Scheduling in Heterogeneous Networks Using Grammar-Based Genetic Programming

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

[1] Natural Computing Research and Applications Group, UCD, Dublin, Ireland
david.lynch.1@ucdconnect.ie
[2] Bell Laboratories, NOKIA, Dublin, Ireland

Abstract. Effective scheduling in Heterogeneous Networks is key to realising the benefits from enhanced Inter-Cell Interference Coordination. In this paper we address the problem using Grammar-based Genetic Programming. Our solution executes on a millisecond timescale so it can track with changing network conditions. Furthermore, the system is trained using only those measurement statistics that are attainable in real networks. Finally, the solution generalises well with respect to dynamic traffic and variable cell placement. Superior results are achieved relative to a benchmark scheme from the literature, illustrating an opportunity for the further use of Genetic Programming in software-defined autonomic wireless communications networks.

Keywords: Scheduling · Heterogeneous networks · Grammar-based genetic programming

1 Introduction

Traditional cellular infrastructure is under significant strain due to exponentially increasing demand [8]. The number of mobile-connected devices is now greater than the world's population and network traffic will grow tenfold by 2019 [3]. Low-powered antennas called Small Cells (SCs) have been proposed as a means of scaling existing deployments to meet these trends [4].

In traditional networks, high-powered Macro Cells (MCs) are distributed on hexagonal grids to provide blanket coverage to User Equipments (UEs). A UE could be a smartphone, tablet or laptop etc. Heterogeneous Networks (HetNets) are comprised of both SCs and MCs. By offloading UEs in traffic hotspots, SCs alleviate strain on the macro tier. Note that hotspots are regions containing a concentration of UEs. Multi-tiered networks exhibit several desirable properties. Firstly, SCs support ad-hoc deployment by operators. Secondly, they are a cost effective means of densifying networks. Finally, HetNets are spectrally efficient as both tiers share the same channel under the current 3rd Generation Partnership Project–Long Term Evolution (3GPP–LTE) framework [1].

Network operators must offer better quality of service than their competitors to attract and retain customers. In particular, they must maximise the data rates

M. Heywood et al. (Eds.): EuroGP 2016, LNCS 9594, pp. 83–98, 2016.
DOI: 10.1007/978-3-319-30668-1_6

delivered by their networks. The metric that most strongly correlates with user experience is the downlink rate which quantifies the amount of data that can be transferred per unit time. Operators must maximise downlink rates for the least advantaged customers (due to location say), sometimes at the expense of the more privileged. Fairness is vital because dropped calls or slow data speeds are unacceptable from a customer satisfaction standpoint.

HetNets present unique challenges vis-à-vis optimisation because they are highly dynamic. In this paper, we employ a grammar-based form of Genetic Programming (GP) to evolve a HetNet scheduling heuristic. This is a difficult real world problem which to date has not been tackled with GP. Operators currently implement highly suboptimal greedy proportionally fair scheduling. Tailoring such methods to corner cases requires much human effort. These inefficiencies can be alleviated by evolving better software at a cost that is negligible compared to cell densification–deploying a single SC can cost several thousand euros.

The paper is organised as follows. Section 2 describes the problem in detail. Previous work is surveyed in Sect. 3. Our simulation environment is described in Sect. 4. Experiments, results and discussion follow in Sects. 5 and 6. Finally, the paper closes with future directions and conclusions in Sect. 7.

2 Problem Definition

The 3GPP–LTE framework outlines a number of high level protocols for making HetNets viable [1]. SCs are typically underutilised because UEs preferentially attach to stronger MCs. The Cell Range Expansion mechanism is proposed to encourage more efficient offloading from MCs. To achieve this, SCs broadcast a Cell Selection Bias (β_i) such that $\beta_i \geq 0$, $\forall i \in \mathcal{S}$, the set of all SCs. There is no need for MCs to implement bias so $\beta_i = 0$, $\forall i \in \mathcal{M}$, the set of all MCs. A UE (u) attaches to and hence receives data from cell k, where,

$$k = \arg\max_i \; (Signal_{u,i} + \beta_i), \;\; \forall i \in \mathcal{M} \cup \mathcal{S}. \tag{1}$$

If a UE attaches to cell $k \in \mathcal{S}$ but would otherwise attach to a MC $m \in \mathcal{M}$, then we say that the UE resides in the expanded region of k. Since the signal from m in the expanded region is by definition larger than that from k, it follows that interference is significant therein. Edge interference is exacerbated in LTE HetNets because MCs broadcast on the same channel as SCs.

Interference mitigation in the time domain is a defining feature of the 3GPP framework. UEs can receive data in 1 ms intervals referred to as subframes (SFs). A contiguous block of 40 SFs defines a 'frame'. Frames constitute a convenient timespan over which network performance can be analysed. The enhanced Inter-Cell Interference Coordination (eICIC) paradigm introduced the notion of Almost Blank Subframes (ABSs) to mitigate cell-edge interference [11]. MCs mute their data transmissions during an ABS so that only minimal control signals are broadcast, allowing neighbouring SCs to transmit with minimal interference. We refer to the sequence of active and muted SFs at MCs as an 'ABS pattern'.

Clearly, UEs at SC edges experience greatly reduced interference when nearby MCs undergo an ABS. However, UEs that are attached to the muting MC cannot receive any data during an ABS. Intelligent resource interleaving strategies are thus required to realise the benefits from eICIC. A key task in this regard is allocating SFs to SC attached UEs–hence, the scheduling problem.

Scheduling is trivial for UEs served by MCs because they enjoy high signal to interference and noise ratios (*SINRs*) and therefore can be allocated to every non-ABS SF. However, SC attached UEs are subjected to high MC interference. Shannon's formula [27] describes how downlink rates (R) depend on bandwidth and *SINR*:

$$R_{u,f} = \frac{B}{N_f} \times \log_2(1 + SINR_{u,f}), \tag{2}$$

where, B is the available bandwidth, N_f is the number of UEs scheduled in SF f and u denotes a UE. From Eq. 2, observe that $R_{u,f}$ is inversely proportional to N_f, where the downlink rate quantifies how much data can be transferred in a unit of time. Each UE will experience reduced rates in any given SF as it becomes more congested. Consequently, scheduling is a non-trivial problem because we would like to schedule each UE for as many SFs as possible but yet minimise per SF congestion.

SF \ UE	'6'	'4'	'7'	'2'	'8'	'9'
1	T	T	T	T	T	T
2	T	T	T	T	T	T
3	T	T	T	T	T	T
4	T	T	T	F	F	F
5	T	T	T	F	F	F
6	T	T	T	F	F	F
7	T	T	T	F	F	F
8	T	T	T	F	F	F

Time

Fig. 1. Depiction of a SC schedule where rows represent SFs and columns store schedules. UE u receives data in all SFs indexed by 'T' in their schedule.

The non-trivial nature of the problem can be appreciated by visualising schedules in the form of a boolean matrix. Figure 1 describes a feasible set of schedules for a SC with six attached UEs (only 8 out of 40 SFs are displayed for concision). Here 'T' indicates that a UE will receive data in the corresponding SF, and 'F' implies the converse. For instance, UE 9 is scheduled in the first 3 SFs. By construction the illustrative schedules in Fig. 1 exhibit sub-optimal properties. SFs 1–3 are fully congested so the bandwidth is divided six-fold. SFs

4–8 are less congested and so UEs 6, 4 and 7 profit from the liberated bandwidth. However, the reduced congestion is at the expense of UEs 2, 8 and 9 because they receive less airtime. Clearly, this SC could employ a vast number of alternative strategies to allocate SFs, despite the fact that it only serves six UEs. We ask if GP can derive a heuristic to compose synergistic schedules on the fly. Our task is to populate a scheduling matrix like Fig. 1 for all SCs in the network.

HetNet control algorithms are typically evaluated using a proportional fair utility of user experience. The sum log of downlink rates, see for example [12, 22, 26, 29], is given by:

$$PF\ Utility = \sum_{u \in \mathcal{M} \cup \mathcal{S}} \log \overline{R}_u, \tag{3}$$

where,

$$\overline{R}_u = \frac{1}{|\mathcal{F}|} \sum_{f=1}^{|\mathcal{F}|} R_{u,f},$$

is the average downlink rate for UE u over $|\mathcal{F}| = 40$ SFs. Equation 3 rewards individuals that fairly allocate resources. Lifting the downlink rates for poorly performing UEs is heavily rewarded by the logarithm. Conversely, losses for the best performing UEs are not penalised severely. Therefore, the fitness function rewards solutions that 'rob from the rich and give to the poor'.

3 Previous Work

An extensive literature exists on scheduling. Such problems arise in domains of operations research ranging from rostering [14] and job shop scheduling [23] to air traffic control [16]. In general, the feasible solution space is explored directly via search-based techniques. However, heuristic rules that can compute solutions on the fly are often motivated by practical constraints.

Bader-El-Den and Fatima (2010) employed an auction inspired scheme for the exam time-tabling problem [6]. They evolved a 'bidding function' that exams use to bid for time-windows. Auctions are held for each available window until all exams have been allocated. We note that the evolved solution operates within the context of a meta-algorithm, in this case inspired by an auction.

Jakovocić and Marasović (2012) identified the limitations of enumerative and search-based techniques [21]. The combinatorial nature of scheduling problems renders a direct search of the solution space impractical when runtime must be minimised. Following the authors in [6], they manually designed meta-algorithms tailored to specific job-shop scheduling environments. Evolved priority functions operate within these meta-algorithms. Thus, domain knowledge informed the solution structure, lending GP a foothold to search for functional forms.

Sun et al. (2006) instrumented a game theoretic approach to allocate channel resources in a wireless network [28] but our literature review has uncovered no previous work addressing scheduling in HetNets using GP. This paper attempts to fill the gap. Evolutionary methods are indicated in this domain because they are known to yield good solutions in dynamic environments [13,31]. Ho and Claussen (2009) used GP to optimise the coverage of femtocell deployments in enterprise environments [20]. 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 setting femtocell powers to optimise coverage [17–19]. The best solutions outperformed human designed heuristics on two of the three objectives.

A number of papers are relevant to our work in the space of eICIC optimisation. Weber and Stanze (2012) compared the performance of two scheduling strategies: strict and dynamic [30]. The former schedules centre UEs in non-ABSs so that ABSs are reserved for expanded region UEs. The latter allows edge UEs to receive both ABS and non-ABS airtime. Experiments showed that the dynamic scheduler achieves a better tradeoff between cell border rates and spectral efficiency.

Pang et al. (2012) proposed a scheduling method based on dynamic programming [26]. Synchronous patterns were assumed so that MCs mute in unison. Exactly two SCs were simulated per MC sector, with UEs uniformly distributed on the map. It is unclear whether their algorithm would perform well under more general conditions. Jiang and Lei (2012) modelled the scheduling problem as a two player Nash bargaining game where protected (ABSs) and normal (non-ABSs) resources at SCs compete for UEs [22]. Each 'player' strives to maximise the total data that it transmits. Performance was improved under the proposed algorithm relative to standard baselines. Edge UEs experienced comparable rates in the proposed and baseline cases. We will demonstrate that a GP evolved heuristic can give considerable gains for edge UEs.

Deb et al. (2014) formulated eICIC optimisation as a non-linear programming instance [12]. Their algorithm computes the airtime UEs should receive from their serving MC and SC during ABS and non-ABS periods. Simulation revealed that cell edge UEs gain the most under eICIC. The authors showed that their algorithm is within 90 % of the optimal but it requires measurement reports from each UE's best SC and MC. In practice UEs only communicate with their serving cell [24].

López-Peréz and Claussen (2013) proposed a heuristic to balance load (number of UEs) between ABSs and non-ABSs at SCs [24]. Load balancing improved the 5^{th} percentile of SC attached UE rates by 55 %, in a scenario with fixed MC ratios and with non-zero biases on SCs. Reduced mean MC throughput under the proposed scheme, relative to the benchmark, was compensated by increased mean SC throughput. In sum, this paper demonstrated the considerable gains achievable with intelligent scheduling. We adopt López-Peréz and Claussen (2013) as a benchmark.

4 Simulation Environment

In order to rapidly evaluate solutions we simulated a HetNet serving 3.61 km^2 of Dublin City Centre. SCs are typically deployed in an ad-hoc fashion because they serve hotspots, whereas MCs are placed on a grid by network operators. As such, we scattered SCs randomly on the map and arrange MCs in a hexagonal pattern. Figure 3 displays a snapshot of the network used for fitness evaluations. A HetNet with 21 MCs, 50 SCs and 1250 UEs was simulated for training.

Fig. 2. Environmental encoding.

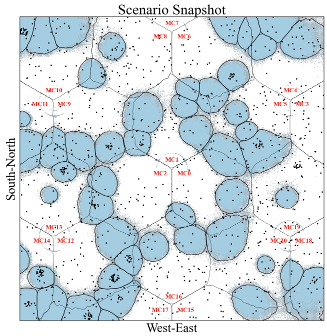

Fig. 3. SCs are shaded blue, MCs white and UEs are indicated by black dots (Color figure online).

4.1 Generating Inputs

The simulation proceeds sequentially. Firstly, an environmental encoding is generated from a Google Maps [2] image of the serviced region (Fig. 2). This encoding captures the distribution of buildings, bodies of water, open spaces and roads. A signal gain path loss matrix G is then computed for all cells. G models the cell gains, shadow fading and environmental obstacles, so that $G[i, x, y]$ represents the path loss from cell i to location $[x, y]$.

Next, UEs are distributed onto the map. Hotspots, 30 in total, are generated containing between 5 and 25 UEs. With probability 0.1 a hotspot will materialise outside of a SC but mostly they appear within SCs. If a UE is not assigned to a hotspot then it is placed at a random point on the map. A total of 1250 UEs are simulated or about 60 per MC sector.

The signal received by UE u from cell i depends on path loss such that:

$$Signal_{u,i} = P_i^{\mathrm{TX}} + G[i, x, y], \tag{4}$$

where, P_i^{TX} is the transmitting power of i in decibel milliwatts (dBm). SC transmit powers and Cell Selection Biases (β_i) are set by an evolved heuristic devised by the authors [15]. MC powers and biases are constant at $P_i^{\mathrm{TX}} = 37$[dBm] and $\beta_i = 0$[dBm], $\forall i \in \mathcal{M}$. Hence, u can identify its serving cell using Eq. 1.

MC ABS ratios are set using the simple heuristic from [29]. The ratio of ABSs to non-ABSs for MC $m \in \mathcal{M}$ is established by:

$$ABS_r = \frac{|\mathcal{SC}_{expanded}|}{|\mathcal{SC}_{expanded}| + |\mathcal{A}_m|}, \tag{5}$$

where, $|\mathcal{A}_m|$ is the number of UEs served by m and $\mathcal{SC}_{expanded}$ is the set of UEs that would attach to m but instead reside in the expanded regions of SCs within m's coverage area. Recall from Sect. 2 that the 'expanded region' contains those UEs that attach to SC i instead of m because β_i is non-zero. Equation 5 is sensible because if the number of expanded region UEs is large relative to $|\mathcal{A}_m|$, then m should surrender more SFs, thus mitigating cell-edge interference.

Each MC constructs a feasible muting pattern from the ABS ratio (Eq. 5) by combining eight base patterns from the standard [1]. Since five SFs are muted in each of the eight base patterns it follows that the ABS_r must be rounded to an element in the set $\{5/40, ..., 35/40\}$. Intra-frame *SINR* variance is reduced by 'front-loading' muted SFs so that if $ABS_r = 10/40$ for $m \in \mathcal{M}$, then m will mute in the first two SFs for every block of eight, e.g. in SFs (1, 2, 9, 10, 17, 18, 25, 26, 33, 34). A MC cannot entirely mute or transmit $\forall f \in \mathcal{F}$.

The most important statistic from a scheduling standpoint is the *SINR* that UEs experience in each SF. $SINR_{u,f}$ is computed by dividing $Signal_{u,serving}$ (the signal in Watts from u's serving cell) in SF f by, the sum of all interfering signals (from all other cells) plus noise. Note that $SINR_{u,f}$ depends on which MCs are muting or transmitting in SF f. The denominator will be reduced during protected SFs because $Signal_{u,m} = 0$ from MC m if it is undergoing an ABS. Therefore, $SINR_{u,f}$ varies over a frame due to the variable number of MCs that mute in different SFs. Our goal is evolve an expression that maps *SINR* related statistics and attachment information to a binary decision for each UE per SF: schedule or don't schedule. The terminal set for GP is derived from the cell attachment information and *SINRs* as annotated in Table 1.

4.2 Calculating Fitness

Algorithm 1 delineates the meta-algorithm (in the sense of [6,11]) used to yield schedules from an individual. The GP tree executes independently on each SC as follows. We loop over SFs and UEs, evaluating the tree at each (u, f) tuple. If the tree outputs a positive value then u will receive transmissions in SF f, else u is not scheduled in f. In this sense the tree performs a binary classification task on every execution.

The schedules are implemented in simulation and summary statistics on the realised downlink rates (accounting for congestion) are computed. Performance is expressed as the improvement in sum log downlink rates relative to a baseline strategy whereby u receives data in every SF f if $SINR_{u,f} \geq 1$. This baseline is naive because whilst airtime is maximised for each UE, so too is congestion. Recalling Eq. 3, the fitness function is expressed thusly:

Algorithm 1. Schedule UEs

function GETSTATISTICS(u, f)
  return Column 2 of Table 1 for UE u in SF f
procedure DOSCHEDULING($\mathcal{S}$, Tree)
  for $SC \in \mathcal{S}$ **do** ▷ Process each SC independently
    $congestion_f \leftarrow 0$ ▷ Track congestion in SF f
    $airtime_u \leftarrow 0$ ▷ Track number of SFs received by u
    $S \leftarrow 0_{(|\mathcal{F}|\times|\mathcal{A}|)}$ ▷ Stores SC schedule, see Fig. 1
    for $f \in \mathcal{F}$ **do** ▷ $\mathcal{F} = \{1, ..., 40\}$ is the set of SFs
      for $u \in \mathcal{A}$ **do** ▷ $\mathcal{A}$ stores the attached UEs
        $inputs \leftarrow$ GETSTATISTICS(f, u)
        Output = $evaluate$(Tree($inputs$)) ▷ Inputs are listed in Table 1
        if Output > 0 **and** $SINR_{u,f} \geq 1$ **then**
          $S_{u,f} \leftarrow$ True ▷ u will receive data in f
          $congestion_f$ += 1
          $airtime_u$ += 1
        else
          $S_{u,f} \leftarrow$ False ▷ For the greater good sacrifice u
  return S

$$Fitness = \frac{1}{10}\sum_{s=1}^{10}\Big(PF\ Utility_s^{\text{tree}} - PF\ Utility_s^{\text{baseline}}\Big). \tag{6}$$

Equation 6 expresses overall fitness as the average performance over ten UE distributions, hereafter scenarios (s). Thus, we provision against overfitting on a single set of UE locations.

5 Experiments

We instrumented a grammar-based form of GP [9,25]. Grammars allow us to incorporate domain knowledge and guarantees that syntactically correct structures are generated. Figure 4 displays the function approximation type grammar used in Backus-Naur Form (BNF). Four non-linear transforms were admitted including 'step' which output -1 if its argument is less than 0, else +1. The logarithm and square root functions were protected via $\log(1 + |x|)$ and $\sqrt{|x|}$ respectively. Random floats in the set $\{-1.0, -0.9, ..., 1.0\}$, statistics on instantaneous rates and memory nodes (airtime and congestion) composed the terminal set. The $SINR$ statistics were mapped to instantaneous downlink rates (assuming no bandwidth splitting) as illustrated in Table 1. Note that `num_viable` is the number of SFs in which a UE can receive data without packet loss (i.e. when $SINR \geq 1$). By contextualising u relative to all attached UEs, we anticipated that GP would uncover cooperation strategies, whereby u sometimes sacrifices a particular SF f for the global objective.

Thirty independent runs were performed for 75 generations. The Ramped Half-and-Half method was used to initialise the population (pop size = 1000) with an initial max depth of 6. We used fair tournament selection (tournament size was 1 % of pop size) so that all individuals had a chance of getting selected. Subtree Crossover, Subtree Mutation and Point Mutation were used to search the space of derivation trees. Subtree crossover was applied with a probability of 0.5

Table 1. GP Terminal Set

Raw Input	Statistic	Terminal Name
SC attached UEs	$\lvert\mathcal{A}\rvert$	`num_att`
$SINR_{u,f}$	$\log_2(1 + SINR_{u,f})$	`downlink`
$SINR_{u,f}, \forall f \in \mathcal{F}$	$\frac{1}{\lvert\mathcal{F}\rvert} \sum_{f \in \mathcal{F}} \log_2(1 + SINR_{u,f})$	`avg_downlink_frame`
"	$\max_f \{\log_2(1 + SINR_{u,f})\}$	`max_downlink_frame`
"	$\min_f \{\log_2(1 + SINR_{u,f})\}$	`min_downlink_frame`
"	$\lvert \{SINR_{u,f} \geq 1\} \rvert_f$	`num_viable`
$SINR_{u,f}, \forall u \in \mathcal{A}$	$\frac{1}{\lvert\mathcal{A}\rvert} \sum_{u \in \mathcal{A}} \log_2(1 + SINR_{u,f})$	`avg_downlink_SF`
"	$\max_u \{\log_2(1 + SINR_{u,f})\}$	`max_downlink_SF`
"	$\min_u \{\log_2(1 + SINR_{u,f})\}$	`min_downlink_SF`
$SINR_{u,f}, \forall f \in \mathcal{F}, \forall u \in \mathcal{A}$	$\frac{1}{\lvert\mathcal{A}\rvert} \sum_{u \in \mathcal{A}} \left(\frac{1}{\lvert\mathcal{F}\rvert} \sum_{f \in \mathcal{F}} \log_2(1 + SINR_{u,f}) \right)$	`avg_downlink_cell`
"	$\max_u \left\{ \max_f \{\log_2(1 + SINR_{u,f})\} \right\}$	`max_downlink_cell`
"	$\min_u \left\{ \min_f \{\log_2(1 + SINR_{u,f})\} \right\}$	`min_downlink_cell`
Previous outputs of tree	#SFs received by current UE	`airtime`
Previous outputs of tree	# other UEs sharing SF	`congestion`

to each pair of selected parents. Sixty percent of the population was subjected to Subtree Mutation. The remaining forty percent underwent point mutation with probability of 0.05, 0.1, 0.2 or 0.3 per node. We used generational replacement and elitism with elite size equal to 1 % of pop size. A run took ten hours on a twelve core hyperthreaded machine operating at 2.66 GHz.

The search space for this problem admitted many local optima. Trees that output exclusively positive or negative values yielded trivial schedules whereby all UEs were always or never scheduled. To avoid local optimum we assigned

```
<expr> ::= <reg> | <reg> | <reg> | <Terminal>
<reg> ::= <arithmetic>(<expr>,<expr>) | <arithmetic>(<expr>,<expr>) |
          <arithmetic>(<expr>,<expr>) | <arithmetic>(<expr>,<expr>) |
          <non-linear>(<expr>) | <non-linear>(<expr>)
<arithmetic> ::= + | - | * | % (protected division)
<non-linear> ::= sin | log | sqrt | step
<Terminal> ::= U(-1, +1, 0.1) | U(-1, +1, 0.1) |
               U(-1, +1, 0.1) | U(-1, +1, 0.1) |
               num_viable | num_att |
               downlink |
               avg_downlink_frame | max_downlink_frame | min_downlink_frame
               avg_downlink_SF | max_downlink_SF | min_downlink_SF
               avg_downlink_cell | max_downlink_cell | min_downlink_cell
               airtime | congestion
```

Fig. 4. BNF Grammar Definition.

zero fitness to such trees. Runtime was reduced substantially, without degrading solution quality, by terminating the simulation early for trivial trees.

6 Results and Discussion

Figure 5 displays the mean best fitness, with a 95 % confidence interval about the mean (grey shading), across 30 runs for 75 generations. Note we are maximising the improvement in sum log rates–Eq. 6. Clearly, the evolved population has a significantly higher mean fitness than the initial random population. Evidently convergence is not achieved in only 75 generations, so better solutions are likely to emerge from longer runs. Figure 5 suggests that the GP system is stable because the variance about mean best fitness is low across many independent runs.

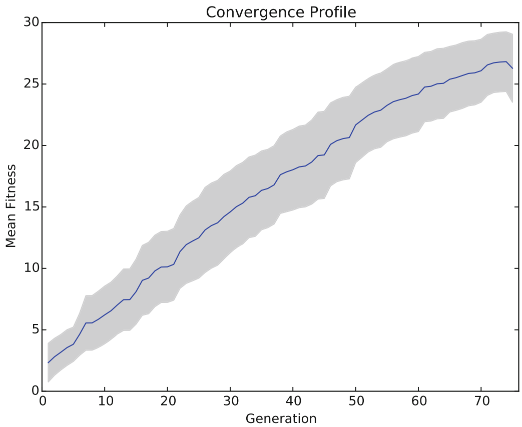

Fig. 5. Mean best fitness on training data including a 95 % confidence interval (Color figure online).

The best overall solution (see Fig. 6[1]) was identified by exposing all 30 best-of-run individuals to unseen test networks. The test fitness of each individual was computed as the average performance across 100 scenarios, in networks with 20, 60 and 100 SCs (each serving 1250 UEs). Most individuals performed well in one or two of the three test networks. A few performed very well across all three topologies. The best individual achieved a fitness of 55.5, 60.4 and 25.8 on the networks containing 20, 60 and 100 SCs respectively. These observations underscore the need for multiple runs despite the fact that we seek only one good solution. That some individuals struggle on one or two of the test networks but perform well on the other(s) may be indicative of overfitting. In future work the use of a validation network will enable intelligent termination if overfitting ensues.

[1] Note that the constants have been obfuscated to protect intellectual property.

$$S_{u,f} \leftarrow ((A + D/B) - C + F)$$

$$A = (\sin(\log(\sin(\texttt{max\_downlink\_cell}))) + (\log(\texttt{downlink} + \texttt{airtime})/$$
$$(\texttt{congestion} - (\texttt{congestion}/\texttt{max\_downlink\_SF}))))$$
$$B = \text{step}(((\text{step}(\texttt{num\_viable}) * \texttt{constant}))/\text{step}(\log(\texttt{max\_downlink\_cell})))$$
$$C = (\sin(\texttt{num\_viable} - (\text{step}(\texttt{min\_downlink\_frame}/\texttt{min\_downlink\_SF}) -$$
$$\text{sqrt}(\texttt{avg\_downlink\_SF} + \texttt{avg\_downlink\_frame}))))$$
$$D = \text{step}(((\text{step}(\texttt{min\_downlink\_frame}) * \texttt{constant})))$$
$$E = ((((\text{step}(\texttt{min\_downlink\_frame})) + ((\texttt{num\_viable} - \texttt{avg\_downlink\_frame})$$
$$/\texttt{avg\_downlink\_cell})) * \texttt{max\_downlink\_SF})$$
$$/\log(\texttt{downlink} + \texttt{downlink}) * \log(\texttt{downlink}))$$
$$F = (\texttt{avg\_downlink\_frame} - (((E/(\texttt{downlink} - \texttt{max\_downlink\_frame}))))$$

Fig. 6. Best Evolved Scheduler. Terminals are distinguished by blue text.

The percentage lift in sum log rates in the test network containing 100 SCs was lower relative to that in the less densified networks. One might expect that a larger number of SCs should afford the scheduler greater scope to improve fairness. However on inspection we found that the proportion of UEs in SC expanded regions decreased with increasing SC density. With 20 SCs on the map about 14 % of the SC attached UEs resided in an expanded region compared to just 7 % with 100 SCs. In addition, the average number of UEs per SC decreased as more SCs were added. In combination these factors diminished the marginal impact of scheduling. As expected, the overall network utility was boosted by SC densification. The sum log downlink rates trended 17912→ 18621→18933 as the number of SCs increased 20→60→100. Densification and scheduling are recognised as key requisites for 5G [7].

6.1 Terminal Utilisation

Figure 7 displays each terminal's count in the fittest 150 individuals at each generation. The instantaneous downlink rate (`downlink`) occurs most frequently. Heavy utilisation of this terminal is unsurprising because it most faithfully predicts realised downlink rates. Antithetically, `num_att` appears least frequently. This too is unsurprising as the number of attached UEs is a constant statistic and hence it bears no differentiating power. Therefore, `num_att` represents a baseline with respect to which the importance of other terminals can be inferred. The terminals describing u's performance across all 40 SFs appear more frequently than the other contextual statistics. This may suggest that although u's context within a SF relative to other UEs is important (`*_downlink_SF`, `*_downlink_cell`), more relevant for u are the attributes of a SF f relative to other SFs (`*_downlink_frame`). Of particular interest is the standing of our mem-

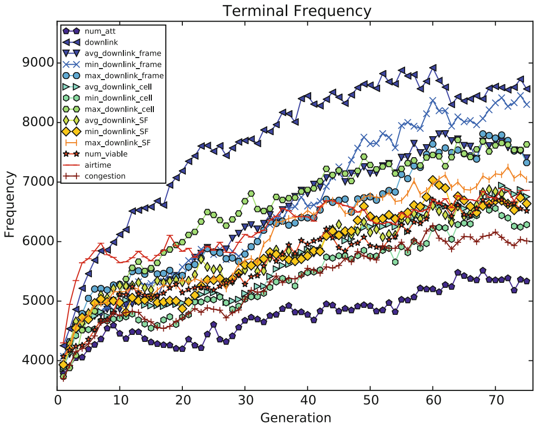

Fig. 7. Count of each terminal in the best 150 individuals over 75 generations.

ory nodes, `airtime` and `congestion`. Their importance is commensurate with that of most SF and cell-wide context nodes. In any case it is clear that they are selected for in the evolving populations. Given the ostensible importance of memory in HetNet scheduling, we are motivated in future work to explicitly factor prior decisions into the current tree output, see Sect. 7.

6.2 Subframe Utilisation

This subsection examines how the scheduler displayed in Fig. 6 behaves semantically. To appreciate how MC muting modulates interference at SCs, consider a toy network with three MCs running ABS ratios of 5/40, 10/40 and 15/40. Since ABSs are front-loaded as described in Subsect. 4.1, it follows that all three MCs will mute in SFs 1, 9, 17, 25 and 33. Two of the three MCs will mute in SFs 2, 10, 18, 26, 34, and only one MC will mute in SFs 3, 11, 19, 27, 35. All three will transmit in the remaining SFs. Thus, SFs 1, 9, 17, 25 and 33 are protected from high MC interference. Perhaps GP would learn to award protected SFs to UEs at the SC cell-edge whilst de-allocating those more advantaged cell-centre UEs (in order to relieve congestion). Indeed, simulation of a network with 60 SCs and 21 MCs revealed stark differences in how edge and centre UEs are scheduled. For both groups, we counted the cumulative number of UEs scheduled per SF, for all SCs over ten scenarios.

Columns 2 and 3 of Table 2 display the proportion of expanded region and centre UEs that are scheduled in each SF. For example, column 2 states that on aggregate 85 % of the expanded region UEs are scheduled in the best SFs (1, 9, 17, 25, 33), 58 % in the second best SFs, etc. Column 3 indicates that the scheduler sacrifices centre UEs by denying them the most protected SFs. Consequently, congestion is dramatically reduced for expanded region UEs where their *SINR* will be greatest. Column 1 confirms that expanded region UEs are awarded more airtime in the better SFs. They can thus leverage both high *SINR*'s and the liberated bandwidth to maximise their achieved downlink rates. Centre UEs make up for the lost premium airtime by dominating less protected SFs where few or no expanded region UEs receive data.

Column 3 shows the cell-wide proportion of UEs scheduled per SF. Clearly, congestion management is of prime importance in SFs 1, 9, 17, 25 and 33 as less that 25 % receive data. In fact quite a few UEs are denied in every SF. The proportion jumps to 91 % in SFs 2, 10, 18, 26 and 34 then decreases monotonically reflecting how the scheduler negotiates trade-offs between airtime and congestion. In sum, non-trivial yet intuitive behaviour is realised by the expression in Fig. 6.

Table 2. Proportion scheduled per SF in various SC regions.

SF	Expanded region UEs	Cell-centre UEs	All attached
1, 9, 17, 25, 33	0.85	0.14	0.23
2, 10, 18, 26, 34	0.58	0.96	0.91
3, 11, 19, 27, 35	0.28	0.94	0.86
4, 12, 20, 28, 36	0.10	0.93	0.82
5, 13, 21, 29, 37	0.02	0.92	0.81
6, 14, 22, 30, 38	0.0	0.92	0.80
7, 15, 23, 31, 39	0.0	0.92	0.80
8, 16, 24, 32, 40	0.0	0.92	0.80

6.3 Benchmarking

The benchmark scheme was proposed by López-Peréz and Claussen (2013) [24]. Based on *SINR*, UEs are split into queues 'overlapping' with host MC ABSs or 'non-overlapping'. The worst UE in both queues is identified. Achievable downlink rates are computed for both worst UEs for both queue types. Next the SC computes target queue lengths based on the expected rates. UEs are transferred iteratively from one queue to the other until convergence. They are then scheduled according to their queue type, i.e. during ABS or non-ABS SFs.

Table 3 compares the evolved GP solution with the benchmark scheme, a Genetic Algorithm (GA) and a Hill Climbing heuristic. The improvements in the 5^{th} and 50^{th} percentile downlink rates and the lifts in sum log rates (Eq. 6) over baseline scheduling are reported. Statistics are generated across 100 scenarios in unseen test networks containing 20, 60 and 100 SCs. The GA was instrumented using a population size of 750 and run for 50 generations for each SC in the test set. The solution space was explored using two-point crossover (probability 1.0) and bit flip mutation on 20 % of the population (probability 0.2 per codon). Similarly, the Hill Climber mutated a randomly generated schedule (bit flip

Table 3. Comparison of Methods

%–tile	GP: Figure 6 20 SCs	60 SCs	100 SCs	Benchmark 20 SCs	60 SCs	100 SCs
50^{th} ($\times 10^3$)	52 ± 41	-130 ± 108	-437 ± 133	-92 ± 51	-368 ± 160	-733 ± 276
5^{th} ($\times 10^3$)	159 ± 44	388 ± 69	649 ± 105	118 ± 38	240 ± 56	209 ± 107
Fitness	55.92 ± 0.69	58.10 ± 0.85	24.58 ± 1.08	41.57 ± 1.04	18.84 ± 1.80	-42.63 ± 4.66

%–tile	GA 20 SCs	60 SCs	100 SCs	Hill Climber 20 SCs	60 SCs	100 SCs
50^{th} ($\times 10^3$)	43 ± 27	45 ± 69	41 ± 96	46 ± 28	42 ± 68	10 ± 89
5^{th} ($\times 10^3$)	154 ± 43	440 ± 64	570 ± 131	159 ± 44	450 ± 60	563 ± 126
Fitness	84.77 ± 0.42	117.45 ± 0.69	87.09 ± 0.91	87.33 ± 0.51	117.69 ± 0.95	82.36 ± 1.12

with probability 0.025 per codon) for 20,000 iterations and greedily accepted improvements to the current best.

The first panel shows that our evolved solution outperforms the benchmark on all metrics. Two sample t-tests confirm that the differences are significant at a confidence level of 0.99. Of particular interest is row 2 of the first panel which compares the benchmark and evolved heuristics with respect to the 5^{th} percentile of downlink rates. Our fitness function does not incorporate this metric explicitly. We simply maximise sum log rates. Therefore, the improved performance of the worst UEs versus the baseline and benchmark schemes emerges naturally as a by-product of the optimisation.

The second panel shows that there is scope to build on this pilot study. Both the GA and Hill Climbing heuristics significantly outperform the evolved solution and benchmark. Note that these direct search methods are orders of magnitude too slow for online operation.

7 Future Work and Conclusions

We implemented a reinforcement learning approach in this work but some pilot experiments suggest that supervised training will yield far better schedulers. In a follow-up study we will compute near optimal schedules offline using a genetic algorithm. Hence, a more informative fitness function can be devised which respects the distance (e.g. Hamming) to the target semantics. Table 3 reveals that significant gains are achievable.

We observed in Sect. 6 that it is expedient to track intermediate scheduling decisions using counters. Conrads et al. (1998) considered a time series problem where previous tree outputs acted as terminals for the current evaluation [10]. Alfaro et al. showed their recurrent system achieves comparable performance with state of the art methods on real world problems [5]. We propose that a recurrent system like [5,10] will yield better results on the present problem.

Genetic Programming lends itself well to the task of evolving schedulers for HetNets implementing eICIC. Indeed, the framework proposed in this paper outperforms a state of the art human engineered approach. Future work will build on this pilot study to close the optimality gap.

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

References

1. 3Gpp, December 2014. http://www.3gpp.org/
2. Google Maps, December 2014
3. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014–2019. Cisco, White Paper (2015)
4. Small Cell Solutions. Alcatel-Lucent (2015). https://www.alcatel-lucent.com/solutions/small-cells
5. Alfaro-Cid, E., Sharman, K., Esparcia-Alcázar, A.I.: Genetic programming and serial processing for time series classification. Evol. Comput. **22**(2), 265–285 (2014)
6. Bader-El-Den, M., Fatima, S.: Genetic programming for auction based scheduling. In: Esparcia-Alcázar, A.I., Ekárt, A., Silva, S., Dignum, S., Uyar, A.Ş. (eds.) EuroGP 2010. LNCS, vol. 6021, pp. 256–267. Springer, Heidelberg (2010)
7. Bhushan, N., Li, J., Malladi, D., Gilmore, R., Brenner, D., Damnjanovic, A., Sukhavasi, R., Patel, C., Geirhofer, S.: Network densification: the dominant theme for wireless evolution into 5G. IEEE Commun. Mag. **52**(2), 82–89 (2014)
8. Bian, Y.Q., Rao, D.: Small Cells Big Opportunities. Global Business Consulting. Huawei Technologies Co., Ltd. (2014)
9. Brabazon, A., O'Neill, M., McGarraghy, S.: Natural Computing Algorithms. Springer, Berlin (2015)
10. Conrads, M., Nordin, P., Banzhaf, W.: Speech sound discrimination with genetic programming. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, pp. 113–129. Springer, Heidelberg (1998)
11. Damnjanovic, A., Montojo, J., Wei, Y., Ji, T., Luo, T., Vajapeyam, M., Yoo, T., Song, O., Malladi, D.: A survey on 3GPP heterogeneous networks. IEEE Wirel. Commun. **18**(3), 10–21 (2011)
12. Deb, S., Monogioudis, P., Miernik, J., Seymour, J.P.: Algorithms for enhanced inter-cell interference coordination (eICIC) in LTE HetNets. IEEE/ACM Trans. Netw. (TON) **22**(1), 137–150 (2014)
13. Dempsey, I., O'Neill, M., Brabazon, A.: Grammatical evolution. In: Dempsey, I., O'Neill, M., Brabazon, A. (eds.) Foundations in Grammatical Evolution for Dynamic Environments. SCI, vol. 194, pp. 9–24. Springer, Heidelberg (2009)
14. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: a review of applications, methods and models. Eur. J. Oper. Res. **153**(1), 3–27 (2004)
15. Fenton, M., Lynch, D., Kucera, S., Claussen, H., O'Neill, M.: Evolving coverage optimisation functions for heterogeneous networks using grammatical genetic programming. In: Proceedings of the 19th International Conference on the Applications of Evolutionary Computation, EvoCOMNET 2016. Springer (2016)
16. Hansen, J.V.: Genetic search methods in air traffic control. Comput. Oper. Res. **31**(3), 445–459 (2004)

17. Hemberg, E., Ho, L., O'Neill, M., Claussen, H.: 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, pp. 639–646. ACM (2011)
18. Hemberg, E., Ho, L., O'Neill, M., Claussen, H.: Evolving femtocell algorithms with dynamic and stationary training scenarios. In: Coello, C.A.C., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) PPSN 2012, Part II. LNCS, vol. 7492, pp. 518–527. Springer, Heidelberg (2012)
19. Hemberg, E., Ho, L., O'Neill, M., Claussen, H.: A comparison of grammatical genetic programming grammars for controlling femtocell network coverage. Genet. Program Evolvable Mach. **14**(1), 65–93 (2013)
20. Ho, L.T., Ashraf, I., Claussen, H.: Evolving femtocell coverage optimization algorithms using genetic programming. In: 2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 2132–2136. IEEE (2009)
21. Jakobović, D., Marasović, K.: Evolving priority scheduling heuristics with genetic programming. Appl. Soft Comput. **12**(9), 2781–2789 (2012)
22. Jiang, L., Lei, M.: Resource allocation for eICIC scheme in heterogeneous networks. In: 2012 IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), pp. 448–453. IEEE (2012)
23. Jones, A., Rabelo, L.C., Sharawi, A.T.: Survey of job shop scheduling techniques In: Wiley Encyclopedia of Electrical and Electronics Engineering (1999)
24. López-Pérez, D., Claussen, H.: Duty cycles and load balancing in hetnets with eICIC almost blank subframes. In: 2013 IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops), pp. 173–178. IEEE (2013)
25. Mckay, R.I., Hoai, N.X., Whigham, P.A., Shan, Y., O'Neill, M.: Grammar-based genetic programming: a survey. Genet. Program Evolvable Mach. **11**(3–4), 365–396 (2010)
26. Pang, J., Wang, J., Wang, D., Shen, G., Jiang, Q., Liu, J.: Optimized time-domain resource partitioning for enhanced inter-cell interference coordination in heterogeneous networks. In: 2012 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1613–1617. IEEE (2012)
27. Shannon, C.E.: Communication in the presence of noise. Proc. IRE **37**(1), 10–21 (1949)
28. Sun, J., Modiano, E., Zheng, L.: Wireless channel allocation using an auction algorithm. IEEE J. Sel. Areas Commun. **24**(5), 1085–1096 (2006)
29. Tall, A., Altman, Z., Altman, E.: Self organizing strategies for enhanced ICIC (eICIC). In: 2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pp. 318–325. IEEE (2014)
30. Weber, A., Stanze, O.: Scheduling strategies for HetNets using eICIC. In: 2012 IEEE International Conference on Communications (ICC), pp. 6787–6791. IEEE (2012)
31. Yang, S., Ong, Y.S., Jin, Y.: Evolutionary Computation in Dynamic and Uncertain Environments. Springer Science & Business Media, New York (2007)