

# Cooperative Code-sharing for UMTS Femtocells

David Stynes, Kenneth N. Brown  
 CTVR, Cork Constraint Computation Centre,  
 Department of Computer Science,  
 University College Cork, Ireland

Eric Jul  
 Bell Labs,  
 Alcatel-Lucent Ireland Ltd,  
 Ireland

**Abstract**—Femtocells are low-cost, user deployed base-stations that can alleviate indoor coverage problems. However as femtocell deployment density increases, inter-femtocell interference can severely limit their effectiveness. In this paper, we present a protocol for coordinated auto-configuration of femtocells in a UMTS environment. We use an iterated heuristic search (GRASP) to assign orthogonal CDMA spreading codes in the downlink to avoid interference between overlapping signals. The protocol is compared against a baseline of current practice on a range of simulated problems and shown to consistently provide greatly improved performance for data users, providing up to 480% of the average data rate the baseline provides on certain problems.

## I. INTRODUCTION

Femtocells [1] (FC) are essentially subscriber-deployed mini base-stations that link to the core mobile network through a subscriber broadband connection. They are intended to provide high quality cellular service in residential and commercial environments where macrocell coverage is poor or non-existent. Because femtocells are installed by the subscriber, they must be able to auto-configure themselves [2] and the operator has no control over their locations. Neighbouring femtocells can cause interference to each other, reducing the quality of service received by their connected User Equipment (UE). At present, femtocells simply ignore the presence of neighbours, and, if interference occurs, the users experience a degraded service. This is acceptable when deployment density is low, but when deployment density increases, the interference problem will increase.

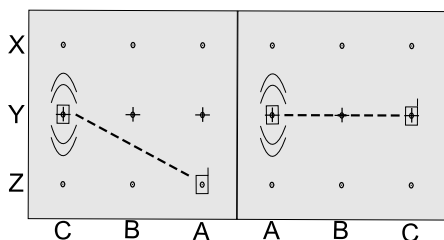


Fig. 1: Layout for a pair of Femtocells in neighbouring rooms.

To estimate this potential interference problem, we simulated two neighbouring femtocells transmitting at fixed power with different scrambling codes in an environment of two 18 m<sup>2</sup> open space rooms with a separating wall equivalent to 10 m of open space, as shown in figure 1. The FCs in each room are placed in row Y at either 3 m from the wall at column A, in the centre of the room at column B, or 15 m from the

wall at column C. We considered nine candidate locations, in rows X, Y and Z and columns A, B and C, for UEs uniformly arranged in a 12 m x 12 m grid around the centre of the room. The results are shown in Figure 2(a), as a heat map of the percentage of the possible data rate that was achieved for each possible location for a UE in each combination of FC positions. The outer y-axis represents the column position of the main femtocell and the outer x-axis represents the column position of the neighbour interfering femtocell. The inner axes represent the coordinates of the user equipment connected to the main femtocell. Our simulations show that on average a UE only receives 32.29% of the data rate possible if there was no interference, and in the worst case, as illustrated in Figure 1, the UE may receive only 5.06% of the possible data rate. To regain this lost bandwidth, some form of interference management is required.

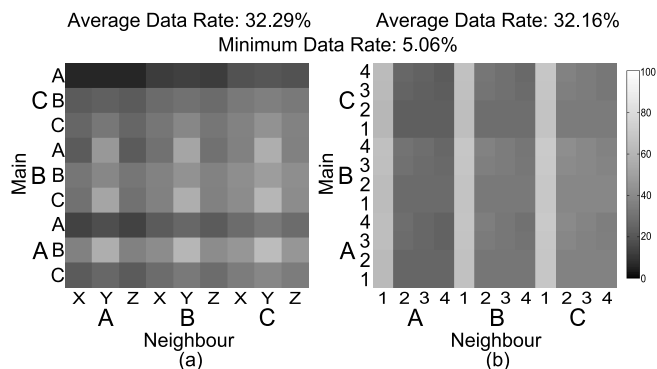


Fig. 2: Inter-Femtocell Interference when (a) using different scrambling codes and (b) using uncoordinated spreading codes

In this paper, we propose a coordinated auto-configuration scheme for the downlink where neighbouring femtocells, in a UMTS network using CDMA, share spectrum through code division to allow the local network to give access to more users and to increase their bandwidth. The femtocells already use the IP backhaul to connect to the core network, and we propose that they could also exchange information with each other through the IP backhaul and agree upon orthogonal CDMA codes for overlapping signals to reduce interference. The central idea is to have cooperating, time-synchronised femtocells use different spreading codes while using the same scrambling code; this differs from current UMTS deployment where each cell uses a different scrambling code that also identifies the cell. Here, a second pilot is used for identifying

the cell. This gives us LTE-like avoidance flexibility in UMTS systems. We specify a protocol for inter-femtocell communication and decision making adapted from a spectrum etiquette proposed for cognitive radio [6]. The initiating femtocell uses an implementation of GRASP [5] (Greedy Randomized Adaptive Search Procedure) to approximate optimal assignments for the code sharing problem in real-time without the need to contact a centralised management system. We describe the algorithm, and evaluate its effectiveness by simulating its performance with various femtocell and user equipment layouts and compare it to a baseline configuration. We show that it can provide a significant increase, of up to 480% of the baseline, in down-link data rate over a range of different problems.

## II. RELATED WORK

In the existing literature, many different forms of interference mitigation for the downlink have already been proposed. In OFDMA networks, the focus has been on using time and frequency division to avoid inter-femtocell interference. In [9] Liang et al. propose a greedy algorithm to assign Physical Resource Blocks efficiently while taking account of QoS demands for different connection types. Their algorithm is centralised and runs at the Femtocell Management System in the core network. Arslan et al. [10] developed a centralised resource management system for allocating near-optimal frequency sub-channels to femtocells and they implemented the system on a WIMAX femtocell testbed. Da and Zhang [4] provide both a centralised scheme and a decentralised scheme, with no femtocell coordination, to allocating sub carriers and transmit powers to the femtocells. Their decentralised approach achieves performance close to that of the centralised scheme. In [8] Ladányi et al. propose a decentralised model for allocating modulation and coding schemes, subchannels and transmit power levels. They do not directly coordinate between the femtocells and provide a two-level decomposition approach for quickly solving the complex resource allocation optimisation problem.

In CDMA networks, the focus has been on using time division to avoid interference. Siddavaatam et al. [11] partition macrocell coverage into concentric regions and assign femtocells time-slot allocations dependant upon the region they reside in. Power control is also used to reduce interference between femtocells within the same region. While in [3], Chandrasekhar and Andrews improve uplink capacity through use of time-hopped CDMA and sectorized receive antennas on the femtocells. However, exploiting code division in CDMA networks has yet to be considered for interference mitigation.

## III. SPREADING CODES AND INTERFERENCE

In UMTS, messages are encoded using a *spreading code* before transmission in the down-link. There is a tree of available spreading codes, and each level of the tree (or *spreading factor*, SF) doubles the length of the spreading code and hence halves the achievable data rate. However, codes that do not share an ancestor/descendant relationship

are *orthogonal*, and orthogonally encoded signals may be transmitted simultaneously without interfering with each other. Each UMTS W-CDMA base station transmits multiple such encoded signals as a combined signal to its connected user equipment. If two neighbouring femtocells using the same scrambling code are synchronised, then the same orthogonality relationship will apply to their neighbouring signals. This time synchronisation is not difficult to achieve, it is already an operational requirement for existing CDMA2000 femtocells. Figure 2(b) shows the average bandwidth received for each UE across all locations when varying between 1-4 UEs per FC when FCs are synchronised and using the same scrambling code but without coordinating their spreading code usage. The inner x-axis represents the number of UEs connected to the neighbour femtocell, and the inner y-axis represents the number of UEs connected to the main femtocell.

As in [12], we model the pathloss  $L_{fu}$  experienced by a signal from a FC  $f$  to UE  $u$  that is  $r$  meters away as:

$$L_{fu} = -28 - 10 * 3.5 * \log_{10}(r), r > 0 \quad (1)$$

The signal  $S_{fu}$  in Watts received by  $u$  is then dependent on the transmit power of  $f$ ,  $P_f$  (in dBm):

$$S_{fu} = 10^{(P_f + L_{fu})/10} \quad (2)$$

We can then calculate the SINR of  $u$  as:

$$SINR_u = \frac{S_{c_x u}}{(\sum_{i=1}^n o_{ui} \alpha_{ui} S_{c_i u}) + N/SF_u} \quad (3)$$

where  $N$  is the background noise,  $c_x$  is the femtocell that UE  $x$  is connected to, and  $i$  is every UE not connected to  $c_x$ .  $o_{ui}$  takes the value 0 if  $u$  and  $i$  use orthogonal spreading codes, and 1 otherwise.  $\alpha_{ui}$  is the ratio of the smaller spreading factor to the larger spreading factor of the 2 codes being used by  $u$  and  $i$ , and  $SF_u$  is the spreading factor of  $u$ 's code.

The final data rate received by  $u$ , with an assumed 3-dB offset from the Shannon theoretical bound, is given as:

$$O_u = \frac{3.86e6 * \log_2(1 + SINR_u/2)}{SF_u} \quad (4)$$

## IV. PROTOCOL DESCRIPTION

We assume the involved FCs are using the same scrambling code, and have a second pilot for identification. Each FC receives reports from its user equipment on potentially interfering FCs, and maintains a neighbour list. Each FC may communicate with any other through the IP backhaul. A FC that wishes to change its spreading codes must contact its neighbours to determine their constraints; if appropriate codes can be found, they are activated and the protocol ends. If no codes are appropriate, then the neighbours may also change codes to accommodate the request, but they in turn need to contact their neighbours to report their constraints. If no codes can be found at this stage, the protocols ends, and the original FC must back-off and try again after a random wait. We assume the original FC is not permitted to directly

alter the codes of its neighbours, and so only advises them on possible changes. We distinguish between two kinds of possible reassignment for the neighbouring FCs. In a *vertical* reassignment, the new code must be lower in the code-subtree of the original code or be unchanged. A vertical change will not cause a new source of interference to other neighbours. In a *horizontal* reassignment, the new code is not in the code-subtree of the current code. This can cause new interference for the femtocell's neighbours. If a neighbour's UE  $e$  receives a horizontal change, every UE connected to FCs that  $e$  can hear is limited to a vertical change.

All FCs maintain a list of known neighbours. Anytime a UE informs FC  $A$  of a new FC  $B$  not in  $A$ 's neighbour list,  $A$  adds  $B$  to the list and informs  $B$ , which then adds  $A$  to its own list. Entries are removed, if no activity has been detected for some time. When a FC that is close to being deleted is detected by a UE, its timer is reset to the maximum, and a message sent to that FC for it to update its corresponding timer. Thus all neighbour lists are synchronised and no pair of FCs should ever be in the situation where only one is aware of the other.

An FC initiates the protocol whenever it requires more bandwidth for its UEs. This could, for example, be because: a UE becoming active and requesting a connection; a connected UE moving into an area with excessive interference from a neighbouring FC; a connected UE receiving excessive interference from a neighbouring FC changing its code use; handover from the macrocell/FC to a FC; or the bandwidth requirements of a connected UE change.

#### A. The Core Protocol

- 1) *Stage 1*: Initiating FC  $A$  contacts all neighbour list FCs requesting their UE codes, the signal strength at which they hear  $A$  and the interference from other sources;
- 2)  $A$  waits until all neighbours have replied;
- 3) If  $A$  can find a code reassignment for its UEs that does not drop neighbours below their minimum required service level, it reassigns its codes;
- 4) Else  $A$  tells all active neighbours to join active group  $G$  and initiate *Stage 2*:
  - a) Each FC  $g$  in  $G$  requests UE info from FCs in its neighbour list, and waits until all FCs have replied;
  - b) Each  $g$  in  $G$  sends  $A$  a message telling which codes each of its UEs must be orthogonal to, due to their neighbours currently occupying them, and states which neighbours cause the restrictions;
  - c) If  $A$  can find a reassignment compatible with the vertical/horizontal assignment restrictions,  $A$  sends a message to all members of  $G$  stating  $A$ 's new codes and the horizontal/vertical restrictions on their UEs,  $A$  and the members of  $G$  then implement their own reassignments;
  - d) Otherwise  $A$  backs off.

Note that in step 4(c) the possible solutions found by  $A$  are restricted by the horizontal limitations. The message sent must provide enough information to allow the neighbours to

locally calculate their own new compatible code reassignment without requiring further inter-FC communication.

We also explore a variation of the above protocol, in which at step 4(b) the members of  $G$  inform  $A$  of all their neighbour's codes and signal strengths. Using this information, we then allow  $A$  to find reassignments that may negatively affect the external neighbours of  $G$  but do not cause any of their UEs to no longer achieve their minimum bandwidth. We refer to this variation as the Extended (EX) version of the protocol.

$A$  and stage 1 neighbours are locked from joining other runs of the protocol. Second degree neighbours are locked from other EX-Stage 2 and Stage 1 protocol runs once EX-Stage 2 is invoked. They are never locked from non-EX Stage 2 runs. If the protocol encounters a locked femtocell, it backs off and waits a random period of time before restarting.

We assume two types of possible users, 'Voice' and 'Data', both having a minimum required bandwidth. Only data users can benefit from bandwidth above their minimum. We also assume that the FCs are trying to ensure all UEs receive their minimum required bandwidth while also maximising either (i) the average bandwidth across all data UEs or (ii) the minimum bandwidth received by any data UE.

#### B. The GRASP Model

The initiating FC  $A$  can use any method to search for code reassignments. In this paper, we use GRASP [5], which starts by generating a greedy random assignment of codes to all UEs, and then performs iterative improvements to the solution through local search. Once a stopping condition is reached (e.g., a local optimum, a fixed number of iterations being exceeded, or a timer expiring), a new initial solution is generated and the process repeats until a final stopping condition is reached (e.g., a certain quality of solution, a total time limit, or a limit to the number of explored solutions).

1) *GRASP Stage 1*: For the random initialisation, we assign every UE a randomly chosen spreading code amongst those codes that are orthogonal to all codes already assigned so far. Data users are assigned a spreading code at depth 5 in the tree (spreading factor 16), and voice users are assigned a spreading code at depth 8 (spreading factor 128). Once any single voice user is assigned a code, we fix the subtree at depth 5 which contains it, and all remaining voice users must take a code in the same subtree, to group the voice users together.

Once every UE has been assigned a code, we begin the iterations to improve the solution. We attempt to replace each UE's code with the parent of that code in the tree. For voice UEs, we continue this procedure until they reach their required minimum bandwidth, or would become non-orthogonal to another connected UE's code. For data UEs, we continue until moving to the parent would reduce the average/minimum bandwidth of the data UEs, or would be non-orthogonal to the other connected UEs. Once no UE can have its code raised any further, we generate a new initial solution and repeat. We stop once the Stage 1 time limit is reached and report the best solution found so far, if any. GRASP Stage 1 has a runtime complexity of  $O(F_{max}U_{max}^2)$ . Where

```

1: Best Solution  $\delta = \emptyset$ 
2: while  $time \leq timeLimit$  do
3:   Generate initial solution  $\alpha$ 
4:   for  $i = 1 \rightarrow numIterations$  do
5:     Select random FC  $f \in G \cup A$ 
6:     Select random UE  $u \in f$ 
7:      $bestCode = 0$ 
8:     for  $n = 1 \rightarrow 10$  do
9:       Select a random code  $c \in validCodes(u)$ 
10:      if  $eval(u, c) > eval(u, bestCode)$  then
11:         $bestCode = c$ 
12:      assign  $u$  code  $bestCode$  in  $\alpha$ 
13:      for  $v \in f, v \neq u$  do
14:        if  $notOrthogonal(bestCode, code_v)$  then
15:           $best_v = 0$ 
16:          for all  $c \in validCodes(v)$  do
17:            if  $eval(v, c) > eval(v, best_v)$  then
18:               $best_v = c$ 
19:            assign  $v$  code  $best_v$  in  $\alpha$ 
20:          if  $evaluate(\alpha) > evaluate(\delta)$  then
21:             $\delta = \alpha$ 

```

Fig. 3: Stage 2 Algorithm

$F_{max}$  is the maximum number of neighbouring femtocells of any femtocell, and  $U_{max}$  is the maximum number of UEs connected to a femtocell.

2) *GRASP Stage 2 (fig 3)*: To generate the initial solution at line 3, we randomly assign a UE  $v$  as being allowed all (horizontal and vertical) moves, and all UEs which require to be vertical if  $v$  is horizontal are limited to vertical moves. The initiating FC's UEs can all make any kind of move. We repeat until all UEs are assigned as eligible to make either all moves or only vertical ones, and then start to assign spreading codes.

We order all UEs in descending order of demand, breaking ties in descending order of the number of neighbouring FCs. We randomly select a UE from the first four members of this list and assign it a spreading code (subject to horizontal or vertical constraints) compatible with any assigned UEs that are connected to the same FC. As with in Stage 1, data UEs are assigned a code at SF16, and voice users are assigned at SF128. We evaluate all possible valid codes the UE could be assigned to and assign it to the highest evaluated code. The evaluation function provides an estimate of the expected average/minimum bandwidth of the data UEs.

When evaluating individual code assignments (lines 10+17), the function returns the estimated improvement to the average/minimum data rate of the data users. When evaluating full solutions (line 20), the function returns the average/minimum bandwidth of all data UEs in the problem. The function  $validCodes(u)$  at lines 9+16 returns the set of all codes that are compatible with the horizontal/vertical restrictions on the UE  $u$  (but not less than SF32 for data users).

We experimented with two approaches, one that takes account of the reduced interference for the ratio of the different spreading factor of codes in use (SF), and one that assumes

that every neighbouring FC is using the same spreading code as the UE, if not orthogonal and so causing it full interference (IMP). Empirically, it appears best for SF to revert all changes to  $\alpha$ , if it was not better than  $\delta$  at line 20. For IMP we always retain the changes to  $\alpha$  regardless.

A single run of GRASP Stage 2 to generate an initial solution and iterate it, has a runtime complexity of  $O(IF_{max}U_{max}^2)$ . Where  $I$  is the number of iterations.

## V. EXPERIMENTS

We evaluate our cooperative code-sharing model against the current practice baseline in simulation, for a set of scenarios with interfering FCs. We compare the number of times all UEs receive at least a minimum service, and compare the quality of service offered to the UEs. Quality of service can be determined by two different measures: (i) the minimum data rate received by any connected UE that requested a data service, and (ii) the average data rate received by the complete set of data UEs. Here we focus on maximising the minimum data rate received by the data UEs.

We assume all femtocells transmit signals of strength 8 dBm. We vary the distance between the femtocells, the maximum distance from their connected FC at which UEs can be located and the fraction of users that are data users. Data users are assumed to require a minimum of 512 kbps, and voice users 12.2 kbps. UEs are added one by one to each FC, and each addition invokes our protocol to assign codes to the new UE and potentially change those of the older UEs. We assume a corporate deployment of FCs and so test on larger numbers of UEs per FC than may be expected in a residential deployment.

We consider a *solution* as having been reached when all UEs are added to the femtocells without ever failing to provide the minimum data rate to every connected UE at each step. We allow a time limit of 100 ms for Stage 1 of the protocol and 1000 ms for Stage 2. We assume a background noise of  $N = 4.8260e - 14$  Watts. For each problem set, we generate 50 random placements of UEs for the FCs.

To reduce problem complexity when trying to optimise code assignments, our algorithm only treats the FCs with a received signal strength above a threshold as neighbours, and all weaker signals are added as increased noise, assumed to be causing the maximum interference possible even if their actual code assignments may result in lesser interference. This means that any SINR evaluations during search are an underestimate of the eventual SINR. In these experiments, we set the threshold to FCs that are within 60 m of a UE. A larger threshold means more FCs will be included in Stage 2 of the protocol, but also results in an on average lower performance within the same time limit. We use the EX version of the protocol for all tests.

The baseline (BL) algorithm attempts to assign the entire code tree to the UEs of each FC, though the root code is never assigned. We compute a complete coverage of the tree using the number of codes equal to the number of connected data UEs plus one if any voice UEs exist, and then the UEs with highest usage demands are assigned the codes highest in the

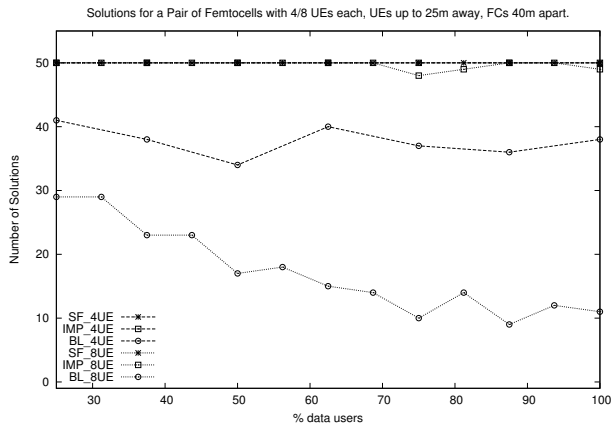


Fig. 4

tree, with the UE acting as a placeholder for all the voice call UEs assumed to have the lowest demand of all data UEs. UEs with the same level of demand are randomly assigned to the codes. When the voice UEs are to be assigned, they are placed in a single sub-tree rooted at SF64. Each femtocell in the baseline uses one of 3 possible scrambling codes, choosing to use the code with the least neighbouring femtocells of the same code when it first activates. Signals from femtocells using different scrambling codes are treated as noise as the interference cannot be avoided even if spreading codes are orthogonal.

#### A. A Pair of Femtocells

Initially we investigate performance for a pair of femtocells placed 40 cm apart, with 4 and 8 UEs each randomly placed within 25 m of their connected femtocell. Figure 4 shows the number of time all UEs received their minimum service level. For 4 UEs we can see that both of our methods find a valid assignment for all UEs, while the baseline fails in nearly 33% of cases, typically due to a UE being closer to the neighbour than its connected FC. For 8 UEs we do not always reach a solution for IMP, but maintain a high success rate while the BL drops significantly at higher fractions of data users.

Increasing the range that UEs can be placed to 50 m from their connected FC gives a similar drop in performance for the BL on problems with 4 UEs per FC. The BL then only reaches a solution in 12% of cases, while IMP maintains a 100% success rate and SF 98%. The SF fails on these problems as some of the optimisations it makes on the intermediate UEs leads to non-robust solutions that cannot be extended to allow new UEs to be added. We note for future work the need to balance optimisation with picking robust solutions that can accept additional UEs or increased service demands.

In Figure 5, we compare our performance on the 4 UE problems in which the baseline found a valid solution. The quality of our solutions is significantly higher, with close to 500% improvement when there are 40% data users. As the data user density increases, the improvement over the baseline decreases because there is less scope to avoid interference.

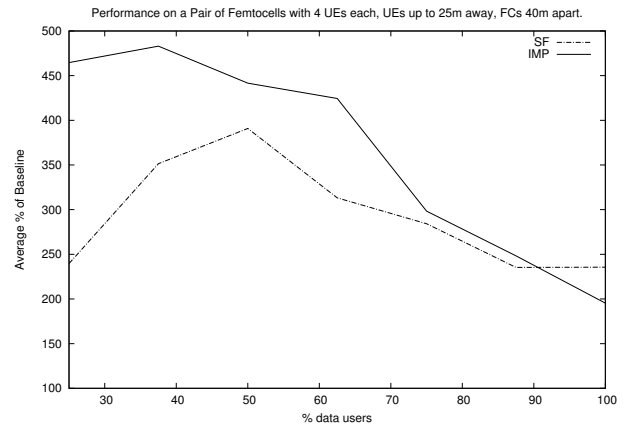


Fig. 5

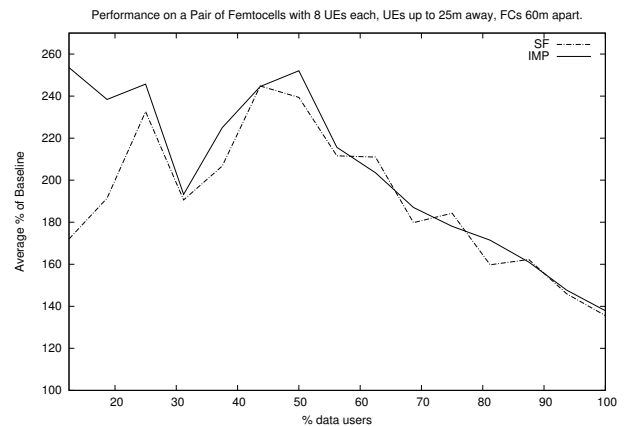


Fig. 6

If we increase the placement of the FCs to 60 m apart, all approaches find a solution to every test case, even when there are 8 UEs per FC, shown in Figure 6. As we can see, even on these easier problems where reaching a valid solution is not difficult, we can still make significant improvements on the data rate received over the baseline.

#### B. A Grid of 9 Femtocells

Finally, we investigate a grid of 9 femtocells, representing a corporate deployment, arranged into 3 equally spaced rows of 3 femtocells. The spacing between the rows and between the femtocells is always the same (initially 40 m for both). Figure 7 shows number of solutions reached for 4 and 8 UEs per FC. The baseline rarely achieves a valid solution. Our algorithms perform well on 4UE grids, but cannot find solutions to 8UE grids with moderate to high data user densities (although solutions might not exist for these problems). We observe that SF performs better than IMP quite often. During search, IMP overestimates the effect of interference, and then corrects it when evaluating the solution. When the constraints are so tight that no solution can be found using the overestimate, SF, which uses the more accurate calculation throughout, can still be able to find a solution. When the spacing between FCs is increased to 60 m, the baseline reaches solutions on every

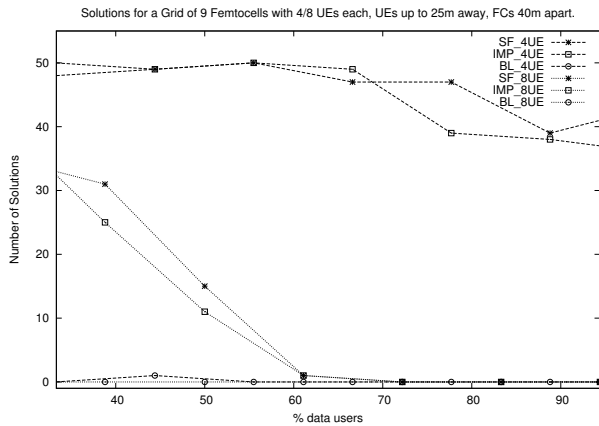


Fig. 7

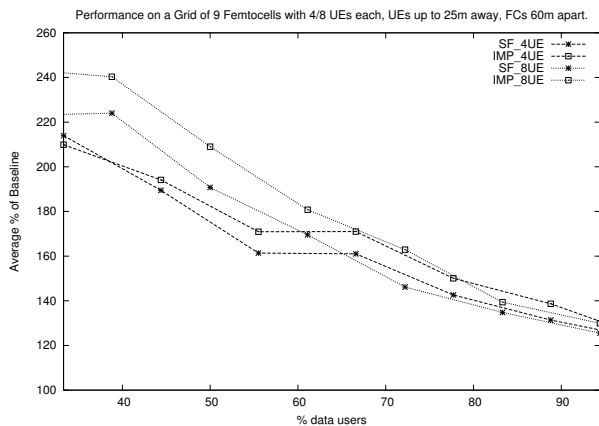


Fig. 8

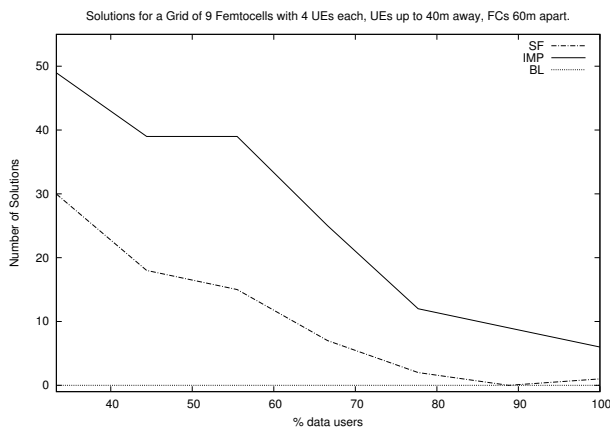


Fig. 9

case with 4UEs per FC, and on 99.5% of cases with 8UEs per FC. Figure 8 shows our performance on these problems, while figure 9 shows solutions when the maximum distance from UE to FC is raised to 40 m.

## VI. CONCLUSION

We have proposed a cooperative protocol for UMTS femtocells communicating over the IP backhaul and searching for coordinated code assignments that reduce inter-femtocell

interference. We have shown the approach outperforms the baseline, providing a minimum service to many more users, and providing significantly higher bandwidth for those data users it connects, up to an average of 480% improvement across 50 cases. For looser problems, a variant (IMP) that simplifies the effect of the spreading codes on interference provides better results, while the SF variant performs best on problems with very little scope for interference tolerance where precise knowledge of the SINR can be the difference between a valid solution existing or not.

In the future, we will develop techniques that allow a femtocell to deduce which of the two variants it should use prior to attempting code assignment optimisation. We also note that greedily optimising code assignments can give a brittle solution unable to accommodate future changes, so we will find a more effective balance between optimisation and robustness. Finally, adapting the transmit power of femtocells can result in large performance gains [7] and we will investigate doing so in conjunction with spreading code assignment.

## ACKNOWLEDGMENT

This work is funded in part by Science Foundation Ireland. We thank Rouzbeh Razavi for his help with formulas for signal propagation, Holger Claussen for insightful discussions and comments on earlier drafts of this paper, and Joe Bater, for his earlier work on the spectrum commons protocol.

## REFERENCES

- [1] V. Chandrasekhar, J. F. Andrew, and A. Gatherer, *Femtocell Networks: A Survey*, IEEE Communications Magazine, pp. 59-67, Sept. 2008
- [2] H. Claussen, *Performance of Macro- and Co-Channel Femtocells in a Hierarchical Cell Structure*, IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1-5, Sept. 2007
- [3] V. Chandrasekhar and J. Andrews, *Uplink capacity and interference avoidance for two-tier femtocell networks*, IEEE Transactions on Wireless Communication, vol.8, no.7, pp.3498-3509, July 2009
- [4] B. Da and R. Zhang, *Cooperative Interference Control for Spectrum Sharing in OFDMA Cellular Systems*, IEEE International Conference on Communications (ICC), pp.1-5, 5-9 June 2011
- [5] T. A. Feo and M. G. C. Resende, *Greedy randomized adaptive search procedure*, Journal of Global Optimization, vol. 6, pp.109-133, 1995
- [6] J. Bater, H.-P. Tan, K. N. Brown and L. E. Doyle, *Maximising Access to a Spectrum Commons using Interference Temperature Constraints*, Proceedings CrownCom, 2nd Intl Conf. Cognitive Radio Oriented Wireless Networks and Communications, Florida, pp. 441-447, 2007
- [7] H. Claussen, L. T. W. Ho, and L. G. Samuel, *Self-optimization of Coverage for Femtocell Deployments*, Wireless Telecommunications Symposium, pp. 278-285, April 2008
- [8] Á. Ladányi, D. López-Pérez, A. Jüttner, X. Chu and J. Zhang, *Distributed Resource Allocation of OFDMA Femtocells in Enterprise Scenarios*, IEEE Globecom Workshop on Enabling Green Wireless Multimedia Communications (MMCOM), pp. 744-749, December 2011
- [9] Y.-S. Liang, W.-H. Chung, G.-K. Ni, I.-Y. Chen, H. Zhang and S.-Y. Kuo, *Resource Allocation with Interference Avoidance in OFDMA Femtocell Networks*, IEEE Transactions on Vehicular Technology, pp. 2243-2255, June 2012
- [10] M. Y. Arslan, J. Yoon, K. Sundaresan, S. V. Krishnamurthy, and S. Banerjee, *FERMI: A Femtocell Resource Management System for Interference Mitigation in OFDMA Networks*, 17th annual international conference on Mobile computing and networking, pp. 25-36, 2011
- [11] R. Siddavaatam, A. Anpalagan, M. Tharma, and J. Nadaraj, *Interference Mitigation in Femtocell CDMA Networks Through Time Reuse Partitioning*, International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), pp. 1-5, December 2011
- [12] 3GPP TR 36.814, *Evolved Universal Terrestrial Radio Access; Further advancements for E-UTRA physical layer aspects*, v9.0 2010.