Comparing the Robustness of Grammatical Genetic Programming Solutions for Femtocell Algorithms

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

ABSTRACT

We compare how multiple of training scenarios in the evolutionary search produce different solutions and performance on training and test scenarios. The experiments use grammar based Genetic Programming on the Femtocell problem with one grammar for generating real-values and another grammar for generating discrete values for changing the pilot power. The use of a grammar which produces discrete changes to the pilot power generate better solutions on the training and the test scenarios.

Categories and Subject Descriptors

D.1.2 [**Programming Techniques**]: Automatic Programming

Keywords

Femtocell, symbolic regression, grammatical evolution

1. INTRODUCTION

In Evolutionary Computation over-fitting is an open issue [7]. A generated solution is fitted to training data, thus reducing its general performance on other data to which the solutions might be applied. Additionally, generalizability and robustness have also been studied [6]. A method for evolving robust solutions is needed when evolving solutions that can be deployed in actual consumer products. One such product is the Femtocell, a low power, low-cost, user-deployed cellular base station with a typical coverage range of tens of meters [1].

Previous studies have evolved Femtocell algorithms using Genetic Programming (GP) [4] and Grammar based Genetic Programming [3] for specific scenarios. We use Grammatical Evolution (GE) [2]. Our aim is to evolve more robust solutions, i.e. solutions that work well over a number of different scenarios, which we measure as the average fitness over out-of-sample test scenarios. In the Femtocell case, this is especially important since each fitness evaluation is a simulation that is quite computationally expensive.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

2. EXPERIMENTS & RESULTS

The problem addresses distributed coverage optimisation by adjusting the pilot power of the base stations in order to alter the coverage of the Femtocells and satisfy the mobility, load and leakage objectives: Mobility, to minimise Femtocell mobility events within the Femtocell group's intended area of coverage; a mobility event is a handover from a Femtocell to a Macrocell or vice versa. Load, to balance the load amongst the Femtocells in the group to prevent overloading or under-utilisation. Leakage,to minimise the leakage of the Femtocell group's coverage outside its intended area of coverage.

2.1 Setup

Simulation model The setup is the same as Hemberg et al. [3]. In total 50, 200 and 400 users in *low* (l), *medium* (m) and *high* (h) load scenarios. **Office (O12, O8, O4)**. There are versions with 12, 8 and 4 Femtocells for the different configurations. *Outdoor (Od4)* There are no walls. The coordinates of the 4 BSs are the same as in the office with 4 BS. *Cross (C5)* There are walls and 5 BS. All the way-points and hot-spots are different. The way-points are set to explicitly model the need for load balancing and a different path loss model is used, where signals bounce off the walls. The training scenarios are *Od4, C5, O12* with medium load and the test scenarios are *O8, O4* at low, medium and high load.

Grammars Two different grammars were tested. A conditional grammar that changes the pilot power with discrete values and a conditional equation grammar changing the pilot power with continuous values calculated from generated equations. The search space is very different for the grammars and a difference in performance is to be expected. *Conditional Statement Grammar (CG)*, the thresholds and the size of the increase and decrease of power needs to be predetermined, the change is 1dBm. The values of the thresholds are mobility ($MT = 0$), leakage ($LeT = 0$) and load ($LT = 7$). *Symbolic Regression and Conditional Statement Grammar (SRCG)* combines the grammars in Fig. 1 and the grammar from [3].

Statistics of mobility, load and leakage are collected over a specified update period. These statistics are then used as inputs into the algorithm, and for calculating the fitness. **Mobility fitness** is based on the number of handovers and relocation of users using the Femtocells' statistics of the mobility events involving Femtocell users. During the simulation in a number of update periods, the mobility events between Femtocells and Macrocells are recorded

```
<CODE> ::= if gt(h, MT) if gt(l, LT) if gt(mrt, LeT)
           <fun> else <fun> else if gt(mrt, LeT)
           <fun> else <fun> else if gt(l, LT) if gt(mrt, LeT)
           <fun> else <fun> else if gt(mrt, LeT)
           <fun> else <fun>
<fun> ::= <terminal><fun> | <terminal>
<terminal> ::= p = increase_power(p); | p = decrease_power(p);
             | p = do_nothing(p);

Symbolic Regression and Conditional Statement Grammar (SRCG)
<function> ::= my_power = <expr_0>;
<expr_0> ::= (<expr><op><expr>) | <pre-op>
<expr> ::= (<expr><op><expr>) | <var> | <var> | <var> | <var>
         | <pre-op> | <pre-op_step> | <pre-op_monotone>
```

Figure 1: Symbolic Regression and Conditional Statement Grammar (SRCG) and Conditional Grammar (CG). Only the differences in CG and SRCG are shown

Table 1: Fitness on test data for the non-extreme solutions on the first front.

Version	Avg Fit	Std	Med	Min	Max
CG STS	0.467	0.033	0.471	0.374	0.521
SRCG STS	0.301	0.077	0.316	0.130	0.458

for each period. Mobility is the ratio of update periods where a mobility event occurs divided by the total number of update periods. It is maximised when there are no handovers or relocation to the Macrocell underlay, 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 **Load fitness** has the objective that the Femtocells should serve enough users. The fitness for the load is based on the ratio of average number of times the load has been greater than a defined maximum load threshold, LT, and the total load, including the Macrocell. If the mean cell load during an update period exceeds this threshold, load fitness is equal to one, else it is equal to zero. **Leakage fitness** is the number of outside users trying to use the Femtocell is measured by leakage. Leakage increases the number of unwanted users captured, which increases the signalling load to the core network. The leakage is the ratio of blocked calls and maximum number of Macrocell users.

GE Setup To find extreme solutions and those which have uniform fitness components we use the index from Jain et al. [5], where a score of one is uniform and zero is non-uniform, $\phi(x) = \frac{(\sum_{i=0}^{n} x_i)^2}{n \sum_{i=0}^{n} x_i^2}$. We penalise the fitness function, $f(x)$ to get $f'(x)$ by multiplying it with its score, $h(x)$, where $h(x) = 1 - \phi \circ f(x)$ and $f'(x) = e^{-h(x)}(1 - h(x)^{1/4})$. It leaves the extreme solutions unpenalised when one of the objectives is zero, $h(x) = 1$ if $x = 0$. The settings for the GE algorithm are as in [3] except for population size of 20, generations are 10, and 28 runs.

2.2 Results

The values from the training fitness of the population are the average fitness of non-extreme solutions, shown in Fig. 2. The fitness is the average of all objectives. The difference between the methods are significant as can be seen by the non-overlapping error-bars. The graphs show that the representation in CG finds good solutions very fast in comparison to SRCG.

The results from the test scenarios are shown in Tab. 1. The non-extreme solutions from the first front for each run are evaluated on the test scenarios. There is a significant difference in fitness according to the non-parametric Wilcoxon rank sum test for equal medians at a 0.05-level for all values.

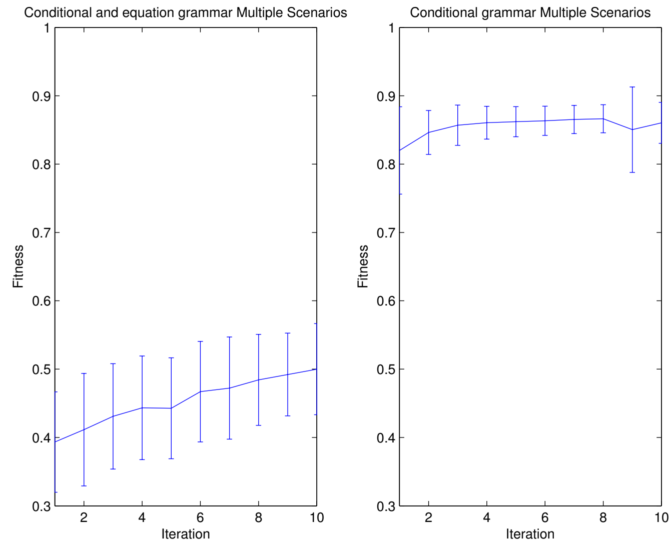

Figure 2: Femtocell average fitness of non-extreme solutions. The fitness is the average of all objectives.

3. CONCLUSIONS & FUTURE WORK

Methods for generating robust solutions are needed when evolving solutions which are deployed in actual consumer products, e.g. Femtocells. The use of a grammar which produces discrete changes to the pilot power generate better solutions on the training and the test scenarios. Future work is to study more problems and setups.

Acknowledgement

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

References

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

[2] Ian Dempsey, Michael O'Neill, and Anthony Brabazon. *Foundations in Grammatical Evolution for Dynamic Environments*, Springer, April 2009.

[3] E. Hemberg, L. Ho, M. O'Neill, and H. Claussen. A symbolic regression approach to manage femtocell coverage using grammatical genetic programming. GECCO, p639–646. 2011.

[4] L 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*, p 2132–2136. IEEE, 2010.

[5] R. Jain, D.M. Chiu, and W.R. Hawe. *A quantitative measure of fairness and discrimination for resource allocation in shared computer system.* Eastern Research Laboratory, Digital Equipment Corp., 1984.

[6] I. Kushchu. Genetic programming and evolutionary generalization. *IEEE TEC*, 6(5):431–442, 2002.

[7] M. O'Neill, L. Vanneschi, S. Gustafson, and W. Banzhaf. Open issues in genetic programming. *Genetic Programming and Evolvable Machines*, 11(3):339–363, 2010.