A Symbolic Regression Approach To Manage Femtocell Coverage Using Grammatical Genetic Programming

Erik Hemberg
Complex & Adaptive Systems Laboratory
School of Computer Science & Informatics
University College Dublin
erik.hemberg@ucd.ie
Lester Ho
Bell Laboratories
Alcatel-Lucent
Dublin
lester.ho@alcatel-lucent.com

Michael O'Neill
Complex & Adaptive Systems Laboratory
School of Computer Science & Informatics
University College Dublin
m.oneill@ucd.ie
Holger Claussen
Bell Laboratories
Alcatel-Lucent
Dublin
holger.claussen@alcatel-lucent.com

February 2, 2012

Abstract

We present a novel application of Grammatical Evolution to the real-world application of femtocell coverage. A symbolic regression approach is adopted in which we wish to uncover an expression to automatically manage the power settings of individual femtocells in a larger femtocell group to optimise the coverage of the network under time varying load. The generation of symbolic expressions is important as it facilitates the analysis of the evolved solutions. Given the multi-objective nature of the problem we hybridise Grammatical Evolution with NSGA-

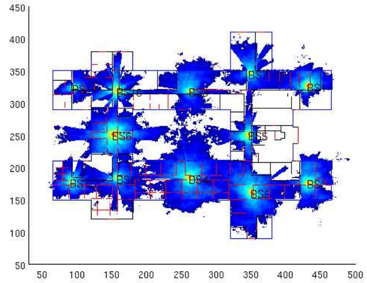


Figure 1: Coverage of a femtocell setup for an office environment with 12 cells, the color shows the pilot channel power in dBm. The base stations are in their initial state with power -40dbm.

II connected to tabu search. The best evolved solutions have superior power consumption characteristics than a fixed coverage femtocell deployment.

D.1.2 Programming Techniques Automatic Programming
Algorithms
Femtocell, symbolic regression, wireless networks, grammatical evolution

1 Introduction

In telecommunication networks, femtocells are low power, low-cost, user- deployed cellular base stations with a typical coverage range of tens of meters [4]. In order to minimize operational expenses, femtocells have considerable self-configuration and self-optimization capability to enable plug-and-play deployment. These capabilities are implemented using algorithms that are designed to automatically change certain network configuration parameters in response to any changes in the environment it is operating in.

Furthermore, to maintain scalability when used in large networks, these algorithms should work in a distributed manner whenever possible, using only local information but achieving good global performance.

Designing these highly distributed algorithms can be difficult, particularly if the network environment varies significantly. Moreover, for femtocell deployments in enterprise environments a group of femtocells are deployed where the individual cells need to work together to jointly provide continuous coverage in a large building or campus. Figure 1 shows the coverage of a femtocell setup for an offices environment with 12 cells, with the colored areas indicating areas of femtocell coverage. The base stations are in their initial state with a power of -40dBm. When femtocell users enter any gaps in the coverage between the femtocells, mobility procedures (handovers or cell re-selections) to the underlying macrocell is performed, or a loss of service occurs if macrocell coverage is unavailable.

In this study we expand on previous work by Ho et al. [9] which adopted Genetic Programming(GP)[15] to optimize femtocell coverage. The evolved solution trees were comprised of a set of conditional statements similar to a decision tree architecture. In this paper, a grammar-based GP setup is adopted which allows us to manipulate the structure of the solutions and incorporate domain knowledge. More explicitly, we aim to widen our search space by investigating if it is possible to fine tune the pilot power increments using equations, instead of conditional programs for increasing or decreasing the pilot power by a preset step (0.5dB). With this approach we also aim to generate solutions that are easier to analyze.

In addition to the novel application of GE to the dynamic, real-world, femtocell coverage optimisation problem we examine the utility of a number of extensions to the basic GE approach, including the use of NSGA-II with tabu search and the use of more refined forms of mutation.

The rest of the paper is structured as follows, in Section 2 previous work and GE is shown. The setup of the algorithm and the experiments are explained in Section 3, and the results are shown in Section 4. Finally, conclusions and future work are discussed in Section 5.

2 Background

This section describes the femtocell coverage problem, previous work and a description of Grammatical Evolution.

2.1 Femtocell Coverage problem

We consider an intended area of coverage, such as an enterprise environment, where a group of femtocells is deployed to jointly provide end-user services. The problem addresses distributed coverage optimization by adjusting the coverage of the femtocells to satisfy the mobility, load and power objectives:

Mobility - To minimize femtocell mobility events within the femtocell group's intended area of coverage.

Load - To balance the load amongst the femtocells in the group to prevent overloading or under-utilization.

Power - To minimize the pilot channel transmission power. This objective is to prevent, whenever possible, leakage of the femtocell group's coverage outside its intended area of coverage.

This is a multi-objective optimization problem with conflicting objectives. For example, increasing the coverage of a femtocell would reduce the amount of mobility events, but doing so may increase the load of the femtocell, and increase its pilot channel power. Thus it is necessary to balance the requirements of all three objectives according to their importance.

2.2 Previous Work

There have been previous studies of telecommunications and EC [1], but only one specifically covering coverage algorithms and GP [9]. Moreover, there have been no studies of GE and femtocell coverage algorithms. First, regarding cellular coverage optimization, most related work in the literature deals with centralized computation methods [9], e.g. the calculation of parameters such as the number and locations of base stations (BS), pilot channel transmit powers or antenna configurations using a central server running an optimization algorithm. Many studies also focus on determining the optimal BS numbers or placements to achieve the operator's quality of service or coverage target. This approach is not always practical because network design is restricted by BS placements. Instead, optimization of the configuration of cellular networks where the locations of the base stations have been fixed can be more realistic.

One example of self-configuration and self-optimization capability in femtocells deployments is coverage optimization. The aim of coverage optimization in residential femtocell deployments is to ensure that leakage of coverage by a single femtocell into public spaces is minimized while at the same time maximizing indoor coverage [8, 5]. The methods for this deployment are not applicable to enterprise environments [9].

Previous work with GP and femtocells [9] automatically derived a distributed algorithm to dynamically optimize the coverage of a femtocell group using Genetic Programming. The resulting evolved algorithm showed the ability to optimize the coverage well, and was able to offer increased overall network capacity compared with a fixed coverage femtocell deployment. The functions and terminal set for GP consisted of conditions checking if the load, overlap and probability of users entering a gap was over a predefined threshold, as well as combining the branches of the conditionals. The terminals were increase power, decrease power or do nothing. Here, we extend the GP approach using grammatical GP, and using an equation form. Thus, the solutions that are evolved in our study are less constrained than in Ho et al. [9] and different fitness functions are used.

For work with a grammar and GP, Lewis et al. [11] enhanced IEEE802.11 DCF. They designed MAC layer algorithms using GP by evolving algorithms instead of optimizing values and tuning parameters,and a wider behavior space was searched. The variation of contention window sizes was explored, the results outperformed standard 802.11 behavior on a variable sized network under standard load, and the throughput performance is comparable to the best aspects of the protocol. For their studies they use a grammar to embed domain knowledge in the algorithms they evolve. One way to extend this work is to use a formal grammar specification, e.g. Backus-Naur Form(BNF), as used in grammar based approaches of GE.

GE has been used by O'Neill and Ryan [13] to automatically evolve caching algorithms, where simple caching algorithm solutions were found. Hu and Goodman [10] used GP for wireless access point configuration, the results improved when they post-processed their solutions to find the minimum spanning tree. Yasuda and Sato [16] used linear GP and a pruning operator on their solutions for wireless LAN access point configuration to gain improved performance and run time speed-up, which shows that pruning can be efficient.

This section concludes that grammatical GP approaches, e.g. GE, are viable for

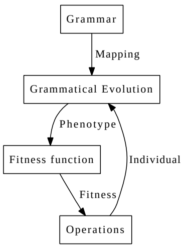


Figure 2: The GE algorithm

telecommunication network studies. It also reveals gaps in the generation of coverage algorithms for femtocells.

2.3 Grammatical Evolution

Grammatical Evolution (GE) [14, 7] is a grammar-based form of GP [12]. It is inspired by representation in molecular biology and combines this with formal grammars. The GE system is flexible and allows the use of alternative search strategies, whether evolutionary, deterministic or of some other approach. This system also includes the ability to bias the search by changing the grammar used. Since a grammar is used to describe the structures that are generated by GE, editing the grammar modifies the output structures. This constraining power is one of GE's main features. The genotype-phenotype, i.e. input-output mapping means that GE allows search operators to be performed on any representation in the algorithm, e.g. on the genotype, as well as on partially generated phenotypes, and on the completely generated derivation trees or phenotypes. Figure 2 shows an overview of the GE algorithm and its components.

The components of GE are the mapping of the genotype to the phenotype via a grammar, the evaluation of the phenotype in the fitness function and the application of evolutionary operators to the individuals.

In GE, the grammar mapping uses a Context Free Grammar, which is a four tuple $G = (N, \Sigma, R, S)$, where:

- N is a finite non-empty set of non-terminal symbols
- Σ is a finite non-empty set of terminal symbols and $N \cap \Sigma = \emptyset$, the empty set
- R is a finite set of production rules of the form $R : N \leftarrow V : A \leftarrow \alpha$ or (A, α) where $A \in N$ and $\alpha \in V$. V is the set of all strings constructed from $N \cup \Sigma$ and $R \subseteq N \times V, R = \emptyset$
- S is the start symbol, $S \in N$.

The genotype is used to map the start symbol into a sentence, by the BNF-grammar. The mapping is done by reading input(codons) from the genotype to generate a corresponding integer value, from which an appropriate production rule is selected by using the mapping function. The production choices is determined by the current codon's

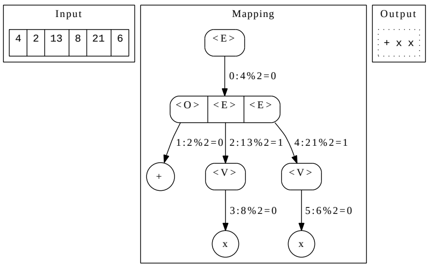


Figure 3: Example of GE genotype-to-phenotype mapping. The derivation order codon value and production choice are shown to the right of the arrows, e.g. from the start symbol $0 : 4\%2 = 0$.

integer value mod the number of production choices of current rule. The genotype is read from left to right, and the codon to be read is shifted every time the current rule has more than one production. The derivation sequence is also expanded left to right (depth-first). An example of a genotype generating a function using a small equation grammar

```
<E> ::= <O><E><E> | <V>
<O> ::= + | -
<V> ::= x | 1
```

is shown in Figure 3.

The basic mapping function of GE has been described in this section. In Section 3 we introduce the setup of the experiments.

3 Setup

This section describes the femtocell simulation, the GE algorithm, the grammar used in the GE algorithm and the fitness function.

3.1 GE Algorithm

The steps in a single iteration of the GE algorithm used here are:

1. Initialization: The genotype input in the initial solutions is generated, uniformly randomly generated integer sequences
2. Mapping: A BNF form Context Free Grammar is used for the mapping
   (a) Integer to String translation where the grammar maps integer value to a sentential form.
   (b) When the end of the genotype is reached and the output contains non-terminal symbols it wraps and is read from the start again. If this still has non- terminals the individual is reinitialized.

3. Evaluation: The individual solutions are evaluated using the femtocell simulation.

4. Selection: Some individuals from the current population are included in a new population using a tournament selection.

5. Variation operators: Individuals are modified by one point crossover and uniform integer codon mutation.

6. Replacement: A new population is created from the selected population and from the current population

The steps 2-7 are repeated until the maximum number of generations(iterations) is reached.

3.2 Grammar Description

The grammars used are for generating coverage algorithm solutions which are equations, differing from the previous conditional expressions used by Ho et al. [9]. The solutions the grammar generates does not use any predefined threshold parameters. A wide range of functions were used to attmempt to capture different behaviours. Moreover, in order to avoid imaginary numbers, we only use the real valued part of the function values. The argument passed to the trigonometric functions sin, cos, tan is in degrees, hence `sind, cosd, tand`, and the unary minus is `uminus`. The grammar adopted in this study is in MATLAB syntax and is presented below.

```
<CODE> ::= value = <expr_0>;
<expr_0> ::= (<expr><op><expr>) | <preop>
<expr> ::= (<expr><op><expr>) | <var> | <preop>
          | <preop_step> | <preop_monotone>
<op> ::= + | - | * | / | ^
<preop> ::= sind(real(<expr>)) | sind(real(<expr>))
           | cosd(real(<expr>)) | log(real(<expr>))
           | tand(real(<expr>))
<preop_monotone> ::= exp(round(real(<expr>)))
                   | uminus(<expr>)
<preop_step> ::= heaviside(<expr>) | sigmoid(<expr>)
<var> ::= my_power | my_load | my_handover | <cnst>
<cnst> ::= <nr><nr> | <nr> | 0.<nr><nr> | 0.<nr>
<nr> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Many solutions found in earlier exploratory runs were of functions that were monotonic (increasing, decreasing or simply constant). Since these monotonic functions are not capable of responding to all environmental changes we bias slightly towards non-monotonic solutions by using `<expr_0>` and also reduce the probability of using `exp`. In addition, we introduce sigmoid and heavyside functions in order to increase the number of non-monotonic solutions. An example of a solution that will always increase the pilot power based on a constant and the current power and load is `value = (5 + (my power + my load))`

3.3 Fitness Function

The fitness function is used by GE to determine the quality of the generated solutions when applied to the femtocell network. The functions are mobility, load and power, and the duration of the simulation is T, the number of femtocells is N, and x is a vector of femtocells. Statistics of mobility, load and power are collected over a specified update period. These statistics are then used as inputs into the algorithm, and for calculating the fitness. The fitness function is a vector comprised of the fitness for each function, $F = [f_M(\overline{M(h,r)}), f_L(\overline{L(x)}), f_P(\overline{P(x)})]$.

We tried two different fitness functions approaches a weighted and a Pareto optimal front. A weighted fitness function would use the weights to modify the impact of each fitness component. The results from using a weighted fitness function showed a very fast convergence to a local optima, with very low diversity among the solutions in experiment 1. Therefore we used a different approach and used the NSGA-II algorithm to calculate fitness. The NSGA- II creates a Pareto optimal front of non-dominated solutions[6].

3.3.1 Mobility fitness

The fitness function for mobility is based on the number of handovers and relocation of users using the statistics of the mobility events of the femtocells involving femtocell users. During the simulation, the number of update periods where the mobility events (handovers and relocation) between femtocells and macrocells is recorded during the update period. The number of femtocell handovers is h, macrocell handovers is h^M, femtocell relocations is r, and macrocell relocations is r^M. Mobility M is composed of

$$M_b^M(h,r) = \sum_{t=0}^{T}\sum_{i=1}^{N} h_{it}^M + \sum_{t=0}^{T}\sum_{i=1}^{N} r_{it}^M$$

$$M_b(h,r) = M_b^M(h,r) + \sum_{t=0}^{T}\sum_{i=1}^{N} h_{it} + \sum_{t=0}^{T}\sum_{i=1}^{N} r_{it}$$

Mobility is the ratio of update periods where a mobility event occurs to the total number of update periods is calculated. It is maximized when there are no handovers or relocation to the macrocell underlay occur, and is 0 when all femtocell user handovers are to or from macrocells. The average mobility is 1 if there are no handovers or relocation, otherwise it is

$$\overline{M(h,r)} = \begin{cases} M_b^M(h,r)/M_b(h,r) & \text{if } M_b(h,r) > 0 \\ 1 & \text{if } M_b(h,r) = 0 \end{cases}$$

The mobility fitness is calculated as

$$f_M(\overline{M(h,r)}) = e^{-\overline{M(h,r)}}(1 - \overline{M(h,r)}^{1/4})$$

3.3.2 Load Fitness

The fitness for the load is based on the average number of times the load has been greater than a defined maximum load threshold, l_{max}. If the mean cell load during an update period exceeds this threshold, L is equal to one, else it is equal to zero. Cell load $0 \leq x \leq 7$ is the load, $l_{max} = 7$ users in this scenario. This is just below the capacity of the femtocell, as the aim is to prevent the femtocell from operating at its capacity.

$$L(x) = \begin{cases} 1 & \text{if } x > l_{max} \\ 0 & \text{if } x \leq l_{max} \end{cases}$$

Average load is

$$\overline{L(x)} = 1/T \sum_{t=0}^{T} \sum_{i=1}^{N} L(x_{it})$$

The fitness function for load is

$$f_L(\overline{L(x)}) = e^{-\overline{L(x)}}(1 - \overline{L(x)}^{1/4})$$

3.3.3 Power Fitness

Finally, the fitness for power P is the normalized power. Power P, $-50dBm \leq x \leq 11dBm$

$$P(x) = \frac{x - x_{min}}{x_{max} - x_{min}}$$

The average power is

$$\overline{P(x)} = 1/T \sum_{t=0}^{T} \sum_{i=0}^{N} P(x_{it})$$

and the fitness function for power is

$$f_P(\overline{P(x)}) = 1 - \overline{P(x)}$$

3.4 Femtocell Scenario

The femtocell problem addresses distributed coverage optimization by adjusting the coverage of the femtocells to satisfy the mobility, load and power objectives:

Mobility: Minimize the number of mobility procedures between femtocells and macrocells (in both directions) for femtocell users within the building.

Load: Balance the load amongst the femtocells in the group to prevent overloading or under-utilization

Power: Minimize the pilot channel transmission power to prevent leakage of the femtocell group's coverage outside its intended area of coverage.

This is a multi-objective optimization problem with conflicting objectives and it is necessary to balance the requirements of all three objectives.

The simulation scenario used to calculate the fitness of the algorithms is an office environment shown in Figure 1. The building is an office with cubicles, closed meeting rooms, and toilets. The exterior of the building is mainly glass and the interior is mostly light interior walls and cubicle partitions. There are four stairwells at each corner with thick concrete walls. The location of the 12 Femtocells were chosen such that they are spaced fairly evenly apart, but without any cell surveying done. This reflects a plug-and-play deployment where some heuristic has been used in the deployment, i.e. the femtocells are not placed too closely to each other. This is a realistic plug-and-play femtocell deployment, but can be sub-optimal due to the lack of exhaustive cell planning. In the simulation each femtocell has a maximum capacity of 8 voice calls, a macrocell underlay coverage is also assumed. A path loss map is generated for the 450m x 500m area for each femtocell. For shorter distances the path loss (dB) at d (meters) from a BS is modeled as $38.5 + 20log_{10}(d) + PL_{walls}$, with a smooth transition to $28 + 35log_{10}(d) + PL_{walls}$ otherwise. A correlated shadow fading with a standard deviation of 8 dB and spatial correlation of $r(x) = e^{x/20}$ for a distance x (meters) is considered. The assumed transmission losses for the explicit building model are a function of the incident angle, the model is taken from Ho et al. [9].

A user mobility and traffic model with the users moving to predefined way points in the map at a speed of $1ms^{-1}$, spending a some time in a way point before moving to another way point. At the start the users are randomly placed at way points. In total 200 users are modeled, and each user has a voice traffic model which produces 0.2 Erlangs of traffic.

When evaluating an algorithm, the scenario is run to simulate 24 hours of operation time, with the algorithm adjusting the femtocell pilot power after collecting statistics for 30 minutes. The algorithm start time for each femtocell is randomly dithered so that the femtocells to avoid synchronous pilot powers updates, with each femtocell's initial pilot channel power set to -40dBm, the maximum pilot channel power is 11dBm and minimum power is -50dBm. In order to keep the users connected to the femtocell network, femtocell to macrocell handovers are triggered when a user terminal's pilot channel receive power goes below -100dBm.

One evaluation of the Femtocell scenario on a core of an Intel i7 2.93GHz processor takes approximately 10 minutes.

3.5 Modifications to the GE Engine

We replace the standard *integer mutation* which randomly modifies an integer codon value to another random value with nodal mutation proposed by Byrne et al. [2, 3]. The nodal mutation operator has a superior property of locality to the standard GE mutation operator which is applied indiscriminately to any codon value irrespective of it's context within the derivation tree. Nodal mutation is only applied to individuals who have not undergone crossover.

Table 1: Experimental Parameter Settings

Parameter	**Value**
MAX_WRAPS	2
CODON_SIZE	128
POPULATION_SIZE	200
INITIALISATION	Ramped half-and-half
GENERATIONS	50
TOURNAMENT_SIZE	2
CROSSOVER_PROBABILITY	0.5
MUTATION	1 event per individual
PARSIMONY_PRESSURE	True
EXTENDED_NODAL_PROBABILITY	0.5
EXTENDED_NODAL_TRIES	1000

NSGA-II is used to rank the individuals according to domination. The top individuals from each front are used until the population is filled, see Deb et al. [6]. When regenerating individuals the depth is picked from the distribution of depths in the first front. This is both an attempt to restrict bloat and search at depths were good solutions have been found. All evaluated solutions are added to a tabu list. If a solution is already on the tabu list it will also be regenerated.

The evolutionary parameter settings for the GE algorithm are presented in Table 1. Due to the long run time to evaluate each individual algorithm in the femtocell scenario, the number of generations was limited to 50.

4 Results

The equations of the two best evolved solutions (GE6 and GE7) are outlined below, and their corresponding power output behaviour during the femtocell simulation are shown in Figure 4.

GE6: In Figure 4(a), $F = [0.685, 0.562, 0.766]$

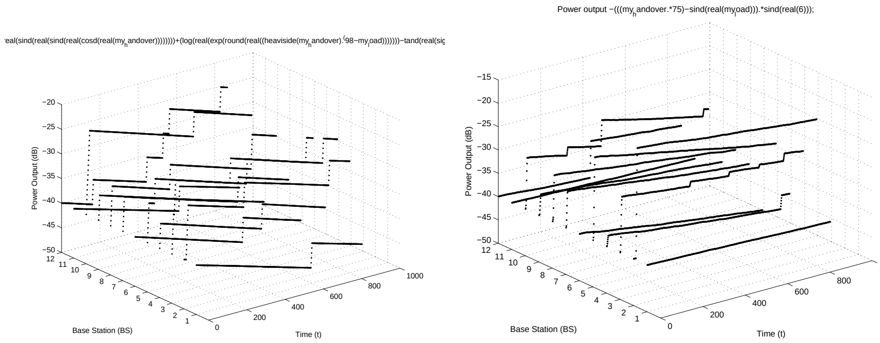
```
tand(real(sind(real(sind(real(cosd(real(
my_handover))))))))+(log(real(exp(round(real(
(heaviside(my_handover)^(98-my_load)))))))
-tand(real(sigmoid(sigmoid(my_load))))));
```

GE7: In Figure 4(b), $F = [0.699, 0.583, 0.741]$

```
((my_handover.*75)-sind(real(my_load)))*
sind(real(6)));
```

This solution only uses mobility and load input data.

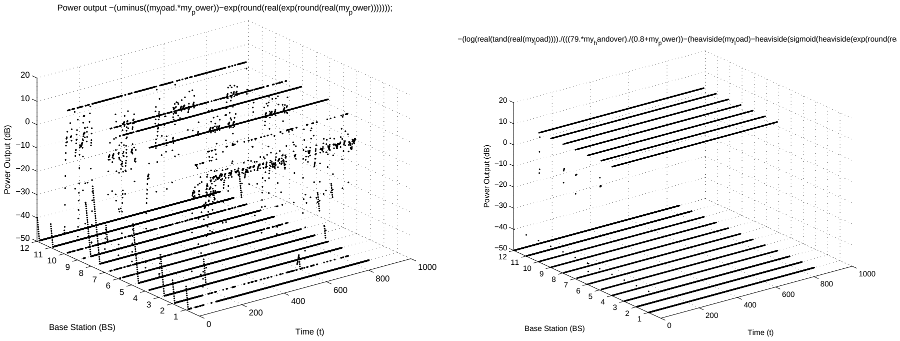
The solutions GE6 and GE7 show a desirable behaviour, where the power is stepwise increased and then slowly decreasing, when compared with a fixed coverage femtocell deployment, or other extreme forms of evolved strategies which simply switch between power on and off.



(a) GE6

(b) GE7

Figure 4: Power output of the Femtocells during simulation of the two best evolved solutions found.



(a) GE4

(b) GE5

Figure 5: Power output of the Femtocells during simulation of two evolved power on/off switching solutions.

Examples of two evolved solutions (GE4 and GE5) which alternate power on and off can be seen in Figure 5, and the resulting expressions are provided below.

GE4: In Figure 5(a), $F = [0.842, 0.450, 0.749]$

```
(log(real(tand(real(my_load))))/(((73*my_handover)/
(0.8+my_power))-(heaviside(my_load)-heaviside(
sigmoid(exp(round(real(my_power))))))));
```

GE5: In Figure 5(b), $F = [0.745, 0.490, 0.770]$

```
(uminus((my_load*my_power))-exp(round(real(exp(
round(real(my_power)))))));
```

When analyzing the correlation between the fitness values there is a positive correlation between power and load, and a negative correlation between mobility and power,

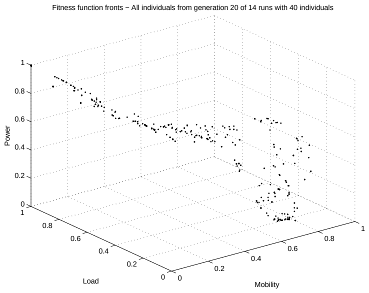


Figure 6: Fitness values for all objectives for all the individuals in the last generation.

as well as mobility and load, see Figure 6. This is to be expected from how the values of the fitness functions are calculated. With higher power it is possible for a cell to have more users and reach max capacity.

Moreover we tried to get an indication of terminal symbol fitness by studying the number of times a terminal had been used in the solutions and the average fitness value of the average of the fitness function values there was no correlation. In Figure 7 we analyse the composition of solutions within a population by focusing on the ratio of production choices for each generation in the first pareto front. The increasing size of the solutions in the first front can be seen from the increase in the ratio of `<var>`.

In summary, the best evolved solutions demonstrate that it is possible to find solutions with good fitness that have non-trivial behaviour. That is, the behaviour is not only switching power on and off or settling in constant states such as a typical fixed coverage deployment.

5 Conclusions & Future Work

We described how to evolve the coverage of femtocells using a symbolic regression approach with GE. In order to maximize the coverage of the femtocells and minimize the power used the algorithm controlling the power settings of the femtocell are evolved. The study contributes to automatic equation generation of coverage optimisation for telecoms networks, and it investigates how GE behaves in a dynamic real-world environment.

The evolved solutions are simpler to understand and read for a human compared to some evolved programs which use conditional statements in Ho et al. [9]. The best evolved solutions are superior on two of the objectives than the current approach which is to use the maximum power setting (11dBm).

We also introduced novel modifications to Grammatical Evolution (GE), such as incorporating NSGA-II and tabu search in the search engine. The use of a non-weighted fitness function allows a more diverse set of solutions to be explored. More points on the Pareto front were found when tabu search was used.

We are currently investigating the use of a hybrid approach where an Evolutionary Strategy is used for a local optimisation of the constants in promising equations. In

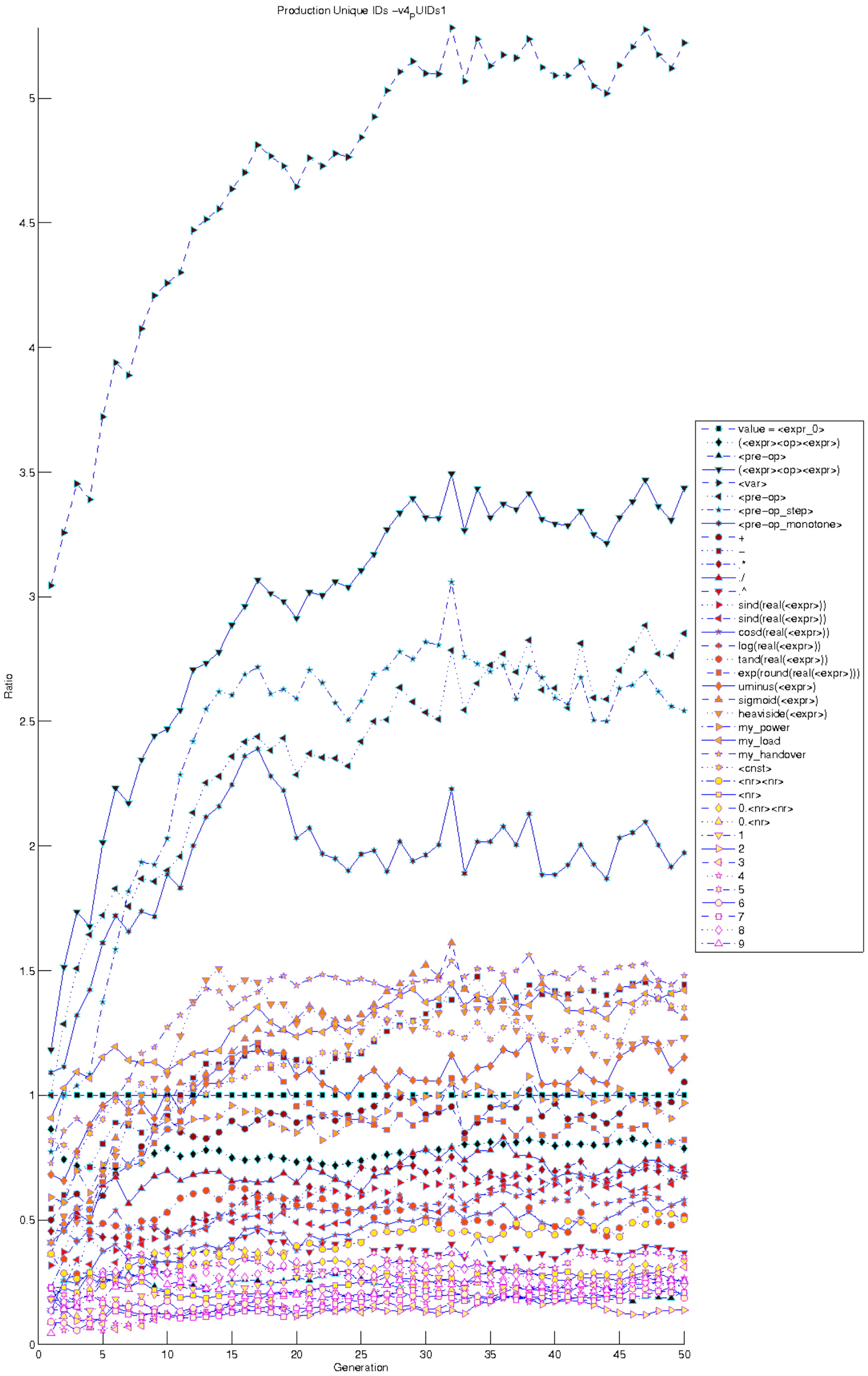


Figure 7: Ratio of production choices used for each generation in the first front

addition, alternative fitness measures are being examined.

Acknowledgments

This research is based upon works supported by the Science Foundation Ireland under Grant No. 08/IN.1/I1868.

References

[1] E. Alba and J.F. Chicano. Evolutionary algorithms in telecommunications. In *Electrotechnical Conference, 2006. MELECON 2006. IEEE Mediterranean*, pages 795–798. IEEE, 2006.

[2] J. Byrne, J. McDermott, E. Galvan-Lopez, and M. O'Neill. Implementing an intuitive mutation operator for interactive evolutionary 3d design. In *IEEE Congress on Evolutionary Computation*, July 2010.

[3] J. Byrne, M.O'Neill, J. McDermott, and A. Brabazon. An analysis of the behaviour of mutation in grammatical evolution. *Genetic Programming*, pages 14–25, 2010.

[4] V. Chandrasekhar, J. Andrews, and A. Gatherer. Femtocell networks: a survey. *Communications Magazine, IEEE*, 46(9):59–67, 2008.

[5] H. Claussen, F. Pivit, and L.T.W. Ho. Self-Optimization of femtocell coverage to minimize the increase in core network mobility signalling. *Bell Labs Technical Journal*, 14(2):155–183, 2009.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[7] Ian Dempsey, Michael O'Neill, and Anthony Brabazon. *Foundations in Grammatical Evolution for Dynamic Environments*, volume 194 of *Studies in Computational Intelligence*. Springer, April 2009.

[8] L.T.W. Ho and H. Claussen. Effects of user-deployed, co-channel femtocells on the call drop probability in a residential scenario. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5. IEEE, 2007.

[9] L.T.W 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*, pages 2132–2136. IEEE, 2010.

[10] J. Hu and E. Goodman. Wireless access point configuration by genetic programming. In *Proceedings IEEE Congress on Evolutionary Computation*, pages 1178–1184, 2004.

[11] T. Lewis, N. Fanning, and G. Clemo. Enhancing IEEE802. 11 DCF using Genetic Programming. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 3, pages 1261–1265. IEEE, 2006.

[12] R.I. Mckay, N.X. Hoai, P.A. Whigham, Y. Shan, and M. O'Neill. Grammar-based Genetic Programming: a survey. *Genetic Programming and Evolvable Machines*, 11(3):365–396, 2010. ISSN 1389-2576.

[13] Michael O'Neill and Conor Ryan. Automatic generation of caching algorithms. In Kaisa Miettinen, Marko M. Mäkelä, Pekka Neittaanmäki, and Jacques Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, pages 127–134, Jyväskylä, Finland, 30 May - 3 June 1999. John Wiley & Sons.

[14] Michael O'Neill and Conor Ryan. *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Kluwer Academic Publishers, Norwell, MA, USA, 2003. ISBN 1402074441.

[15] Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. *A field guide to genetic programming*. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008.

[16] Yuta Yasuda and Yuji Sato. Using genetic programming to improve the performance of wireless LAN access point configuration. In The Long Pham, Hai Khoi Le, and Xuan Hoai Nguyen, editors, *Proceedings of the Third Asian-Pacific workshop on Genetic Programming*, pages 57–68, Military Technical Academy, Hanoi, Vietnam, 2006.