A Graph Coloring Approach for Scheduling Undo Actions in Self-Organizing Networks

Tsvetko Tsvetkov*, Henning Sanneck†, and Georg Carle*
*Department of Computer Science, Technische Universität München
Email: {tsvetko.tsvetkov, carle}@in.tum.de
†Nokia, Munich, Germany
Email: henning.sanneck@nsn.com

***Abstract*—In a mobile Self-Organizing Network (SON) a coordinator is necessary to avoid the execution of conflicting SON function instances. Typically, such a coordinator bases its decision to accept or reject a network parameter change request on a rule set that considers only known conflicts. Moreover, it does not observe the impact of approved changes on the network. For this reason, SON verification approaches have been specified to assess the impact of deployed configuration changes and identify those that are causing an undesired network behavior. Similarly to anomaly detection techniques, a SON verification mechanism has a mathematical model that specifies how the network behavior should look like and defines any behavior that significantly deviates form the expectations as abnormal. Furthermore, the outcome is a corrective action, also called an undo action, that sets network parameters to some previous configuration.**

The question that often remains unanswered is how conflicting undo actions should be scheduled. A SON coordinator does not have the knowledge to resolve them and may, therefore, prevent such from being deployed. In this paper we present a scheduling approach of such undo actions that uses minimum graph coloring in order to identify the sets of cells whose configuration can be safely rolled back. Our evaluation is split in two parts. In the first part we highlight the importance of our approach by observing a real Long Term Evolution (LTE) network. The second part is based on simulation data in which we show the ability of our method to keep the performance of the network at a high level.

I. Introduction

Nowadays, mobile operators need to find an efficient way of managing the increasing complexity of their communication networks. The rapid adaptation of mobile services by users significantly increases the generated data volume, the amount of signaling in the network, and number of generated control events. Hence, Self-Organizing Network (SON) functionalities have been specified and developed to deal with the complex nature of network standards like Long Term Evolution (LTE) and LTE-Advanced. Usually, they are designed to optimize the operation of the network, supervise the configuration and auto-connectivity of newly deployed Network Elements (NEs), and are in addition to that responsible for fault detection and resolution [1].

A network enhanced with SON features is typically controlled by a set of autonomous functions performing specific Network Management (NM) tasks. These SON functions are designed as control loops which monitor Performance Management (PM) and Fault Management (FM) data, and based on their objectives adjust Configuration Management (CM) parameters. For example, the Mobility Load Balancing (MLB) function tries to move traffic from high loaded cells to neighbors as far as coverage and interference allows by optimizing the Cell Individual Offset (CIO) [2].

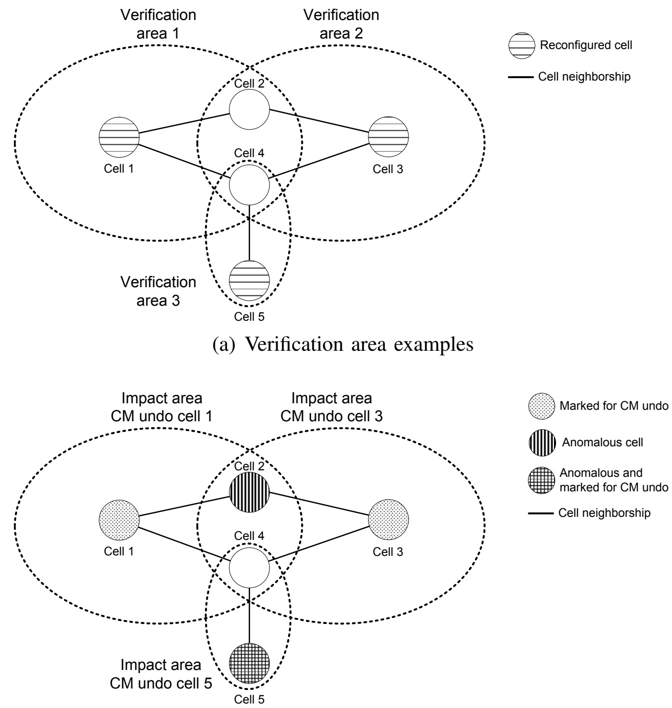
However, the increasing reliance on SON features to perform the correct optimization tasks introduces a new set of challenges. In a SON, the impact of each function's action on the environment depends upon the actions of other functions as well. For instance, if the Coverage and Capacity Optimization (CCO) function modifies the antenna tilt, the cell border changes physically which means that the received signal quality changes as well. Obviously, this affects the handover performance of the neighboring cells which is typically monitored by an optimization function like Mobility Robustness Optimization (MRO). Therefore, an inappropriate change of the physical cell borders induced by CCO may negatively impact the handover performance and, therefore, all upcoming decisions taken by the MRO function.

SON coordination can be considered as the first approach that has addressed these function dependencies. It defines rules used to avoid *known* conflicts between SON function instances. In literature, three conflicts classes have been proposed: (1) configuration, (2) measurement, and (3) characteristic conflicts [3]. The first type includes conflicts that occur when instances of SON functions operate on shared CM parameters. The second type addresses cases where the activity of one SON function instance affects the input measurements of another one. The third type deals with situations where two instances are in a direct conflict, e.g., both try to change the cell coverage area of two neighboring cells, or in a logical dependency, e.g., the above-mentioned CCO / MRO dependency. To prevent such conflicts, all running SON function instances are required to send a request to a SON coordinator before performing any changes to the network. The decision to accept or reject a request depends on whether another, higher prioritized and conflicting SON function instance has been recently active within the same area.

SON verification is a special type of anomaly detection. It aims at computing statistical measures on performance indicators at a relevant spatial and temporal aggregation level to assess the impact of a set of (SON-induced) CM changes. The verification process is a three step procedure comprising of (1) defining the scope, (2) running an anomaly detection algorithm, and (3) diagnosing the problem [4], [5]. During the first phase the verification area is computed which defines the set of cells that are being under assessment. During the second phase anomaly detection techniques are employed which may vary significantly in the underlying mathematical models and the assumptions about the data they are observing. For instance, in [6] performance indicator normalization is used to detect whether cells are showing an expected behavior or not. During the third phase root cause analysis is performed with a possible outcome of a corrective action. Typically, this action is a CM undo of the network parameters that have caused an undesired network behavior [7]. In addition, to improve the correctness of the diagnosis a scoring system can be used that rewards a corrective action if it has had a positive effect on the network [8].

The scheduling of CM undo actions is not a trivial and often underestimated task. If we let the SON coordinator handle this task, it may suppress undo actions that are in conflict with each other. This is caused by the fact that a coordinator does not have the knowledge to resolve such kind of conflicts. For example, if the verification mechanism has the desire to undo the antenna tilt change of two neighboring cells, one of them may get blocked because they are in a characteristic conflict. What we propose in this paper is a CM undo scheduling approach that builds upon graph coloring theory. Our method depicts the mobile network as a graph and applies minimum graph coloring in order to identify the set of cells whose CM settings can be safely undone.

The rest of the paper is organized as follows. In Section II we highlight the importance of the CM undo scheduling problem and show why it can occur in a real mobile network. In Section III we give an overview of the SON verification function we have developed. In Section IV is solely devoted to our new scheduling approach. In Section V we outline the results from our experimental case study which is based on real network as well as simulation data. Our paper concludes with the related work and a summary.



(a) Verification area examples

(b) Example of the coordination based CM undo approach

Figure 1. CM undo scheduling problem

II. Scheduling of CM Undo Actions

A. Background

Before going any further into the problem, we should at first get familiar of how SON functions are coordinated and how assembled CM configurations are deployed in a mobile network. In a SON, every function instance comes with two essential properties required for coordination: the impact area and the impact time. The impact area consists of the function area (set of cells that are configured by the instance), the input area (set of cells where the instance takes its measurements from), the effect area (set of cells that are possibly affected by the activity of the instance), and the safety margin (an extension to the impact area). The impact time is defined as the additional time interval after the execution time, during which a SON function instance needs to be considered to allow a successful conflict detection and prevention. Every time a SON function instance decides to change a network parameter, it contacts the SON coordinator by sending a CM change request. The latter one acknowledges the change only if there has not been another conflicting function activity for the given impact area and time.

SON verification approaches operate within verification areas, also sometimes called observation areas. In research, several approaches of how to specify them have been introduced. A common technique is to compute a verification area by taking the impact area of the SON function instance whose activity is being under assessment [4]. Furthermore, areas of dense traffic, difficult environments and known trouble spots can be considered during the selection process as well [9]. Another possible solution is to consider the cell neighbor relations, e.g., by taking the first degree neighbors of the reconfigured cell [7]. In a mobile network two cells are neighbors when they have a common coverage area so that a handover of User Equipments (UEs) can be made.

In case an anomaly is detected (e.g., degradation in performance) an undo request is sent to the SON coordinator. The impact area of the undo request equals the verification area since a coordinator has to prevent other functions from adjusting parameters for the area that is being under assessment. Note that in this paper we use the term *coordination based CM undo approach* to refer to this kind of workflow.

B. Problem Description

The coordination based CM undo approach has two major drawbacks which we are going to introduce by giving an example. Suppose that we have a network consisting of five cells, as shown in Figure 1(a). The neighbors of cell 1 as well as cell 3 are 2 and 4, and the neighbor of cell 5 is cell 4. For simplicity reasons, let us assume that a single CM parameter has been changed within cells 1, 3, and 5. If we compute the verification area by taking the reconfigured cell and the direct neighbors, and cells 2 and 5 start to show an anomalous behavior, we will have three overlapping undo requests as shown in Figure 1(b). The question that arises here is how we should schedule CM undo actions, especially when they are in conflict with each other. In addition, how should we treat *verification collisions*, i.e., situations in which there is an equivocality when scheduling several CM undo requests at the same time, like those of cell 1 and 3?

C. Solutions

One possible solution is to follow an aggressive approach by undoing all changes. The main disadvantage of this approach is the treatment of verification collisions. We may undo a change that was required and did not harm performance, e.g., the change made within cell 3 in Figure 1(b) might not be the cause for the anomalous behavior of cell 2.

In contrast, a conservative strategy would perform a stepwise undo of the overlapping areas. If we take the simplified scenario from above, it would mean that we first undo the CM change of cell 5, then undo the one of cell 1, and if required proceed with cell 3. Such an approach may work perfectly fine when we have a small number of overlapping verification areas and few active SON function instances. However, this changes as soon as those conditions are no longer met. Every time we undo a CM parameter, other SON functions instances may get active. For example, if we undo a tilt change the MRO function running on a neighboring cell might get active to adapt the handover parameters. As a consequence, such an activity will interfere which may prevent the verification process from achieving its goal.

A possible way to improve the latter approach is to block the areas that are being under verification until all required CM undo actions are executed. For this purpose we require a SON coordinator to prevent other function instances from performing any changes for those areas. However, this might not be always a suitable solution for real world scenarios since the verification area may comprise of more than hundred cells if we just take the direct neighbors of a single cell [7].

D. Causes

There are three major reasons why we may have overlapping undo requests and verification collisions. The first reason is the *location of the verification mechanism*. In order to have a wide view on the mobile network and the running SON function instances, it resides at the Domain Management (DM) or even the NM level of the Operation, Administration and Management (OAM) architecture. However, being at that level prevents a verification mechanism from being able to instantly verify the action of every running SON function instance in the network. To do so, it would require frequent transfer of data from the NE to the DM/NM level which would consume OAM bandwidth and induce additional delays within the network of the operator.

The second reason is the *high number of network cell adjacencies*, i.e., cell neighborships where a handover can potentially occur. In Figure 2 we have outlined the cell out-degree distribution of an LTE network consisting of 3028 cells. As it can be seen, cells in a mobile network tend to have a high number of neighbors which is due to several reasons. As in every new technology, the size of a cell shrinks which as a result leads more neighboring cells to appear, including those from other technologies towards which handovers are also possible. Furthermore, a high number of neighbor relations can be caused by an increased cell density.

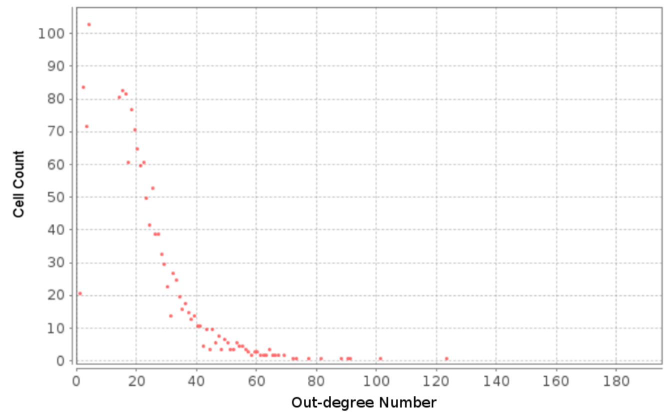


Figure 2. Cell out-degree distribution in an LTE network

The third reason is *offline optimization*, i.e., waiting until all required data is collected out of the network, running the optimization algorithm, and manually deploying the resulting CM change sets on the network. In Figure 3 we have outlined the number of CM parameter modifications that have occurred over a time period of three weeks. The shown statistics derive from the same LTE network. As the figure shows, the operator has performed numerous optimization changes, including a large number of cell adjacency adjustments. In a network with a high cell out-degree this creates a potential for having many overlapping verification areas.

III. The SON Verification Function

In a previous work of ours [4], we have proposed a SON verification approach that is tightly integrated with SON coordination. We designed it as a SON function that analyzes the network performance for acknowledged action request of SON function instances. In case an undesired network behavior is detected, for example, caused by the activity of a given function, it requests permission to undo the responsible CM changes from the SON coordinator for the affected area. To achieve its task the SON verification function makes use of four helper functions: (1) an anomaly level, (2) a cell level, (3) an area resolver, and (4) an area analyzer function.

The anomaly level function is designed to differentiate between normal and abnormal cell Key Performance Indicator (KPI) values. Typical KPIs are the number of radio link failures, Handover Success Rate (HOSR), the Channel Quality Indicator (CQI) and so on. The output is a *KPI anomaly level* which depicts the deviation of a KPI from its expectation. In order to compute it, we have defined a verification training phase during which we collect samples $X_1 \dots X_t$ for each KPI, where t marks a training period. During this phase the network has to show an expected behavior. Furthermore, the duration of a period depends on the granularity for gathering PM data from the network. For instance, it can correspond to an hour if KPIs are exported on hourly basis as presented in [7]. Then, we standardize the gathered data by computing the z-score of each data point $X_1 \dots X_t, X_{t+1}$. Here, X_{t+1} corresponds to the current sample that we want to evaluate. The anomaly level of a KPI corresponds to the z-score of X_{t+1}.

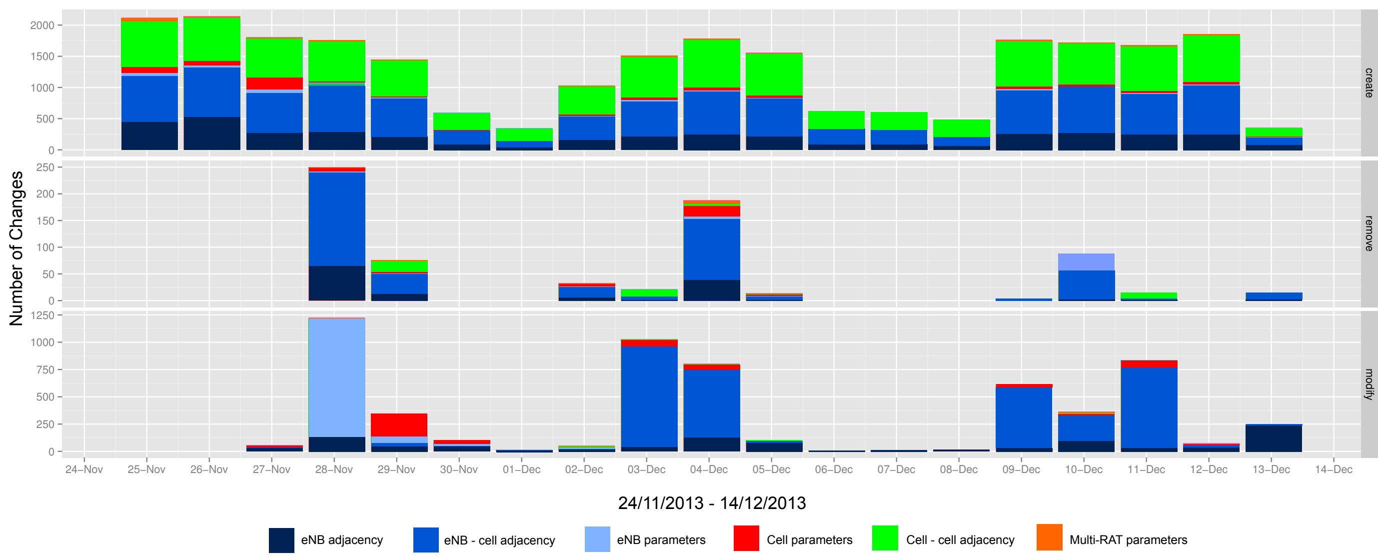


Figure 3. Typical LTE parameter modification scenario

The cell level function creates an overall performance metric of individual cells. The output is the sum of the weighted KPI anomaly levels which we have named the *cell level*. The ability to change those weighting factors allows us to test a cell for different anomaly types. For example, we may take only handover related KPI anomaly levels into consideration when we assess the changes made by the MRO function.

The area resolver function computes the verification area. It consists of a set Σ_b that includes the cells that have been reconfigured by a SON function instance and a set of cells Σ_e that have been influenced by that reconfiguration process. We call Σ_b the *CM change base* and Σ_e the *CM change extension area*. The union $\Sigma_b \cup \Sigma_e$ composes the verification area. The computation itself is based on the impact area of the SON function instance that has made the change. The set Σ_b equals the function area, the cells that are most prone for experiencing anomalies. The extension Σ_e consists of cells included in the effect area and the safety margin. The motivation of taking the effect area into account is because it has all cells that are supposed to experience side-effects after the execution of a function instance. For instance, if CCO changes the transmission power of a cell, the load of a neighboring cell (that is part of the effect area) may change as well. The idea behind taking the safety margin is that the effect area can differ from its original specification. The safety margin extends the border of the impact area which should provide a higher degree of protection against undesired effects. For example, due to an increased network density the effect area can be much larger than it has been initially assumed which as a result may require the second degree neighbors of the reconfigured cell to be considered as well.

The purpose of the area analyzer function is determine whether an area shows a significant divination from the expected cell level. Furthermore, it is responsible for generating a CM undo of the CM change base and sending it to the SON coordinator.

IV. Our CM undo scheduling approach

In this section we present our CM undo scheduling approach which operates in three phases. An example of applying it is also given.

A. Verification Graph Construction

First of all, we construct an undirected graph $G = (V, E)$ which comprises of a set V of verification areas and a set E of verification edges. How such areas are defined has been outlined in Section II. Verification edges are added by applying dependency function d from Equation 1.

$$d\colon V \times V \rightarrow E \cup \{\varnothing\} \tag{1}$$

The criteria based on which d adds an edge between two areas are verification collisions. If we denote the set of all cells as Σ and the set of all anomalous cells as Σ_a, where $\Sigma_a \subseteq \Sigma \cup \{\varnothing\}$, an edge (v_i, v_j) is added only when $f_e(v_i) \cap f_e(v_j) \subseteq \Sigma_a$ for all i and j. Note that f_e is an extraction function that returns the cells of a verification area, i.e., $f_e\colon V \rightarrow \mathcal{P}(\Sigma) \setminus \{\varnothing\}$.

B. Identification of Collision Free Undos

In the next step, we determine the nodes that are *collision free*. To do so, we perform minimum vertex coloring on the verification graph G which assigns each vertex a color such that no edge connects two vertexes having the same color. Formally, we apply the map function m from Equation 2, where C is the set of available colors and $|C| = |V|$. The algorithm we are using is minimal vertex coloring with backtracking, as defined in [10].

$$m : V \rightarrow C \subseteq \mathbb{N}_0 \tag{2}$$

Verification areas that have been colored with the same color are collision free and any CM undo operations within them can be safely executed. In addition, the smallest number of colors required to color G, also known as the chromatic number $\chi(G)$, equals the *verification collision grade* of the verification graph. It shows the optimal number of slots that are required to execute all collision free CM undo operations.

Furthermore, we call G *collision complete* if $m(v_i) \neq m(v_j)$ and *collision free* if $m(v_i) = m(v_j)$ for all adjacent v_i and v_j.

C. Scheduling of Collision Free Undos

After applying the coloring function m, i.e., assigning each verification area a positive integer, we have to distinguish between the following three outcomes:

- $\chi(G) = 1$
- $\chi(G) = |C|$
- $\chi(G) \in [2; |C|)$ for $|V| > 2$

In the first case G is marked as being collision free, which means that our method resembles the aggressive approach introduced in Section II-C. The second case implies that every verification node has a different color (under the assumption that we have at least two verification nodes), which as a result marks G as collision complete. Consequently, we start processing the one having the highest number of anomalous cells. In the third case, we see the frequency of the used colors as the main criteria when defining the execution order, i.e., the nodes having the most frequently used color are scheduled at first place.

D. Example

The network consists of 12 cells and 18 cell adjacencies, as shown in Figure 4. Furthermore, three CM changes have been deployed: CIO modification within cell 7, and an antenna tilt change within cell 4 and 10. Because of those changes, cells 5 and 10 begin to experience a degradation in performance. If we compute the verification area by taking the reconfigured cell and its first degree neighbors, three areas (v_1, v_2, and v_3) are added to V. Since areas v_1 and v_2 share anomalous cells, the verification edge (v_1, v_2) is added to the set E. After applying minimum vertex coloring two of the areas, namely v_1 and v_3, get the most frequently used color. As a consequence the CM undos of cell 4 and 10 are marked as collision free and executed. The verification collision grade equals to 2.

V. Evaluation

The evaluation of our concept is based on real network as well as simulation data. In the real data evaluation we show an example of conflicting CM undo actions caused by a verification collision. As for the simulation part, we show the impact of the whole closed-loop verification process by observing the effect of executing the most frequently colored collision free CM undo requests.

A. Real-data based

A crucial parameter in today's mobile LTE networks is the Physical Cell Identity (PCI). It is a low level identifier broadcasted by a cell in the System Information Block (SIB) [11], [12]. The value of a PCI can range between 0 and 503, and is used as a primary identifier for handover procedures, initiated based on the PCI reported by the UE. To

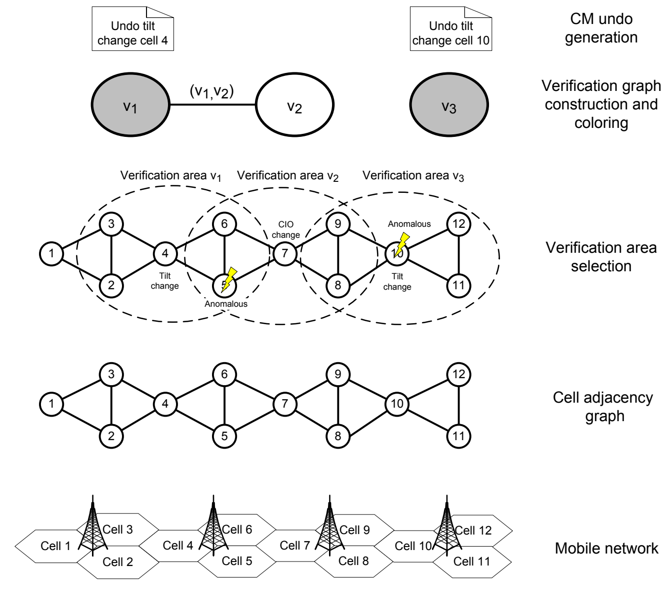


Figure 4. Example of our CM undo scheduling approach

have a successful handover, the PCI allocation procedure has to make sure that the network is PCI collision and confusion free. A PCI collision occurs when two adjacent cells share the same identifier. A confusion on the other side happens when a cell has two neighbors with the same PCI value. An improper PCI allocation can occur due to several reasons. For instance, in cell outage management a common approach to close a coverage hole is to extend the coverage area of the surrounding cells. However, by doing so cells that were not assigned to be neighbors may start to have a common coverage area [13]. Another reason that can have the same negative impact can be the deployment of new cells in the network.

This particular CM parameter is of high interest for us since an improper PCI allocation has typically a negative impact on more than just one cell which as a consequence may lead to a verification collision. If the allocation procedure fails and assigns a cell a PCI value that is already used by one of its neighbors, both cells will start to experience a handover performance drop as they try to move UEs between each other. If at the same time a second degree neighbor of the reconfigured cell is optimized, for example, by the MRO function, a verification mechanism that constructs the verification area by taking the adjusted cell and its direct neighbors will not be able to tell which of the two changes is responsible for the performance drop. A similar observation can be made when we face a PCI confusion. If one cell suddenly starts to see two neighbors having the PCI value, its handover performance may start to suffer even though it has not been directly reconfigured. As a result, any CM change occurring at the same time in the neighborhood of the cell can potentially be blamed for that.

Let us take a closer look at the results shown in Figure 3 and observe whether verification collisions can occur in a real network. During the whole three week observation period, we monitored a high number cell adjacencies adjustments triggered over the whole LTE network. The main reason for this to happen was the introduction of 400 new cells by the

Table I. SON SIMULATION SYSTEM SETUP

Component	Parameter	Value
LTE simulator	Network Frequency	2000 MHz
	Bandwidth	20 MHz
	Total cells	32
	Simulated area	50 km^2
	User speed	6 km/h
	Handover hysteresis threshold	2.0 dB
	Radio Link Failure threshold	-6.0 dB.
	Constant bit rate requirement	256 kbps
	Radio propagation model	Signal to interference plus noise
SON Function Engine	Active SON functions	SON verification function, MRO, TXP, RET
SON coordinator	Initial priority setup	SON verification function $>$ RET $>$ TXP $>$ MRO
	Tokens	10 for each bucket
	Token change for Acks	-2 tokens
	Token change for Nacks	+1 token
SON verification function	Training rounds	70
	Cell level degradation range	$(\infty; -2.0]$
	Cell level weights	0.5 for CQI and HOSR

operator. In one particular case, a PCI collision was detected which led a cell to show an anomalous behavior. The cell reported a degradation of the inter Evolved NodeB (eNB) handover and the radio resource control connection setup success rate. In addition, it was also experiencing an unusual high rate of radio link control Protocol Data Units (PDUs) retransmissions. As a result, the verification mechanism was triggered to observe the affected scope. The cell that was showing an anomalous behavior was not the cell what was newly added, but a direct neighbor, i.e., the anomalous cell was part of the CM change extension area of the corresponding verification area. Furthermore, we were able to spot at the same time another CM change at a neighbor of the anomalous cell which led to the creation of a second verification area. As a consequence, we had two overlapping verification areas, sharing the same anomalous cell whose CM settings were not changed.

B. Simulation based

Environment: The SON Simulation System (S3) consists of an LTE radio network simulator, a SON Function Engine (SFE), and a SON coordinator [4]. The parameter setup of those components is outlined in Table I.

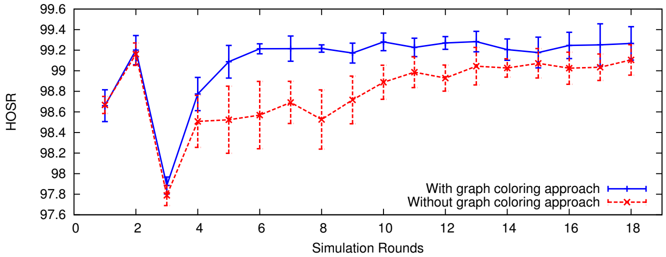
The LTE simulator, as part of the SON simulator/emulator suite [14], performs continuous simulation by tracking the network changes over time. The time is partitioned into time slices, called simulation rounds, and the state of the network is updated according to the set of activities that have occurred in a time slice. A simulation round corresponds to 100 minutes in real time. At the beginning of a round, the simulator configures the network as defined by the CM parameter setup. During a round, 1500 uniformly distributed users follow a random walk mobility model and actively use the mobile network. At the end of a round, PM data is exported for every cell . The simulated scenario covers the area around ”Louhenpuisto” park located in Helsinki (Finland).

The SFE is a runtime environment for SON functions that handles their communication and configuration. Every time the LTE network simulator completes a round, the SFE triggers the monitoring phase of all SON functions. The SFE controls the MRO, Remote Electrical Tilt (RET), and Transmission Power (TXP) optimization function, as defined in [1]. Furthermore, it includes the SON verification function as described in Section III. CM change request generated by those functions are forwarded to the SON coordinator.

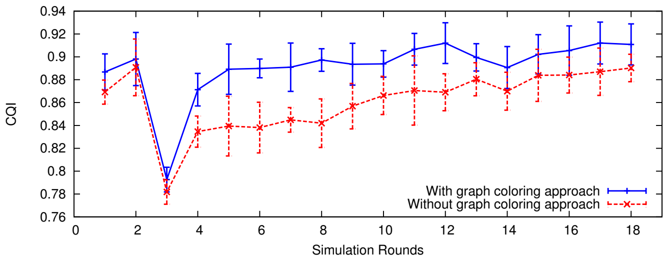
The SON coordinator performs pre-action coordination by employing the batch coordination concept with dynamic priorities, as defined in [15]. Each SON function instance has an assigned bucket and dynamic priority. The bucket of an instance contains a number of tokens that are reduced every time a request is accepted and increased if a request is rejected. In case of an empty bucket, the priority is set to minimum. It gets increased if requests begin to get rejected.

For our SON verification function a sampling period equals a simulation round. The training data is gathered during a separate test run where optimal network settings are used. In addition, the KPIs we consider for computing the cell level are the HOSR and CQI. The CQI is computed as the weighted harmonic mean of the CQI channel efficiency. The efficiency values are defined in [16].

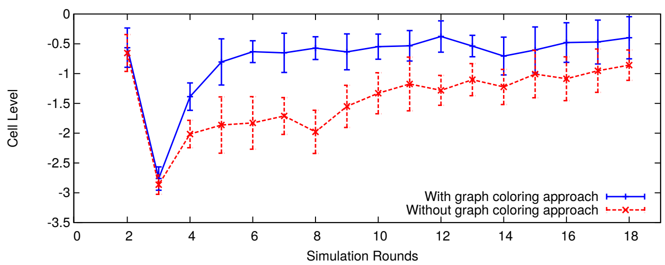
Results: In the first part of the simulation experiment we want to recreate a similar parameter modification deployment as discussed above. However, the purpose of this case study is not only the detection of an anomaly and figuring out the CM change responsible for this to happen, but also to observe how the network actually behaves when we execute the CM changes suggested by our CM undo scheduling approach. Furthermore, we are interested of how our new concept compares to the co-ordination based CM undo approach, which we have described in Section II. For this purpose we selected four eNBs (three sectors for each eNB) surrounding the park and performed offline coverage optimization. However, we used obsolete data during the optimization phase which can actually happen in a real mobile network. As stated in [1], when the environment changes from the assumptions made when the network was planed and set up, the coverage of the system can be reduced to what we could achieve with optimal settings. They can occur due to new or demolished buildings, insertion or deletion of base stations, and season changes.



(a) Handover success rate

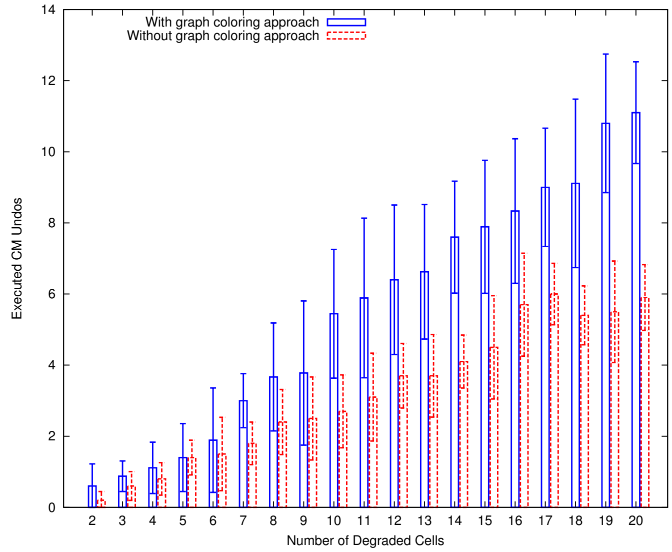


(b) Channel Quality Indicator

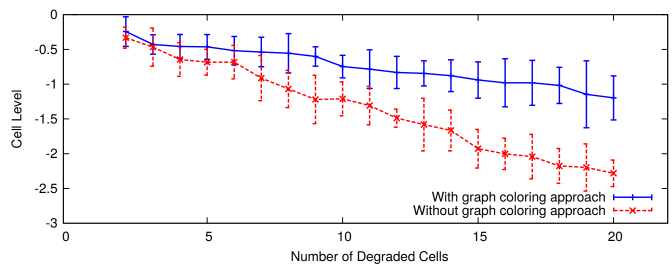


(c) Cell level

Figure 5. Part I of the simulation experiment



(a) Number of executed CM undo requests



(b) Correlation between the number of executed CM undos and the cell level

Figure 6. Part II of the simulation experiment

The results of this case study are outlined in Figure 5. Figures 5(a) to 5(c) show the handover success rate, the CQI, and the resulting cell level of all four eNBs. The shown 99 % confidence intervals are computed around the sample mean of five consecutive test runs. Note that a single test runs lasts 18 rounds which corresponds of a simulated time of approximately 30 hours. We applied the new transmission power and antenna tilt settings in simulation round 2 and observed the resulting impact on the network. As the results depict, our CM undo scheduling approach requires two simulation rounds to return the observed network area to the expected cell level. The coordination based CM undo approach on the other side does not manage to do that. Our observations show that CM undo actions have been suppressed due to overlapping verification areas. What is even more interesting here is that the SON system is not able to completely return the cell level as reported before round 2. The exported configuration data indicates that this is due to the dynamic coordination mechanism. The coordinator simply starts to reject the requests of a SON function instance if it has frequently been executed. In this particular case, the RET function is blocked as it tried to adjust the antenna tilt so that the TXP function can get its turn. The same happens to TXP after a certain period of time.

In the second part of the simulation experiment we study how the number of CM undo actions varies when the number of degraded cells changes. The number of degraded cells ranges between 2 and 20. Furthermore, the cells marked for degradation are selected based on an uniform distribution. The degradation itself is done by deploying a cell configuration that is unusual for the used network setup: a transmission power of 42 dBm and an antenna tilt of 3 degrees. Our results show that when we have a low number of degraded cells in the given network (up to six cells) the advantage of employing our approach is minor. This, however, changes when at least seven degraded cells, as it can be seen in Figure 6(a). As we increase the number of degraded cells, conflicting CM undo actions start to occur more often. As a consequence, the coordination based CM undo approach starts to suppress them more frequently because of the reasons outlined in Section II. Our strategy on the other side lets more such actions through which also has a positive impact on the cell level as it can be seen in Figure 6(b). Note that the presented confidence intervals are computed in the same way as described above.

VI. Related Work

The concept of pre-action SON coordination defines rules used to anticipate and avoid known conflicts between SON function instances. In addition, the idea of verifying a CM change triggered by a SON function instance and rolling it back based on certain rules has been introduced in [15]. Such rules, however, are defined with regard to SON coordination, i.e., only the priorities of the SON function instances are taken into consideration. As a result, an accepted CM change made by a given SON function instance is rolled back only if another higher prioritized and conflicting instance triggers a CM change request within the same area and time.

In [17] an anomaly detection and diagnosis framework for mobile communication systems is proposed. The authors have developed a framework that analyses performance indicators measured by a NE, observes them for anomalous behavior and suggests a corrective action to the operator. Typical counters are the number of successful circuit or packet switched calls. The suggested system consists of three main building blocks: a profile learning, an anomaly detection and a diagnosis module. The profile learning module analyzes historical data and learns all possible realizations of normal network operation. These realizations are represented by profiles which describe the usual (faultless) behavior of a KPI. The anomaly detection module monitors the current network performance and compares it to the profiles. Should a significant difference be detected the diagnosis module is contacted. Based on a knowledge database populated with fault cases by the operator, it tries to identify the possible cause. Furthermore, a performance report containing the suggested corrective action is provided to the operator who can optionally provide feedback to the system so the underlying models of the system get improved.

Graph coloring itself is a known method in mobile networks. In [13] it is used for PCI allocation where the set of colors represents the number of available PCI values. As we have already mentioned the assignment has to be confusion and conflict free, i.e., there is no cell in the network that has two or more neighbors with identical PCIs, and there are no two neighboring cells that have the same PCI. The authors have proposed a graph coloring approach where the vertexes represent the cells in the network. In addition, for any two neighboring cells an edge is added to the graph which fulfills the collision free requirement. In order to satisfy the confusion free requirement, an edge is added for every two neighboring cells of second degree.

In [18] another example of using graph coloring is given. The authors employ this technique for the assignment of frequencies for wireless LAN by constructing a so-called interference graph. The vertexes of the graph represent the wireless access points whereas the edges connect two access point that would interfere with each other. In addition, a set of maximum possible colors is defined by collecting the number of channels available to the access points.

VII. Conclusion

In a mobile Self-Organizing Network (SON), SON functions are actively analyzing Performance Management (PM) and Fault Management (FM), and based on their objectives are performing changes to one or more Configuration Management (CM) parameters. Such changes, however, might be in conflict with each other if we deploy them at the same time on the network. For example, one function may try to change the antenna tilt within a cell as another one adjusts handover parameters between the cell and its neighbors. For this reason, the SON coordination concept has been proposed to resolve known conflicts between SON function instances. SON verification mechanisms have been developed to monitor the network (or certain parts of it) and trigger an undo action for the parameters that have caused an anomaly like a degradation in performance. Two CM undo requests, however, can be in conflict with each other which means that we need a scheduling strategy. Relying solely on a SON coordinator is not feasible since it is does not have the knowledge to resolve unknown conflicting actions.

Instead of relying on a SON coordinator to figure out the right way to deploy CM undo actions, we have developed an alternative approach that is based on graph coloring theory. Since verification approaches operate on verification areas (set of cells that are being under assessment) we decided to define them as vertexes in our verification graph. Two vertexes are connected when they are in a verification collision, i.e. , there is an equivocality when we try to schedule an undo for both cell sets. The nodes that have the most frequently used color are marked as collision free and scheduled for an undo.

The evaluation of our concept is twofold. On the one side, we show why overlapping CM undo actions and verification collisions can occur in a real Long Term Evolution (LTE) network. On the other side, we compare our method with the coordination based CM undo approach in a simulated environment based on a real network setup. The results show that our approach is able to let more CM undo action through and return the network to the expected performance state.

Our future work will be devoted to the specification of verification areas. Furthermore, of particular interest would be also the resolving of verification collisions as well adding retraining capabilities to our SON verification function.

Acknowledgment

The authors would like to thank Szabolcs Nováczki (Nokia, Budapest, Hungary) for the data analysis support. Furthermore, the authors would like to thank Janne Ali-Tolppa (Nokia, Munich, Germany) and Jürgen Goerge (Nokia, Munich, Germany) for the valuable feedback.

References

[1] S. Hämäläinen, H. Sanneck, and C. Sartori, Eds., *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Chichester, UK: John Wiley & Sons, Dec. 2011.

[2] T. Yamamoto, T. Komine, and S. Konishi, "Mobility Load Balancing Scheme based on Cell Reselection," in *The 8th International Conference on Wireless and Mobile Communications (ICWMC 2012)*, Venice, Italy, Jun. 2012.

[3] T. Bandh, "Coordination of autonomic function execution in Self-Organizing Networks," PhD Thesis, Technische Universität München, Apr. 2013.

[4] T. Tsvetkov, S. Nováczki, H. Sanneck, and G. Carle, "A Post-Action Verification Approach for Automatic Configuration Parameter Changes in Self-Organizing Networks," in *6th International Conference on Mobile Networks and Management (MONAMI 2014)*, Würzburg, Germany, Sep. 2014.

[5] G. Ciocarlie, C.-C. Cheng, C. Connolly, U. Lindqvist, S. Nováczi, H. Sanneck, and M. N. ul Islam, "Managing Scope Changes for Cellular Network-level Anomaly Detection," in *International Workshop on Self-Organizing Networks (IWSON 2014)*, Barcelona, Spain, Aug. 2014.

[6] P. Kumpulainen, M. Särkioja, M. Kylväjä, and K. Hätönen, "Finding 3G Mobile Network Cells with Similar Radio Interface Quality Problems," in *Engineering Applications of Neural Networks, L. Iliadis and C. Jayne, Eds. Springer Berlin Heidelberg*, 2011, pp. 392–401.

[7] T. Tsvetkov, S. Nováczki, H. Sanneck, and G. Carle, "A Configuration Management Assessment Method for SON Verification," in *International Workshop on Self-Organizing Networks (IWSON 2014)*, Barcelona, Spain, Aug. 2014.

[8] P. Szilágyi and S. Nováczki, “An Automatic Detection and Diagnosis Framework for Mobile Communication Systems,” *IEEE Trans. Netw. Serv. Manag.*, vol. 9, no. 2, pp. 184–197, Jun. 2012.

[9] Ericsson, “Transparent Network-Performance Verification For LTE Rollouts,” White Paper, 284 23-3179 Uen, Sep. 2012.

[10] E. W. Weisstein, “Minimum Vertex Coloring From MathWorld – A Wolfram Web Resource,” http://mathworld.wolfram.com/MinimumVertexColoring.html, Jul. 2014.

[11] P. Szilágyi, T. Bandh, and H. Sanneck, “Physical Cell ID Allocation in Multi-layer, Multi-vendor LTE Networks,” in *4th International Conference on Mobile Networks and Management (MONAMI 2012)*, Hamburg, Germany, Sep. 2012.

[12] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification.” 3rd Generation Partnership Project (3GPP), Technical Specification 36.331 V10, Mar. 2012.

[13] T. Bandh, R. Romeikat, H. Sanneck, L. C. Schmelz, B. Bauer, and G. Carle., “Optimized Network Configuration Parameter Assignment Based on Graph Coloring,” in *Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS 2010)*, Osaka, Japan, Apr. 2010.

[14] NSN, “Self-Organizing Network (SON): Introducing the Nokia Siemens Networks SON Suite - an efficient, future-proof platform for SON,” White Paper, Oct. 2009.

[15] R. Romeikat, H. Sanneck, and T. Bandh, “Efficient , Dynamic Coordination of Request Batches in C-SON Systems,” in *IEEE Veh. Technol. Conf. (VTC Spring 2013)*, Dresden, Germany, Jun. 2013.

[16] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures,” 3rd Generation Partnership Project (3GPP), Technical Specification 36.213 V12.1.0, Mar. 2014.

[17] S. Nováczki, “An Improved Anomaly Detection and Diagnosis Framework for Mobile Network Operators,” in *9th International Conference on Design of Reliable Communication Networks (DRCN 2013)*, Mar. 2013.

[18] J. Riihijarvi, M. Petrova, and P. Mahonen, “Frequency allocation for WLANs using graph colouring techniques,” in *Wireless On-demand Network Systems and Services, 2005 (WONS 2005)*, St. Moritz, Switzerland, Jan. 2005, pp. 216–222.