Evolving Coverage Optimisation Functions for Heterogeneous Networks Using Grammatical Genetic Programming

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

[1] Natural Computing Research and Applications Group, UCD, Dublin, Ireland
michaelfenton1@gmail.com
[2] Bell Laboratories, Alcatel-Lucent, Dublin, Ireland

Abstract. Heterogeneous Cellular Networks are multi-tiered cellular networks comprised of Macro Cells and Small Cells in which all cells occupy the same bandwidth. User Equipments greedily attach to whichever cell provides the best signal strength. While Macro Cells are invariant, the power and selection bias for each Small Cell can be increased or decreased (subject to pre-defined limits) such that more or fewer UEs attach to that cell. Setting optimal power and selection bias levels for Small Cells is key for good network performance. The application of Genetic Programming techniques has been proven to produce good results in the control of Heterogenous Networks. Expanding on previous works, this paper uses grammatical GP to evolve distributed control functions for Small Cells in order to vary their power and bias settings. The objective of these control functions is to evolve control functions that maximise a proportional fair utility of UE throughputs.

1 Introduction

Recent technological advancements have created a paradigm shift in the way that mobile phones are used. The advent of large screens on handheld devices has prompted a shift from voice traffic to video and data streaming [1]. Until recently network operators have prioritised power management and cost minimisation over capacity maximisation [2,3]. However, the recent surge in high data-rate traffic [1] has prompted a switch from cost minimisation to capacity maximisation as carriers and operators struggle to cope with the increased demand.

Traditional network deployments are characterised by a distribution of high-powered transmitters (known as Macro Cells) placed on a hexagonal grid pattern to maximise coverage [4]. User Equipments (UEs) greedily attach to whichever cell provides the strongest signal. Cells then transmit data to all attached UEs by sub-dividing available bandwidth amongst them. A higher number of attached UEs per cell results in higher congestion, thus reducing the bandwidth. Consequently, each UE receives less data as congestion increases [4]. A standard

G. Squillero and P. Burelli (Eds.): EvoApplications 2016, Part I, LNCS 9597, pp. 219–234, 2016.
DOI: 10.1007/978-3-319-31204-0_15

method for dealing with increased demand is to increase the density of cells in the network.

Heterogenous Networks (HetNets) are multi-tiered cellular network deployments comprised of Macro Cells (MCs) and Small Cells (SCs) where both cell tiers operate on the same bandwidth. SCs transmit at a lower power and have a smaller operational range than MCs. They are employed to supplement the MC tier by offloading User Equipments (UEs), thus easing network congestion. SCs are often deployed in an ad-hoc manner by business owners in high-traffic areas such as restaurants, cafés, and shopping malls. As such, network operators may not necessarily have control over their placement [5].

Since SCs will be sub-optimally distributed, network operators seek to vary cell parameters in order to optimise the overall state of the network. One method of network optimisation is the notion of control algorithms [2,3,6–8]. These control algorithms either operate locally on individual cells or globally through a central server to manage the state of the network. In this study we expand on previous works by the authors [9] which used Grammatical Evolution (GE) [10,11] to evolve optimal SC settings for network load balancing. While results were highly optimal, the major limitation of this work was that it was necessary to do a full evolutionary run to find good settings. In this paper we adopt a symbolic regression approach to search the space of SC control functions, allowing for optimisation of SC settings on the fly, in a fraction of the time.

The remainder of this paper is structured as follows. A detailed definition of the problem is given in Sect. 2. Section 3 will give an overview of HetNet optimisation under the 3GPP standard, including a description of grammatical GP. Our approach summary is detailed in Sect. 4, including descriptions of our simulation environment (Sect. 4.1), grammatical representation (Sect. 4.2) and our fitness function (Sect. 4.3). Experimental studies are described in Sect. 4.4, and the results are discussed in Sect. 5. Finally, our conclusions and recommendations for future work are given in Sect. 6.

2 Problem Definition

Optimization of HetNets involves varying parameters of the network such that some objective is satisfied, usually the maximisation of overall UE throughput (with fairness). While MCs are invariant, SCs have adjustable parameters that can affect UE attachment [5]. Each SC s can vary its power P_s in order to modulate its operational range (the area in which it is the strongest serving cell). However, under the 3rd Generation Partnership Project - Long Term Evolution (3GPP-LTE) framework [4], SCs have an additional variable parameter that affects UE attachment, namely the Cell Selection Bias (CSB), β_s.

The CSB is a mechanism which artificially increases the effective range of the SC. UEs in this "expanded region" of the SC will attach to the cell in deference to their better serving MC for the global good of the network. A UE u will therefore attach to a cell k in accordance with the attachment rule in Eq. 1:

$$k = \arg\max_{i}(S_{ui} + \beta_i). \tag{1}$$

where S_{ui} is the perceived signal strength of cell i for UE u, $\beta_i = 0, \forall i \in \mathcal{M}$, the set of all MCs, and $\beta_i \geq 0, \forall i \in \mathcal{S}$, the set of all SCs [8]. Note that P_i (the transmitting power of cell i) is subject to path loss such that the signal strength perceived by u is given by:

$$S_{ui} = P_i[\text{dBm}] + G_{ui}[\text{dB}], \tag{2}$$

such that G_{ui} is the signal gain from cell i to u, see Sect. 4.1.

Cells transmit data during 1 ms intervals with each interval referred to as a subframe (f). During each subframe, a cell will transmit data across the available bandwidth to all attached UEs. Transmitted data primarily consists of packets of data, along with some minimal control signals. A full frame $\mathcal{F}$ consists of 40 subframes (i.e. 40 ms of network run-time).

The performance of UE u in any given subframe f is quantified by the downlink rate $R_{u,f}$ (in bits/sec, bps) from a cell i to u. Shannon's formula gives the downlink rate for wireless transmission in the presence of noise as [12]:

$$R_{u,f} = \frac{B_i}{N_{i,f}} * \log_2(1 + \mathit{SINR}_{ui,f}) \tag{3}$$

where B_i is the available bandwidth, $N_{i,f}$ is the total number of scheduled UEs attached to cell i for subframe f, and $\mathit{SINR}_{ui,f}$ is the Signal to Interference and Noise Ratio (the ratio of the received signal strength to the sum of all interfering signal strengths from all other cells in the network including background thermal noise) from source cell i to UE u in subframe f. Note that the available bandwidth across which the cell can transmit is divided by the total number of UEs attached to that particular cell. Therefore, the greater the number of attached UEs to a cell, the less bandwidth will be available to each individual UE [13].

Since by definition any UE within the expanded region of a SC (the additional area served by the SC due to its non-zero CSB) must experience significant interference from their strongest serving MC, provision has been made in recent 3GPP releases [4] for an enhanced Inter-Cell Interference Coordination (eICIC) mechanism for HetNets implementing CSBs. This mechanism mitigates inter-tier cell edge interference by employing Almost Blank Subframes (ABSs) [8]. With ABSs, MCs periodically mute across their entire bandwidth (save for some negligible but necessary control signals), thus giving nearby SCs quiet subframes in which they can transmit with greatly reduced interference. UEs in the expanded region of SCs (those UEs who are most vulnerable to interference from neighboring MCs) experience greatly improved *SINR* during ABSs and, therefore, receive greater throughput. Unfortunately, UEs attached to MCs that are implementing ABSs will receive no data transmissions during muted SFs.

Control algorithms which adjust SC power and bias settings have been used to optimise HetNets [6]. Increasing or decreasing the operational range of individual SCs can change the number of attached UEs, thereby affecting global network performance. Control algorithms can be designed to operate centrally across an entire network or independently on individual cells [6,7]. This study will focus on generating control algorithms for individual SCs. Each SC will run the same algorithm that is capable of adjusting both power and bias settings based measurement reports collected by the SCs.

3 Previous Work

Release 10 of 3GPP [4] describes eICIC conceptually but does not specify methods for configuring ABS patterns, setting SC powers and CSBs or scheduling UEs, as these are non-trivial tasks. Deb *et al.* prove (Sect. 4-A) that optimising ABS patterns alone is an NP-hard problem, even for minimal networks with a single MC and multiple SCs [8]. Huge growth forecasts [14] for the SC market motivate algorithms which can maximise the benefits from eICIC. The literature describes three resource allocation problems which jointly determine the performance of HetNets implementing eICIC [15]. They are:

1. setting SC powers and CSBs to ensure optimal offloading from the MC tier,
2. setting ABSrs to protect UEs at SC edges, and
3. scheduling SC attached UEs so that protected resources (ABSs) are optimally utilised.

A number of contributions address one or multiple components of this joint optimisation problem.

Tall *et al.* proposed a stochastic approximation technique to optimise SC CSBs and the ABSrs of MCs [3]. They first derived a Self Organising Network (SON) load balancing update function that minimised the load imbalance between MCs and SCs. It computed CSB adjustments based on averaged load statistics and operated in a distributed manner across SCs. Stochastic Approximation theorems from Combes *et al.* [16] proved that their SON converges to the set of optimal CSBs. The authors also derived update equations from a proportional fair utility of UE throughputs in order to optimise MC ABSrs. Simulations showed that the load balancing SON, in combination with the ABSr optimisation SON from the second implementation above, achieved the best tradeoff between overall network throughput and cell edge throughput (i.e. fairness).

Deb *et al.* formulated the eICIC optimisation problem as a non-linear programming (NLP) problem [8]. They adopted the proportional fair sum logarithm of UE throughputs used by [3]. This utility function negotiates a tradeoff between fairness for cell edge UEs and maximisation of overall network throughput [17]. The authors simulated their algorithm using a realistic HetNet deployment in Manhattan. The 5^{th} percentile of UE throughputs was improved by more than 50 % under eICIC without significant throughput losses for MC attached UEs. However, their algorithm requires measurement statistics from each UE's best serving SC and MC, but in reality UEs only attach to one cell [18].

As the problem is NP-hard and the structure of the solution is unclear it presents an opportunity for Genetic Programming (GP), a heuristic technique in Evolutionary Computation (EC).

3.1 Grammatical Genetic Programming

Grammatical Genetic Programming is a subset of GP techniques which use a formal grammar to define the terminal sets [19]. The use of a grammar means that programs can be generated in an arbitrary language [10,11,20]. Grammatical GP methods draw metaphorical inspiration from the principles of evolutionary and molecular biology to create machine executable solutions for a diverse spectrum of problems [20]. In contrast to canonical GP [21], where solutions are represented directed by parse trees, Grammatical GP techniques use a formal grammar to map from genotype to phenotype. Solutions can be generated using derivation trees [19,22] or using variable-length integer strings (chromosomes) [10,11] to map to programs (phenotypes) using a Backus-Naur Form (BNF) [23] grammar definition [24]. A key strength of these grammar-based techniques is that bias can be incorporated into the grammar to guide the search towards more desired solutions.

Grammatical GP methods such as Context-Free Grammatical GP (CFG-GP) [19,22] and Grammatical Evolution (GE) [10,11] have been successfully applied to financial modelling [25], structural engineering [26–28] and indeed HetNet optimisation [2,9,29,30]. Such flexibility is possible because problem specific domain knowledge can be incorporated into the grammars. This heuristic approach is appropriate for problems that do not easily admit analytic treatment, i.e. those where complete domain knowledge is lacking, or for dynamic environments [20].

3.2 Coverage Optimisation

A number of EC techniques have been applied in the field of telecommunications networks [31], but there have been relatively few in the area of coverage optimisation. Ho *et al.* applied GP to optimise the coverage of femtocell deployments (SCs with a range of several meters that are designed to support plug-and-play deployment) in enterprise environments [6]. Cell powers must be set in order to achieve load balancing and minimisation of coverage gaps and signal leakage. This problem is multi-objective with conflicting objectives since, for example, increasing power to reduce coverage gaps may increase leakage. The authors evolved programs that adjusted the power on individual femtocells based on local measurement statistics. Solutions responded sensibly to network conditions. This study represents a proof of concept that controllers can be automatically generated for wireless networks.

Hemberg *et al.* also examined a variety of different grammars on the related HetNet coverage optimisation problem [2,29,30]. In these instances, the three conflicting objectives of mobility minimisation (number of UE hand-overs), load balancing, and cell power minimisation (leakage) were jointly optimised for various indoor femtocell deployment scenarios using the multi-objective optimisation algorithm NSGA-II [32]. The authors found in [29] that the weighted fitness function used in [6] caused premature convergence to local optima, and they employed a symbolic regression approach in [2] to evolve femtocell power control equations. The grammar combined smooth and non-linear functions so that a wide range of non-trivial behaviours were accessible to evolved solutions. In [30] the authors compared a symbolic regression grammar, a grammar consisting exclusively of conditional statements and a hybrid combining both conditionals and functions. The purely conditional grammar allowed discrete power changes and was found to converge faster than the less constrained symbolic regression grammar. The combined grammar was slowest to converge and evolved solutions exhibited significantly worse fitness over all scenarios. It was noted that less domain knowledge is required for symbolic regression grammars but engineers favour the easily interpretable conditional solutions [6]. Finally, the utilities of control programs evolved using GE were found to match and sometimes exceed those achieved by partial enumeration of the search spaces.

There are two main differences between coverage optimisation and eICIC optimisation problems. Firstly, the objective function is univariate in eICIC because our goal is simply to maximise network capacity with fairness, while coverage optimisation observes a multivariate objective. Secondly, we currently have three degrees of freedom in eICIC: SC powers, SC CSBs and MC ABS patterns, as opposed to the single variable of SC powers. With this in mind, we now describe our simulation environment for a HetNet that implements eICIC.

4 Experimental Setup

4.1 Simulation

The simulation environment covers a 3.61 km^2 area of Dublin City Centre (Fig. 1), with a resolution of 2 m^2. A total of 21 MCs are placed on a hexagonal grid, with 79 SCs scattered randomly across the map. The random placement of SCs accounts for their ad-hoc deployment, since the manufacturer (or indeed the carrier) may have little control over their placement. UEs are distributed on the map with an average density of 60 UEs per MC sector, giving a total of 1,260 UEs. UE hotspots are modeled as dense congregations of between 5 to 25 UEs, such that 20 % of all UEs are located in hotspots. Hotspots are distributed at randomly selected SC locations with a probability of 90 %. Otherwise they are placed randomly on the map, thus simulating the tendency of business owners to deploy SCs in high-traffic areas.

An environmental encoding is generated from a Google Maps [33] screenshot of the region served by the network. The encoding recognises four environmental categories: buildings, bodies of water, parks, and roads/footpaths. UEs and SCs are not placed in bodies of water, but their placement in all other locations respects a uniform distribution (subject to the distribution of hotspots). A 2-dimensional signal gain path loss matrix G_i is then calculated for each cell i. Path loss is based on cell location, cell gain, shadow fading, and environmental obstacles such as buildings. The signal gain from a cell i to a location $[x, y]$ is thus indexed by $G[i, x, y]$. UE locations do not change throughout the optimisation procedure, with all UEs requesting data constantly (a “full-buffer” model).

The power range for a SC is 23–35 dBm, while the bias β_s can vary from 0–15 dBm. No cell can be completely turned off/muted. In this study we allow

Fig. 1. Simulated coverage area of network deployment.

SCs to adopt non-zero CSB only if their power is already at a maximum, thus minimising the number of SC expanded regions, more formally defined in Eq. 4:

$$\beta_s = \begin{cases} 0 & \text{if } P_s \leq P_{s\_max} \\ \geq 0 & \text{if } P_s = P_{s\_\max} \end{cases}, \forall s \in \mathcal{S}. \tag{4}$$

The set of UEs attached to a cell i is denoted by $\mathcal{A}_i$. The set of UEs for whom cell i is the strongest serving cell of its tier based solely on received power (those UEs that have the *potential* to attach to i, i.e. Eq. 1 when $\beta_i = 0$) is denoted by $\mathcal{P}_i$. The set of UEs attached to SC s who are in the expanded region of s is denoted by $\mathcal{E}_s$. Note that $\mathcal{A}_i \subset \mathcal{P}_i$ and $\mathcal{E}_s \subset \mathcal{A}_s$. We denote by $N_{\mathcal{X}}$ the cardinality of the set $\mathcal{X}$.

ABS. The ratio of the number of ABSs to non-ABSs in a full frame is known as the ABS ratio (ABSr) [3]. This is defined as:

$$ABSr_m = \frac{\left\lceil \frac{\sum_{s \in \mathcal{S}} N_{(\mathcal{E}_s \cap \mathcal{P}_m)}}{\left(\sum_{s \in \mathcal{S}} N_{(\mathcal{E}_s \cap \mathcal{P}_m)}\right) + N_{\mathcal{A}_m}} \times 8 \right\rceil}{8}. \tag{5}$$

Note that this ratio is sensitive only to the ratio of UEs within a single MC sector.

Release 10 of the 3GPP-LTE framework [4] cites eight distinct ABS patterns that can be used (shown in Table 1). While these patterns can be combined in any fashion to suit given ABSrs, since there are only eight patterns ABS ratios must consequently be multiples of 0.125. A MC can never be completely blanked for a full frame (i.e. no MC can run an ABSr of 1; the maximum ABSr is 0.875), and for this study a minimum ABSr of 0.125 is set for all MCs (i.e. no MC can run an ABSr of 0).

Table 1. There are eight possible ABS patterns [4]. The patterns are isomorphic to a full frame of 40 SFs. 1 indicates MC transmission, 0 indicates an ABS.

ABS pattern 1	0111111101	1111110111	1111011111	1101111111
ABS pattern 2	1011111110	1111111011	1111101111	1110111111
ABS pattern 3	1101111111	0111111101	1111110111	1111011111
ABS pattern 4	1110111111	1011111110	1111111011	1111101111
ABS pattern 5	1111011111	1101111111	0111111101	1111110111
ABS pattern 6	1111101111	1110111111	1011111110	1111111011
ABS pattern 7	1111110111	1111011111	1101111111	0111111101
ABS pattern 8	1111111011	1111101111	1110111111	1011111110

For the purposes of this study, each UE u attached to cell i is scheduled to receive data transmissions in all subframes within a full frame for which they have an $SINR_{ui,f} > 1$ for every subframe $f \in \mathcal{F}$, the full frame (i.e. they are receiving more signal from their host cell than interference from the rest of the network). Note that muting of MCs during ABSs affects $SINR$ for all UEs as the sum of all interfering signals diminishes. A UE u can not be scheduled to receive data transmissions from a cell i during a subframe f if their $SINR_{ui,f}$ is less than 1 as this would result in a transmission outage and their data packets would be dropped. Since intelligent scheduling can have a significant effect on UE downlink performance [34], we wish to remove any variability in the simulation so that any performance improvement may be definitively ascribed to the evolved optimisation algorithm.

Finally, since the Shannon formula defined in Eq. 3 gives the downlink rate (in $bits/sec$)for a UE for one particular subframe, the total received downlink for a UE across a full frame $\mathcal{F}$ is averaged across all 40 subframes in the frame:

$$R_{u\_avg} = \frac{\sum\limits_{f \in \mathcal{F}} R_{u,f}}{40}. \tag{6}$$

4.2 Grammar

Since MC powers are invariant, it is desirable that SCs should operate autonomously and regulate their output in accordance with some measurements about their environment. SCs obtain this environmental information through reports from both attached UEs and local MCs. Each UE u attached to a SC s reports specific information back to the SC:

- $SINR_{us,\mathcal{F}}$, the array of all SINRs across the full frame for UE u
- the id of the UE's strongest serving MC
- R_{u_avg} (Eq. 6)

With this information the SC has a profile of its surrounding environment. The SC collates a list of the nearby MCs from all attached UEs. The SC area is then sub-divided into individual MC sectors (i.e. areas where individual MCs are the strongest serving MCs).

Each MC m within the SC s region is then queried for further information:

$$N_{\mathcal{A}_m}, \tag{7}$$

the number of m attached UEs,

$$N_{(\mathcal{P}_m \cap \mathcal{A}_s)}, \tag{8}$$

the number of s attached UEs who are in the MC m sector,

$$avg\_R_m, \tag{9}$$

the average downlink rates of all m attached UEs,

$$\sum_{u \in \mathcal{A}_m} \log(R_{u\_avg}), \tag{10}$$

the sum of the log of the downlink rates of all m attached UEs (a logarithmic scale is used to reward solutions that achieve fairness; changes in the throughput of the worst performing UEs will be highlighted but decreases in performance of the best UEs are less critical),

$$avg\_R_{(\mathcal{P}_m \in \mathcal{A}_s)}, \tag{11}$$

the average downlink rates of all s attached UEs who are in the MC m sector, and

$$\sum_{u \in (\mathcal{P}_m \cap \mathcal{A}_s)} \log(R_{u\_avg}), \tag{12}$$

the sum of the log of the downlink rates of all s attached UEs who are in the MC m sector.

Furthermore, each SC s can provide further information about itself:

$$N_{\mathcal{A}_s}, \tag{13}$$

the number of s attached UEs,

$$avg\_R_s, \tag{14}$$

the average downlink rates of all s attached UEs, and

$$\sum_{u \in \mathcal{A}_s} \log(R_{u\_avg}), \tag{15}$$

the sum of the log of the downlink rates of all s attached UEs.

A grammar (as shown in Fig. 2) was then written for a typical symbolic regression application [21]. The grammar is capable of producing formulae using basic mathematical expressions: $+, -, \times, \%, sin, cos, tan, log, sqrt$. These are used to build arithmetic compositions of the SC-specific values defined in Eqs. 7 to 15, as outlined in Table 2.

Since the values given in Table 2 are defined per MC in a SC sector, evaluation of the function derived from the grammar generates a numerical value for each MC sector within the SC region. The sum of all such generated values per SC is then used as an update value U_s for the power and bias of SC s (as defined in Eq. 16, subject to Eq. 4).

```
<expr> ::= <reg> | <reg> | <val> <op> <val> | <val>
<reg> ::= (<expr> <op> <expr>) | <fn>(<expr>)
<op> ::= + | - | * | %
<fn> ::= sin | cos | tan | log | sqrt
<val> ::= <epsilon> | N_m | N_s | N_ms |
          s_log_R | m_log_R | ms_log_R |
          avg_R_s | avg_R_m | avg_R_ms
<epsilon> ::= 0.000<n><n><n>
<n> ::= 0|1|2|3|4|5|6|7|8|9
```

Fig. 2. Grammar for evolution of HetNet control algorithms.

Table 2. Cell-dependent grammar elements

Grammar element	Meaning
N_m	$N_{\mathcal{A}_m}$
N_s	$N_{\mathcal{A}_s}$
N_ms	$N_{(\mathcal{P}_m \cap \mathcal{A}_s)}$
avg_R_m	avg_R_m
avg_R_s	avg_R_s
avg_R_ms	$avg\_R_{(\mathcal{P}_m \cap \mathcal{A}_s)}$
s_log_R	$\sum_{u \in \mathcal{A}_s} \log(R_{u\_avg})$
m_log_R	$\sum_{u \in \mathcal{A}_m} \log(R_{u\_avg})$
ms_log_R	$\sum_{u \in (\mathcal{P}_m \cap \mathcal{A}_s)} \log(R_{u\_avg})$
epsilon	Constant, from 0.000000 to 0.000999

$$U_s = \sum_{m \in \mathcal{M}} \text{eval}(expr_m). \tag{16}$$

All SCs are initialised with minimum power levels and bias of 0 in order to fairly compare solutions between individuals. Note from Eq. 4 that the power and bias of a SC can be treated as a single entity, as any increase in power past the upper power limit of a SC is translated as an increase in bias.

The objective of this grammar is to provide a single optimization algorithm which can be applied to all SCs in the network deployment (rather than individual algorithms tailored for specific cells). This control algorithm will set the power or bias of the SC based on the evolved parameters. The same algorithm will be run on all SCs across the entire network (i.e. solutions must be generalized and not tailored to individual cells).

4.3 Fitness Function

The performance of a network instance (i.e. a network state with fixed powers, biases, and UE attachments, running for a single full frame) is calculated from the data throughputs of the UEs via:

$$Performance = \sum_{u \in \mathcal{U}} \log(R_{u\_avg}). \tag{17}$$

A logarithmic scale is used in order to magnify the changes in R_{u_avg} of the worst performing UEs. Any decrease in throughput of the best performing UEs is deemed relatively unimportant by this fitness metric, as the focus is on improving the performance of those worst performing UEs. The objective of evolved control algorithms is to maximise this performance metric. The fitness of an individual solution is given by the change in the performance of the network from its initial state to its optimized state after the algorithm is run:

$$Fitness = avg\left(\sum_{u \in \mathcal{U}} \log(R_{u\_avg})_{post\ opt} - \sum_{u \in \mathcal{U}} \log(R_{u\_avg})_{pre\ opt}\right) \tag{18}$$

This fitness is averaged across multiple scenarios in order to evolve solutions which are generalisable.

4.4 Evolutionary Setup

Experiments in this paper adhere to the recommendations of Hemberg *et al.* [2]. Each individual is evaluated across 10 different scenarios in order to ensure generalizability of solutions. Each scenario is characterized by a different UE distribution (subject to the distribution of hotspots, as described in Sect. 4.1). Evolutionary parameters are shown in Table 3. SC powers and biases for each scenario were initialized at 23 dBm and 0 dBm respectively.

Table 3. Evolutionary parameter settings

Number of runs:	50
Pop. Size:	100
Generations:	100
Initialisation:	Sensible
Max tree depth:	10
Crossover type:	Subtree
Crossover probability:	70 %
Mutation types:	Subtree & Point
Selection:	Tournament
Tournament size:	2
Replacement:	Generational with Elites
Elite size:	1

A list of all previously evaluated phenotypes per evolutionary run is retained. If a phenotype has already been evaluated in the evolutionary process then it is discarded and a new randomly initialised individual is inserted into the population in its place. This ensures that all 10,000 evaluations in a single run are unique. The net result of this is that a much wider search area is covered, since the evolutionary process does not waste time re-evaluating known solutions as the best of these exist in the form of elites [29,35].

5 Results and Discussion

The results of the evolutionary runs are shown in the graph in Fig. 3, which displays the average of the best fitnesses across each run as generations progress.

The best evolved algorithm is described in Eq. 19. Note that the control algorithm is run independently on each SC $s \in \mathcal{S}$, and that it must be run once for each MC $m \in \mathcal{M}$ which intersects with the area of influence of s (as described in Sect. 4.2).

$$U_s = \sum_{m \in \mathcal{M}} \frac{\left(\sum_{u \in (\mathcal{P}_m \cap \mathcal{A}_s)} \log(R_{u\_avg}) \right) + N_{(\mathcal{P}_m \cap \mathcal{A}_s)}}{N_{\mathcal{A}_s}} \tag{19}$$

The primary method to evaluate success of a strategy is through analysis of a Cumulative Distribution Function (CDF) graph of the log of the downlink of all UEs in a scenario of the network. Figure 4 compares the evolved strategy against a number of benchmark methods of setting SC powers and biases. A number of observations can be made from Fig. 4:

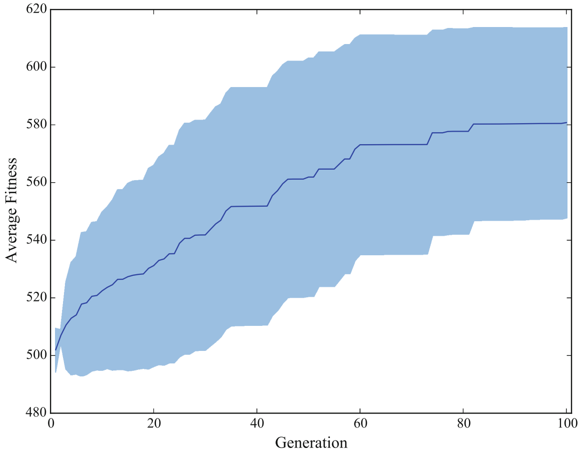

Fig. 3. Average best fitness across all runs, including standard deviation of average best fitness across all runs.

- The height of the lowest portion of any curve represents the total number of UEs in the network for whom $SINR_{max} < 1$ and, thus, cannot be scheduled for any data transmissions.
- The red curve (minimum power, and bias of 0 dBm) represents the CDF of pre-optimised $\log R_{u\_avg}, \forall\ u \in \mathcal{U}$. It can be seen to begin not at the origin of the y-axis, but at a level of $y = 0.0722$. This indicates that there are 91 UEs in the original pre-optimised network for whom $SINR_{max} < 1$ and, thus, cannot be scheduled for any data transmissions.
- Since $\log 0$ is undefined, the performance metric of a network state (as described in Eq. 17) does not account for those UEs who cannot be scheduled to receive any data transmissions (i.e. those UEs that have a data throughput of 0 bps). Thus, easy improvements can be made to the fitness of the network by increasing the number of scheduled UEs. The evolved algorithm generates significant improvement in fitness by increasing the overall number of scheduled UEs, more than any of the benchmark methods.
- Since the fitness function (as defined in Eq. 18) operates on a logarithmic scale, the greatest gains can be made by increasing the throughput of the worst performing UEs in the network. It can be seen that the black curve lies to the right of and below the majority of the other curves, indicating a significant increase in performance for all UEs. This desirable behaviour emerges automatically and is not explicitly rewarded by the fitness function.
- While setting all powers and biases to maximum levels (as shown by the dashed green line) results in marginally improved performance for the majority of UEs over the evolved method, the evolved method can schedule more UEs overall, leading to a better fitness.

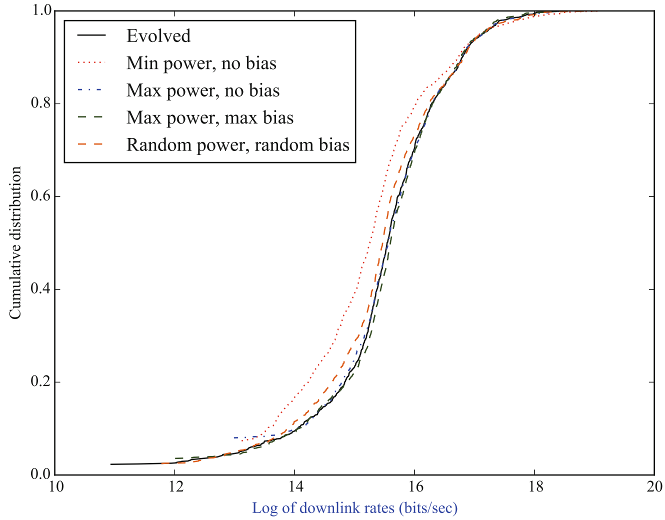

Fig. 4. Cumulative Distribution Function graph of $\log R_{u\_avg}, \forall u \in \mathcal{U}$, the set of all UEs. The black line represents the performance of all UEs under the evolved algorithm. Red, orange, green, and blue lines represent various baseline methods of setting powers and biases(Color figure online).

6 Conclusions and Future Work

Grammatical GP has been shown to be capable of successfully evolving control algorithms for HetNets which give acceptable performance. Analysis of the performance of best evolved algorithm shows that not only does it improve the throughput of the worst performing UEs in the network, but that it also reconfigures the network such that the number of UEs who cannot be scheduled (due to $SINR_{u,max} < 1$) is minimised. This is a critical issue, as fairness is a desirable quality in cellular network performance [4].

A number of areas for further study have been identified as a result of this work:

- At present, all UEs are scheduled to receive data transmissions from their hosting cell in every subframe f where they have an $SINR_{ui,f} > 1$. Significant improvements in UE throughput are expected by more intelligent scheduling methods.
- The function for describing ABS is a simple model. It is possible that further network improvements can be made by evolving optimal ABS functions.
- Optimisation of networks for minimal transmission outages (i.e. minimise the number of UEs for whom $SINR_{max} < 1$) has been described in this paper, but has not been fully realised here as this was not the main objective of the fitness function. The evolution of control algorithms for minimisation of transmission outages would be a worthwhile study.

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

References

1. Cisco: Cisco visual networking index: global mobile data traffic forecast update, 2014 2019. Technical report (2015). http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html
2. 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)
3. Tall, A., Altman, Z., Altman, E.: Self organizing strategies for enhanced ICIC (eICIC). arXiv preprint arXiv:1401.2369 (2014)
4. 3gpp (2014). http://www.3gpp.org/
5. Chandrasekhar, V., Andrews, J., Gatherer, A.: Femtocell networks: a survey. Commun. Mag. **46**, 59–67 (2008)
6. Ho, L.T., Ashraf, I., Claussen, H.: Evolving femtocell coverage optimization algorithms using genetic programming. In: 20th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC), pp. 2132–2136. IEEE (2009)
7. Fagen, D., Vicharelli, P.A., Weitzen, J.A.: Automated wireless coverage optimization with controlled overlap. IEEE Trans. Veh. Technol. **57**, 2395–2403 (2008)
8. Deb, S., Monogioudis, P., Miernik, J., Seymour, J.P.: Algorithms for enhanced inter-cell interference coordination (eICIC) in LTE HetNets. IEEE/ACM Trans. Networking (TON) **22**, 137–150 (2014)
9. Fenton, M., Lynch, D., Kucera, S., Claussen, H., O'Neill, M.: Load balancing in heterogeneous networks using an evolutionary algorithm. In: Proceedings of IEEE Conference on Evolutionary Computation, pp. 70–76, Sendai, Japan (2015)
10. Ryan, C., Collins, J.J., Neill, M.O.: Grammatical evolution: evolving programs for an arbitrary language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, pp. 83–96. Springer, Heidelberg (1998)
11. O'Neill, M., Ryan, C.: Grammatical evolution. IEEE Trans. Evol. Comput. **5**, 349–358 (2001)
12. Shannon, C.E.: Communication in the presence of noise. Proc. Inst. Radio Eng. **37**, 10–21 (1949)
13. M. R1-060877: Frequency domain scheduling for E-UTRA. Technical report TSG RAN1#44bis, March (2006)
14. Bian, Y.Q., Rao, D.: Small Cells big opportunities (2014). www.huawei.com/ilink/en/download/HW_330984
15. Bedekar, A., Agrawal, R.: Optimal muting and load balancing for eICIC. In: 11th International Symposium on Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt), pp. 280–287. IEEE (2013)
16. Combes, R., Altman, Z., Altman, E.: Self-organization in wireless networks: a flow-level perspective. In: Proceedings of INFOCOM, pp. 2946–2950. IEEE (2012)
17. Wu, Q., Esteves, E., Proprietary, Q.: The CDMA2000 High Rate Packet Data System. Artech House, Norwood, MA, USA (2002)

18. López-Pérez, D., Claussen, H.: 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), pp. 173–178. IEEE (2013)
19. Whigham, P.A.: Grammatically-based genetic programming. In: Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications. vol. 16, no. 3, pp. 33–41 (1995)
20. Dempsey, I., O'Neill, M., Brabazon, A.: Foundations in Grammatical Evolution for Dynamic Environments. SCI, vol. 194. Springer, Heidelberg (2009)
21. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
22. Whigham, P.A.: Inductive bias and genetic programming. In: 1st International Conference on Genetic Algoritms in Engineering Systems: Innovations and Applications (GALESIA), pp. 461–466. IET (1995)
23. Backus, J.W., Nauer, F.L., Green, J., Katz, C., McCarthy, J., Naur, P., Perlis, A.J., Rutishauser, H., Samelson, K., Vauquois, B.: Revised report on the algorithmic language ALGOL 60. Comput. J. **5**, 349–367 (1963)
24. 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**, 365–396 (2010)
25. Brabazon, A., O'Neill, M.: Biologically Inspired Algorithms for Financial Modelling. Springer, Berlin (2006)
26. Byrne, J., Fenton, M., Hemberg, E., McDermott, J., O'Neill, M., Shotton, E., Nally, C.: Combining structural analysis and multi-objective criteria for evolutionary architectural design. In: Di Chio, C., Brabazon, A., Di Caro, G.A., Drechsler, R., Farooq, M., Grahl, J., Greenfield, G., Prins, C., Romero, J., Squillero, G., Tarantino, E., Tettamanzi, A.G.B., Urquhart, N., Uyar, A.Ş. (eds.) EvoApplications 2011, Part II. LNCS, vol. 6625, pp. 204–213. Springer, Heidelberg (2011)
27. Fenton, M., McNally, C., Byrne, J., Hemberg, E., McDermott, J., O'Neill, M.: Automatic innovative truss design using grammatical evolution. Autom. Constr. **39**, 59–69 (2014)
28. Fenton, M., McNally, C., Byrne, J., Hemberg, E., McDermott, J., O'Neill, M.: Discrete planar truss optimization by node position variation using grammatical evolution. In: IEEE Transactions on Evolutionary Computation (2016, in press)
29. 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)
30. 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**, 65–93 (2013)
31. Alba, E., Chicano, J.F.: Evolutionary algorithms in telecommunications. In: Electrotechnical Conference (MELECON), pp. 795–798. IEEE (2006)
32. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**, 182–197 (2002)
33. Google maps (2014). https://maps.google.com
34. Weber, A., Stanze, O.: Scheduling strategies for HetNets using eICIC. In: International Conference on Communications (IEEE ICC), pp. 6787–6791. IEEE (2012)
35. Glover, F.: Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. **13**, 533–549 (1986)