2013 27th International Conference on Advanced Information Networking and Applications Workshops

Mobility-Sensitive Power Control for MBSFN Cellular Networks

Konstantinos Asimakis†, Christos Bouras*†, Vasileios Kokkinos*† and Andreas Papazois*†
* Computer Technology Institute & Press "Diophantus", Patras, Greece
† Computer Engineering & Informatics Dept., University of Patras, Greece
asimakis@ceid.upatras.gr, bouras@cti.gr, kokkinos@cti.gr, papazois@ceid.upatras.gr

***Abstract*—Cellular networks' Multimedia Broadcast/Multicast Service (MBMS) over a Single Frequency Network (MBSFN) allows broadcasting cell signals to constructively interfere and users to take advantage of increased bitrates compared to other broadcasting technologies. In this paper, we present our work on the optimization of the power control for future cellular networks that employ MBSFN transmission scheme. We propose a novel simulator and optimizer that can minimize the transmission power of individual cells taking into account the changing positions of users. In order to read descriptions of the various scenarios, the simulator uses the eXtendable Temporal Network Description Language (XTNDL), a language that is defined in this paper. The optimizer's engine can be used in conjunction with the simulator or can be modified to read real time measurements from a real network.**

I. Introduction

There are many technologies created to face the ever increasing problem of cellular network's resource usage when data have to be broadcasted to many recipients. Usually the data that need to be broadcasted are multimedia and with the increasing need for higher quality, we constantly need better ways to transmit said data using the fewest possible resources. Multimedia Broadcast/Multicast Service (MBMS) is one such technology introduced in the standard for cellular networks by 3^{rd} Generation Partnership Project (3GPP). MBMS will achieve high transfer rates by employing the MBMS over a Single Frequency Network (MBSFN) transmission, where the cells transmit synchronized data so that the interference of the cell is no longer destructive. The signals of the cells that form what we call an MBSFN area [1], [2] interfere constructively, increasing the signal to noise ratio. In turn this will increase the Spectral Efficiency (SE), i.e., the throughput achieved with a certain frequency bandwidth, of the data channel and therefore the total available bitrate. If we remove the synchronization of cells from MBSFN we basically have a Point-to-Multipoint (PTM) transmission, which although by itself is fairly resource efficient, it suffers from inter-site destructive interferences, especially at the cell edges, unless guard zones are used [3], i.e., each cell transmitting in a frequency that shares with none of its neighbors. The synchronization of MBSFN can be done either over the air using up minimal bandwidth or using a wired network, in either case it requires special equipment.

In the past, different aspects of MBSFN were studied in order to increase efficiency. For example, the authors of [4] propose analytical approaches to evaluate MBSFN. The Modulation and Coding Scheme selection is studied under various scenarios by the authors of [5] and [6]. The authors of [7] present interference coordination methods for MBMS in 4G cellular networks for better SE, especially at the cell borders. Additionally, 3GPP in [3] assesses the SE and the Resource Efficiency (RE) under varying numbers of MBSFN assisting rings. The combination of MBSFN and PTM is studied in [8] using a novel algorithm. The algorithm is able to simulate networks composed of a hexagonal grid and is based on a simple extrapolation of the input data. The simulation results show that the usage of strategically placed assisting cells can increase the RE and that assisting rings generally reduce the RE of a network. The first version can only be used in theoretical topologies and does not take the number of people in each cell into account. The second version of the algorithm which supported real coordinates for users and cells and included a genetic algorithm for the power optimization of the network is presented in [9].

In this paper we present a major upgrade of our optimization algorithm, which has been extended and modularized to support more metrics and more complex network topologies and its capabilities. The most notable one is the support for moving users and the use of an innovative network description language as input. The algorithm is implemented in a graphical tool, named DAGraCO [10], which is written in LÖVE [11], a framework for writing 2D applications using Lua [12] programming language. The simulator takes into account the characteristics of the MBSFN. Nearby cells that transmit media using MBSFN form the MBSFN area. Those cells transmit in sync independently of whether they contain receivers or not. Even cells that do not contain users assist by increasing the SE of users in nearby cells. MBSFN signals of different cells, because of the synchronized transmission can constructively interfere to some degree that depends on the distance each signal travels before it reaches the user. This is the main difference between MBSFN and PTM transmissions; PTM transmissions cause interference to any other nearby cells that transmit at the same frequency. In our past research works [8] and [9], we saw that in many cases the usage of assisting cells is rarely needed. It is actually better to increase the power of cells that actually contain users instead of using nearby assisting cells to increase the received signal of users from a distance. For this reason, in this paper we do not use any PTM transmitting cells. Another important contribution of this

978-0-7695-4952-1/13 $26.00 © 2013 IEEE
DOI 10.1109/WAINA.2013.73

work is the definition of the eXtendable Temporal Network Description Language (XTNDL), a language that can be used for the description of cellular network's topology as well as the user mobility scenarios.

The remainder of the paper is structured as follows: in Section II, we briefly describe the XTNDL language without listing unnecessary technical details in the description. In Section III we present the proposed optimization algorithm. Section IV describes the experiments demonstrating the operation of the proposed algorithm. Finally, our conclusions and some possible next steps are described in Section V.

II. eXTendable Temporal Network Description Language

The input to the algorithm is basically a list of associative arrays. Every cell in the list represents a single user and the list is indexed using the user ID. Each cell holds an associative array that is indexed by time (which is a floating point number) and contains the position of that user at that point in time.

XTNDL provides functions to populate those tables easily. It is invented as a scenario description language for cellular networks. It should be extendable, defining only the structure and leaving to the simulator the minimum data requirements. Each scenario is described fully by a set of associative arrays. Each array in the set describes fully the position, movement and all related data for a specific type of entity. For example a single array describes users while another array describes cell bases. The arrays store data in entityID-entityData pairs. The entityID can be any string, e.g., "User1", or even a number and will be used as a reference to a specific entity. In either case the entity array stores in every index another associative array which is indexed by timestamps (floating point seconds) and contains arrays of data that describe the entity in that point in time. Additionally each entity array may contain any number of generic options in the "general" field. This field should store data about the entity that is valid always. The special field interpolate may provide a custom function that will be used for interpolating the data between points in time. If this function doesn't exist, linear interpolation should be used by the simulator for every named piece of numerical data that appears in one or more points in time. For non numerical data the default is to interpolate by using the value of the oldest of the two values.

III. Proposed Algorithm

The algorithm emulates the network in real time. It does small, semi-random changes to the transmission power of the cells and after simulating the network it decides whether the changes should be kept or rolled back. Each user is able to mark the three closest (in terms of path loss) cells to him. Marked cells tend to increase their power while unmarked cells tend to decrease it. For example a marked cell that transmits with 10W can be altered randomly to transmit with 9.8W-10.4W in and step of 0.1W. If it was unmarked it would be randomly changed to some power level between 9.6W and 10.2W.

If the algorithm was applied to a live network it would constantly need to know the bitrate that each client achieves and the path loss between each user and cell. In our case the user positions are known so we estimate the bitrate for each user and feed it to the optimization part of the algorithm in real time. That means that 1 second in simulation equals one second of running the algorithm. It is possible to change that ratio if needed for the purpose is to show how the algorithm could cope with real time updates of the user bitrates and not to actually optimize the network.

In an real network we would be unable to have instantaneous evaluations of the network. To accommodate for that, each simulation and evaluation is broken up into phases.

Phase 1: The current state of the network is evaluated. A snapshot of the current network configuration (an array containing the transmission power of every cell in our case) as well as the current score of the network is saved in memory for use during the next phase. The network topology is randomly changed in terms of transmission power of each cell. This optimization round ends. The next round will be a phase 2 round.

Phase 2: The current state of the network is evaluated. If it is better then the last known state then the last known network coverage is dumped and the current network coverage is saved in memory for use during the next round. The current round ends and the next round will be a phase 2 round again. If, on the other hand, the last known network coverage had a better score than the current coverage, then we roll back to that network coverage and end this round. The next round will be a Phase 1 round.

Between rounds a small time passes to redraw the screen. This time in a real live application of the algorithm would be spent evaluating the bitrates achieved by each user. If this measurement is bigger than the amount that the algorithm spends drawing the screen then it would be trivial to delay the screen redraw to simulate the actual delay of user bitrate measurements.

Whenever the network topology needs to be evaluated the algorithm runs separately for each user, following these steps: First, the distance of each user from each base station is measured. Based on that distance we estimate the received signal power using the COST 231 path loss model. The next step is to estimate the SE of each user. We can then see if the SE is over a threshold value that would also make the bitrate be over some acceptable service threshold. If the algorithm was to be applied on a live network then we would need a way to measure each user's bitrate directly. Initially we used a (function with many parts) where if more than 95% of the users has a satisfying bitrate, the evaluation of the network will be 1+1/power which is a number in $(1,\infty)$ and the higher this score the better. If less than 95% of the users are satisfied then the score is the portion of the users satisfied which is a number in [0,0.95), again the higher the better. The non-continuity in the score graph in (0.95,1] did not affect the optimization algorithm since it just tries to maximize the score. Taking into account that this a real time algorithm we have to

make the transition smooth and continuous because otherwise the algorithm would be have trouble being efficient when users move. Specifically, since the algorithm keeps the amount of satisfied users barely over 95% in a real time scenario where users move, their movement would often cause the satisfied users to fall slightly below 95%. This in turn would cause the algorithm to suddenly increase the network power consumption to compensate for that. But in the meantime the specific solution that the algorithm might have converged into could be possibly destroyed. We therefore need the transition to be smooth so that in the case where the satisfied users go slightly below 95% the algorithm will not destroy the overall solution, since, if it tries to do so, the now visible (part?) of the score function would cause a big decrease in the score.

IV. Simulation Experiments

In out past work we used a genetic algorithm to optimize a static network. In the current version of our algorithm the data we process are not only spatial but also temporal. If we used our algorithm in a real network and not just in a simulation, a genetic algorithm would be impractical since it's hard, if possible at all, to keep a population of possible network coverages (as genetic algorithms require) and be able to test them for fitness in the network. Even if we did test each atom in the population of possible network coverages sequentially, the resulting performance would be mediocre since many solutions would result in bad network coverage and interrupts in service. Therefore we need to use an algorithm that makes conservative changes which will not cause serious service disruptions and which will slowly drift towards better network coverage.

In a real network the algorithm will be able to gather statistics about the achieved bitrates of each user and therefore will not rely on path loss models. The metric that the algorithm will try to maximize will be the same with the one described in Section III but instead of having to simulate the network it will now percentage of satisfied users.

A. Vehicular users close to village

Our first experiment involves the scenario where a road passes by a village. A total of 36 base station antennae form a hex grid with inter-site distance equal to 1000m. On the west side of the grid we place a static group of 100 mobile users in a square area with side equal to 1000m. A road with mobile users traveling at 50km/h passes at a distance of 2000m from the center of the village. Every 20s a new mobile user enters the area from the north side of the road and stays visible for 382.6s when he exits the simulated area from the south side of the road. The experiment spans a period of 10 minutes. In Fig. 1 we can see a snapshot of the simulated area during the experiment. The red lines connecting users to base stations show which antennae each users votes for a power increase. Antennae with users voting for them have the tendency to increase their power more easily compared to those that no one voted for. The grey lines, when visible, indicate that the user is mainly served by some remote antenna instead of one

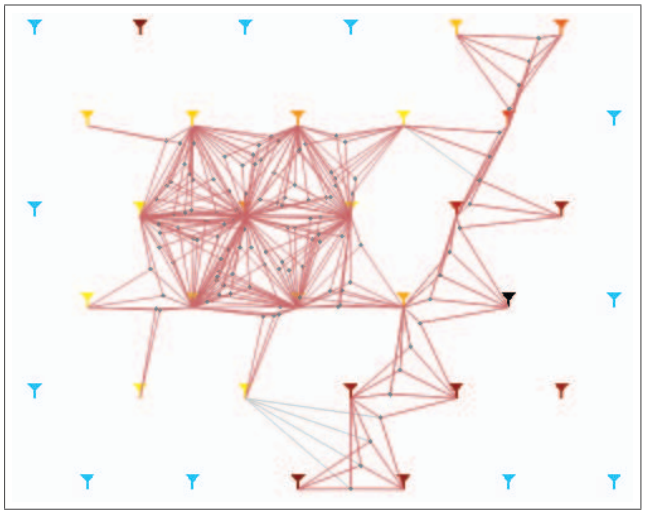

Fig. 1. A topology with users moving on a road near a populated area.

of those that he votes for. In Fig. 2 we can see that the algorithm indeed manages to lower the power consumption of the algorithm during the experiment. The initial state of the network is obviously using way more power than needed and in the first ∼200s the algorithm reduces the power usage of the network very fast. Between 200s and ∼750s the algorithm cannot decrease the consumption further without leaving users unsatisfied. After ∼750s no new users enter the simulation area while the ones already traveling on the road start leaving the area. For that reason the algorithm manages to further lower the power consumption by turning of the northern antennae. Finally we can see that during the experiment the algorithm manages to keep all users satisfied since the green line is constantly at 1.

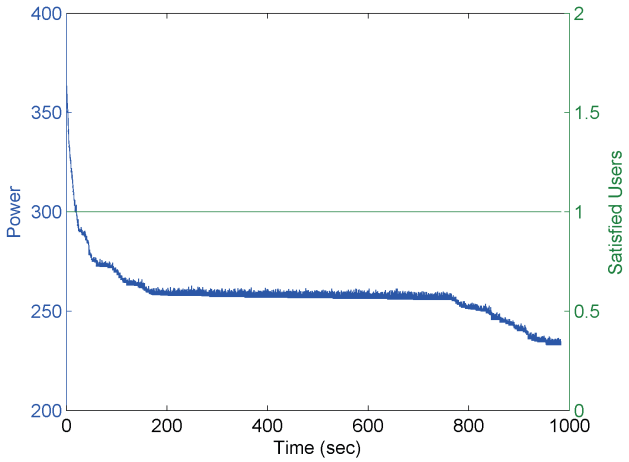

Fig. 2. Power and satisfied users as a function of time.

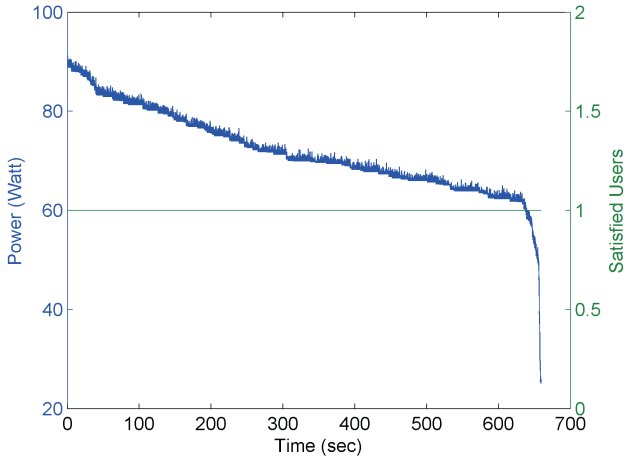

Fig. 3. Power and satisfied users as a function of time.

B. Random movement within confined area

In the next experiment we examine system's behavior for a confined area where low mobility users pass through. For this experiment we define an 1km^2 area containing 9 base station antennae, which are placed uniformly on a hex grid. Inside this area, 100 mobile users move in a random fashion. Their speed has a constant value of 3.5km/h. The figure for this topology is omitted since it is straight forward. The experiment lasts 10 minutes. In Fig. 3 we can see how the algorithm constantly lowers the power without even leaving unsatisfied users and when users start disappearing during the last seconds of the experiment it almost turns all MBSFN transmissions off.

C. Vehicular users in heavy traffic

In our last experiment we examine more closely the efficiency of the algorithm for users moving over a road. In this case the road has 3 lanes (each lane at a distance of 4 meters from the neighboring ones) and users move slowly. The experiment lasts 10 minutes. In this case the algorithm had trouble fully optimizing the network. The reason is that the initial state of the network is random and usually most cells transmit with more power than needed. Therefore during the first ∼20s the algorithm lowers rapidly the power of most cells and the network power drops from ∼400W to less than 70W. During that time a few users have just entered the network on the north-north-west side and the three cells closest to the entry point tend to stay on and serve those users. These are the three yellow antennae in Fig. 4. The mistake done by the algorithm here is that it starts decreasing power to every cell but those three so fast that the total power is lowered a lot even though it increases the power of those three cells way more than needed. This is easy to happen because these three cells are voted by users to increase power and compared to every other cell they are the only ones that tend to increase power while the whole network decreases. The result is that the algorithm sees an overall and very fast reduction of network power (optimization). This is fine until users start leaving the area of these three cells. The users start voting for new cells to increase power but new users entering the simulation area keep voting for the first three cells as seen in Fig. 5. It is hard for algorithm therefore to "realize" that those three cells must be decreased an their power would be more useful if it was transferred to another cell. We can see then in Fig. 6 that even when users have reached the other side of the simulated area the initial three cells are more powerful than needed and the even serve a few users away from them. The problematic situation is slowly improved between ∼350s and ∼700s. We can notice that this fact also causes some temporary disturbances in the satisfaction of users. Nevertheless, this situation is rapidly resolved when users stop entering the simulated area and no one votes for these three cells.

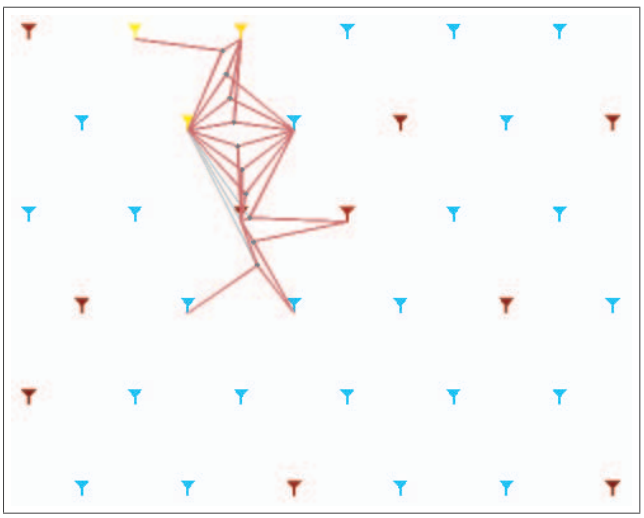

Fig. 4. Slow response when users leave the initial area.

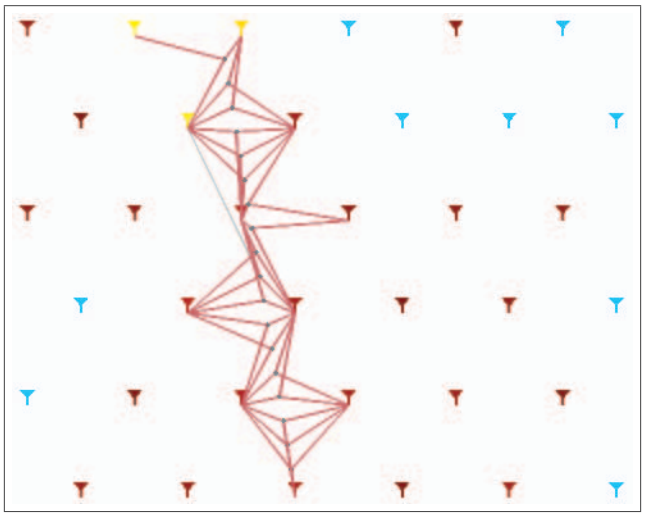

Fig. 5. Slow response when users leave the initial area.

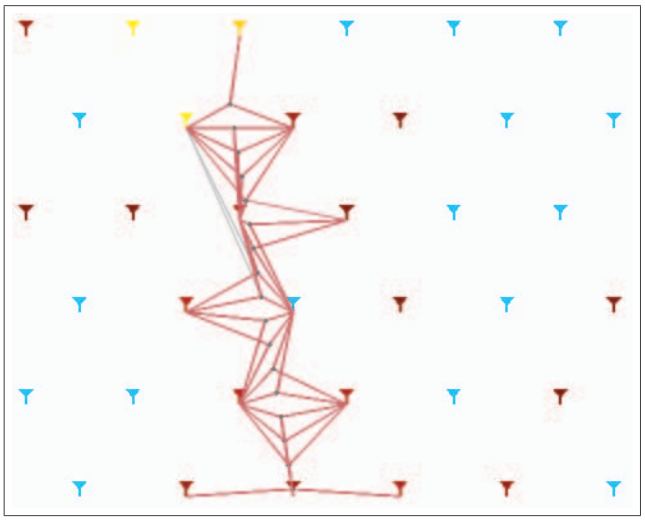

Fig. 6. Slow response when users leave the initial area.

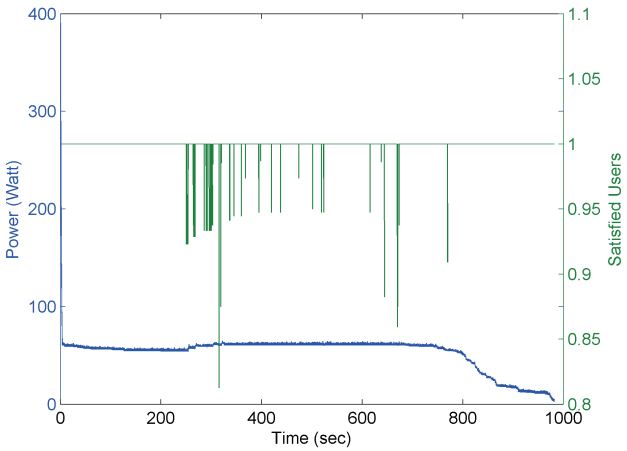

Fig. 7. Power and satisfied users as a function of time.

V. Conclusions & Future Work

The results showed that our algorithm can improve the power efficiency of an MBSFN network. This is achieved with the assumption that input information for the bitrate of each user is always available in an almost real time manner. A dilemma we had during the design of our algorithm was how "smart" the optimization should be. When we used a completely "dumb", brute force approach where mutations were completely random, the algorithm was practically too slow to be used in real system deployments. On the other hand, creating a very "smart" algorithm with mutations being more deterministic and based on an analysis of the network might make the results skewed. The reason for this is because we are probably hardcoding a bias in the algorithm based on what we consider a better solution. This could possibly make the algorithm look like it is actually doing a efficient work while it is missing the globally optimal solution, favoring other solutions that agree with the "smart" algorithm we made. For now we use a partially "smart" algorithm which, while faster than brute force, still uses almost random mutations.

More experiments can be conducted with different sampling rates for the bitrate to simulate more closely actual networks or the algorithm could be applied to a small test network. In the future XTNDL could be extended and used in different simulators. Being an extension of Lua, XTNDL could be easily embedded into existing simulators written in C/C++ or Java. A library to create agents that behave like humans by walking around or getting into a car and drive around could be created to provide even more realistic scenarios. Finally XTNDL could also include small extensions to alter the path loss of each user individually based on whether they are in a vehicle, inside a house or on the street, to further improve the simulation. Finally, the optimization part should be further improved to be faster in cases like the third experiment where the algorithm was almost trapped in a local optimum.

References

[1] D. H. Holma and D. A. Toskala, *LTE for UMTS - OFDMA and SC-FDMA Based Radio Access*. Wiley Publishing, 2009.

[2] 3GPP, "TS 36.300, v9.3.0, Technical specification group radio access network; evolved universal terrestrial radio access (E-UTRA) and evolved universal terrestrial radio access network (E-UTRAN); overall description; stage 2 (release 9)," 3rd Generation Partnership Project, Tech. Rep., 2010.

[3] ——, "TSG RAN WG1#47-bis R1-070051 Performance of MBMS transmission configurations," 3rd Generation Partnership Project, Tech. Rep., 2007.

[4] L. Rong, O. B. Haddada, and S.-E. Elayoubi, "Analytical analysis of the coverage of a MBSFN OFDMA network," in *IEEE Global Communications Conference (GLOBECOM'08)*, 2008, pp. 2388 – 2392.

[5] A. G. Alexiou, C. Bouras, V. Kokkinos, A. Papazois, and G. Tsichritzis, "Efficient MCS selection for MBSFN transmissions over LTE networks," in *Proc. of IFIP Wireless Days 2010 (WD'10)*, 2010. [Online]. Available: http://dx.doi.org/10.1109/WD.2010.5657749

[6] A. Alexiou, C. Bouras, V. Kokkinos, A. Papazois, and G. Tsichritzis, "Spectral efficiency performance of MBSFN-enabled LTE networks," in *Proc. of IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'10)*, Oct. 2010, pp. 361 – 367.

[7] A. Lopes, J. Seguro, P. Gomes, N. Souto, and A. Correia, "Interference coordination for E-MBMS transmissions in LTE-Advanced," *International Journal of Digital Multimedia Broadcasting*, 2010, Article ID 689705. [Online]. Available: http://dx.doi.org/10.1155/2010/689705

[8] A. Alexiou, K. Asimakis, C. Bouras, V. Kokkinos, and A. Papazois, "Combining MBSFN and PTM transmission schemes for resource efficiency in LTE networks," in *Proc. of 9th IFIP TC 6 International Conference on Wired/Wireless Internet Communications (WWIC'11)*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 56 – 67. [Online]. Available: http://dl.acm.org/citation.cfm?id=2023094.2023100

[9] K. Asimakis, C. Bouras, V. Kokkinos, and A. Papazois, "Genetic optimization for spectral efficient multicasting in LTE systems," in *Proc. of 5th joint IFIP Wireless and Mobile Networking Conference (WMNC'12)*, 2012, (to appear).

[10] "Distributed Accurate Graphical Coverage Optimizer (DAGraCO)." [Online]. Available: http://ru6.cti.gr/ru6/DAGraCO.zip

[11] "LOVE - Free 2D Game Engine." [Online]. Available: http://love2d.org/

[12] "The Programming Language Lua." [Online]. Available: http://www.lua.org/