parameter optimisation sweep was not undertaken, our earlier research in this domain did undertake a coarse-grained sampling of the parameter space (Fenton et al., 2015) and these are basis of the settings employed here.

Table 3: Evolutionary parameter settings.

Initialization:	Ramped Half-Half
Max initialized derivation tree depth:	20
Overall max tree depth:	20
Number of runs:	100
Population size:	1000
Number of generations:	200
Selection:	Tournament
Tournament size:	1% of population
Replacement:	Generational with elites
Elite size:	1% of population
Crossover type & probability:	Subtree, 70%
Mutation type:	Subtree

In order to find solutions which generalise well, each solution is evaluated on 10 training snapshots of network run-time. As described in Section 4.1, a snapshot is defined by a unique distribution of UEs across a full frame $\mathcal{F}$. Since the simulation area contains 30 SCs, 10 unique snapshots results in a training set of 300 unique SCs. Model selection was performed by subjecting the best evolved solution from each run (as defined by training performance) to unseen test data of 100 snapshots (i.e. a test set of 3,000 SCs). The best solution on test data across all runs was presented as the best overall solution.

5.1 Fitness Function

The industry standard metric for measuring fairness in network performance is given by summing the natural logarithm of the average downlink rates of all UEs, known as the "Sum-Log-Rates" or SLR, as shown in Eq. 4:

$$SLR = \sum_{u'=1}^{|\mathcal{U}|} \left(\log_e \left(\overrightarrow{R}_{u'} \right) \right) \tag{4}$$

where $\overrightarrow{R}_{u'}$ is the average downlink rate of UE u' (as described in Eq. 3) across the full frame of 40 subframes. The target for network optimisation is typically to maximise this fairness utility. Fairness is ensured by taking the logarithm of average rates; the logarithm means that increases in average downlink rates for those with the worst performance will be more heavily weighted than changes in rates for those with the best performance. Essentially, maximising the SLR from Eq. 4 can be interpreted as a "Robin-Hood" scenario, where fairness is maximised by removing resources from those best performers and giving them to those worst performers.

The performance of all evolved solutions is compared against that of a simple greedy baseline scheduling scenario whereby every UE is scheduled for all subframes where their estimated $SINR > Q_{out}$. The overall fitness of a candidate scheduling solution is given by the percentage increase in global network SLR over this baseline scheduling method, as shown in Eq. 5.

$$fitness = \frac{SLR_{evolved}}{SLR_{baseline}} \times \frac{100}{1} \tag{5}$$

In order to compute this, the network must be run once for a full frame of 40 subframes under baseline scheduling methods in order to calculate the baseline SLR for each snapshot. The evolved scheduler is then applied and the network run for a further full frame for that same snapshot in order to obtain the percentage change in global network SLR[8].

5.2 Grammar Design

In grammar-based Genetic Programming, a BNF grammar defines the space which can be searched by evolution. Each SC $s \in \mathcal{S}$ needs to compute optimal schedules for all attached UEs $\mathcal{A}_s$ such that cell throughput is maximised with respect to proportional fairness (i.e. maximise per-cell SLR). As such, the terminal set encompasses various statistics from the domain $(u, f) \in \{\mathcal{A}_s\} \times \mathcal{F},\ \forall s \in \mathcal{S}$, where $\mathcal{A}_s$ represents the set of UEs attached to s. Solutions can use the following terminal set, comprised solely of information available to real-world SC schedulers:

- $|\mathcal{A}_s|$, the number of attached UEs for SC s,

[8]Note that for evolutionary methods the baseline performance only needs to be computed once, rather than at every individual evaluation.

- $|\{SINR_{u,*} \geq Q_{out}\}|$, the maximum number of subframes in which UE u can receive data,
- $\log_2(1 + SINR_{u,f})$, the downlink rate that u would receive in f ignoring congestion[9],
- $\max, \min, \text{avg}, P_{25}, P_{75}\,\{\log_2(1 + SINR_{u,*})\}$, statistics on the uncongested rates experienced by u over all subframes,
- $\max, \min, \text{avg}, P_{25}, P_{75}\,\{\log_2(1 + SINR_{*,f})\}$, statistics on the uncongested rates experienced across all UEs $\mathcal{A}_s$ sharing a given subframe f,
- $\max, \min, \text{avg}, P_{25}, P_{75}\,\{\log_2(1 + SINR_{*,*})\}$, statistics on the uncongested rates experienced across all UEs $\mathcal{A}_s$ across the full frame $\mathcal{F}$,
- an indicator of which subframes each UE is permitted to be scheduled in (i.e. -1 for $SINR \leq Q_{out}$, +1 for $SINR > Q_{out}$), and
- $-0.9, -0.8, \ldots, +0.8, +0.9$, ephemeral constants.

Operators $\max, \min, \text{avg}, P_{25}, P_{75}$ return the maximum, minimum, average, 25th and 75th percentiles of their arguments. Note that each terminal is an array of shape [1, 8] in order to efficiently schedule the entire full frame (a block of 8 subframes repeated 4 times) simultaneously.

For simplicity, each item from the terminal set described above has been given an alias of the range T1 - T21 in the grammar, as defined in Table 4. Since it is desirable that evolved solutions generalise well, and noting that regression-based grammars require less domain knowledge than conditional grammars (as reported in (Hemberg et al., 2013)), a symbolic regression-style grammar was designed for this application. Standard symbolic regression function sets were used, including protected operators of log ($plog(x) = \log(1 + |x|)$), square root ($psqrt(x) = \sqrt{|x|}$), and division (division by 0 returns the numerator). The full grammar is shown in Fig. 3. Bias has been given towards the selection of recursive production choices from the production rule <e> in order to increase the probability of evolving larger solutions. This grammar has a maximum branching factor of 23 (from non-terminal <T>), and can generate a total of 3.280 $\times 10^{733}$ unique solutions up to and including its maximum derivation depth of 20.

```
<s>     ::= <c_t><e>
<c_t>   ::= constrained | threshold
<e>     ::= <r> | <r> | <r> | <T>
<r>     ::= (<e><o><e>) | <ops>(<e>)
<o>     ::= + | - | * | %
<ops>   ::= plog | sin | psqrt | sign
<T>     ::= <sign>0.<n> | <sign>0.<n> | T1 | T2 | T3 | T4 | T5 | T6 |
            T7 | T8 | T9 | T10 | T11 | T12 | T13 | T14 | T15 | T16 |
            T17 | T18 | T19 | T20 | T21
<sign>  ::= - | +
<n>     ::= 0|1|2|3|4|5|6|7|8|9
```

Figure 3: Grammar for evolution of SC scheduling algorithms.

[9] Note that uncongested downlink rates do not include terms for bandwidth or resource blocks.

Table 4: Cell-dependent grammar variables.

Element	Translation	Domain
T1	$\lvert\mathcal{A}_s\rvert$	$\mathbb{Z}_{\geq 0}$
T2	$\lvert\{SINR_{u,*} \geq Q_{out}\}\rvert$	$\mathbb{Z}_{\geq 0}$
T3	$\log_2(1 + SINR_{u,f})$	$\mathbb{R}_{\geq 0}$
T4 - T8	$\max, \min, \text{avg}, P_{25}, P_{75}\,\{\log_2(1 + SINR_{u,*})\}$	$\mathbb{R}_{\geq 0}$
T9 - T13	$\max, \min, \text{avg}, P_{25}, P_{75}\,\{\log_2(1 + SINR_{*,f})\}$	$\mathbb{R}_{\geq 0}$
T14 - T18	$\max, \min, \text{avg}, P_{25}, P_{75}\,\{\log_2(1 + SINR_{*,*})\}$	$\mathbb{R}_{\geq 0}$
T19	UE ID	$\mathbb{Z}_{\geq 0}$
T20	Subframe ID	$\mathbb{Z}_{\geq 0}$
T21	-1 for $SINR_{u,f} \leq Q_{out}$ +1 for $SINR_{u,f} > Q_{out}$	$\in [-1, 1]$

5.3 Mapping Schemes

Since the grammar defined in Fig. 3 is intended for regression-style applications, the solutions described in Section 5.2 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 are therefore considered:

i threshold mapping, where any positive real-valued output of the solution is evaluated to True (with negative outputs evaluated to False), and

ii constrained mapping, where the four subframes (out of a repeating block of eight subframes) with the largest outputs are set to True.

The grammar described in Fig. 3 contains a terminal (`c_t`) which defines which mapping scheme will be used by a specific solution. Figure 4 illustrates these mapping processes. Panel 1 details two UEs, of ID '6' and '2'. The values in each cell represent the real-valued outputs of an arbitrary solution generated by the grammar in Fig. 3 in each subframe 1-8 (recall from Section 2.4 that schedules for each block of 8 subframes are repeated 4-fold in order to complete the full frame of 40 subframes).

Panel 2 shows the decisions made by a threshold mapper based on outputs described in Panel 1. If $Output_{u,f} > 0$ then $schedule_{u,f} \rightarrow$ True else $schedule_{u,f} \rightarrow$ False. Notice that in this instance UE2 will not receive any data because $Output_{2,f} \leq 0, \forall f \in \mathcal{F}$. Threshold Mapping was used to good effect in (Lynch et al., 2016b), however it was noted in (Lynch et al., 2016a) that although each UE effectively gets a different 'Airtime Ratio', it can give rise to solutions that 'play it safe' at the expense of performance.

Panel 3 of Fig. 4 shows how the constrained mapping method sets the largest four cells to True in each column (i.e. each UE will be scheduled for exactly 4 subframes out of $|\mathcal{F}| = 8$). Exploratory experiments conducted in (Lynch et al., 2016a) suggested that an Airtime Ratio of 4/8 gave the best performance (i.e. all UEs receive data in half of the total available subframes). (Lynch et al., 2016a) also noted pros and cons to both methods:

- Each UE is guaranteed to receive data under the threshold scheme.
- Variable airtime ratios across UEs achievable under the threshold scheme can enforce fairness.

Output

Subframe	UE '6'	UE '2'
1	2.4	-1.1
2	3	-2
3	1	-3
4	4	-5.1
5	8	-4
6	2	-8
7	4.4	-9
8	-8.3	-7

Threshold

Subframe	UE '6'	UE '2'
1	T	F
2	T	F
3	T	F
4	T	F
5	T	F
6	T	F
7	T	F
8	F	F

Constrained

Subframe	UE '6'	UE '2'
1	F	T
2	T	T
3	F	T
4	T	F
5	T	T
6	F	F
7	T	F
8	F	F

Figure 4: Mapping schemes.

- Congestion is guaranteed to be low with a constrained scheme with lower Airtime Ratios.
- Better solutions are found earlier in runs when the constrained mapping scheme is adopted.

6 Results & Augmentation

Training and test performance across 100 independent runs are shown in Figs. 5a and 5b respectively. Runs were parallelized across 80 cores of a Mac Pro cluster, each at 2.66 GHz. The total cumulative CPU time for all 100 runs was 17 days, 2 hours, 23 minutes, and 53 seconds. Average completion time for a single run was 4 hours and 10 minutes. It can be seen from Fig. 5a that evolution is indeed capable of evolving viable solutions given quantized, averaged, and limited information about the true state of the network. Furthermore, Fig. 5b shows that all best-of-run solutions show positive performance on unseen test data (i.e. all evolved solutions are capable of improving on the naive baseline scheduling technique described in Section 5.1).

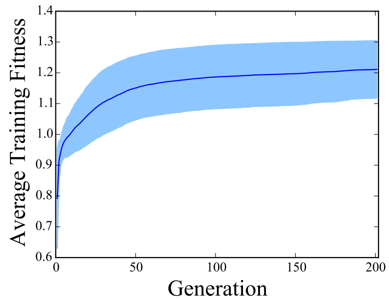
The best evolved solution across 100 independent runs (based on test performance, as described in Section 5) is shown in Fig. 6. This solution uses the "threshold" mapping scheme, meaning the number of subframes in which a UE can be scheduled is not fixed. While it appears to be a grey-box solution, it can actually be simplified quite considerably, resulting in the solution shown in Eq. 6:

$$\frac{\log(1 + |T12 - T17|) + \left(T3 \times \sqrt{|T10|}\right) - (T20 \times T6)}{0.5 - T4 + (T6 \times T17 \times (T4 - T5))} \tag{6}$$

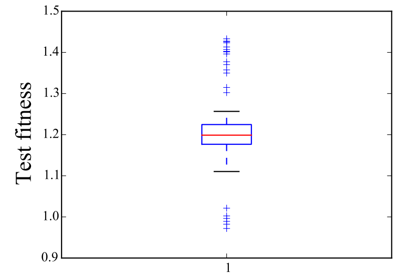
A particular strength of EC techniques such as the one utilized in this paper is that solutions are typically transparent and can be examined by domain experts in order to understand their behaviour. This is analysis is performed in the following sections.

6.1 Examination of Best Evolved Solution

The solution described in Eq. 6 uses 8 separate terminals: T3, T4, T5, T6, T10, T12, T17 and T20. All terminals are used exactly once, except for terminals T4 and T6, which



(a) Graph of average evolutionary performance on training dataset. The blue region indicates standard deviation.



(b) Box plot of distribution of results on test dataset.

Figure 5: Plots of evolutionary performance on training and test datasets.

```
sin((sign(pdiv(((log(1+abs((T12-T17)))+(T3*sqrt(abs(T10))))-(T20*T6)),
(log(1+abs(log(1+abs(sin(T6)))))+(((T6*T17)*(T4-T5))-T4))))+log(1+
abs((sign(pdiv(pdiv((T9-T10),(T5+T6)),sign(log(1+abs(T12)))))*(log(1+
abs(sqrt(abs((T5*T20)))))-T15))))))
```

Figure 6: Best evolved solution, unsimplified. This solution has a maximum depth of 20, with 159 nodes in the derivation tree.

appear twice. One constant is present[10], and all terminals (with the exception of T20) directly reference uncongested downlink rates. The solution mainly consists of a single division operator, and as such its performance can be analysed to some degree. Since the "threshold" mapping scheme is used, the equation is interpreted solely with respect to the sign of its output. Therefore, both the numerator and denominator of the equation can be examined to assess the sign of their respective outputs.

Both the numerator and denominator can be broken down into distinct parts:

Numerator

$$\log\left(1 + |(T12 - T17)|\right) \tag{7}$$

$$\left(T3 \times \sqrt{|T10|}\right) \tag{8}$$

$$-\left(T20 \times T6\right) \tag{9}$$

Denominator

$$0.5 - T4 \tag{10}$$

$$T6 * T17 * (T4 - T5) \tag{11}$$

The terminal T3 (the uncongested downlink rate of UE u in subframe f; $\log_2(1 + SINR_{u,f})$) occurs only once, in expression 8. Its effect on the overall outcome

[10] This constant was generated in the simplification process.

of the solution is low since it is multiplied by the square root of terminal T10 ($\min\{\log_2(1+SINR_{*,f})\}$), resulting in very small numbers. However, terminal T20 (the subframe ID) has a large effect on the output of the numerator, since it is multiplied by the average uncongested downlink rate for UE u across the full frame $\mathcal{F}$ (T6, as seen in expression 9, whereas the other two components of the numerator are relatively small, being composed of logarithms and square roots). Since python indexes from zero, T20 (i.e. expression 9) will be zero in the first subframe, and negative thereafter. Therefore, the numerator of the equation will always be positive in the first subframe (since expressions 7 and 8 will always return a positive result). However, in subsequent subframes the numerator is almost guaranteed to be negative as the T20 component increases; experimental observations have shown that the numerator of the equation from expression 6 is always negative for $f \geq 2$, and is negative for $f = 1$ in the vast majority of cases.

The denominator is similarly straightforward to examine. Since T4 ($\max\{\log_2(1+SINR_{u,*})\}$) is greater than 0.5 for all but the very worst-performing UEs whose $SINR$ is less than -3dB, expression 10 will in general be positive. Expression 11 then essentially separates UEs by performance. Since each UE only reports two $SINR$ values to the hosting cell, and since maximum $SINR$ values are capped at 23 dB, it is possible for high-$SINR$ UEs to report the same $SINR$ values during both ABS-overlapping and non-ABS-overlapping subframes (i.e. all terminals T4 - T8 will be identical across all subframes). Therefore, expression 11 will evaluate to 0 for these UEs since both their maximum (T4; $\max\{\log_2(1+SINR_{u,*})\}$) and minimum (T5; $\min\{\log_2(1+SINR_{u,*})\}$) $SINR$ values will be identical. It can therefore be appreciated that lower-$SINR$ UEs (i.e. those for whom T4 and T5 are different) will impose a gradient on the denominator.

In general, it can be said that the numerator of Eq. 6 describes subframe quality, whereas the denominator describes UE performance. However, unlike in (Weber and Stanze, 2012) evolution has included the ability to address corner cases, mainly by imposing a gradient on UE performance in the denominator through expression 11. Broadly speaking, if a UE's maximum uncongested downlink rate (T4) is less than 0.5 plus expression 11, they will be scheduled during ABS-overlapping slots (the benchmark scheduling method operates in a similar fashion (López-Pérez and Claussen, 2013)). The evolved method generally separates UEs into two discrete groups; those to be scheduled during ABS-overlapping subframes, and those to be scheduled during non-ABS-overlapping subframes. Knowing this, it is possible to further generalise and abstract the solution.

6.2 Further Simplification and Augmentation

Given that we can readily interpret the output solutions from evolutionary computation (which themselves out-perform the state-of-the-art solutions), these solutions can be adapted by human experts to produce further enhancements facilitating a process of augmented design. The evolved solution presented in Eq. 6 presents a heuristic for scheduling UEs that is highly fit for its environment. However, further examination of the solution indicates potential pitfalls and indicators of over-fitting to the incubation environment. It has already been explained that the T20 component of the numerator (i.e. the subframe ID) has a large effect on the output of the equation by always resulting in a positive numerator during the first subframe. However, the good performance realised by the solution is mainly due to the fact that:

i ABS patterns are front-loaded in our simulation, and ii ABS ratios rarely exceed 1/8 in our simulation.

Good performance is observed from this solution as the first subframe (i.e. subframe ID 0) is always guaranteed to offer the highest channel quality due to the front-loading of ABS frames in our simulation environment (as detailed in Section 4.2). As reported in (Hemberg et al., 2013), the solution is exploiting the assumptions of the simulation model; it is providing the best results given its incubating environment. It can therefore be appreciated that this method will break down either when:

i ABS patterns are *not* front-loaded (meaning the first subframe is not guaranteed to provide the best performance), or

ii the ABS ratio is greater than 1/8.

Both of these issues can be resolved by completely abstracting the numerator of Eq. 6 and replacing it with its implicit meaning. A far more robust solution is to replace the entire numerator with +1 during ABS overlapping subframes and -1 during non-ABS overlapping subframes, as shown in Eq. 12.

$$\frac{[+1\, during\, ABS;\ -1\, during\, non\, ABS]}{0.5 - T4 + (T6 \times T17 \times (T4 - T5))} \tag{12}$$

The resultant solution has far greater generalisation than the original, and is more robust to changes in ABS patterns. Furthermore, only 4 terminals are used in the entire equation:

i T4, the maximum uncongested downlink rate for UE u,

ii T5, the minimum uncongested downlink rate for UE u,

iii T6, the average uncongested downlink rate for UE u, and

iv T17, the 25$^{\text{th}}$ percentile of all uncongested downlink rates $\forall u \in \mathcal{A}_s, \forall f \in \mathcal{F}$.

The final solution therefore becomes very simple and can be easily interpreted. As with (Jiang and Lei, 2012; Weber and Stanze, 2012; López-Pérez and Claussen, 2013), the numerator merely initialises two separate queues (ABS-overlapping and non-ABS-overlapping), while the denominator defines which UEs are scheduled in the respective queues. A positive output from the denominator means the UE will be scheduled during ABS-overlapping subframes, a negative output means the UE will be scheduled during non-ABS-overlapping subframes. Note that this methodology is nearly identical to the benchmark technique, except it does not require a recursive function to populate the queues. This solution does not break down as a result of changes to ABS patterns or ratios, it addresses corner cases through its use of gradient, and is shown in subsequent sections to out-perform both baseline and benchmark scheduling methods on all observed snapshots. Importantly, the performance gains remain similar between the original evolved solution and the generalised simplified solution on the test snapshots described in Section 4.1. The performance evaluation discussed hereafter details the generalised solution shown in Eq. 12.

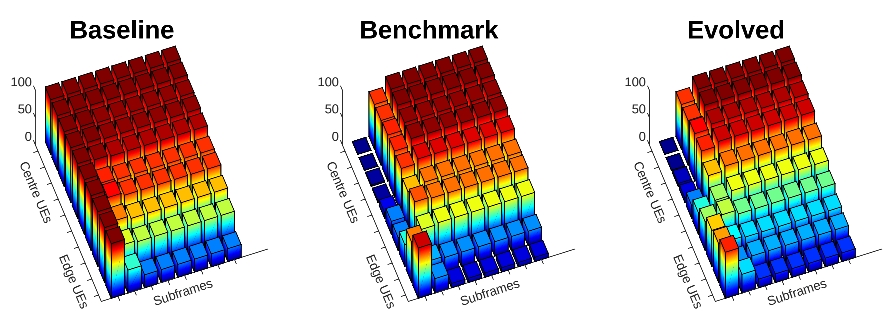
7 Performance Evaluation

The best evolved solution was run on an unseen test set of 100 network snapshots (as described in Section 4.1). A number of insights can be made into the performance of the best evolved scheduler by examining a variety of different metrics: the scheduling semantics of the evolved method (discussed in Section 7.1), the generalisation of the evolved method (how well it performs on cells of varying sizes, discussed in Section 7.2), and improvements in ultimate data rates and Sum-Log-Rates (discussed in Section 7.3). The performance of the best evolved solution is compared against:

1. the baseline scheduling method, as described in Section 5.1, and
2. a state-of-the-art human-designed benchmark scheduling method, as described in (López-Pérez and Claussen, 2013).

7.1 Scheduling Semantics

The plots in Fig. 7 display heatmaps of the scheduling semantics for the first 8 out of 40 subframes[11] of SCs with exactly 10 attached UEs, averaged across 100 network snapshots. The 10 attached UEs are sorted from worst to best with respect to average *SINR*. Recall that ABS patterns are front-loaded in our simulation (as described in Section 4), meaning the first few subframes in every repeating block of 8 subframes are guaranteed to have the least MC interference. It can be seen from the heatmaps that in general the worst performing UEs in every cell (the leftmost columns on the heatmaps) are scheduled by all methods in the best available subframes (the topmost rows on the heatmaps). Conversely, the best performing UEs are relegated to those later subframes where interference is high.



(a) Baseline scheduling semantics

(b) Benchmark scheduling semantics

(c) Evolved scheduling semantics

Figure 7: Heatmaps of UE scheduling semantics for baseline, benchmark (López-Pérez and Claussen, 2013), and evolved scheduling methods. Each heatmap represents the conglomeration of the scheduling semantics of all cells with exactly 10 attached UEs across 100 network snapshots. Note that the color scale across all methods is identical.

The semantics for the baseline method of scheduling (as described in Sec. 5.1) are shown in Fig. 7a. The dark red colors indicate where individual UEs are consistently

[11] Recall that since ABS patterns repeat every 8 subframes (as shown in Table 1), and assuming a static "full buffer" traffic model, each repeated block of 8 subframes in a full frame of 40 subframes will be identical, as detailed in Section 4.2.

scheduled. As expected, it can be seen that all UEs are scheduled in the first subframe since there is no color variation in the heatmap. However, in subsequent subframes MC interference increases as fewer MCs implement ABSs. Thus, $SINR$s decrease and less UEs are eligible to be scheduled (due to their $SINR \leq Q_{out}$) in each subsequent subframe. It is only the very best performing UEs (those UEs with the highest $SINR$) that can be scheduled consistently for all subframes.

While both Figs. 7b and 7c are broadly similar in their approach, subtle differences can be appreciated between the semantics of the benchmark and those of the evolved scheduling method. Analysis of the network simulation (not discussed here) revealed that the maximum ABS ratio used by any MC across 100 network snapshots was 2/8. Since ABSs are front-loaded in our simulation, it follows that all MCs are transmitting for all snapshots in subframes 3-8. Thus, there are no changes in $SINR$ values for static UEs (as network interference does not change).

As described in Section 6.2, both the benchmark and evolved scheduling methods only schedule UEs in ABS-overlapping or non-ABS-overlapping subframes (i.e. they cannot distinguish between subframes with identical $SINR$ values). This is evident in Figs. 7b and 7c, where subframes 3-8 have identical semantics. The difference between the two methods, however, lies in *which* UEs are scheduled for which slots. Whereas the benchmark method schedules UEs in either ABS-overlapping or non-ABS-overlapping slots such that the performance of the worst UE in either slot is equalised (López-Pérez and Claussen, 2013), the evolved method selects for either queue UEs based on a comparison of their average performance against the cell-wide 25$^{\text{th}}$ percentile performance. It would appear that the evolved method schedules fewer cell-center UEs than the benchmark, giving greater preference to low-$SINR$ cell-edge UEs.

One interesting observation is that neither the benchmark nor the evolved methods unilaterally schedule the singe worst UE in the single best available subframe. In Fig. 7a it can be seen that the very left-most UE (the worst performing UE in the cell) is scheduled consistently in subframe 1 (as indicated by the deep red color). Indeed, for maximising throughput with respect to proportional fairness it would be expected that the worst-performing UE would be given the strongest airtime advantage. However, Figs. 7b and 7c show a much lighter shade of red in that same cell, indicating that the worst performing UE is not guaranteed airtime in the best available slot under these methods. With both benchmark and evolved methods, it is only the best-performing UEs (those right-most cell center UEs on the plots) that are consistently scheduled in any of the available subframes (the subframes with the highest MC interference).

7.2 Solution Generalisation

The generalisation of the evolved scheduling method can be inferred by examining the performance of cells of specific sizes (based on UE attachment numbers). Each cell is examined before the evolved scheduling method is applied (i.e. when the baseline scheduling method is applied, as described in Section 5.1), and once more after the evolved method is applied in order to ascertain the percentage performance improvement with regards to cell SLR in that cell. The average percentage performance improvements of cells of corresponding sizes are then calculated in order to investigate the performance of the evolved scheduler across all cell sizes (an indication of good generalisation).[12]

Figure 8 compares the average percentage performance improvement of cells of varying UE attachment numbers when running:

[12] Note that all results are on unseen test data.

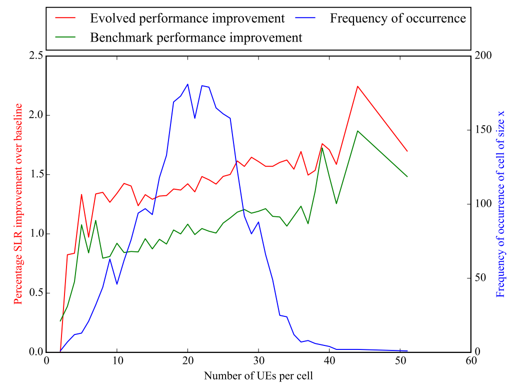


Figure 8: Standard simulation scenario.

i the baseline scheduling method,

ii the benchmark scheduling method, and

iii the evolved scheduling method.

Overlaid on the plot is the distribution of the frequency of occurrence of cell sizes, as described in Section 4.1. A straight horizontal line across the plot would indicate minimal variation across different cell sizes, signifying good robustness with respect to cell load and similar performance improvements regardless of UE attachment numbers.

It can be seen from Fig. 8 that the evolved scheduling method out-performs both the baseline and benchmark scheduling methods for all cases, regardless of cell attachment numbers. Taking benchmark improvements over baseline scheduling as 100%, the evolved scheduling method produces an average improvement of 26% in performance over the benchmark. Furthermore, the performance of the evolved scheduler can be seen to improve with larger cell sizes, indicating it can cope with high network congestion. This indicates that the evolved & augmented solution is highly generalisable.

7.3 Downlink Rates

Figure 9 shows the percentage changes in downlink rates over the baseline scheduling method. In terms of outright downlink rates expressed as a percentage improvement over the baseline rate, the single best improved UE in the network (with respect to percentage improvement) sees on average around a 200% increase in downlink rates. Up to the 5^{th} percentile, all UEs see greater than a 100% increase in downlink rate performance over the baseline scheduling method. Furthermore, the top 60% of UEs in the network see an average downlink rate improvement of 15% over both baseline and benchmark. Unlike the benchmark scheduling method, no UEs under the evolved scheme see worse performance than the baseline.

Notably, when compared to the benchmark the evolved scheduling heuristic sees smaller average improvements in downlink rates for lower-percentile UEs. This is because the evolved method was trained to maximise cell throughput with respect to proportional fairness (through the use of the SLR fitness function defined in Section 5.1), whereas the benchmark method eschews global fairness in favor of equalising the

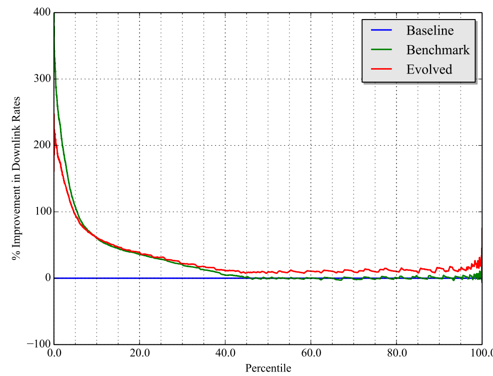
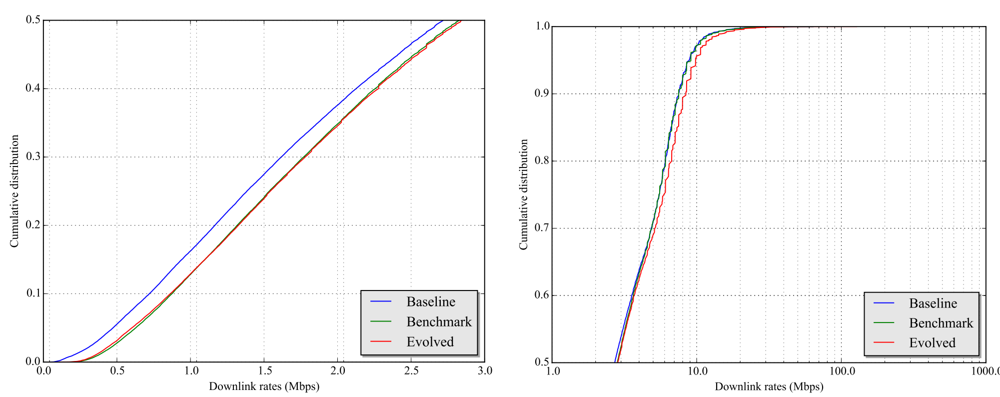


Figure 9: Percentage improvement in downlink rates over baseline for SC-attached UEs.

performance of two select UEs per cell. Therefore, while the lower percentile downlink rates may be marginally higher for the benchmark, the evolved method actually produces a fairer network environment (in terms of the industry-standard SLR metric) as the performance of all UEs is taken into account. This can be seen in Fig. 8, which shows that all SCs in the network simulation see an improvement in SLR over both baseline and benchmark in all observed cases.



(a) CDF plot of UE downlink rates, low-$SINR$ UEs.

(b) CDF plot of UE downlink rates, high-$SINR$ UEs.

Figure 10: Cumulative Distribution Function plots of downlink rates for all SC-attached UEs. Figure 10a shows the first 50% of UEs, with the second 50% shown in Fig. 10b.

While Fig. 8 shows the average improvement in SLR for cells of varying sizes under the different methods, it only describes part of the performance of the examined scheduling methods. Figures 10a and 10b show the Cumulative Distribution Function (CDF) plots of the downlink rates of all SC-attached UEs on the network across all 100 test snapshots for the three observed methods (baseline, benchmark, and evolved). UEs are plotted on the y-axis, with their average downlink rates plotted on the x-axis[13]. As

[13]Note that Figs. 10a and 10b plot two halves of the same data, with Fig. 10a detailing the worst performing

such, these graphs directly describe the average downlink performance of each individual SC-attached UE in the network.

Figure 10a shows that the evolved scheduling method is able to match the performance gains for low-$SINR$ UEs that the benchmark method is able to achieve over baseline scheduling methods. Figure 10b shows that the benchmark method is only capable of matching the performance of the baseline scheduling method for top-end UEs (those UEs with the highest $SINR$). In effect, the benchmark technique excessively sacrifices the performance of these UEs in order to improve the performance of low-end UEs. However, the evolved method is able to provide a $\sim$1 MB/s performance improvement for the top 30% of UEs over both baseline and benchmark. Significantly, this implies that the evolved method is able to provide similar improvements to the performance of low-$SINR$ UEs to the benchmark, *without* its attendant sacrifice in top-end UE performance. Since the objective of the evolved method was to maximise network throughput with respect to proportional fairness, the end result for the evolved scheduling method is a higher average global network SLR than with both baseline and benchmark methods.

A two-sample Kolmogorov-Smirnov test was performed on the data from Fig. 10 in order to check for statistically significant differences between the performance of the evolved method over both the baseline and the state-of-the-art human-designed benchmark techniques. Taking an alpha value of 0.05, a p-value of 1.07e-05 means we can confidently reject the null hypothesis that the evolved solution produces the same performance to that of the baseline. Similarly, a p-value of 8.48e-08 means that we can confidently reject the null hypothesis that the evolved solution produces the same performance to that of the benchmark technique. We can therefore conclude with a high confidence that the performance of the evolved technique is statistically significantly better than both the simple baseline and the state-of-the-art human-designed benchmark.

8 Method Generalisation

As it is not commercially viable for network operators to develop unique human-designed algorithms for an array of unique scenarios, operators are forced to utilize potentially sub-optimal "one size fits all" solutions which can cater for all eventualities. However, one of the main advantages of using an automatic algorithmic method over human design to generate solutions is that the parameters of the problem can easily be changed without the need for re-investing in human input. Thus far in this paper, Sections 4 and 5 have described how an experiment can be set up, with the results of a single run being examined in detail in Section 6. It is a simple matter to change certain input parameters for the simulation setup, thus changing the specialties of the evolved solutions.

Figure 8 described the general performance of the evolved algorithm from Eq. 12. However, this solution was trained and tested on variations of the normal UE distribution shown in blue in Fig. 8. By changing the distribution density of UEs or the number of SCs in the network simulation, we can easily change the distribution patterns to simulate certain scenarios. Once a new distribution is set, a new problem is effectively produced. It is then a simple matter to re-run the evolutionary process described in Section 5 to evolve a new solution for this particular scenario[14]. Thus, we are

---

UEs (i.e. up to the 50th percentile), and Fig. 10b detailing the best performing UEs (i.e. from the 50th to the 100th percentile).

[14]Following examination of the evolved solution from Fig. 6, a new terminal "ABS" was added to the

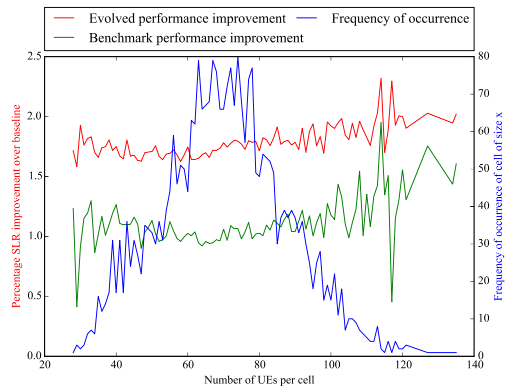


Figure 11: Generalisation on high congestion network.

able to examine the ability of the evolutionary system described in Sections 4 and 5 to produce solutions which generalize well to different UE distributions. The following results summarize the performance of individual solutions evolved for their respective distributions. The performance results of the evolved schedulers are similar to those discussed in Section 6, and as such the following Sections 8.1 and 8.2 focus more on the examination, simplification, and augmentation of the solutions themselves rather than their outright performance.

8.1 High-Congestion Network Scenario

Figure 11 represents the performance of an evolved model from a network of highly congested (i.e. highly overloaded) SCs. Such a scenario might be indicative of a high traffic situation such as football stadia, city-center parades, or festivals. This high congestion was achieved by increasing the average number of UEs per MC sector from 60 (the industry-accepted standard (3GPP, 2014)) to 238, i.e. 5,000 total UEs in the simulation environment, with only 30 SCs. Taking benchmark SLR improvements over baseline scheduling as 100%, the evolved scheduling method for high congestion scenarios produces an average SLR improvement of 37.81% in performance over the benchmark on the same scenario.

The best evolved solution is shown in unsimplified form in Fig. 12. As with the previous solution, this solution also uses the threshold mapping scheme. What is immediately notable is the size of the solution; it is far larger than that shown in Fig. 6. The size of the solution alone makes augmentation and simplification far more difficult than with the previous case.[15] However, a number of insights can still be made.

The form of the solution shown in Fig. 12 follows that of Eq. 12, namely that it consists of a single fraction. Therefore, as the threshold mapping scheme is used it is simply the *sign* of the solution which dictates whether or not a UE u will be scheduled in a subframe f. As described in Section 6.1, we only need to examine the outputs of both numerator and denominator of this solution to gain deeper insight into its overall *modus operandi*.

Examination of the denominator of the solution reveals that it works in exactly the same fashion as the numerator from Eq. 12. Since the encompassing function of the

terminal set. This new terminal operates in the manner of the numerator from Eq. 12.

[15]Note that the only form of bloat control in use is the maximum overall depth limit of derivation trees.

```
pdiv(((log(1+abs((T14-pdiv((pdiv(T21,T17)-log(1+abs(T16))),pdiv((T5+
T5),(T17*T17))))))-(((log(1+abs((T5+T11)))+sqrt(abs(T16)))*log(1+
abs(sqrt(abs(log(1+abs(T4)))))))-log(1+abs((T6-sin(sqrt(abs(
T4))))))))*sqrt(abs((pdiv(pdiv((sin(T5)*(T5*T16)),sin(+0.0)),(sqrt(
abs(sin(T10)))+sqrt(abs(log(1+abs(T4))))))*pdiv(((sign(T21)-T2)*
sin(log(1+abs(T5)))),pdiv(pdiv((T20-T6),sin(T18)),T4)))))),

sign(pdiv(pdiv(sqrt(abs(pdiv(sign(sign(T21)),(T3+(T14+T9))))),pdiv(
sin(sign(pdiv(ABS,T19))),(((T4-T21)+(T6*T5))*(log(1+abs(T10))*
sign(T4))))),(((sqrt(abs(log(1+abs(T6))))+(T6*T5))*log(1+abs(T5)))+
pdiv(((pdiv(T5,T1)+(T21*T10))-sign((T2-T12))),((sqrt(abs(T12))+(T19*
T5))+pdiv((T17*T20),sqrt(abs(T12))))))))
```

Figure 12: Best evolved high congestion solution, unsimplified. The solution comprises a single fraction, with numerator and denominator distinguished by the paragraph break above. This solution has a maximum depth of 20, with 527 nodes in the derivation tree.

$$\frac{\log\left(1+\left|T14-\frac{\frac{T21}{T17}-\log(1+T16)}{\frac{2\times T5}{T17^2}}\right|\right)-\left(\left(\log(1+T5+T11)+\sqrt{T16}\right)\times\log\left(1+\sqrt{\log(1+T14)}\right)\right)-\log\left(1+\left|T6-\sin\left(\sqrt{T4}\right)\right|\right)}{[+1\,during\,ABS;-1\,during\,non\,ABS]} \tag{13}$$

denominator is the sign of the expression, the denominator only needs to be examined for the sign of its outputs. The denominator uses 15 different terminals, including the newly introduced "ABS" terminal as described in Section 8. However, upon deep analysis it transpires that the only component of the entire denominator that affects the actual sign of its output is the single use of the terminal "ABS" itself. Therefore, the entire denominator can be replaced by the single terminal "ABS", i.e. the numerator of Eq. 12. Thus, it can be appreciated that the evolved solution from Fig. 12 operates in the same manner as both that of Eq. 12 and of the benchmark, i.e. by scheduling UEs in either ABS or non-ABS overlapping subframes. Identification of *which* UEs are scheduled during either slot comes from examination of the numerator.

While the denominator of the solution from Fig. 12 can be entirely reduced and replaced, the numerator is less easy to simplify beyond simple contractions and removing of obsolete functions such as unnecessary "absolute value" calls on terminals which are always positive. The final simplified and augmented form of the high congestion solution shown in Fig. 12 is described in Eq. 13.

Opaque though it may seem, deeper insight into the operation of Eq. 13 can be gained by examining which terminals are used. Overall, the numerator from Eq. 13 uses 12 terminals. By consulting the cell-dependent grammar variables table shown in Table 4, clusters of terminals become apparent. The numerator obtains the majority of its information from terminal groups T4-T8 (UE-specific data across all subframes) and T14-T18 (global data across all UEs and all subframes). By comparing the max (T4), min (T5) and average (T6) performance of a single UE across all subframes against the max (T14), average (T16) and 25^{th}-percentile (T17) performance of all UEs across all subframes, a gradient is imposed on which UEs are to be scheduled during specific subframes.

This is similar to how Eq. 12 operates; so much so, in fact, that the entire numerator from Eq. 13 can be replaced with the entire denominator from Eq. 12 with nigh on identical performance. Significantly, the converse is true for the standard distribution described in Section 4.1; Eqs. 12 and 13 can be entirely swapped to run on their respective distributions with no appreciable loss in performance.[16] The implications of this are significant. The main difference between the normal UE congestion and high UE congestion scenarios is in the mean and variance of their respective normal distributions. Therefore, it can be inferred that if the distribution of SC attachment numbers is normal, Eqs. 12 or 13 can provide a successful scheduling solution. Furthermore, since evolution has evolved two highly similar solutions for these two problems, one can assume that similar solutions will be successful for similar distributions.

8.2 Low-Congestion Network Scenario

As with Figs. 8 and 11, Fig. 13 shows the generalisation performance of a solution evolved under a low congestion scenario, created by increasing the number of SCs in the network from 30 to 100 while retaining the original density of 60 UEs per MC sector. This scenario would be in line with standard network practice of cell densification to decrease congestion (Bian and Rao, 2014). This scenario has average SC attachment numbers of 9.56 UEs, but the distribution is heavily right-tailed (as cells cannot have less than 0 attached UEs). Thus, the distribution differs significantly from the normal distributions discussed previously. Taking benchmark SLR improvements over baseline scheduling as 100%, the scheduling method presented in Eq. 14 produces an average SLR improvement of 50.84% in performance over the benchmark. Furthermore, it can be seen from Fig. 13 that in some cases of extremely low attachment numbers, the benchmark method breaks down (i.e. the green line dips below the x-axis, indicating that the benchmark performs worse than the simple baseline method described in Section 5.1).

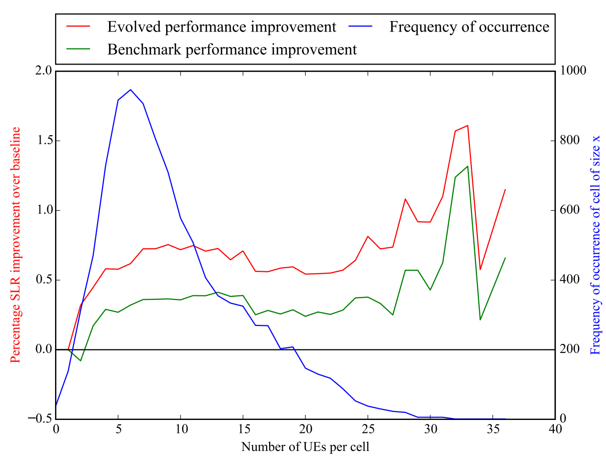


Figure 13: Generalisation on low congestion network.

The best evolved solution for the low congestion distribution is shown in its unsimplified form in Fig. 14. As with all previous solutions, this solution uses the "threshold" mapping scheme. Similarly to the original solution shown in Fig. 6, while difficult to interpret from a first glance this solution can be somewhat simplified and augmented

[16]The performance is so similar that there is difficulty distinguishing between generalisation graphs (e.g. Fig. 11) produced by the two solutions on the same dataset.

```
T20-(pdiv(T13,T14)-pdiv((((sign(sin(T15))+sign((T16*T10)))*(T6*pdiv(
pdiv(T17,T5),pdiv(T5,T18))))+(((T13*(T4-T4))-(sqrt(abs(T9))+(T21+T18)))
+pdiv(((T21+T20)-T17),sqrt(abs(pdiv(T11,ABS)))))),(((ABS+pdiv(ABS, T4))
-pdiv(pdiv(pdiv(T9,T8),sqrt(abs(T19))),(T13*T9)))+((sign((T16*T5))+
pdiv(ABS,sign(T20)))-sin((log(1+abs(T5))*log(1+abs(T20))))))))
```

Figure 14: Best evolved low congestion solution, unsimplified. This solution has a maximum depth of 20, with 293 nodes in the derivation tree.

$$T20 - \frac{T13}{T14} - \frac{(2 \times T6 \times \frac{\frac{T17}{T5}}{\frac{T5}{T18}}) - \sqrt{T9} - T21 - T18 + \frac{T20+T21-T17}{\sqrt{T11}}}{2 \times ABS + \frac{ABS}{T4} + 1 - sign(T20)} \tag{14}$$

by domain experts. The augmented solution presented in Eq. 14 retains the same performance characteristics of the original, but with a smaller representation. However, there are major differences between this solution and the solutions discussed in Sections 6 and 8.1.

Firstly, the solution shown in Eq. 14 contains three separate components, meaning it differs greatly in its operation from the form of both the previous augmented solutions described in Eqs. 12 & 13, and of the benchmark scheduling method. Whereas these methods schedule UEs in either ABS or non-ABS overlapping subframes, the standalone use of T20 (subframe ID) at the beginning of Eq. 14 indicates that the output scheduling semantics of this solution will vary across all subframes.

The final augmented solution contains 12 unique terminals, with 19 terminals overall being used. Again, further contrast can be made between the operation of this solution and that of the previous solutions from Eqs. 12, 13:

1. The solutions examined previously require far less terminals to evaluate their solutions,

2. They use clear clusters of terminals (as indicated by Table 4) while Eq. 14 uses a more even spread of terminals, and

3. They use more information about the relative performance of individual UEs, whereas Eq. 14 makes wide use of terminals T20 (subframe ID), T21 (SINR quality indicator), and ABS, implying it is relying more heavily on the specific attributes of individual subframes than either of the previous solutions in order to accurately schedule UEs.

The complex relationships between these terminals make the solution difficult to interpret further; merely changing or removing any single terminal significantly degrades the performance of the solution. Furthermore, it is not possible to use this solution on anything other than the right-tailed distribution shown in Fig. 13. Indeed, implementing the solutions from Eqs. 12 or 13 on the low congestion scenario sees performance worse than the simple baseline. It can therefore be inferred that the right-tailed low congestion distribution is a special case scenario.

9 Conclusions and Future Work

Evolutionary computation has been shown to be capable of producing human-competitive solutions that improve upon the performance of a state-of-the-art human-designed benchmark across a variety of scenarios, despite being given very poor quality information about the true state of the problem. Extensive analysis of these solutions reveal that EC has uncovered a new technique for scheduling SC-attached UEs which is not only generalisable but is both intuitive and easy to implement. Furthermore, evolution has been shown to have *twice* produced a solution which conforms to accepted scheduling frameworks which match the literature, despite evolution being given no information about this form of solution and despite being trained on different datasets. These solutions do not break down as a result of changes to ABS patterns or ratios, and address corner cases through their use of gradient.

These presented methods are human-competitive in the traditional Koza sense (Koza, 2010), as:

1. they are equal to or better than a result that was accepted as a new scientific result at the time when it was published in a peer-reviewed scientific journal, and

2. they are equal to or better than the most recent human-created solution to a long-standing problem for which there has been a succession of increasingly better human-created solutions.

Specifically, the evolved solutions manage to significantly increase cell throughput with respect to proportional fairness over a state-of-the-art human-designed benchmark without excessively sacrificing the performance of high-$SINR$ UEs. In the standard scenario, 30% of SC attached users are shown to achieve a $\sim$1 MB/s performance improvement under the evolved scheme, while the top 60% of all SC-attached users see an average downlink improvement of 15% over the benchmark. Taking benchmark improvements over a simple baseline scheduling method as 100%, the presented method produces an average improvement of 26% in per-cell Sum-Log-Rate performance over the benchmark scheme. Low UE congestion network scenarios show average per-cell SLR improvements of 50.84%, while high congestion scenarios show average improvements of 37.81%.

As network demand rises, SC densification is seemingly the most cost-effective method for operators to increase capacity within their networks. However, evolutionary computation provides a means to not only automatically generate tailored algorithms for specific scenarios, but for human experts to further augment and enhance these solutions. Targeted solutions can be evolved for different network deployments that are capable of handling highly congested/overloaded SCs. This presents a low-cost software alternative to hardware upgrades, thus postponing the need for network operators to supplement their networks with additional SCs. Moreover, higher attachment numbers allow for more fine-grained performance trade-offs, enabling increased fairness.

Future 5G systems are increasingly moving towards software-defined networks. Furthermore, existing 4G architecture will remain in concurrent operation with newly implemented 5G networks. As such, there remains a need for automatic tools such as the ones presented in this paper in future networks. Future work may look at evolving schedulers on multiple different UE distributions, rather than solely on the normal distribution described in Section 4.1. In theory, this should lead to an even more robust solution. In addition, a more robust model selection could be performed with the use of validation sets and by subjecting the entire final population of each run to test data. Finally, bloat control methods could be utilized in order to remove inactive aspects of solutions and to reduce overall solution size.

Acknowledgements

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

References

3GPP (2014). 3rd Generation Partnership Project (3GPP). http://www.3gpp.org/.

3GPP E-UTRA (2010). Further advancements of E-UTRA physical layer aspects. Technical Report TR 36.814 v9.0.0, 3GPP Technical Specification Group Radio Access Network, Evolved Universal Terrestrial Radio Access (E-UTRA).

3GPP E-UTRA (2012). Mobility enhancements in heterogeneous networks. Technical Report TS 36.839 v11.0.0, 3GPP Technical Specification Group Radio Access Network, Evolved Universal Terrestrial Radio Access (E-UTRA).

Aliu, O. G., Imran, A., Imran, M. A., and Evans, B. (2013). A survey of self organisation in future cellular networks. *IEEE Communications Surveys Tutorials*, 15(1):336–361.

Andrews, J. G., Singh, S., Ye, Q., Lin, X., and Dhillon, H. S. (2014). An overview of load balancing in hetnets: Old myths and open problems. *IEEE Wireless Communications*, 21(2):18–25.

Bian, Y. Q. and Rao, D. (2014). Small Cells Big Opportunities. Global Business Consulting, Huawei Technologies Co. Ltd.

Cisco (2015). Cisco visual networking index: Global mobile data traffic forecast update, 2014 2019. Technical report. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

Cisco (2016). At-a-glance: The internet of things. Technical report. http://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf.

ComReg (2012). ComReg announces results of its multi-band spectrum auction. Technical report.

Deb, S., Monogioudis, P., Miernik, J., and Seymour, J. P. (2014). Algorithms for enhanced inter-cell interference coordination (eICIC) in LTE HetNets. *IEEE/ACM Transactions on Networking*, 22(1):137–150.

ETSI (2016). User equipment (UE) radio transmission and reception (TDD). Technical Report TR 125.102 v13.0.0, European Telecommunications Standards Institute (ETSI).

Fenton, M., Lynch, D., Kucera, S., Claussen, H., and O'Neill, M. (2015). Load balancing in heterogeneous networks using an evolutionary algorithm. In *Proceedings of IEEE Conference on Evolutionary Computation*, pages 70–76, Sendai, Japan.

Fenton, M., Lynch, D., Kucera, S., Claussen, H., and O'Neill, M. (2017a). Multilayer optimization of heterogeneous networks using grammatical genetic programming. *IEEE Transactions on Cybernetics*, 47(9):2938–2950.

Fenton, M., McDermott, J., Fagan, D., Forstenlechner, S., Hemberg, E., and O'Neill, M. (2017b). PonyGE2: Grammatical Evolution in Python. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1194–1201. ACM.

GSMA (2017). The mobile economy. Technical report. https://www.gsmaintelligence.com/research/?file=9e927fd6896724e7b26f33f61db5b9d5&download.

Hämäläinen, S., Sanneck, H., and Sartori, C. (2012). *LTE self-organising networks (SON): network management automation for operational efficiency*. John Wiley & Sons.

Hemberg, E., Ho, L., O'Neill, M., and Claussen, H. (2011). 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.

Hemberg, E., Ho, L., O'Neill, M., and Claussen, H. (2013). A comparison of grammatical genetic programming grammars for controlling femtocell network coverage. *Genetic Programming and Evolvable Machines*, 14(1):65–93.

Jain, R., Chiu, D.-M., and Hawe, W. R. (1984). *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*, volume 38. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA.

Jiang, L. and Lei, M. (2012). Resource allocation for eICIC scheme in heterogeneous networks. In *23rd International Symposium on Personal Indoor and Mobile Radio Communications (IEEE PIMRC)*, pages 448–453. IEEE.

Koza, J. R. (2010). Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines*, 11(3-4):251–284.

López-Pérez, D. and Claussen, H. (2013). Duty cycles and load balancing in HetNets with eICIC almost blank subframes. In *24th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC Workshops)*, pages 173–178. IEEE.

Lynch, D., Fenton, M., Kucera, S., Claussen, H., and O'Neill, M. (2016a). Evolutionary learning of scheduling heuristics for heterogeneous wireless communications networks. In *Proceedings of the 2016 Annual Conference on Genetic and Evolutionary Computation*, pages 949–956. ACM.

Lynch, D., Fenton, M., Kucera, S., Claussen, H., and O'Neill, M. (2016b). Scheduling in heterogeneous networks using grammar-based genetic programming. In *Genetic Programming*, pages 83–98. Springer. LNCS 9594.

Lynch, D., Fenton, M., Kucera, S., Claussen, H., and O'Neill, M. (2017). Ensemble techniques for scheduling in heterogeneous wireless communications networks. In *Operations Research Proceedings 2016*, pages 399–404. Springer.

Madan, R., Borran, J., Sampath, A., Bhushan, N., Khandekar, A., and Ji, T. (2010). Cell association and interference coordination in heterogeneous LTE-a cellular networks. *IEEE Journal on Selected Areas in Communications*, 28(9):1479–1489.

McKay, R., Nguyen, X., Whigham, P., Shan, Y., and O'Neill, M. (2010). Grammar-based genetic programming - a survey. *Genetic Programming and Evolvable Machines*, 11(3-4):365–396.

Motorola (2006). Frequency domain scheduling for E-UTRA. Technical Report R1-060877. TSG RAN1#44bis.

O'Neill, M. and Ryan, C. (2003). *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Kluwer Academic Publishers.

Peng, M., Liang, D., Wei, Y., Li, J., and Chen, H.-H. (2013). Self-configuration and self-optimization in LTE-advanced heterogeneous networks. *IEEE Communications Magazine*, 51(5):36–45.

Shannon, C. E. (1949). Communication in the presence of noise. *Proc. Institute of Radio Engineers*, 37(1):10–21.

Siomina, I. and Yuan, D. (2012). Load balancing in heterogeneous LTE: Range optimization via cell offset and load-coupling characterization. In *2012 IEEE International Conference on Communications (ICC)*, pages 1357–1361.

Tang, J., So, D. K., Alsusa, E., Hamdi, K. A., and Shojaeifard, A. (2015). Resource allocation for energy efficiency optimization in heterogeneous networks. *IEEE Journal on Selected Areas in Communications*, 33(10):2104–2117.

Weber, A. and Stanze, O. (2012). Scheduling strategies for HetNets using eICIC. In *International Conference on Communications (IEEE ICC)*, pages 6787–6791. IEEE.