Load Balancing in Heterogeneous Networks using an Evolutionary Algorithm

Michael Fenton*, David Lynch†, Stepan Kucera‡, Holger Claussen§ and Michael O'Neill¶
*†¶Natural Computing Research & Applications Group
Complex & Adaptive Systems Laboratory
School of Business
University College Dublin
Email: * michaelfenton1@gmail.com, † david.lynch.1@ucdconnect.ie, ¶ m.oneill@ucd.ie

‡§Bell Laboratories
Alcatel-Lucent Dublin
Ireland
Email: ‡ stepan.kucera@alcatel-lucent.com, § holger.claussen@alcatel-lucent.com

***Abstract*—Grammatical Evolution (GE) is applied to the problem of load balancing in heterogeneous cellular network deployments (HetNets). HetNets are multi-tiered cellular networks for which load balancing is a scalable means to maximise network capacity, assuming similar traffic from all users. This paper describes a proof of concept study in which GE is used in a genetic algorithm-like way to evolve constants which represent cell power and selection bias in order to achieve load balancing in HetNets. A fitness metric is derived to achieve load balancing both locally in sectors and globally across tiers. Initial results show promise for GE as a heuristic for load balancing. This finding motivates a more sophisticated grammar to bring enhanced Inter-Cell Interference Coordination optimisation into an evolutionary framework.**

I. Introduction

Cellular networks are subject to exponentially increasing strain from the proliferation of mobile devices and the shift from voice traffic to multimedia streaming [1]. In order to keep pace with demand, network operators are supplementing their existing Macro Cell (MC) deployments with Small Cells (SCs). SCs are low-power cells located within the operational reach of MCs, with which they share a common channel. The overall performance of a network can be improved by offloading user equipment (UE) from the MC tier to the SC tier, thereby easing congestion and increasing data throughput.

Multi-tiered heterogeneous networks (HetNets) have been identified as an efficient infrastructure for 3rd Generation Partnership Project-Long Term Evolution (3GPP-LTE), the current standard in wireless technology [2]. Channel sharing is attractive to operators because bandwidth is costly and scarce. However, high power MCs pose two problems for SCs within their operational reach. Firstly, low-power SCs are unable to offload enough UEs from the MC tier because UEs attach based on signal strength. Therefore, the SC tier is typically underutilised. Secondly, SC attached UEs in the cell edges are subject to significant interference from the signals of nearby MCs in co-channel deployments.

Cell Range Expansion (CRE) is employed as a mechanism to prevent under-utilisation of the SC tier. Attachment to the SC tier is promoted by adding a bias to the SC pilot signal strengths. SC coverage areas are thereby expanded because UEs attach to whichever cell offers the strongest pilot signal. Under the CRE mechanism UEs may attach to a SC in deference to their better serving local MC for the global good of the network. This is achieved by adding a cell selection bias (CSB) β_i to the reference-signal received power (P_{ui}) of cells, so that UE u attaches to cell k if:

$$k = \arg\max_i (P_{ui} + \beta_i). \tag{1}$$

Where, $\beta_i = 0, \forall i \in \mathcal{M}$, the set of all MCs, and $\beta_i \geq 0, \forall i \in \mathcal{S}$, the set of all SCs [3]. Note that P_i^{TX} (transmitting power of cell i) is subject to path loss so that the signal strength perceived by u is given by,

$$P_{ui} = P_i^{\mathrm{TX}}[\mathrm{dBm}] + G_{ui}[\mathrm{dB}], \tag{2}$$

such that G_{ui} is the signal gain to u, see Section III.

Since UEs greedily attach to their best serving cell per attachment rule (1) intelligent inter-tier load balancing, by setting optimal SC powers and biases, is crucial in HetNets. If a large imbalance exists between the number of UEs (load) attached between MCs and SCs it follows that per-cell congestion will limit downlink rates for UEs attached to congested cells. Expression (1) describes how UEs may be offloaded onto the SC tier by increasing SC CSBs (assuming that SCs are in hotspots). However, UEs in the expanded region of a SC will always receive lower signal strength from their serving SC than from the local MC. It follows that UEs in SC expanded regions will experience strong interference from nearby MCs. Note that the expanded region is the additional overage area realised when CSB is positive. The enhanced Inter-Cell Interference Coordination (eICIC) mechanism has been proposed to mitigate inter-tier cell-edge interference in

978-1-4799-7492-4/15/$31.00

HetNets [2]. A key interference mitigation feature of eICIC is the notion of Almost Blank Subframes (ABSs) [3].

ABSs are quiet subframes during which MCs mute, save for minimal but indispensable control signals, thus providing clear subframes in which SCs can transmit without interference. Note that a full frame consists of 10 subframes that are $1\,\mathrm{ms}$ in duration. As the number of UEs in an expanded region increases due to increasing CSBs, so must the proportion of ABSs sacrificed by nearby MCs so that their influence is minimised. In this sense ABS patterns and CSBs are coupled.

ABSs are necessary in HetNets implementing eICIC because SCs operate at much lower power than MCs. Therefore, UEs at the cell edges experience significant interference from nearby MCs and require quiet periods so that they experience acceptable signal to interference and noise ratios (SINRs). It is important that the user-experience is acceptable for all users including those at cell edges.

The combination of ABSs and CRE enable effective interference management and load balancing in HetNets, see for example [3]–[5]. We assess Grammatical Evolution (GE) [6]–[8] as a potential framework for automatically evolving eICIC algorithms. Since load balancing is a prerequisite for maximising overall network capacity, we first apply GE to the problem of load balancing in a non-eICIC context (i.e. without the use of ABSs). Furthermore, as a first approximation we are assuming a 'full buffer' model where all UEs are static and request data at the same constant rate. Results are encouraging and motivate several directions for future work.

II. Previous Work

Release 10 of 3GPP [9] describes eICIC at the conceptual level but does not specify algorithms for setting ABS patterns and CSBs. This gap has generated a number of research contributions which address the NP-hard problem (see proof Section IV-A in [3]) of setting ABS configurations. Tall et al. (2014) derived an algorithm to jointly balance load and optimise ABS ratios (the proportion of ABSs in a frame) [5]. Their distributed self-organising network algorithm is based on stochastic approximation. A key finding is that network throughput is maximised when, for each SC, UEs attached to the SC centre (i.e. UEs not within the SC expanded region) are scheduled during both ABSs and non-ABSs. Using this result, Deb et al. (2014) employed a non-linear programming formulation which achieved better utility than the standard approach using fixed ABS ratios and CSBs [3]. In both cases convergence is guaranteed by the mathematical structure of the model. As the problem is NP-hard it presents an opportunity for Grammatical Evolution (GE), a heuristic technique in evolutionary computation [8], [10].

GE takes metaphorical inspiration from the principles of evolutionary and molecular biology to create machine executable solutions for a diverse spectrum of problems [6], [7]. GE has been successfully applied to financial modelling, structural engineering and indeed HetNet optimisation [4], [11]–[14]. 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 [15]. A variant of Genetic Programming (GP) [16] but using formal grammars [17], GE is flexible and allows

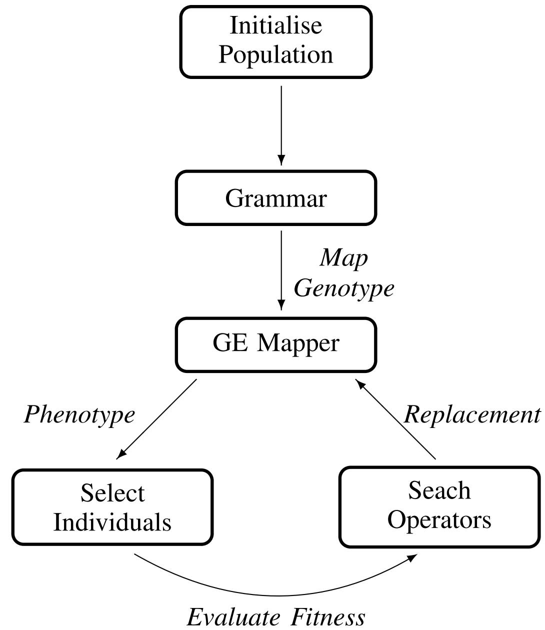

Figure 1. Flow of the GE algorithm. Adapted from [13].

the practitioner to constrain and bias the search by simply modifying the grammar. A strength of GE is that domain knowledge can be incorporated into the grammar to guide the search. Furthermore, simplicity of the crossover and mutation search operators, a key feature of GP, is preserved in GE but the operators can be applied at both the genotypic and phenotypic levels.

The process flow of GE (as illustrated in Figure 1) can be described thusly [6]:

1) A set of chromosomes (integer strings, also known as Genotypes) are generated to initialise the population.
2) Genotypes are then mapped to phenotypes which are executable blocks of code. This mapping is accomplished by reading an individual's chromosome left to right. Each codon (integer) on the chromosome selects a production choice based on the codon value modulo the number of production choices associated with the current non-terminal.
3) Individual solutions are evaluated or executed and a fitness is assigned based on some assessment of their performance.
4) Populations of solutions are varied using selection (tournament, roulette wheel, etc.), variation (crossover and mutation), and replacement (steady-state, generational, etc.).
5) Step 4 is repeated until a terminating condition is satisfied.

Hemberg et al. (2011, 2013) examined a variety of different grammars on the related HetNet coverage optimisation problem [4], [13]. In this instance, the three conflicting objectives of mobility (number of UE hand-overs) minimisation, load balancing, and cell power minimisation were jointly optimised for various indoor femtocell deployment scenarios using the multi-objective optimisation algorithm NSGA-II [18]. The authors employed a symbolic regression approach in [13] 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 [4] the authors compare 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 [19]. 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 the coverage optimisation and the eICIC optimisation problems. Firstly, the objective function is univariate in eICIC because the goal is simply to maximise network capacity (load balancing is a prerequisite for this), 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 GE implementation of eICIC load balancing.

III. Approach Summary

This section describes the fitness model used to evaluate load balancing GE individuals and the simulation set-up. Note that a GE individual describes an array of power and CSB settings for SCs, see Section IV.

A. Fitness Model

We define a typical MC sector for $m \in \mathcal{M}$ as the region in which UEs would attach to m if no SCs were present. Individuals that balance UE load between the SC and MC tiers in all such sectors are assigned a good fitness.

More formally, the sets $\mathcal{P}_m$ and $\mathcal{P}_s$ are populated $\forall m \in \mathcal{M}$ and $\forall s \in \mathcal{S}$, where $\mathcal{P}_m$ describes all UEs for which MC m proves their best serving MC according to equation (1) and $\mathcal{P}_s$ describes all UEs for which SC s proves their best serving SC. In other words, $\mathcal{P}_m$ describes all UEs which would attach to MC m if no SCs were present and $\mathcal{P}_s$ describes all UEs which would attach to SC s if no MCs were present. Attachment is set via equation (1), giving $\mathcal{A}_m \subseteq \mathcal{P}_m$ and $\mathcal{A}_s \subseteq \mathcal{P}_s$, the sets of attached UEs. Observe that $\mathcal{P}_m$ and $\mathcal{P}_s$ store UEs in the 'potential' coverage regions of MCs (hence MC sectors) and SCs but $\mathcal{A}_m$ and $\mathcal{A}_s$ are subsets storing those UEs that actually attach. The former sets depend on the realisation of cell and UE locations, cell powers and G, while the latter additionally depend on SC CSBs.

Figure 2 illustrates the regions in which UEs would populate these sets for a typical MC m. In the upper left quadrant the potential coverage area for m is indicated by the shaded region. Herein the pilot power from m is greater than the pilot powers from all other MCs in the network. In the lower left quadrant the shaded region indicates where the pilot power from m exceeds that of all other MCs *and* all other SCs, herein UEs would populate $\mathcal{A}_m$. Similarly, the shaded regions in the upper right and lower right quadrants depict where UEs would populate $\mathcal{P}_{s_1}$, $\mathcal{P}_{s_2}$ and $\mathcal{A}_{s_1}$, $\mathcal{A}_{s_2}$ respectively. Our goal is to evolve power and CSB constants for SCs so that UEs are evenly shared between the MC and SCs in each MC sector.

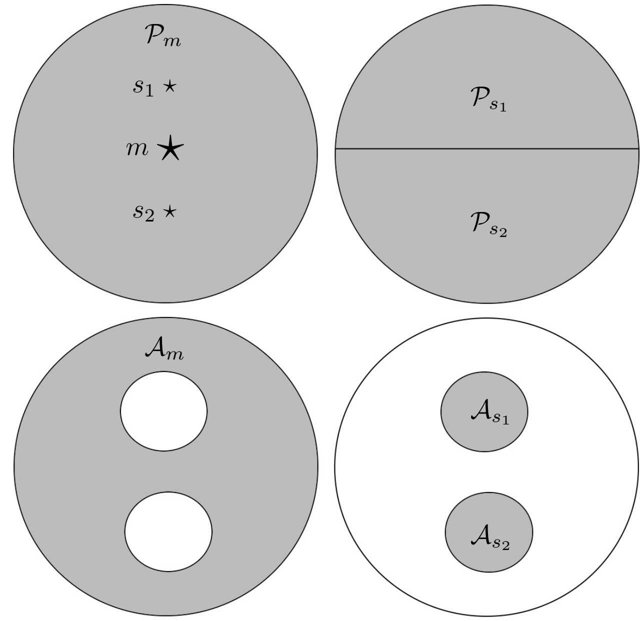

Figure 2. The sets $\mathcal{P}_m$, $\mathcal{A}_m$ (left) and $\mathcal{P}_s$, $\mathcal{A}_s$ (right) are visualised for a MC m and two SCs s_1 and s_2.

A sector is well balanced if $|\mathcal{A}_m| \approx \sum_s |\mathcal{P}_m \cap \mathcal{A}_s| \approx \frac{|\mathcal{P}_m|}{2}$, for SCs s that overlap with that sector. We sum the difference of the square of the imbalances $\forall m \in \mathcal{M}$ to compute the overall fitness of an individual. Thus, a phenotype's fitness is given by:

$$\text{fitness} \leftarrow \sum_{m \in \mathcal{M}} \left(|\mathcal{A}_m|^2 - \left(\sum_{\substack{s:\mathcal{P}_m \cap \mathcal{A}_s \neq \emptyset \\ s \in \mathcal{S}}} |\mathcal{P}_m \cap \mathcal{A}_s| \right)^2 \right) \quad (3)$$

This fitness metric ensures that network settings resulting in high degrees of MC/SC load imbalance are penalised; the greater the imbalance of a particular MC sector, the greater the penalty. Consequently, a low fitness value is desirable as it indicates a well balanced network. The best possible fitness is zero which occurs when load is perfectly balanced in all MC sectors. Equation (3) penalises the overall fitness if any one sector exhibits high imbalance so that resources are shared in a proportionally fair manner. Furthermore, equation (3) tends to achieve global inter-tier balance so that the same number of UEs attach between the SC and MC tiers. In Figure 2 if $|\mathcal{A}_m| = |\mathcal{A}_{s_1}| + |\mathcal{A}_{s_2}|$ then equation (3) would return a fitness of zero because load is perfectly balanced (assuming m is the only MC in this toy network).

Ultimately the goal of eICIC is to provide acceptable downlink rates for all customers served by the HetNet. By load balancing across sectors we approximate this objective because downlink rates improve as congestion eases. Finally, the fitness model described in this section assumes the full-buffer model whereby UEs request data at a constant rate, in reality traffic is variable.

B. Simulation

A network deployment consisting of 21 MCs and 79 SCs is simulated in a $3.61\,\mathrm{km}^2$ sector of Dublin City Centre. MCs are distributed in a regular hexagonal pattern and SCs are distributed randomly to reflect their user-deployed nature, see Figure 3. A Google Maps [20] screen shot of the physical terrain is processed to model obstacles such as buildings, parks and open water, see Figure 4. Signal gain path losses are computed accounting for the location of cells, cell gain, shadow fading, and environmental obstacles.

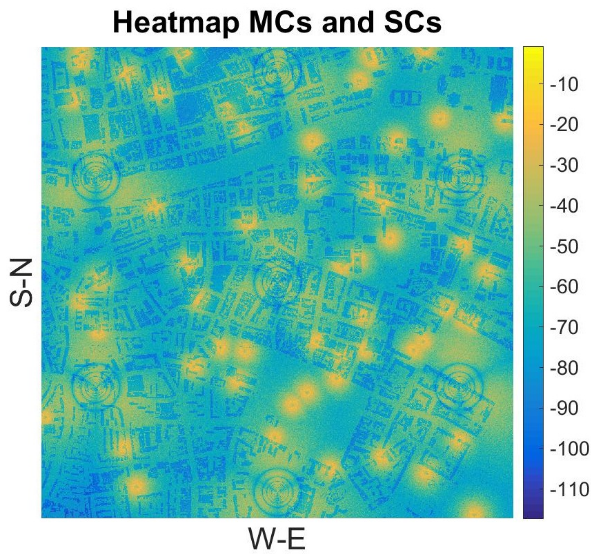

Figure 3. MC sites (concentric circles) are distributed in a hexagonal pattern. SCs are distributed randomly. Cell gains [dB] are visualised as a heat map.

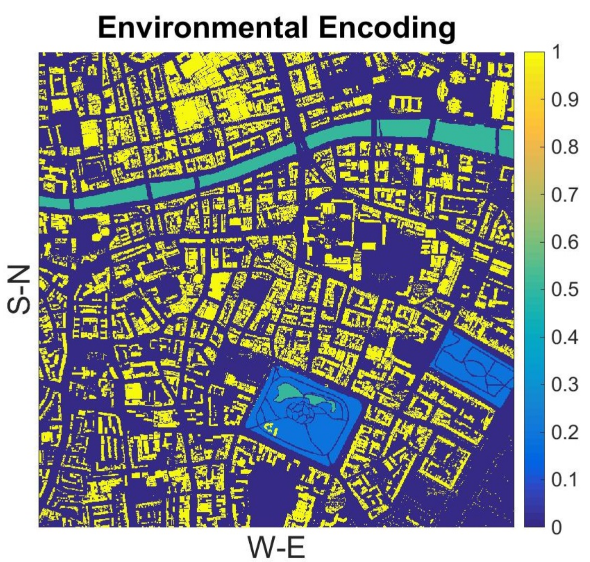

Figure 4. False colour map of Dublin City Centre showing the coding of buildings (yellow), park lands, water and open spaces (shades of blue).

Cell gain data is stored in a layered [900 x 900 x 100] matrix G where $G[i,j,k]$ stores the signal gain in decibels from cell k to the $4\,\mathrm{m}^2$ sector $[i,j]$. Since the grid resolution of the environment map is $2\,\mathrm{m} \times 2\,\mathrm{m}$, and since gain levels are stored for all of the cells for all sectors, it follows that the gain matrix is highly memory intensive, at over 650 MB. This makes optimisation of the network environment quite a computationally intensive problem. Finally, UEs are placed randomly on the map at an average density of 60 UEs per MC sector. MC power is invariant and only SCs can take on non-zero CSBs. SC powers and CSBs are restricted to the intervals 23–35 and 0–5 [dBm] respectively. The simulation parameters are summarised in Table I.

Table I. SIMULATION PARAMETERS

Parameters	Value
Scenario	
Type	capacity problem
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	79 eNodeBs (uniformly randomly distributed)
Inter-MC BS distance	800 m
Scenario resolution	2 m
Transmit power	$P_{\mathrm{tx},n} = 21.6\,\mathrm{W}$ (MC), $1\,\mathrm{W}$ (SC)
LPC power weight set	$\{0, 1\}$ (binary on/off power control)
Noise density	$-174\,\mathrm{dBm/Hz}$
SC REB	$5 - 35$ dB
Channel	
Carrier frequency	2 GHz
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) [21]
	$G_{\mathrm{Pn}} = -30.5 - 36.7\log_{10}(d)$ (SC) [21]
LOS path-loss	$G_{\mathrm{Pl}} = -34.02 - 22\log_{10}(d)$ [21]
Shadow fading (SF)	6 dB std dev. [22]
SF correlation	$R = \mathrm{e}^{-1/20d}$, 50% inter-site
Environment loss	$G_{\mathrm{E},n} = -20$ dB if indoor, 0 dB if outdoor
Antenna	
Height	$25\,\mathrm{m}$ (MC), $10\,\mathrm{m}$ (SC)
Maximum gain	$G_{\max} = 15.5$ dBi (MC), 7.06 dBi (SC)
H. halfpow. beamwidth	$\alpha = 65^\circ$
V. halfpow. beamwidth	$\beta = 11.5^\circ$ (MC)
Front-to-back ratio	$\kappa = 30$ dB (MC)
Downtilt	$\delta_1 = 8.47^\circ$ (MC)
Elements & spacing	4 element dipole, $d_{\mathrm{elem.}}$=0.6λ (SC)
Phase difference	$\delta_{\mathrm{phase}} = 95^\circ$ (SC)
Element amplitude	$a_{\mathrm{elem.}} = [0.9691, 1.0768, 1.0768, 0.8614]$ (SC)

IV. EXPERIMENTS

Constants to represent SC powers and CSBs are evolved using the following simple grammar:

$$
\begin{aligned}
\langle \mathrm{E} \rangle &::= \langle \mathrm{power} \rangle , \langle \mathrm{bias} \rangle \\
\langle \mathrm{power} \rangle &::= \langle \mathrm{P} \rangle , \langle \mathrm{P} \rangle , ..., \langle \mathrm{P} \rangle \\
\langle \mathrm{bias} \rangle &::= \langle \mathrm{B} \rangle , \langle \mathrm{B} \rangle , ..., \langle \mathrm{B} \rangle \\
\langle \mathrm{P} \rangle &::= 23 + \frac{12 \times \langle \mathrm{n} \rangle \langle \mathrm{n} \rangle . \langle \mathrm{n} \rangle}{100} \\
\langle \mathrm{B} \rangle &::= 0 + \frac{5 \times \langle \mathrm{n} \rangle \langle \mathrm{n} \rangle . \langle \mathrm{n} \rangle}{100} \\
\langle \mathrm{n} \rangle &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{aligned}
$$

This grammar allows GE to express SC powers and CSBs (in Decibels, see Table I) as a percentage within the range of their minimum and maximum permissible values, allowing for adjustments in increments of 0.01%. Three codons each are used for both the power and CSB values for each SC, meaning the required chromosome length is six times the number of SCs (474 in total for this simulation). Since the only production rule which produces non-terminals has exactly 10 non-terminals, the maximum codon value is set at 10 so as to avoid any bias towards a particular outcome. It should be noted that while GE operates in a similar fashion to a standard genetic algorithm,

Table II. Evolutionary parameter settings

Number of Runs:	30
Initialisation:	Random
Pop. Size:	500
Generations:	100
Total Evaluations:	50000
Crossover Type:	Single Point
Crossover Probability p_{cross}:	70%
Mutation Type:	Per-Codon
Mutation Probability p_{mut}:	1%
Selection:	Tournament
Tournament Size:	2
Replacement:	Generational with Elites
Elite Size:	3

the above grammar allows for finer control over the ranges from which to choose than a traditional GA implementation.

Evolutionary parameter settings are listed in Table II. Single-point crossover is used so that the same point is chosen on the chromosomes of both individuals. This ensures that the chromosome length is preserved. A randomly selected pair of individuals has a probability p_{cross} of crossing over genetic material, otherwise the pair are carried over unchanged. Mutation then occurs on each individual. Mutation operates on a per-codon basis, where each codon has a probability p_{mut} of being randomly mutated.

The genotype to phenotype mapping process yields two arrays which describe power and bias settings for all SCs in the network. These settings are then passed to the simulation described in Section III which computes fitness via equation (3). The best individual at the end of a run is identified as the optimised network configuration.

A set of parameter sweep experiments were conducted to ascertain the best evolutionary population settings for this problem. Six different settings were examined, with successively increasing population sizes, as delineated in Table III. The total number of evaluations was kept constant at 50000 across all runs and 10 individual runs were executed for each setting. As suggested by Figure 5, no statistically significant differences in the end of run (best) fitnesses were observed across all six settings.

Table III. Parameter sweep experiments

	Population	Generation
1	500	100
2	400	125
3	250	200
4	200	250
5	400	125
6	100	500

Computing the fitness of a single individual takes approximately 0.04 seconds and a single run of 100 generations with a population size of 500 individuals completes in approximately 37 minutes using four cores on a 3.2 GHz Intel Core i5 processor. Note that while a significant speed-up is realised by evaluating individuals in parallel across multiple cores when performing a single run, it is more computationally efficient to execute one run per core when performing multiple runs.

V. Results and Discussion

Grammatical evolution is an effective load balancing heuristic for HetNets. Figure 6 illustrates how the average best fitness across 30 runs, with fixed population size of

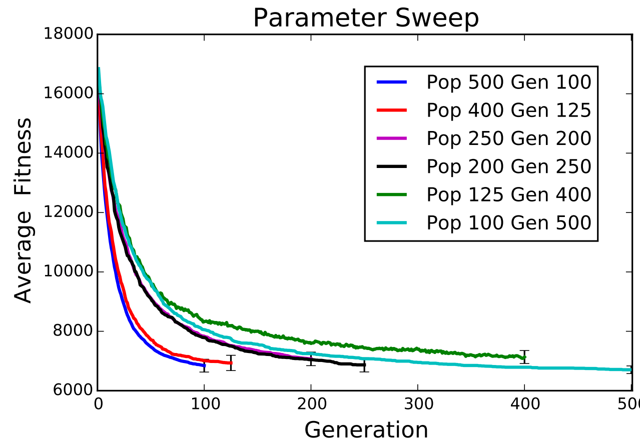

Figure 5. The heuristic is robust with respect to a range of parameter settings.

500, improves monotonically over 100 generations. Recall that low fitness indicates a well balanced network. Furthermore, low variance in fitness across multiple runs (± 216.13 at generation 100) indicates that the method is stable. Figures 5 and 6 illustrate how the heuristic aggressively explores the search space before exploiting promising regions as the fitness converges to an asymptote.

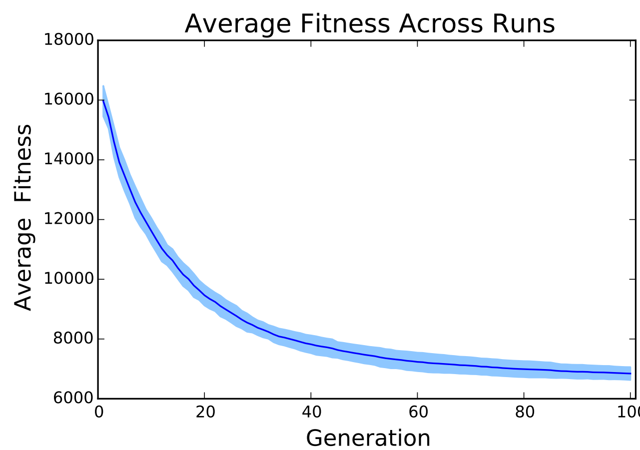

Figure 6. Average best fitness across 30 runs over 100 generations. The shaded region encloses one standard deviation above and below the mean.

Figure 7 compares the empirical distribution function (EDF) of load imbalances for the ten fittest evolved phenotypes against ten randomly generated phenotypes. The leftmost (blue) trace corresponds to the evolved phenotypes, with all MC sectors (10 different configurations, each with 21 MC sectors) achieving an imbalance of 38 or less. Approximately 75 % of all evolved MC sectors have an imbalance of 13 or less, with over 15 % realising perfect balance (imbalance of 0). In contrast, the randomly generated phenotypes (which are representative of non-evolved solutions) give rise to 0 perfectly balanced MC sectors, that is, no sectors have an imbalance of 0, and indeed 10 % of sectors exhibit an imbalance of 40 or more. We can surmise that GE significantly outperforms the random baseline because the EDF trace for the former is shifted left relative to the latter. That is, evolved settings for SC power and bias result in lower load imbalances, where most are

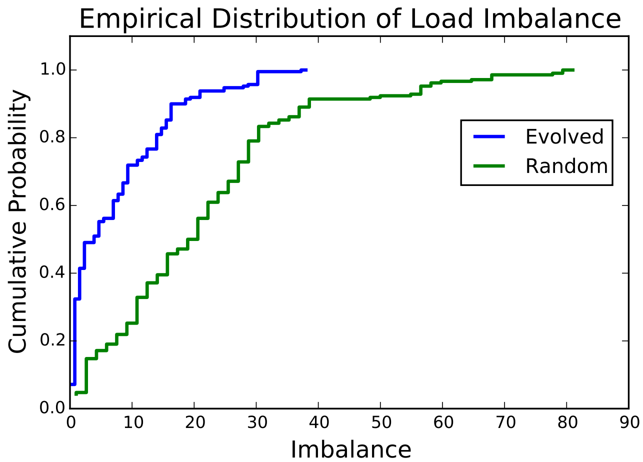

Figure 7. Empirical Distribution Function of MC sector load imbalances for ten evolved phenotypes (leftmost curve) and ten randomly generated phenotypes (rightmost curve). GE clearly outperforms the random baseline.

only a few UEs. As expected the realisation of more extreme imbalances in the randomly configured network indicates very poor performance for some cells.

The sets $\mathcal{P}_m$, $\mathcal{A}_m$ and $\mathcal{A}_s$ summarised in Table IV are realised by the best phenotype (with a fitness of 6549, shown in Table V) over all evolutionary runs. Consider for example MC 2 of Table IV. The optimised settings give $|\mathcal{P}_2| = 70$ UEs for which this is their best serving MC, 35 of whom are attached to the MC itself ($|\mathcal{A}_m| = 35$) with the remaining 35 attaching to SCs in the sector ($|\mathcal{A}_s| = 35$). The load on MC 2 can therefore be considered to be perfectly balanced between the MC and SC tiers in this MC sector. At the other extreme, the imbalance for MC 12 is 38, as the SC tier only offloads 1 UE from the 40 total UEs in this MC's sector.

Table IV. LOAD BALANCING RESULTS

MC ID	$\lvert\mathcal{P}_m\rvert$	$\lvert\mathcal{A}_m\rvert$	$\lvert\mathcal{A}_s\rvert$	|imbalance|	No. Local SCs
1	71	35	36	1	8
2	70	35	35	0	2
3	91	46	45	1	5
4	32	20	12	8	1
5	70	35	35	0	4
6	80	40	40	0	9
7	83	42	41	1	6
8	4	4	0	4	0
9	70	35	35	0	6
10	71	42	29	13	2
11	67	34	33	1	7
12	40	39	1	38	0
13	75	48	27	21	3
14	70	35	35	0	4
15	37	20	17	3	1
16	34	17	17	0	4
17	68	34	34	0	7
18	40	34	6	28	0
19	45	29	16	13	1
20	69	34	35	1	7
21	73	36	37	1	2

For the majority of evolved MCs, it can be seen that the load is well balanced. However, for MCs 4, 10, 12, 13, 18 and 19 it is clear that the load is very unevenly distributed between tiers. Figure 8 depicts a pathological scenario to illustrate why load balancing (by way of adjusting SC powers and CSBs) is a non-trivial problem.

Consider the toy network described in Figure 8. In MC 2 sector, the UE imbalance is 6 (with 7 attached to the MC tier and only 1 attached to the SC tier). In this case it is clear that increasing either the power or CSB (or both) of SC 2 will enable UE offloading from the MC tier, thereby reducing the overall imbalance. However, MC 3 and MC 4 impose conflicting demands on SC 3 which overlaps with both MC sectors. In order to achieve balance for the MC 4 sector, UEs must be offloaded *from* SC 3. However, MC 3 requires more UEs to be offloaded *onto* SC 3 in order to mitigate its imbalance. Hence, it is not possible to satisfy both demands and to perfectly balance the load on this network.

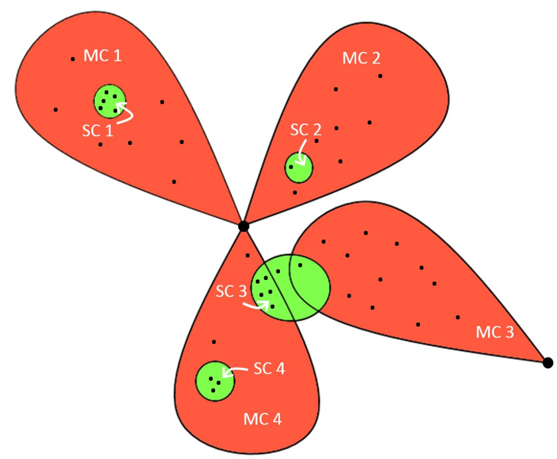

Figure 8. Lobe (red) and circular (green) shaped regions indicate MC and SC coverage areas respectively. UEs are indicated by black dots.

Load balancing algorithms must strike appropriate compromises for SCs that overlap multiple MC sectors. It is therefore unlikely that a network can ever achieve perfect global balance between the MC and SC tiers. In practice conflicting demands will be imposed on those SCs that overlap multiple MC sectors. Note also that in our simulation SCs are randomly distributed on the map. The large imbalances observed in MC sectors 12 and 18 are simply due to a paucity of SCs within their coverage areas. In practice network operators would deploy SCs near traffic hotspots to maximise their offloading potential.

VI. CONCLUSIONS AND FUTURE WORK

GE is capable of evolving well-balanced networks by varying SC power and CSB settings directly through a grammar. However, it is not possible to fully satisfy the need for a balanced network due to conflicting demands on individual SCs which service multiple MC sectors. Compromises must be made in order to create an acceptably balanced network. Furthermore, in reality UEs generate variable traffic bursts so that one 'heavy' user may create a larger strain at a cell than multiple 'light' users. Our assumption of constant static traffic is a first approximation to the ideal model with dynamic traffic. The results provide a proof of concept for the suitability of GE as a heuristic for HetNet optimisation. However, GE is too slow for online deployment when used in a GA-like way to evolve parameters. In future work GE will be used to evolve control algorithms that can optimise network parameters on a much shorter timescale.

The next challenge is to implement full eICIC optimisation in an evolutionary framework with the addition of ABS patterns for MCs. Finally, the evolved algorithms will operate in a social setting because the decisions at an arbitrary cell will have knock-on consequences for all other cells in the network. This motivates future work where the role of social search in evolutionary computation will be investigated. By addressing the socio component of the evo-devo-socio paradigm we aim to evolve eICIC algorithms that work well as a collective.

Acknowledgements

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

References

[1] C. V. N. Index. Global mobile data traffic forecast update, cisco white paper.

[2] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, "A survey on 3gpp heterogeneous networks," *Wireless Communications, IEEE*, vol. 18, no. 3, pp. 10–21, 2011.

[3] S. Deb, P. Monogioudis, J. Miernik, and J. P. Seymour, "Algorithms for enhanced inter-cell interference coordination (eicic) in lte hetnets," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 1, pp. 137–150, 2014.

[4] E. Hemberg, L. Ho, M. ONeill, and H. Claussen, "A comparison of grammatical genetic programming grammars for controlling femtocell network coverage," *Genetic Programming and Evolvable Machines*, vol. 14, no. 1, pp. 65–93, 2013.

[5] A. Tall, Z. Altman, and E. Altman, "Self organizing strategies for enhanced icic (eicic)," *arXiv preprint arXiv:1401.2369*, 2014.

[6] M. O'Neill and C. Ryan, *Grammatical evolution: evolutionary automatic programming in an arbitrary language*. Springer, 2003, vol. 4.

[7] I. Dempsey, M. O'Neill, and A. Brabazon, *Foundations in grammatical evolution for dynamic environments*. Springer, 2009, vol. 194.

[8] M. O'Neill and C. Ryan, "Grammatical evolution," *Evolutionary Computation, IEEE Transactions on*, vol. 5, no. 4, pp. 349–358, april 2001.

[9] (2014, Dec.) 3gpp. [Online]. Available: http://www.3gpp.org/

[10] C. Ryan, J. Collins, and M. O. Neill, "Grammatical evolution: Evolving programs for an arbitrary language," in *Genetic Programming*. Springer, 1998, pp. 83–96.

[11] A. Brabazon and M. O'Neill, *Biologically inspired algorithms for financial modelling*. Springer, 2006.

[12] J. Byrne, M. Fenton, E. Hemberg, J. McDermott, M. ONeill, E. Shotton, and C. Nally, "Combining structural analysis and multi-objective criteria for evolutionary architectural design," in *Applications of Evolutionary Computation*. Springer, 2011, pp. 204–213.

[13] E. Hemberg, L. Ho, M. O'Neill, and H. Claussen, "A symbolic regression approach to manage femtocell coverage using grammatical genetic programming," in *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*. ACM, 2011, pp. 639–646.

[14] M. Fenton, C. McNally, J. Byrne, E. Hemberg, J. McDermott, and M. O'Neill, "Automatic innovative truss design using grammatical evolution," *Automation in Construction*, vol. 39, pp. 59–69, 2014.

[15] I. Dempsey, M. O'Neill, and A. Brabazon, *Foundations in grammatical evolution for dynamic environments*. Springer, 2009, vol. 194.

[16] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992, vol. 1.

[17] R. I. Mckay, N. X. Hoai, P. A. Whigham, Y. Shan, and M. ONeill, "Grammar-based genetic programming: a survey," *Genetic Programming and Evolvable Machines*, vol. 11, no. 3-4, pp. 365–396, 2010.

[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[19] L. T. Ho, I. Ashraf, and H. Claussen, "Evolving femtocell coverage optimization algorithms using genetic programming," in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*. IEEE, 2009, pp. 2132–2136.

[20] (2014, Dec.) Google maps. [Online]. Available: https://www.google.ie/maps/@53.3402244,-6.2646475,2229m/data=!3m1!1e3?hl=en

[21] "Further advancements of E-UTRA physical layer aspects," 3GPP Technical Specification Group Radio Access Network, Evolved Universal Terrestrial Radio Access (E-UTRA), Tech. Rep. TR 36.814 v9.0.0, Mar. 2010. [Online]. Available: http://www.3gpp.org

[22] "Mobility enhancements in heterogeneous networks," 3GPP Technical Specification Group Radio Access Network, Evolved Universal Terrestrial Radio Access (E-UTRA), Tech. Rep. TS 36.839 v11.0.0, Sep. 2012. [Online]. Available: http://www.3gpp.org

Appendix

Table V. Best GE phenotype

SC ID	CSB	Power	Cell ID	CSB	Power
1	4.15	26.73	41	4.36	30.26
2	2.37	23.29	42	4.83	33.46
3	2.15	30.30	43	2.08	25.83
4	4.27	34.63	44	2.44	27.00
5	1.52	27.44	45	0.03	34.66
6	1.14	23.76	46	1.09	23.07
7	3.91	32.64	47	0.73	29.83
8	0.92	23.24	48	4.65	32.28
9	4.60	33.54	49	0.98	28.18
10	1.04	26.64	50	0.87	25.68
11	4.08	23.86	51	2.23	30.93
12	4.88	23.04	52	2.32	32.28
13	4.35	33.56	53	4.07	31.34
14	2.71	28.65	54	4.32	33.78
15	2.23	23.28	55	3.65	32.30
16	4.48	23.92	56	4.01	34.28
17	4.15	33.18	57	1.81	24.56
18	0.48	28.32	58	0.87	24.49
19	1.27	23.25	59	1.39	23.92
20	0.76	25.57	60	1.32	24.44
21	4.41	33.26	61	2.45	26.95
22	1.14	30.69	62	3.63	32.92
23	4.28	33.34	63	1.92	25.72
24	1.54	25.40	64	0.92	24.82
25	0.99	25.64	65	2.35	31.72
26	1.49	30.52	66	1.61	26.85
27	0.88	26.06	67	4.73	34.93
28	4.90	34.95	68	0.36	23.32
29	1.69	27.82	69	0.59	24.45
30	4.93	33.74	70	0.13	24.31
31	4.96	27.62	71	4.17	33.15
32	2.17	23.34	72	0.56	24.60
33	4.30	34.10	73	3.63	26.96
34	2.57	32.65	74	4.18	23.95
35	1.92	33.64	75	1.93	31.10
36	4.94	34.60	76	4.75	33.52
37	0.09	24.01	77	0.16	25.41
38	4.97	34.05	78	3.99	23.35
39	4.26	30.37	79	0.63	26.96
40	0.06	25.45	—	—	—