A Hierarchical Approach to Grammar-Guided Genetic Programming: The Case of Scheduling in Heterogeneous Networks

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

[1] Natural Computing Research and Applications Group, School of Business, University College Dublin, Ireland
{takfarinas.saber, david.fagan, m.oneill}@ucd.ie, david.lynch@ucdconnect.ie

[2] Bell Laboratories, Nokia, Dublin, Ireland
{stepan.kucera, holger.claussen}@nokia-bell-labs.com

Abstract. Grammar-Guided Genetic Programming has shown its capability to evolve beyond human-competitive transmission schedulers for the benefit of large and heterogeneous communications networks. Despite this performance, a large margin of improvement is demonstrated to still exist. We have recently proposed a multi-level grammar approach which evolves structurally interesting individuals using a small grammar, before introducing a thorough grammar to probe a larger search space and evolve better-performing individuals. We investigate the advantage of using a hierarchical approach with multiple small grammars at the lower level instead of a unique one, in conjunction with a full grammar at the upper level. While we confirm in our experiment that the multi-level approach outperforms the use of a unique grammar, we demonstrate that two hierarchical grammar configurations achieve significantly better results than the multi-level approach. We also show the existence of an ideal number of small grammars that could be used in the lower level of the hierarchical approach to achieve the best performance.

Keywords: Genetic Programming, Telecommunications, Hierarchical Grammar-Guided Genetic Programming, Heterogeneous Network.

1 Introduction

The number of mobile phone users is constantly increasing and is expected to exceed 5 billion by 2019 [16]. Communication network companies strive to retain their current subscribers and attract new ones by diversifying their technology offerings. The progress in the number of clients, the criticality of services, and the hike in consumed data drove network operators away from operation cost reduction to Quality of Service (QoS) improvement [17]. A significant part of the QoS improvement is derived from the use of more elaborate optimisation techniques to manage various parts of the network (e.g., antenna duty cycle, and signal strength variation). In parallel to software solutions, network operators also densify their infrastructure with more performing cells [3] leading to an infrastructure heterogeneity.

Traditional cellular networks only employ Macro Cells (MCs) to cover User Equipments (UEs) such as smart-phones. However, MCs alone struggle to cope with the explosion in the number of devices, and Small Cells (i.e., SCs, low-powered cells) have to be installed alongside them, thus creating a Heterogeneous Network (HetNet). SCs are commonly deployed in hot traffic areas (e.g., parks) to attract UEs in their surrounding and offload MCs. While SCs are low cost and can be deployed in an ad-hoc fashion, they are more prone to interference. The 3^{rd} Generation Partnership Project (3GPP [1]) provisioned a mechanism to mitigate these inter-cell interference called Almost Blank Subframes (ABSs), under which, time is split into one-millisecond sub-frames and MCs are muted during some of them. Muting MCs for a certain duration alleviates the interference experienced by SCs and allows them to communicate with their UEs. HetNets face several problems that necessitate on-line and real-time solutions [15], particularly in our work, we focus on the definition of ABS sub-frames and the scheduling of communications between UEs and their attached cells.

Grammar-Guided Genetic Programming (G3P) algorithm by Lynch et al. [5]) is the first autonomic solution for the scheduling in HetNets working in a millisecond timescale. G3P evolves an expression that maps network statistics to a transmission schedule and achieves results beyond expert-agent heuristics. The same authors also demonstrated, using a genetic algorithm run for a long period, that further improvements are possible. In our previous work [9], we have proposed a multi-level grammar approach to G3P as a means to improve its performance. We designed different grammar levels starting from a small grammar containing a restricted number of terminals (the most important ones), to a full grammar containing all the suitable terminals. We run G3P with the small grammar for a few generations to evolve structurally interesting individuals, before expanding the grammar (changing the grammar to a more thorough one) to further explore the search space and improve the fitness.

In the current work, we investigate the advantage of using a hierarchical grammar (with two levels): multiple small grammars instead of a unique one at the lower level, and one full grammar at the upper level. The idea is to (i) independently run G3P with each of the small grammars for a few generations to evolve different structurally interesting individuals, (ii) gather the best-obtained individuals from each independent run, and (iii) evolve them using the full grammar for the rest of the evolution to improve the fitness function. While several works have previously proposed to use greedy approaches to improve the performance of evolutionary algorithms (e.g., heuristics [13, 12, 11, 8] and exact [14, 10]), this work is the first to use different grammar hierarchies for that purpose.

The rest of this paper is organised as follows: Section 2 formally defines the scheduling in heterogeneous networks problem. Section 3 details the G3P algorithm, the state-of-the-art multi-level G3P approach and our proposed hierarchical grammar strategy. Section 4 describes the experimental environment, whereas Section 5 reports and analyses the results of our evaluation. Section 6 concludes this work.

2 Formal Problem Definition

We consider a HetNet $\mathcal{H}$ composed of a set of MCs $\mathcal{M}$ and SCs $\mathcal{S}$ with $\mathcal{M} \cup \mathcal{S} = \mathcal{C}$, and a set of UEs $u_i \in \mathcal{U}$ receiving a wireless signal σ_i^j from every $c_j \in \mathcal{C}$.

2.1 Heterogeneous Networks

UEs often attach to the cell that provides them with the strongest wireless signal. However, SCs are low powered devices, thus only attach few UEs on that basis.

The 3GPP framework provisioned a bias mechanism i.e., Range Expansion Bias (REB) in order to increase the number of UEs that attach to SCs. REB enables SCs to attach UEs located in areas where their signal is not the strongest. REB biases the signal σ_i^j of $c_j \in \mathcal{C}$ to $u_i \in \mathcal{U}$ by a value β_j, with $\beta_j = 0$ for $c_j \in \mathcal{M}$. Therefore, every UE $u_i \in \mathcal{U}$ gets attached to a cell $c_j \in \mathcal{C}$ such that:

$$c_j = \arg\max_{k=1}^{|\mathcal{C}|}(\sigma_i^k + \beta_k) \tag{1}$$

The area in which UEs would attach to c_j when using the bias β_j, but not attach to c_j when ignoring the bias β_j is called the 'Expanded Region' E_j of a SC $c_j \in \mathcal{S}$. A UE u_i belongs to E_j of $c_j \in \mathcal{S}$ if:

$$c_j = \arg\max_{k=1}^{|\mathcal{C}|}(\sigma_i^k + \beta_k) \quad \wedge \quad c_j \neq \arg\max_{k=1}^{|\mathcal{C}|}(\sigma_i^k) \tag{2}$$

One of the main advantages of adding SCs to the network is the fact that they share the same wireless channel as MCs, thus maintaining the network spectrum and necessitating neither heavy network upgrade nor safety regulations/authorisations. However, that same advantage (i.e., sharing the same channel) leads to severe cell-edge interference at the expended regions. To cope with this, the 3GPP standard splits the time by frames $\mathcal{F}$ of 40 sub-frames (SFs) of 1 ms duration each. Thanks to this standardised time domain and using the ABS mechanism, network operators can mute MCs at given SFs, thus allowing SCs to communicate with their UEs with reduced interference from MCs. Although, while UEs at expanded regions experience a reduction in interference during ABS, UEs attached to MC are not communicating in the meantime.

Figure 1 shows an example of a HetNet composed of 1 MC, 1 SC and 21 UEs. Subfigure 1 shows that only a few UEs attach to the SC due to the weakness of its signal, while most UEs attach to the MC. Subfigure 2 shows the REB mechanism at work, by which the SC expands its attaching region reaching more UEs and mitigating the load on the MC. However, at the same time, the REB introduces severe interference in the expanded region of the SC. Subfigure 3 introduces the

ABS mechanism and mutes the MC at the given sub-frame. Therefore, avoiding the interference at the SC's expanded region, while keeping UEs attached to the MC with no transmission.

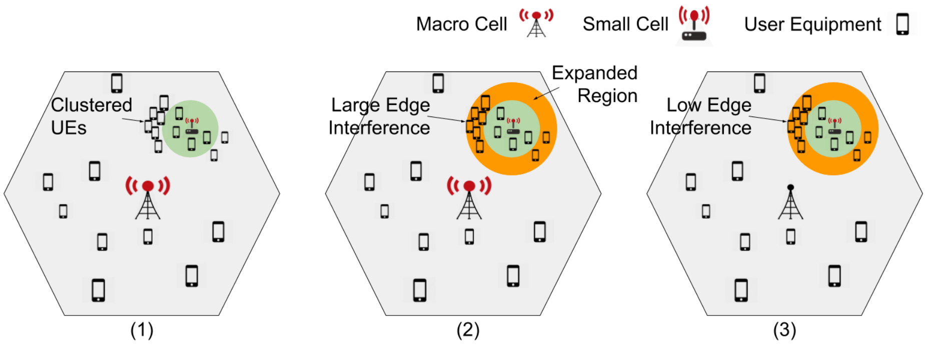


Fig. 1. Example of a HetNet with 1 MC, 1 SC, and 21 UEs. Subfigure 1 shows a few UEs are attached to the SC, while the rest of UEs are attached to the MC, thus overloading the MC. Subfigure 2 shows the SC's expanded region which allows the SC to attach clustered UEs and mitigate the load on the MC. However, this leads to a large interference at the edge of the expanded region. Subfigure 3 shows the muting process which reduces the interference at the edge of the expanded region. However, this also reduced the communication time of UEs attached to the MC.

2.2 Scheduling in Heterogeneous Networks

Let us consider that a UE u_i is able to download an amount of data R_i^f during the SF S_f. This downlink rate R_i^f is well-approximated by Eq., (3) using the bandwidth B, number N_f of UEs communicating at the given SF S_f and the Signal to Interference and Noise Ratio (SINR).

$$R_i^f = \frac{B}{N_f} \times log_2 \left(1 + SINR_i^f\right) \tag{3}$$

MCs have a strong signal, which makes their attached UEs experience high SINR and provides them with high downlink rates whenever the MC is not overloaded. Therefore, all UEs attached to MCs could be scheduled for transmission during all SFs at which the MCs are active, making their scheduling trivial. On the other hand, SCs are low powered devices, making UEs that are attached to them experience a relatively weak signal. Additionally, UEs attached to SCs would be subject to a large interference by MCs during their active SFs.

The bandwidth is hard to improve as it is a scarce and expensive resource. This leaves two levers to act on (i.e., $SINR_i^f$ and N_f). We could improve $SINR_i^f$ of UEs attached to SCs by more often muting MCs. While this would lead to an improvement $SINR_f$ for UEs attached to SCs, it also penalises the downlink rate of UEs attached to MCs (which may be more numerous). We could also attempt to reduce the number of UEs attached to SCs and communicating at the same SF. This would improve the downlink rate for the scheduled UEs. However, it would also penalise the non-scheduled UEs.

All these aspects make transmissions scheduling in HetNets a non-trivial problem. We require a system that defines both the SFs at which MCs are muted and schedules the SFs at which UEs communicate.

2.3 Fitness Function

HetNets operators often aim to optimise the fairness of experienced average downlink rates by all UEs [18] that is expressed in Eq., (4) as it improves low average downlink rates and does not reward high downlink rates. Fairness is the fitness function we aim to optimise. Furthermore, it is the fitness function optimised by works on which ours is based [5, 9].

$$Fairness = \sum_{u_i \in \mathcal{U}} log\left(\bar{R}_i\right) \quad | \quad \bar{R}_i = \frac{1}{|\mathcal{F}|} \sum_{S_f \in \mathcal{F}} R_i^f \tag{4}$$

3 Previous Work and Proposed Approach

In this section, we describe the G3P algorithm for scheduling in HetNets [5], the multi-level grammar approach to G3P [9] and our proposed approach (i.e., the hierarchical grammar approach).

3.1 Grammar-Guided Genetic Programming

The majority of works on transmissions scheduling in HetNets report algorithms designed by expert network operators [2]. The most common techniques partition UEs attached to SCs into two clusters based on SFs at which they are scheduled to transmit.

The first autonomic algorithm that was brought to the problem of scheduling in HetNets is a G3P [5] algorithm. G3P proposed by Lynch et al. [5] is an adaptation of a grammar-based form of GP [6] as implemented in the PonyGE 2 framework [4]. G3P evolves an expression according to a unique grammar F in a Backus-Naur Form (BNF). The grammar F includes arithmetic production rules that are common to the GP community. Additionally, it includes statistics from the networking domain as a means to incorporate domain knowledge. We refer the reader to the original paper [5] for a formal definition of each of them:

```
<expr> ::= <reg> | <reg> | <reg> | <Terminal>
<reg> ::= <expr><op><expr> | <expr><op><expr> | <expr><op><expr> | <expr><op><expr> |
        <non-linear>(<expr>) | <non-linear>(<expr>)
<op> ::= + | - | * | / (protected)
<non-linear> ::= sin | log (protected) | sqrt (protected) | step
<Terminal> ::= <sign><const> | <statistic>
<sign> ::= - | +
```

```
<const> ::= 0.0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0
<statistic> ::= downlink | num_variable | num_att | airtime | congestion |
                avg_downlink_frame | max_downlink_frame | min_downlink_frame |
                avg_downlink_SF | max_downlink_SF | min_downlink_SF |
                avg_downlink_cell | max_downlink_cell | min_downlink_cell
```

G3P maps the evolved expressions and the network statistics to a transmission 'interest' every time a scheduling decision has to be made: whether to schedule the UE to communicate at the given SF or not. For each UE u at every SF f, the expression is evaluated using the network statistics at that SF, and u is scheduled providing there is a positive interest and a sufficiently high SINR. Please refer to the original paper [5] for a more detailed description of the mapping algorithm.

3.2 Multi-Level Grammar

In addition to the full and more thorough grammar (i.e., F as outlined above) defined by Lynch et al. [5], we have previously defined a smaller and more restricted grammar (i.e., S_1) by only keeping a subset of terminals that we believe are the most important [9].

The small grammar is defined by modifying <const> and <statistic>. The number of terminals is reduced to the strict minimum by only keeping a small subset of constants and what seems to be the most important statistics. The downlink is what we would like to optimise. Whereas maximising the value of min_downlink_frame would improve the smallest downlinks. Therefore, improving it would have a better impact on the fitness function. We set in S_1:

S_1 <const> ::= 0.0 | 0.5 | 1.0
<statistic> ::= downlink | min_downlink_frame

After defining the grammars S_1 and F, we adapted the G3P algorithm to take the grammar S_1 at the start of the evolution and dynamically modify the grammar to F after a certain number of generations (in our case, after 10 generations). All individuals obtained using the grammar S_1 are seeded [7] as an initial population to G3P using the following grammar (i.e., F).

While updating the grammar, we do not require any modification in the representation of the individuals as G3P uses a tree representation of individuals and the grammar S_1 is included in the full grammar F. The individuals also do not require the re-evaluation of their fitness as we use the same mapping algorithm and fitness function.

3.3 Hierarchical Grammar

In this work, we also design two grammar levels. However, unlike in the multi-level grammar approach, we design several small grammars for the lower level. Therefore, in addition to the full grammar F from [5], we define multiple small grammars $S_i \mid i \in \{1, .., 5\}$. While the small grammar S_1 is taken from [9], we design by hand four other small grammars S_2, S_3, S_4 and S_5 in a similar way as S_1 by varying their terminals. All $S_i \mid i \in \{2, .., 5\}$ are a subset of F and their production rules <const> and <statistic> have between 2 and 4 terminals each:

S_2 <const> ::= 0.1 | 0.4 | 0.7 | 1.0
<statistic> ::= downlink | max_downlink_frame

S_3 <const> ::= 0.3 | 0.45 | 0.55 | 0.7
<statistic> ::= downlink | min_downlink_cell

S_4 <const> ::= 0.0 | 0.2 | 0.4 | 0.6
<statistic> ::= downlink | min_downlink_frame | max_downlink_frame

S_5 <const> ::= 0.4 | 0.6 | 0.8 | 1.0
<statistic> ::= downlink | min_downlink_frame | min_downlink_cell

Note that <statistic> rules always contain the terminal 'downlink' as it is the most important statistic [9] (we try to improve the downlink). In addition, we include one to two other relevant terminals from {min_downlink_frame, max_downlink_frame or min_downlink_cell} that have been shown to have an impact on the fitness function [5]. Rules <const> are designed to cover different parts of the search range (whole, centre range, higher range, and lower range).

Figure 2 shows an example of the hierarchical grammar approach with two small grammars (S1 and S2) and a full one (F). G3P generates two initial populations (one using each small grammar) of size '$PopulationSize$' each (in our case: 100 individuals per initial population). G3P independently evolves each of them for 'x' generations with the same grammar used to generate them. Afterwards, the $\frac{PopulationSize}{\#SmallGrammars}$ best individuals (in our case, $\frac{PopulationSize}{2}$) from each resulting population are selected and aggregated to form the initial population with F, which is then evolved using the full grammar for 'y' generations. Note that 2x is the computational budget for the lower level, whereas 2x + y is the computational budget for the entire evolutionary process.

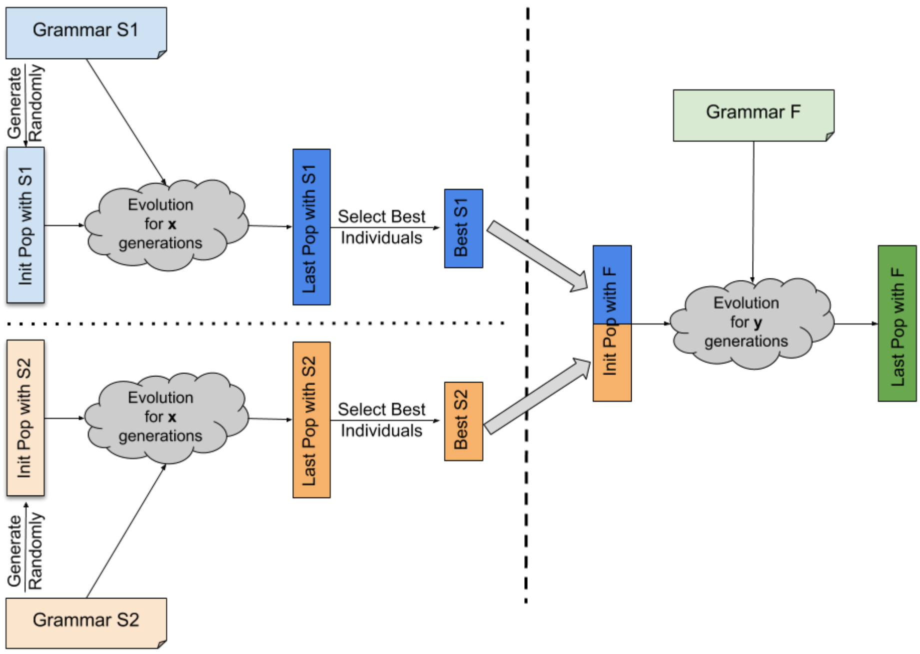


Fig. 2. Overview of the hierarchical grammar approach to G3P with 2 levels (2 small grammars S1 and S2 at the lower level, and one full grammar F at the upper level).

4 Experimental Design

In this section, we describe the dataset, the setup and the statistical test used to assess the significance of our results.

We use in our work the same three HetNets as those used in the works we are comparing to [5, 9]. All the HetNets simulate 21 MCs spread in a hexagonal pattern in a 3.61 km^2 area of Dublin city centre. The three scenarios, however, differ in their number of SCs. The least dense HetNet contains 21 SCs (1 SC per MC on average). The average density HetNet contains 63 SCs (3 SCs per MC on average). The densest HetNet contains 105 SCs (5 SCs per MC on average). Furthermore, 1250 UEs are considered in each of the scenarios. Each of the UEs is attached to either a MC or a SC.

We use the G3P algorithm provided by the authors [5] that is implemented using the PonyGE 2 framework [4]. We set the evolutionary parameters as shown in Table 1.

Table 1. Evolutionary parameters defined for the different G3P approaches: single grammar, multi-level grammar and hierarchical grammar.

Initialisation	Ramped Half-Half
Max initial tree depth	20
Overall max tree depth	20
Population size	100
Number of generations	100
Selection	Fair tournament
Tournament size	1% of population
Replacement	Generational with elites
Elite size	1% of population
Crossover type	Sub-tree with a 70% probability
Mutation type	Sub-tree once per individual
Number of runs	30

We perform the non-parametric test i.e., two-tailed Mann-Whitney U test (MWU) to check the significance of our results. MWU takes performance values (best fitness function values) obtained by two algorithms from each run (in our case: 30) and returns the p-value that one algorithm achieves different results than the other. We consider tests significant with p-values below 5%.

5 Evaluation

We would like to evaluate in this section the advantage of using a hierarchical grammar approach over both a multi-level grammar approach and the original G3P (with one full grammar). Therefore, we consider 6 configurations:

- F: G3P with the full grammar from the start to the end of the evolution [5].

- $S^1$10F: the multi-level grammar approach [9] with G3P starting with one small grammar (i.e., S_1) and introducing the full grammar at generation 10.
- S^i10F with $i \in \{2,3,4,5\}$: the hierarchical grammar approach with G3P starting with i small grammars (i.e., $S_1,...,S_i$) and independently evolving a population with each of them for a number of generations $\lfloor \frac{10}{i} \rfloor$, before gathering $\frac{PopulationSize}{i}$ of the best individuals from each of the independent runs to create a full population that is evolved with the full grammar F for the remaining generations.

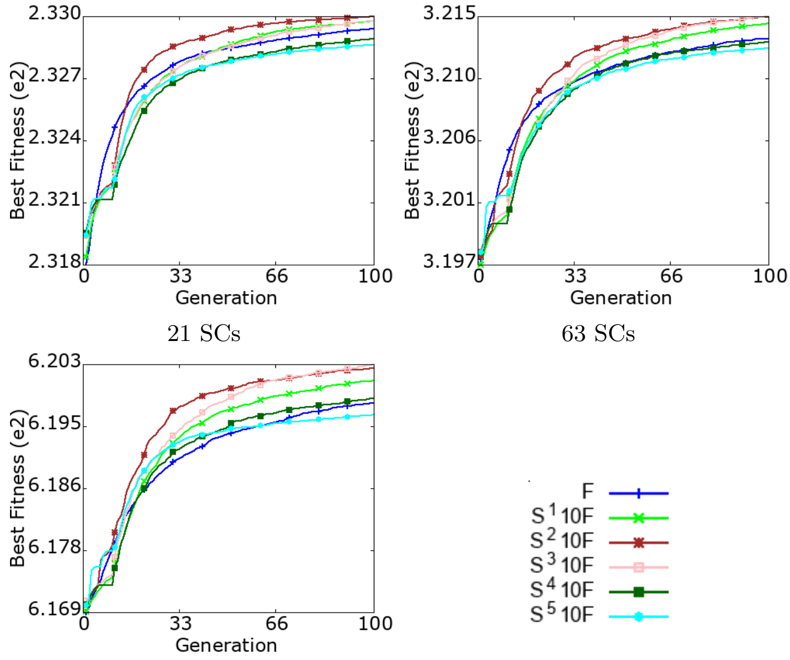
Note that S_i refers to the small grammar S_i, whereas S^i refers to the set of small grammars $\{S_1,S_2,...,S_i\}$. Furthermore, we decided to set the parameters of the various approaches to the same values. More particularly, we set the computational budget of the lower level in the hierarchical approaches to the same value as for the multi-level approach (i.e., 10 generations). While fine-tuning this parameter would likely yield better results, we seek to limit the number of varying elements in our experiments. Therefore, by keeping the same computational budget, we make sure that any improvement is the result of the new approach.

Figure 3 shows the evolution per generation of the best fitness on each instance, obtained by G3P when using the different grammar configurations (results are averaged over 30 runs).

We see from Figure 3 that G3P successfully improves the best fitness using all the grammar configurations. We also see that 100 generations are not sufficient for a full convergence and running more generations is likely to yield a better performance.

Figure 3 confirms that using the multi-level grammar approach $S^1$10F outperforms the single grammar F in all instances. It also shows that the hierarchical approach $S^2$10F yields a better performance over all instances (jointly with $S^3$10F on 63 and 105 SCs) than both the single and the multi-level grammar strategies. However, it also shows that other hierarchical approaches (i.e., $S^4$10F and $S^5$10F) perform poorly as they are outperformed by the multi-level grammar approach in all instances and achieve worse results than the single grammar approach in most cases. This is largely due to the fact that using too many small grammars means that G3P is only allowed a small number of generations to optimise the populations that were generated with each of these grammars (remember that the lower level has to share a computational budget of 10 generations). This is more acute in the case of $S^5$10F where each small grammar is allowed 2 generations (10 generations divided by 5 small grammars) to evolve its population.

We notice that using a hierarchical approach can outperform the performance of a G3P algorithm and outperforms the use of a single or a multi-level grammar approach. However, the number of grammars at the lower level (i.e., number of small grammars) has to be tailored so as to allow G3P to evolve the population that is generated using each of these grammars. In our work, we decided to use the same number of generations allowed to the lower level as in the multi-level approach [9] (i.e., 10) to mitigate the effect of modifying this parameter and make sure that any improvement would be the result of the hierarchical approach.



21 SCs 63 SCs

105 SCs

Fig. 3. Average over 30 runs of the evolution of the best fitness obtained by G3P on the different instances using various grammar hierarchies.

Furthermore, we defined the 5 small grammars $S_i \in \{1, ..., 5\}$ and chose to use them in a particular order (i.e., we have to select S_i to be able to select S_{i+1} for every $i \in \{1, ..., 4\}$). We anticipate that increasing the computation budget for the lower level, choosing different grammars or setting a different grammar selection order might affect the ideal number of small grammars at the lower level. In our case, we have seen that using 2 or 3 small grammars is ideal.

Table 2 reports the mean and the standard deviation of the results obtained by G3P using the different grammar configurations over 30 runs. It also includes the p-value between each of the grammar configurations and either the full grammar F alone or the multi-level grammar approach $S^1$10F.

Table 2 confirms that the hierarchical grammar approach $S^2$10F significantly outperforms both the single grammar F and the multi-level grammar $S^1$10F approaches on all instances. It also shows that $S^3$10F significantly outperforms F and $S^1$10F on all instances (except on 21SCs where results are not statistically significant). Furthermore, while $S^2$10F achieves the best overall mean results on the least dense instance (i.e., 21SCs), $S^3$10F achieves the best mean results on the densest instances (i.e., 63 SCs and 105SCs).

Table 2 also shows high standard deviations with respect to the difference in mean values. However, the standard deviation with $S^2$10F is the lowest in

Table 2. Mean and standard deviation (Sd) over 30 runs obtained by G3P using each of the 6 grammar configurations on the various instances. We also report the p-value (using MWU) between using each grammar configuration against G3P with either F or $S^1$10F. We put in bold best mean performance and significant p-values. We also put '–' when computing the p-value between a grammar configuration and itself.

Instance	Function	F	$S^1$10F	$S^2$10F	$S^3$10F	$S^4$10F	$S^5$10F
21SCs	Mean	232.957	232.995	**233.018**	232.996	232.908	232.877
	Sd	0.051	0.063	0.043	0.065	0.069	0.118
	MWU F	–	**4.43E-04**	**4.49E-08**	**1.20E-03**	**6.05E-04**	**1.71E-04**
	MWU $S^1$10F	**4.43E-04**	–	**2.17E-02**	4.27E-01	**2.27E-09**	**8.28E-10**
63SCs	Mean	321.334	321.443	321.493	**321.497**	321.308	321.260
	Sd	0.075	0.131	0.055	0.092	0.071	0.077
	MWU F	–	**6.76E-06**	**3.63E-12**	**6.92E-10**	**7.84E-02**	**2.13E-04**
	MWU $S^1$10F	**6.76E-06**	–	**1.95E-02**	**1.65E-02**	**1.83E-06**	**1.58E-09**
105SCs	Mean	619.799	620.101	620.274	**620.326**	619.859	619.634
	Sd	0.160	0.244	0.081	0.152	0.150	0.221
	MWU F	–	**2.75E-10**	**4.72E-14**	**1.44E-13**	5.42E-02	**3.38E-03**
	MWU $S^1$10F	**2.75E-10**	–	**8.01E-04**	**1.85E-05**	**7.24E-06**	**3.07E-11**

every instance and is a sign of more stable behaviour. The standard deviation increases from $S^2$10F to $S^5$10F (except between $S^3$10F and $S^4$10F on 63SCs). This indicates that using more small grammars at the lower level either makes G3P behave more erratically (converges to different fitness values) or not fully converge in the given computational budget (requires more generations to fully converge).

6 Conclusion

G3P has been shown to evolve high performing schedulers in HetNets. However, a large potential improvement still exists. We have recently proposed a multi-level grammar approach as a means to improve the performance of G3P by (i) evolving structurally interesting individuals using a small grammar first, and (ii) introducing a more thorough grammar to investigate the full search space and evolve individuals with a better performance.

In this work, we proposed a hierarchical approach whereby we use multiple small grammars independently at the lower level instead of a unique one, before gathering the best individuals and continuing the evolution using the full grammar. While we confirmed that the multi-level approach outperforms the use of a unique grammar, we demonstrated that the hierarchical grammar approach with two or three small grammars at the lower level outperforms the multi-level approach. We also showed the existence of an ideal number of small grammars that could be used at the lower level of the hierarchical approach (in our case 2 or 3) beyond which results are significantly degraded.

As future work, we would like to investigate the automatic design/selection of grammars and the effect of the computational budget on the performance of the lower level of the hierarchical approach. We also would like to apply our hierarchical grammar approach to other problem domains.

Acknowledgement

This research is based upon works supported by the Science Foundation Ireland under Grant No. 13/IA/1850.

References

1. 3gpp: The 3rd generation partnership project. www.3gpp.org
2. Fagan, D., Fenton, M., Lynch, D., Kucera, S., Claussen, H., O'Neill, M.: Deep learning through evolution: A hybrid approach to scheduling in a dynamic environment. In: IJCNN. pp. 775–782 (2017)
3. Fenton, M., Lynch, D., Kucera, S., Claussen, H., O'Neill, M.: Multilayer optimization of heterogeneous networks using grammatical genetic programming. IEEE transactions on cybernetics pp. 2938–2950 (2017)
4. Fenton, M., McDermott, J., Fagan, D., Forstenlechner, S., Hemberg, E., O'Neill, M.: Ponyge2: Grammatical evolution in python. In: GECCO. pp. 1194–1201 (2017)
5. Lynch, D., Fenton, M., Kucera, S., Claussen, H., O'Neill, M.: Scheduling in heterogeneous networks using grammar-based genetic programming. In: EuroGP. pp. 83–98 (2016)
6. Mckay, R.I., Hoai, N.X., Whigham, P.A., Shan, Y., O'neill, M.: Grammar-based genetic programming: a survey. Genetic Programming and Evolvable Machines pp. 365–396 (2010)
7. Saber, T., Brevet, D., Botterweck, G., Ventresque, A.: Is seeding a good strategy in multi-objective feature selection when feature models evolve? Information and Software Technology (2017)
8. Saber, T., Delavernhe, F., Papadakis, M., O'Neill, M., Ventresque, A.: A hybrid algorithm for multi-objective test case selection. In: CEC (2018)
9. Saber, T., Fagan, D., Lynch, D., Kucera, S., Claussen, H., O'Neill, M.: Multi-level grammar genetic programming for scheduling in heterogeneous networks. In: EuroGP. pp. 118–134 (2018)
10. Saber, T., Marques-Silva, J., Thorburn, J., Ventresque, A.: Exact and hybrid solutions for the multi-objective vm reassignment problem. International Journal on Artificial Intelligence Tools p. 1760004 (2017)
11. Saber, T., Thorburn, J., Murphy, L., Ventresque, A.: Vm reassignment in hybrid clouds for large decentralised companies: A multi-objective challenge. Future Generation Computer Systems pp. 751–764 (2018)
12. Saber, T., Ventresque, A., Brandic, I., Thorburn, J., Murphy, L.: Towards a multi-objective vm reassignment for large decentralised data centres. In: UCC. pp. 65–74 (2015)
13. Saber, T., Ventresque, A., Gandibleux, X., Murphy, L.: Genepi: A multi-objective machine reassignment algorithm for data centres. In: HM. pp. 115–129 (2014)
14. Saber, T., Ventresque, A., Marques-Silva, J., Thorburn, J., Murphy, L.: Milp for the multi-objective vm reassignment problem. In: ICTAI. pp. 41–48 (2015)
15. Saber, T., Ventresque, A., Murphy, J.: Rothar: real-time on-line traffic assignment with load estimation. In: DS-RT. pp. 79–86 (2013)
16. Statista: Forecast of mobile phone users worldwide 2018. www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/ (2018)
17. Tall, A., Altman, Z., Altman, E.: Self organizing strategies for enhanced icic (eicic). In: WiOpt. pp. 318–325 (2014)
18. Weber, A., Stanze, O.: Scheduling strategies for hetnets using eicic. In: ICC. pp. 6787–6791 (2012)