Evolutionary Learning of Link Allocation Algorithms for 5G Heterogeneous Wireless Communications Networks

David Lynch
Natural Computing Research and Applications Group
School of Business
University College Dublin
david.lynch@ucd.ie

Takfarinas Saber
Natural Computing Research and Applications Group
School of Computer Science
University College Dublin
takfarinas.saber@ucd.ie

Stepan Kucera
Nokia Bell Labs
Blanchardstown Business and Technology Park
Dublin
stepan.kucera@nokia-bell-labs.com

Holger Claussen
Nokia Bell Labs
Blanchardstown Business and Technology Park
Dublin
holger.claussen@nokia-bell-labs.com

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

ABSTRACT

Wireless communications networks are operating at breaking point during an era of relentless traffic growth. Network operators must utilize scarce and expensive wireless spectrum efficiently in order to satisfy demand. Spectrum on the links between cells and user equipments ('users': smartphones, tablets, etc.) frequently becomes congested. Capacity can be increased by transmitting data packets via multiple links. Packets can be routed through multiple Long Term Evolution (LTE) links in existing fourth generation (4G) networks. In future 5G deployments, users will be equipped to receive packets over LTE, WiFi, and millimetre wave links simultaneously.

How can we allocate spectrum on links, so that all customers experience an acceptable quality of service? Building effective schedulers for link allocation requires considerable human expertise. We automate the design process through the novel application of evolutionary algorithms. Evolved schedulers boost downlink rates by over 150% for the worst-performing users, relative to a single-link baseline. The proposed techniques significantly outperform a benchmark algorithm from the literature. The experiments illustrate the promise of evolutionary algorithms as a paradigm for managing 5G software-defined wireless communications networks.

KEYWORDS

Link Allocation, Genetic Programming, Scheduling, 5G

ACM Reference format:
David Lynch, Takfarinas Saber, Stepan Kucera, Holger Claussen, and Michael O'Neill. 2019. Evolutionary Learning of Link Allocation Algorithms for 5G Heterogeneous Wireless Communications Networks. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019, Prague, Czech Republic, July 13–17, 2019 (GECCO '19)*, 8 pages.
DOI: 10.1145/3321707.3321853

GECCO '19, Prague, Czech Republic

DOI: 10.1145/3321707.3321853

1 INTRODUCTION

Wireless traffic has been exploding in recent years due to the emergence of data-hungry smart devices. Operators such as AT&T Inc. must increase the capacity of their deployments in order to satisfy demand. One approach is to purchase additional wireless spectrum (i.e. radio bandwidth). However, spectrum is a scarce and expensive resource. A more cost-effective tactic is to utilize the available spectrum more efficiently.

The steadily increasing sophistication of wireless technology has been driven by unrelenting traffic growth. In the first generation (1G) analog systems, a single elevated antenna provided blanket coverage across an entire metropolitan region. Researchers at Bell Laboratories invented the cellular concept in the 1950s [11]. Here, multiple antennas form distinct cells in which the same spectrum can be reused. Increasingly innovating strategies for managing spectrum were developed in the digital 2G and 3G systems [13]. The idea of simultaneously routing packets through multiple LTE links was incorporated into the 4G standard [17]. Future 5G deployments will have a heterogeneous architecture whereby LTE, WiFi, and millimeter wave cells will coexist [3]. Schedulers for allocating spectrum on LTE and WiFi links have been presented in previous work [9, 15]. However, a unified framework for automatically constructing schedulers has not yet been proposed in the literature.

Evolutionary algorithms lend themselves well to the 5G context for two main reasons. Firstly, they have been successfully employed to optimize 4G networks [7, 10]. Secondly, simulated evolution produces robust solutions in dynamic and uncertain environments [2, 14]. The main contributions of this paper are as follows:

(1) A novel data-driven framework for synthesizing 5G network controllers is presented. Building network controllers is a complex task that requires significant human effort [6]. Automation through evolution dispenses with the need for costly manual design.

(2) Evolution learns tailored strategies by leveraging training data collected from within a deployment scenario. Inflexible manually designed controllers are highly suboptimal one-size-fits-all solutions.
(3) Extensive simulations reveal that significant performance gains are achieved versus benchmark techniques.
(4) An analysis of the evolved strategies for allocating spectrum is presented.

We define the optimization problem in the next section. The evolutionary techniques and benchmark algorithm are presented in Section 3. The simulation environment and experiments are described in Section 4. A discussion of the results follows in Section 5. Finally, the paper concludes with a discussion of the contributions and limitations of the research, and directions for future work.

2 PROBLEM DEFINITION

The benefits of exploiting multiple links are assessed in this section. In single-link (SL) mode, users receive packets over just one link (i.e. from only one cell). By contrast in multiple-link (ML) mode, packets are routed via multiple links simultaneously. Sample calculations show that users can experience an improved quality of service in ML mode. Hence, the optimization problem addressed by this paper is motivated.

2.1 Downlink Rates

Consider the toy heterogeneous network (HetNet) containing two LTE cells $c_1, c_2 \in \mathcal{L}$ and one WiFi cell $c_3 \in \mathcal{W}$ that is depicted in Figure 1. Three users $u_1, u_2, u_3 \in \mathcal{U}_t$ are active during some time slot t. A time slot is a brief interval lasting several milliseconds during which cells assimilate measurement data from users and render optimization decisions. The signal strength experienced by a user from any cell depends on two factors: the transmitting power of the cell, and how strongly the signal is attenuated before reaching the user. More formally, the signal received by user u from cell c is given by

$$signal_{c,u} = power_c + gain_{c,u}, \tag{1}$$

where $power_c$ is the power of cell c in decibel-milliwatts [dBm], and $gain_{c,u}$ is the signal gain on the link from c to u in decibels [dB]. LTE and WiFi cells typically transmit at powers of 35 [dBm] and 20 [dBm] respectively. Link gains depend on weather conditions and the distribution of buildings etc, since these factors affect how radio waves propagate.

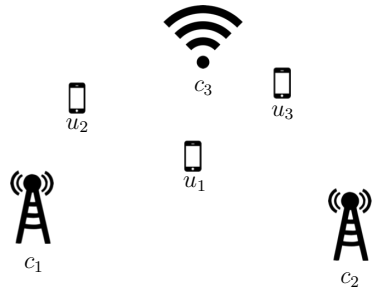


Figure 1: Toy HetNet with WiFi and LTE cells.

	u_1	u_2	u_3
c_1	−182	−173	−180
c_2	−180	−179	−170
c_3	−160	−165	−150

(a) $gain_{c,u}$ [dB]

	u_1	u_2	u_3
c_1	−147	−138	−145
c_2	−145	−144	−135
c_3	−140	−145	−130

(b) $signal_{c,u}$ [dBm]

Table 1: The signal gains on links between cells c and users u in decibels (LHS) in the toy HetNet, and the signal strengths in decibel-milliwatts (RHS) are displayed.

The link gains in the toy HetNet, and the signal strengths provided by each link are displayed in Table 1. For instance, the signal strength experienced by user u_1 from cell c_1 is given by

$$\begin{aligned} signal_{c_1,u_1} &= power_{c_1} + gain_{c_1,u_1} \\ &= 35 \text{ [dBm]} - 182 \text{ [dB]} \\ &= -147 \text{ [dBm]}. \end{aligned}$$

Now, cells that transmit across the same radio band will mutually interfere. For example, a user that receives data from an LTE cell is subject to interference from other nearby LTE cells. The signal to interference and noise ratio for a user u served by an LTE cell $c \in \mathcal{L}$ is given by

$$SINR_{c,u} = \frac{dBm\_w\left(signal_{c,u}\right)}{dBm\_w\left(noise\right) + \sum\limits_{c' \in \mathcal{L} \setminus c} dBm\_w\left(signal_{c',u}\right)}, \tag{2}$$

where $dBm\_w(\cdot)$ is a function that converts the signal strength in decibel-milliwatts to watts, and $noise = -124$ [dBm] is background electromagnetic noise. Set $\mathcal{L}$ in the denominator is replaced by set $\mathcal{W}$ if c is a WiFi cell. Finally, it is convenient to define the quality of the link between c and u:

$$quality_{c,u} = \log_2\left(1 + SINR_{c,u}\right). \tag{3}$$

The downlink rate quantifies how much data is transferred over a wireless link per unit time. Let $rate_{c,u}$ denote the downlink rate on the link between c and u. The value of $rate_{c,u}$ depends on three factors: the link quality $\left(quality_{c,u}\right)$, the spectrum over which c can transmit $\left(spectrum_c\right)$, and the proportion of spectrum allocated to u by c $\left(schedule_{c,u}\right)$. Hence, Shannon's formula yields the downlink link rate as follows:

$$rate_{c,u} = quality_{c,u} \times spectrum_c \times schedule_{c,u}. \tag{4}$$

The values $schedule_{c,u} \in [0, 1]$ specify how much spectrum c allocates to each user u in the HetNet. LTE and WiFi cells typically utilize spectral bandwidths of 20 [MHz] and 22 [MHz] respectively. Our goal is to optimize the schedules for each cell in the HetNet. Updated schedules must be computed on a millisecond timescale.

2.2 Quality of Service and Fairness

The sample calculations in Table 2 illustrate how downlink rates are computed for users in the toy HetNet. Downlink rates are compared in the SL and ML modes. Recall that each user connects to their best-serving cell only in SL mode. That is, a user receives packets

over whatever link offers the highest signal quality. On the other hand, a user can receive packets via multiple cells in ML mode.

	u_1	u_2	u_3
c_1	0.01	0.06	0.01
c_2	0.01	0.01	0.11
c_3	0.04	0.01	0.32

(a) $quality_{c,u}$

	u_1	u_2	u_3
c_1	0.00	1.00	0.00
c_2	0.00	0.00	0.00
c_3	0.50	0.00	0.50

(b) $schedule^{SL}_{c,u}$

	u_1	u_2	u_3
c_1	0.26	0.74	0.00
c_2	0.68	0.25	0.07
c_3	0.87	0.01	0.12

(c) $schedule^{ML}_{c,u}$

c_1	c_2	c_3
20.0	20.0	22.0

(d) $spectrum_c$ [MHz]

	u_1	u_2	u_3
c_1	0.00	1.20	0.00
c_2	0.00	0.00	0.00
c_3	0.44	0.00	3.52

(e) $rate^{SL}_{c,u}$ [Mbps]

	u_1	u_2	u_3
c_1	0.05	0.89	0.00
c_2	0.14	0.05	0.15
c_3	0.77	0.00	0.84

(f) $rate^{ML}_{c,u}$ [Mbps]

Table 2: Downlink rates are computed for users in the toy HetNet. The SL schedules in panel (b) and the ML schedules in panel (c) are compared. A user receives spectrum on a single link in SL mode, but multiple links in ML mode. Shannon's formula yields the downlink rates displayed in panels (e) and (f).

The link qualities between cells and users in the toy HetNet are displayed in Table 2a. Table 2d indicates that LTE cells $c_1, c_2 \in \mathcal{L}$ transmit over a 20 [MHz] band, whereas the WiFi cell $c_3 \in \mathcal{W}$ utilizes a distinct 22 [MHz] band. The SL schedule in Table 2b can be interpreted as follows. Users u_1 and u_3 connect to cell c_3 because it offers them the best signal quality; both receive an equal 50% share of the WiFi spectrum. Similarly, u_2 connects to its best-serving cell c_1. All of the LTE spectrum is allocated to u_2 on the link to c_1, since no other users connect to c_1. A feasible ML schedule is displayed in Table 2c. Each cell allocates a tunable proportion of its spectrum to users u_1, u_2, and u_3. For instance, c_1 allocates 26% of the LTE spectrum on the link to u_1, the remaining 74% of the LTE spectrum on the link to u_2, and 0% of the LTE spectrum on the link to u_3.

Shannon's formula yields the downlink rates on each link. For instance, the downlink rate on the link between c_1 and u_1 under the ML schedule is given by

$$\begin{aligned} rate^{ML}_{c_1,u_1} &= quality_{c_1,u_1} \times spectrum_{c_1} \times schedule^{ML}_{c_1,u_1} \\ &= 0.01 \times 20.0 \text{ [MHz]} \times 0.26 \\ &= 0.05 \text{ [Mbps]}. \end{aligned}$$

The aggregate downlink rate for u_1 over all available links is:

$$\begin{aligned} rate^{ML}_{u_1} &= rate^{ML}_{c_1,u_1} + rate^{ML}_{c_2,u_1} + rate^{ML}_{c_3,u_1} \\ &= 0.05 \text{ [Mbps]} + 0.14 \text{ [Mbps]} + 0.77 \text{ [Mbps]} \\ &= 0.96 \text{ [Mbps]}. \end{aligned}$$

Figure 2 visualizes the downlink rates that are received when the SL (red bars) and ML (blue bars) schedules from Table 2 are implemented. Intensive applications like video streaming require some minimal downlink rate to function properly. Let the threshold rate to support such an application be 0.75 [Mbps] (as indicated by the dashed green line in Figure 2). The SL schedule results in a high downlink rate for u_2 and u_3, but at the expense of an unacceptably low rate for u_1. By contrast, the ML schedule allocates spectrum more fairly, so that all three users receive a downlink rate that exceeds the threshold.

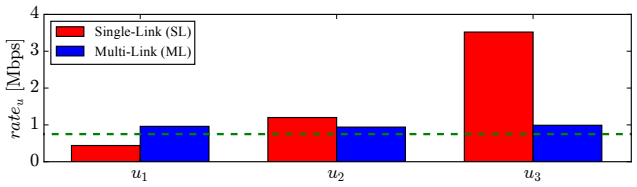


Figure 2: Aggregate downlink rates received by users in the toy HetNet are plotted. The dashed green line indicates a threshold downlink rate of 0.75 [Mbps] needed to run data-hungry applications. User u_1 will experience service disruption if cells execute the SL schedules. However, all users experience an acceptable quality of service in ML mode.

2.3 Defining the Optimization Task

In summary, operators strive to allocate resources fairly, so that all users experience an acceptable quality of service. The traditional approach, whereby data is transferred over a single communications link, exhibits poor fairness. Improved fairness can be achieved when data is routed via multiple links. However, schedulers for intelligently allocating spectrum on the available links are needed. Our goal is to evolve schedulers using evolutionary algorithms. Evolved schedulers must map link quality reports from users to optimized schedules in real time.

3 METHOD

In this section, an algorithm for generating schedules in every time slot is presented. The algorithm first computes statistical features extracted from link quality measurement reports. Hence, a scheduler maps the features to an optimized schedule for each cell. Two different model classes are proposed. Firstly, Grammar-based Genetic Programming [1, 12] is employed to explore the space of symbolic expressions defined by a grammar. Secondly, the weights of a fixed topology multilayer perceptron are optimized using a Genetic Algorithm [5].

3.1 Generating Schedules

We assume that the users $u \in \mathcal{U}_t$ active in time slot t send link quality reports to cells after t. Cells assimilated these data in order to intelligently allocate links in the subsequent time slot. Table 3 defines fourteen statistical features that are relevant for deciding how much spectrum cell c should allocate to user u. For instance, it may be sensible for c to grant u less spectrum if the link quality $(F1_{c,u})$ is large, relative to the average link quality from c to all other users $(F5_{c,u})$. The average, min, and max statistical moments have their usual interpretation. The function $rank(x, \mathcal{X})$ returns the relative magnitude of variable x in the set $\mathcal{X}$. For example, $rank(20, \{30, 20, 10, 0\}) = 2$ because 20 is the second largest value in the set $\{30, 20, 10, 0\}$. The rightmost column displays the values attained by each feature for the link between c_1 and u_1 in the toy HetNet (see Table 2a for the values of $Q_{c,u}$).

The features defined in Table 3 reflect human intuition about what information is relevant when computing a schedule. However, they were chosen in a largely arbitrary manner. Our goal is to automatically learn useful higher-order representations through the mechanism of simulated evolution.

Feature	Definition	F_{c_1,u_1}
$F1_{c,u}$	$Q_{c,u}$	0.01
$F2_{c,u}$	average$\{Q_{c',u} \mid c' \in \mathcal{L} \cup \mathcal{W}\}$	0.02
$F3_{c,u}$	min$\{Q_{c',u} \mid c' \in \mathcal{L} \cup \mathcal{W}\}$	0.01
$F4_{c,u}$	max$\{Q_{c',u} \mid c' \in \mathcal{L} \cup \mathcal{W}\}$	0.04
$F5_{c,u}$	average$\{Q_{c,u'} \mid u' \in \mathcal{U}_t\}$	0.03
$F6_{c,u}$	min$\{Q_{c,u'} \mid u' \in \mathcal{U}_t\}$	0.01
$F7_{c,u}$	max$\{Q_{c,u'} \mid u' \in \mathcal{U}_t\}$	0.06
$F8_{c,u}$	average$\{Q_{c',u} \mid c' \in \mathcal{L}\}$ if $c \in \mathcal{L}$, else average$\{Q_{c',u} \mid c' \in \mathcal{W}\}$ if $c \in \mathcal{W}$	0.01
$F9_{c,u}$	min$\{Q_{c',u} \mid c' \in \mathcal{L}\}$ if $c \in \mathcal{L}$, else min$\{Q_{c',u} \mid c' \in \mathcal{W}\}$ if $c \in \mathcal{W}$	0.01
$F10_{c,u}$	max$\{Q_{c',u} \mid c' \in \mathcal{L}\}$ if $c \in \mathcal{L}$, else max$\{Q_{c',u} \mid c' \in \mathcal{W}\}$ if $c \in \mathcal{W}$	0.01
$F11_{c,u}$	rank $(Q_{c,u}, \{Q_{c',u} \mid c' \in \mathcal{L} \cup \mathcal{W}\})$	3
$F12_{c,u}$	rank $(Q_{c,u}, \{Q_{c,u'} \mid u' \in \mathcal{U}_t\})$	3
$F13_{c,u}$	rank $(Q_{c,u}, \{Q_{c',u} \mid c' \in \mathcal{L}\})$ if $c \in \mathcal{L}$, else rank $(Q_{c,u}, \{Q_{c',u} \mid c' \in \mathcal{W}\})$ if $c \in \mathcal{W}$	2
$F14_{c,u}$	0 if $c \in \mathcal{L}$, else 1 if $c \in \mathcal{W}$	0

Table 3: Statistical features are extracted from link quality reports. Features are then mapped to the proportion of spectrum cell c allocates to user u.

Algorithm 1 Generating a schedule using an arbitrary scheduler.

1: **procedure** GENERATE SCHEDULE($Q_{c,u}$)
2: **for** $c \in \mathcal{L} \cup \mathcal{W}$ **do** #for each LTE and WiFi cell
3: **for** $u \in \mathcal{U}_t$ **do** #for each active user in time slot t
4: Compute the features $F1_{c,u}, F2_{c,u}, \ldots, F14_{c,u}$;
5: Arrange features into a vector $\overrightarrow{F}_{c,u}$;
6: **for** $c \in \mathcal{L} \cup \mathcal{W}$ **do** #cells use exactly 100% of the spectrum
7: **for** $u \in \mathcal{U}_t$ **do**
8: $$schedule_{c,u} = \frac{\exp\left(scheduler_{c,u}\left(\overrightarrow{F}_{c,u}\right)\right)}{\sum_{u \in \mathcal{U}_t} \exp\left(scheduler_{c,u}\left(\overrightarrow{F}_{c,u}\right)\right)};$$

Algorithm 1 describes how the features are transformed into a schedule for each cell. Feature vectors are first computed for all links in the HetNet (lines 2–5). Hence, the scheduler transforms the feature vectors into a feasible schedule (lines 6–8). A cell cannot use more than the available spectrum, and should it not under-utilize spectrum. Thus, the scaling operation on line 8 ensures that exactly 100% of the spectrum is allocated by each cell. The form of the scheduler has not been specified in Algorithm 1.

3.2 Evaluating Schedules

Schedulers are evaluated based on their ability to increase cell edge throughput. As outlined in Section 2, this can be achieved by allocating spectrum fairly on links between cells and users. The abstract concept of fairness is given a precise interpretation here.

Consider a time slot t during which users $u \in \mathcal{U}_t$ are active in the HetNet. Let the schedules for all cells be generated using Algorithm 1. Hence, obtain the received downlink rates with Equation 4. The fitness of the scheduler that is utilized in Algorithm 1 is given by

$$fitness(t) = \sum_{u'=1}^{|\mathcal{U}_t|} \left(\log_e\left(\overrightarrow{rate}_{u'}\right)\right) \times e^{\frac{\gamma \times u'}{|\mathcal{U}_t|}}, \tag{5}$$

where $\overrightarrow{rate}$ is an ordered array storing the downlink rates $\forall u \in \mathcal{U}_t$, such that $\overrightarrow{rate}_1 \geq \overrightarrow{rate}_2 \geq \ldots \geq \overrightarrow{rate}_{|\mathcal{U}_t|}$. The logarithm makes Equation 5 sensitive to changes in the lowest downlink rates. The exponential term further increases sensitivity to cell edge throughput. Pilot experiments indicated that fairness is maximized when the hyperparameter $\gamma \in \mathbb{R}$ is set to the value $\gamma = 10$.

3.3 Genetic Programming

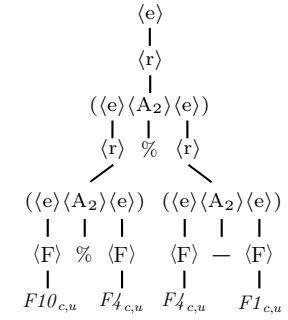
Grammar-based Genetic Programming (GP) is a powerful technique for automatically constructing schedulers given only minimal domain knowledge. The Backus-Naur form grammar that is displayed in Figure 3a defines a search space of symbolic expressions. GP explores this space using the mechanisms of evolutionary search. The grammar contains arithmetic and non-linear functions, numerical constants, and the statistical features that were defined in Table 3. The non-linear functions are defined in Figure 3b.

$\langle e \rangle$ ::= $\langle r \rangle$ | $\langle r \rangle$ | $\langle r \rangle$ | $\langle F \rangle$
$\langle r \rangle$ ::= $\langle A_1 \rangle (\langle e \rangle)$ | $(\langle e \rangle \langle A_2 \rangle \langle e \rangle)$
$\langle A_1 \rangle$ ::= $plog$ | $sine$ | $psqrt$ | $sign$
$\langle A_2 \rangle$ ::= + | − | × | %
$\langle F \rangle$ ::= $F1_{c,u}$ | $F2_{c,u}$ | ,..., | $F14_{c,u}$ | $\langle n \rangle$
$\langle n \rangle$ ::= −1.0 | −0.9 | ,..., | 0.9 | 1.0

(a) Backus-Naur Form Grammar.

$$plog(x) = \log_e(1 + |x|)$$
$$sine(x) = sine(x)$$
$$psqrt(x) = \sqrt{|x|}$$
$$sign(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases}$$
$$x\%y = \frac{x}{\sqrt{1+y^2}}$$

(b) Protected Functions.



(c) Derivation Tree.

Figure 3: Grammar definition (a), mathematical functions (b), and a randomly generated derivation tree (c).

An example of a randomly generated derivation tree is displayed in Figure 3c. The non-terminal symbol $\langle e \rangle$ is expanded using the production rules of the grammar. Derivations continue until all non-terminal symbols have been replaced by terminal symbols. The executable scheduler is given by concatenating leaf nodes:

$$scheduler^{GP}_{u,c}\left(\overrightarrow{F}_{c,u}\right) = \left(F10_{c,u}\%F4_{c,u}\right) \% \left(F4_{c,u} - F1_{c,u}\right).$$

A single GP run proceeds as follows:

(1) A population of 1000 randomly generated derivation trees (individuals) is initialized using the ramped half and half method [16]. The maximum initial derivation tree depth is set to 20.
(2) Individuals are evaluated in simulation. The training and validation sets contain link quality reports from 30 and 10 time slots respectively. Schedules are first generated by executing Algorithm 1, and hence Shannon's formula (Equation 4) yields the received downlink rates. The fitness is given by computing the average of Equation 5 over all time slots.
(3) Parents are selected from the current population using tournament selection with tournament size 5. Individuals are selected based on their training fitness.
(4) With a probability of 0.7, each pair of selected parents undergoes subtree crossover. Subtrees rooted at randomly selected non-terminals in both parents are exchanged.
(5) All of the resulting children are mutated using subtree mutation. A randomly selected non-terminal in the derivation tree is replaced by a new random subtree.
(6) The worst 990 individuals in the current population are replaced with the children from subtree crossover and mutation. The remaining 10 'elites' enter the next generation unchanged.
(7) The individual with highest fitness on the validation set is returned after 500 generations. Performing model selection on the validation set helps to prevent overfitting.

The implementation of GP that was developed by the authors in [8] was instrumented to evolve schedulers.

3.4 Multilayer Perceptron

GP searches a space of variable-sized tree structures. By contrast, the multilayer perceptron (MLP) neural network that is defined by Equations 6–8 has a fixed topology. The MLP consists of an input layer i with 14 units (one for each feature), two hidden layers h_1 and h_2 with 25 units, and an output layer o with one unit. The $tanh(\cdot)$ non-linearity is the hyperbolic tangent. Constraining the topology may simplify the optimization task. Now evolution can focus on optimizing the network parameters, without also having to search for an optimal topology.

$$a_{h_1} = tanh\left(W_{h_1,i} \cdot \overrightarrow{F}_{c,u} + b_{h_1}\right) \quad (6)$$

$$a_{h_2} = tanh\left(W_{h_2,h_1} \cdot a_{h_1} + b_{h_2}\right) \quad (7)$$

$$scheduler^{MLP}_{c,u}\left(\overrightarrow{F}_{c,u}\right) = W_{o,h_2} \cdot a_{h_2} + b_o \quad (8)$$

The 1051 weights W and biases b are optimized using a Genetic Algorithm (GA). The GA and GP search loops are identical except for the initialization, crossover, and mutation steps. An individual is initialized as a vector of 1051 elements. Each element is sampled from the normal distribution $\mathcal{N}(0, 1)$. An individual is parsed during fitness evaluation to yield the weights and biases of the MLP. All pairs of selected parents undergo uniform crossover: elements are exchanged between both parents with a probability of 0.5 per element. Finally, all of the resulting children undergo mutation. A random variable sampled from the normal distribution $\mathcal{N}(0, 0.01)$ is added to elements, with a probability of 0.01 per element.

We benchmark GP against a MLP because deep learning has been recognized as a promising paradigm for network optimization [18]. Unlike many alternative machine learning techniques, both GP and deep learning infer suitable feature representations from data.

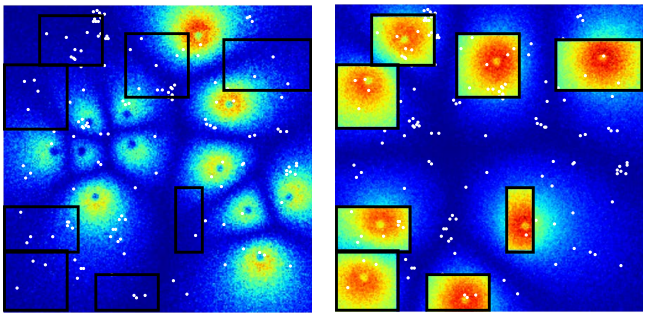
3.5 Benchmark

The benchmark was proposed by Hartung and Buddhikot in [9]. The authors note that link allocation is an NP-complete problem. They present a greedy hill climbing heuristic. Initially, all links between cells and users are assumed to be active. A single iteration of hill climbing involves the following steps. All of the links are visited in a random order. A link is either added or removed if doing so increases fitness (the fitness of a solution is computed using Equation 5). Cells transmit data to users on whatever links remain active after a set number of iterations. A cell allocates an equal share of the spectrum to all connected users.

Updated schedules are required every few milliseconds in a real network. Therefore, it is only feasible to execute one iteration of hill climbing in online operation. In offline mode, the steps are iterated until no further improvements are possible.

4 EXPERIMENTAL DESIGN

The research questions posed in Section 1 were addressed using an experimental methodology based on simulation. The simulation environment and experiments are described in this section.



(a) LTE cells. (b) WiFi cells.

Figure 4: LTE and WiFi cells transmit data to users (white dots) in an enterprise environment. Black lines indicate the exterior walls of buildings. Signal qualities are lowest at cell edges.

4.1 Simulation Environment

Figure 4 visualizes the distribution of 12 LTE and 8 WiFi cells that were simulated in an enterprise environment spanning 500 $\left[\text{m}^2\right]$. LTE cells were located in between buildings, and one WiFi cell was placed inside each building. The LTE and WiFi cells transmitted at fixed powers of 35 [dBm] and 20 [dBm] respectively. The colorbar in Figure 4 shows how the signal quality drops sharply at cell edges.

Brief time slots of activity lasting several milliseconds were simulated. In each time slot, between 100 and 300 users (white dots) were distributed onto the map. A user was either placed into one of 20 hotspots, or else at a randomly chosen location. Hotspots materialize in regions of high traffic density such as transport hubs. Users were not permitted within 10 [m] of the boundary, where interference is artificially lower due to the finite region that is simulated.

Training, validation, and test sets were formed in two steps. Firstly, link gains $(gain_{c,u})$ were computed using the path loss model described in [4]. Secondly, the link qualities $(quality_{c,u})$ were determined.

Step 1: The gain on the link from cell c to user u was modelled as the sum of four separate terms (gains are summed in ratio not decibel scale):

$$gain_{c,u} = gain_{c}^{antennae} + gain_{c,u}^{building} + gain_{c,u}^{distance} + gain_{c,u}^{fading}.$$

Each term models the losses that a signal from c undergoes before it reaches u:

(1) $gain_{c}^{antennae}$ – is the signal loss incurred by focusing the beam at c.
(2) $gain_{c,u}^{building}$ – signals are attenuated by 20 [dB] if they pass through buildings on their path to u (black lines in Figure 4).
(3) $gain_{c,u}^{distance}$ – the loss increases with the distance between c and u.
(4) $gain_{c,u}^{fading}$ – shadow fading occurs when reflected copies of the same signal destructively interfere at u's location.

Step 2: The link qualities $(quality_{c,u})$ were computed using Equation 1 (received signals), Equation 2 (signal to interference and noise ratios), and Equation 3 (link qualities). Schedulers map the reported link qualities to an optimized schedule for each cell. The link quality data were arranged into matrices (see for example Table 2a). A total of 70 such matrices were saved by redistributing users in 70 different time slots. Each link quality matrix had dimensions $\#cells \times |\mathcal{U}_t|$, where $\mathcal{U}_t$ denotes the set of users active in time slot t. A training set was formed with 30 of these matrices, another 10 were reserved as a validation set for model selection, and the remaining 30 constituted a test set for model evaluation.

4.2 Experiments

Downlink rates are lower at cell edges due to increased interference from neighbouring cells. Thirty runs of the GP algorithm from Section 3.3 were carried out. MLPs were trained by executing thirty runs of the GA, as outlined in Section 3.4. The best evolved schedulers were compared on unseen test cases. In addition, the evolved schedulers were compared with a baseline and benchmark heuristic. The baseline permitted each user to receive data over whatever link offered the best signal quality. The benchmark hill climbing heuristic was described in Section 3.5.

It is hypothesized that GP constructs tailored schedulers. This hypothesis was tested by simulating two different deployment scenarios. A different scheduler was evolved for each scenario. The first scenario was described in Section 4.1. Cells and buildings were placed at different locations in the second scenario. Table 4 summarizes how the simulation parameters differed.

	Scenario 1	Scenario 2
Number of LTE Cells	12	8
Number of WiFi Cells	8	12
LTE spectrum	20 [MHz]	15 [MHz]
WiFi spectrum	22 [MHz]	22 [MHz]
LTE power	35 [dBm]	35 [dBm]
WiFi power	20 [dBm]	14 [dBm]
Traffic Density / Time Slot	100 – 300 users	300 – 500 users

Table 4: Two different deployment scenarios are simulated. A different scheduler is evolved for each scenario using GP.

5 RESULTS AND DISCUSSION

The training performance of both evolutionary techniques is analyzed in this section. Generalization performance on unseen test cases is then assessed. Grammar-based Genetic Programming (GP) and the multi-layer perceptrons (MLPs) are compared to a baseline and benchmark algorithm on test data. Several properties of GP are explored including its ability to design tailored schedulers, balance fairness tradeoffs, and discover sensible strategies for allocating spectrum.

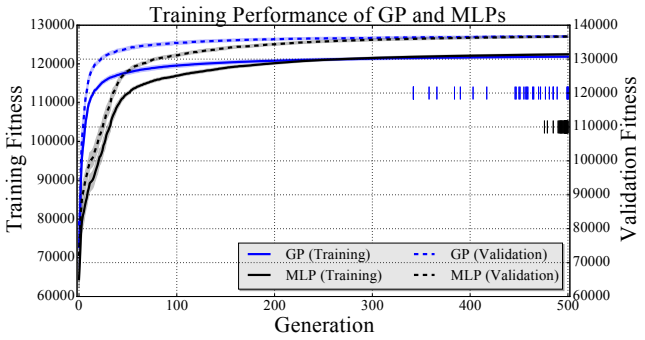


Figure 5: The average training and validation fitness for the 30 GP and MLP runs is plotted over generations. Shaded 95% confidence intervals enclose the average fitness. Vertical strikes indicate when the best scheduler is discovered.

6 TRAINING

Schedulers were evolved for deployment scenario 1 (see Figure 4 and Table 4) using the evolutionary techniques described in Section 3. The convergence of GP, and the Genetic Algorithm (GA) for training MLPs, is visualized in Figure 5. The average best-of-generation fitness is plotted over 500 generations. Training (solid lines) and validation (dashed lines) fitness is given by the average of Equation 5 computed over the 30 training cases and 10 validation cases respectively. A single case is a matrix of link quality reports like that displayed in Table 2a. The best-of-run scheduler is selected based on validation fitness. Performing model selection on the validation set helps ensure that schedulers generalize well to unseen test cases.

MLPs converge to a slightly higher average training fitness. However, both approaches have identical performance on the validation set (the 95% confidence intervals overlap). There is no evidence of overfitting because the validation fitness is monotonically increasing. GP discovers high-performance schedulers in fewer generations than the GA. However, the GA catches up to GP at around generation 300. Vertical strikes reveal that the best scheduler is discovered towards the end of each run.

7 TESTING

The performance of various link allocation techniques on test data is summarized in Table 5. The baseline allows each user to receive data on whatever link provides the highest signal quality. The benchmark (BM) is implemented in online and offline mode as described in Section 3.5. The benchmark is compared with the best scheduler from all 30 GP runs, and the best MLP. Finally, the GA for training MLPs is co-opted to directly optimize schedules offline. A GA is much too slow for online optimization, but it provides an upper bound on the performance that can be expected from online algorithms.

	Fitness	5^{th} Percentile [Mbps]
GA (offline)	138607 ± 28430	7.51 ± 2.61
GP	125639 ± 26974	5.24 ± 1.77
MLP	125123 ± 26677	5.32 ± 1.78
BM (offline)	124387 ± 25195	5.22 ± 1.77
BM (online)	110631 ± 21163	4.00 ± 1.38
Baseline	97534 ± 18134	2.80 ± 0.94

Table 5: Performance on test data. The average of Equation 5 over 30 test cases is tabulated in the first column. The average 5^{th} percentile of downlink rates in [Mbps] **is indicated in the second column.**

The first column of Table 5 displays the average fitness (Equation 5) over 30 test cases. Paired t-tests imply that the best GP and MLP schedulers achieve a statistically significantly higher fitness than the benchmark in online mode. The differences between GP and the MLP are significant at a confidence level of $\alpha = 0.05$. All of the schemes that exploit multiple links are significantly better than the baseline at $\alpha = 0.05$, but they are significantly worse than the highly optimized schedules produced by the GA.

Operators typically strive to increase the downlink rates for poorly performing users at cell edges. A commonly quoted indicator of cell edge throughput is the 5^{th} percentile of downlink rates. The second column of Table 5 displays the average 5^{th} percentile rates over 30 test cases. All of the techniques for intelligently allocating spectrum on multiple links achieve higher cell edge throughput than the single-link baseline.

The cumulative distribution function (CDF) plots in Figure 6 compare the distributions of downlink rates for all users in the test set. The percentage change versus baseline is plotted for each percentile. For instance, the CDF for GP is generated as follows. Downlink rates are computed for all users in the test set. Let $\overrightarrow{rate}^{GP}$ and $\overrightarrow{rate}^{baseline}$ denote vectors of the received downlink rates under optimized schedules (GP) and the baseline. Furthermore, let $\mathcal{P}_p(\overrightarrow{x})$ return the p^{th} percentile of all the elements in vector $\overrightarrow{x}$. The CDF

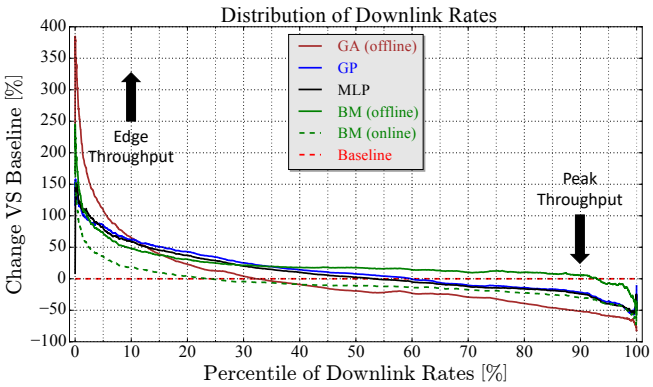


Figure 6: The distribution of downlink rates under various schemes for link allocation are compared relative to a single-link baseline technique. The goal is to increase cell edge throughput versus the baseline. An inherent tradeoff between cell edge and peak throughput is evident.

plot for GP is given by evaluating the following expression for all percentiles $p \in [0, 100]$:

$$CDF_p^{GP} = \frac{\mathcal{P}_p\left(\overrightarrow{rate}^{GP}\right) - \mathcal{P}_p\left(\overrightarrow{rate}^{baseline}\right)}{\mathcal{P}_p\left(\overrightarrow{rate}^{baseline}\right)} \times 100\%. \quad (9)$$

Equation 9 gives the percentage change in the p^{th} percentile of downlink rates achieved by GP versus the baseline.

The left hand side region of Figure 6 corresponds to 'cell edge' users with the lowest downlink rates. The evolutionary techniques (blue and black lines) boost far cell edge throughput by over 150% compared to baseline (dashed red line). By contrast, the benchmark (dashed green line) realizes lower cell edge gains. Gains of up to 380% are achieved when a GA is used to compute idealized schedule offline. However, the GA is too slow for online optimization in real time. The gap between the GA and GP reveals scope for achieving better results in future work. The gap between the GA and offline BM suggests that link allocation is a non-trivial problem, for which greedy search is highly suboptimal.

8 EVOLVING TAILORED SCHEDULERS

A manually designed scheduler may become obsolete due to the deployment of additional cells or a new technology in the network. The main advantage of GP is that updated schedulers can be evolved when required. GP leverages training data to infer a specialized strategy. The degree to which GP generates tailored schedulers was assessed by simulating two different deployment scenarios D1 and D2. Two schedulers were then obtained as follows.

(1) A scheduler S1 was evolved using training data generated in D1.
(2) A different scheduler S2 was evolved using training data from D2.

The CDF plot in Figure 7a confirms that S1 works best in D1 (the scenario for which it was evolved). Similarly, Figure 7b confirms

that S2 works best in D2. As before, the goal is to maximize cell edge throughput.

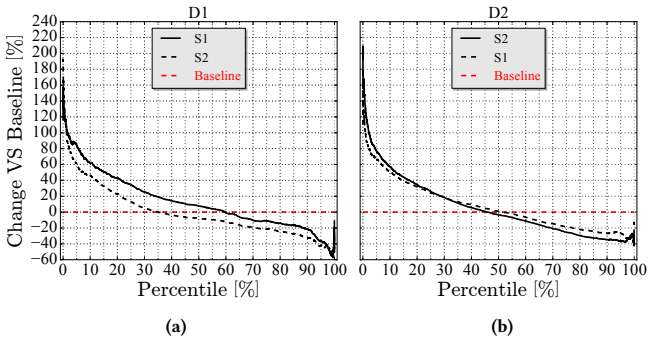


Figure 7: GP generates specialized schedulers that are adapted to their deployment context. Downlink rates are computed for users in 30 time slots.

9 BEHAVIOURAL ANALYSIS

Figure 8 reveals the basic strategy that is discovered by GP for allocating spectrum. Users were split into ten groups based on their aggregate link quality. Cell edge users with the lowest aggregate link quality were placed in group 1. The scheduler tends to award more overall spectrum (red line) to these cell edges users. The users in group 10 have the highest aggregate link quality because they are located close to cell centres. Fairness is achieved by allocating this group much less spectrum on the available links.

Most of the users in groups 1–7 are located outside buildings, where they have more direct access to LTE cells. The scheduler allocates this group the majority of the LTE spectrum. By contrast, those users in groups 8–10 are primarily inside buildings. Hence, they receive most of the WiFi spectrum, since WiFi cells are placed inside the buildings (see Figure 4b).

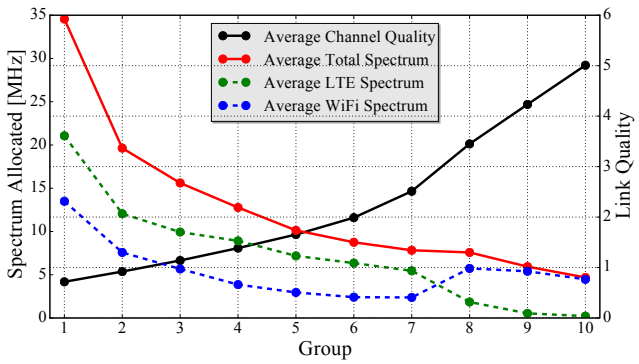


Figure 8: Fairness is achieved by allocating more spectrum to the users with lower aggregate link quality. Users in groups 8, 9, and 10 are allocated more WiFi spectrum because they are primarily inside buildings.

10 CONCLUSIONS

The capacity of a wireless network is significantly increased by transmitting data over multiple links. Schedulers for intelligently allocating spectrum on the available links can be automatically constructed using evolutionary algorithms. Two different techniques were compared in this paper. GP learns a functional mapping from statistical features to optimized schedules by exploring a space of tree structures. A fixed-topology MLP achieves comparable performance to GP, when its weights are optimized using a GA. The best evolved schedulers outperform a greedy hill climbing benchmark, and a baseline which utilizes only one link per user.

The purpose of this paper was to assess the suitability of evolutionary algorithms as a tool for network control. As such, a simplified model of a real wireless network was simulated. In future work, schedulers could be evolved using measurement data from a real network. The task of minimizing latency could be also addressed in future work. Reduced latency will be critical in 5G deployments because it will enable technologies like self-driving cars and virtual reality.

REFERENCES

[1] Anthony Brabazon, Michael O'Neill, and Seán McGarraghy. 2015. *Natural computing algorithms.* Springer.
[2] Jürgen Branke and Hartmut Schmeck. 2003. Designing Evolutionary Algorithms for Dynamic Optimization Problems. In *Advances in evolutionary computing.* Springer, 239–262.
[3] Woon Hau Chin, Zhong Fan, and Russell Haines. 2014. Emerging technologies and research challenges for 5G wireless networks. *IEEE Wireless Communications* 21, 2 (2014), 106–112.
[4] Holger Claussen and Lester Ho. 2012. Multi-carrier cell structures with angular offset. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on.* IEEE, 1179–1184.
[5] Goldberg David. 1989. Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley Longman, Inc.
[6] Supratim Deb, Pantelis Monogioudis, Jerzy Miernik, and James P Seymour. 2014. Algorithms for enhanced inter-cell interference coordination (eICIC) in LTE HetNets. *IEEE/ACM Transactions on Networking* 22, 1 (2014), 137–150.
[7] Michael Fenton, David Lynch, Stepan Kucera, Holger Claussen, and Michael OfiNeill. 2017. Multilayer optimization of heterogeneous networks using grammatical genetic programming. *IEEE transactions on cybernetics* 47, 9 (2017), 2938–2950.
[8] Michael Fenton, James McDermott, David Fagan, Stefan Forstenlechner, Erik Hemberg, and Michael O'Neill. 2017. PonyGE2: Grammatical evolution in python. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion.* ACM, 1194–1201.
[9] Lance Hartung and M Milind. 2015. Policy Driven Multi-band Spectrum Aggregation for Ultra-broadband Wireless Networks. In *Dynamic Spectrum Access Networks (DySPAN), 2015 IEEE International Symposium on.* IEEE, 82–93.
[10] David Lynch, Michael Fenton, David Fagan, Stepan Kucera, Holger Claussen, and Michael O'Neill. 2019. Automated Self-Optimization in Heterogeneous Wireless Communications Networks. IEEE Transactions in Network, in press.
[11] Verne H Mac Donald. 1979. Advanced mobile phone service: The cellular concept. *The bell system technical Journal* 58, 1 (1979), 15–41.
[12] Robert I Mckay, Nguyen Xuan Hoai, Peter Alexander Whigham, Yin Shan, and Michael Ofineill. 2010. Grammar-based genetic programming: a survey. *Genetic Programming and Evolvable Machines* 11, 3-4 (2010), 365–396.
[13] Andreas F Molisch. 2012. *Wireless communications.* Vol. 34. John Wiley & Sons.
[14] Ronald Walter Morrison. 2002. *Designing Evolutionary Algorithms for Dynamic Environments.* George Mason University.
[15] Bahar Partov and Douglas J Leith. 2017. Utility fair rate allocation in LTE/802.11 networks. *IEEE/ACM Transactions on Networking* 25, 2 (2017), 1076–1088.
[16] Riccardo Poli, William B Langdon, Nicholas F McPhee, and John R Koza. 2008. *A field guide to genetic programming.* Lulu. com.
[17] Soheil Rostami, Kamran Arshad, and Predrag Rapajic. 2015. A joint resource allocation and link adaptation algorithm with carrier aggregation for 5G LTE-Advanced network. In *Telecommunications (ICT), 2015 22nd International Conference on.* IEEE, 102–106.
[18] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. 2019. Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys & Tutorials* (2019).