A Framework for Classification of Self-Organising Network Conflicts and Coordination Algorithms

Hafiz Yasar Lateef, *Member IEEE*, Ali Imran, *Member IEEE*, Adnan Abu-dayya, *Senior Member IEEE*
Qatar Mobility Innovations Center (QMIC), Qatar Science & Technology Park (QSTP), Doha, Qatar
Email: {yasarl, alii, adnan}@qmic.com

***Abstract*—The next generation Long Term Evolution (LTE) & LTE-Advanced cellular networks will be equipped with numerous Self-Organizing (SO) functions. These SO functions are being envisioned to be inevitable for technical as well as commercial viability of LTE/LTE-Advanced networks. Therefore, a lot of research effort is currently being channeled to the design of various SO functions. However, given the convoluted and complex interrelationships among cellular system design and operational parameters, a large number of these SO functions are highly susceptible to parametric or logical inter-dependencies. These inter-dependencies can induce various types of conflicts among them, thereby undermining the smooth and optimal network operation. Therefore, an implicit or explicit self-coordination framework is essential, not only to avoid potential objective or parametric conflicts among SO functions, but also to ensure the stable operation of wireless networks. In this paper we present such a self-coordination framework. Our framework builds on the comprehensive identification and classification of potential conflicts that are possible among the major SO functions envisioned by Third Generation Partnership Project (3GPP) so far. This classification is achieved by analyzing network topology mutation, temporal and spatial scopes, parametric dependencies, and logical relations that can affect the operation of SO functions in reality. We also outline a solution approach for a conflict-free implementation of multiple SO functions in LTE/LTE-Advanced networks. Moreover, as an example, we highlight future research challenges for optimum design of Mobility Load Balancing (MLB) and Mobility Robustness Optimisation (MRO).**

***Index Terms*—Self-Organising Networks (SON), LTE/LTE-Advanced, Self-Coordination, SON conflicts, Mobility Robustness Optimisation (MRO), Mobility Load Balancing (MLB)**

I. INTRODUCTION

WIRELESS network management by human operators is not only time consuming but also inefficient and expensive. For Long Term Evolution (LTE) networks, proposed by the standardisation body Third Generation Partnership Project (3GPP), operators demand a significant reduction of manual effort in network deployment, configuration and maintenance stages in order to reduce their operational expenditures (OPEX). For providing high quality services in future wireless networks and to meet the operators' OPEX requirements, automation of wireless network management is crucial. A possible solution for achieving automation in wireless network management is the introduction of Self-Organising functionalities into LTE [1], [2]. Self-Organising Network (SON) solution for wireless network management is endorsed by both wireless operator consortium Next Generation Mobile Networks (NGMN) and 3GPP in numerous technical specifications [3], [4].

Most of the previous work on SON was focused on designing standalone Self-Organising (SO) functions. However, Self-Coordination is essential in those situations where SO function actions impact upon other SO functions in such a way that the originally intended objective of any one of the SO functions is affected. As a result, the network performance may be different from what was intended by the operator. Hence, Self-Coordination, in conjunction with Self-Optimisation is the key to achieving operators' overall objectives on network performance.

A brief description of conflicts between control parameters of certain SO functions is given in [5]. Some basic inter-related parameters are arranged into functional parameter groups in [6]. Some initial challenges on parameter, characteristics and measurement conflicts when integrating SO functions into future wireless networks are described in [7]. However, neither of these papers cover key SO function conflicts especially regarding network nodes mutation, performance metrics, cell outage, measurement and interference perspectives. Two basic policies for base station Self-Configuration and Coverage and Capacity Optimisation (CCO) are described in [8]. However, the policies presented in [8] do not seem to be generalizable to key optimisation aspects, such as handover and load balancing. Some coordination mechanisms are described in [9]. However, [9] does not identify the application of these coordination mechanisms for specific SO function conflicts. To sum up, to the best of the authors' knowledge, a comprehensive framework for classification of SO function conflicts and coordination has never been proposed before in the literature.

In this paper we identify, annotate and categorise SO function conflicts on the basis of network topology mutation, Key Performance Indicators (KPIs), output parameters' direction/magnitude, measurement and logical dependency. Moreover, we propose Trigger-Condition-Action (TCA) policies of Mobility Robustness Optimisation (MRO) and Mobility Load Balancing (MLB) functions for possible evolution of Self-Coordination. We also identify optimum ways of interaction between Self-Coordination and Self-Optimisation.

This paper is organised as follows. A comprehensive classification of SON conflicts is presented in Section 2. A framework for Self-Coordination evolution is proposed in Section 3. The optimum ways of interaction between Self-Coordination and Self-Optimisation are proposed in Section 4. Conclusion and future work is presented in Section 5.

Notation: ANR: Automatic Neighbour Relation, AP: Absolute Priorities, BB: Basic Biasing, CCO: Coverage and

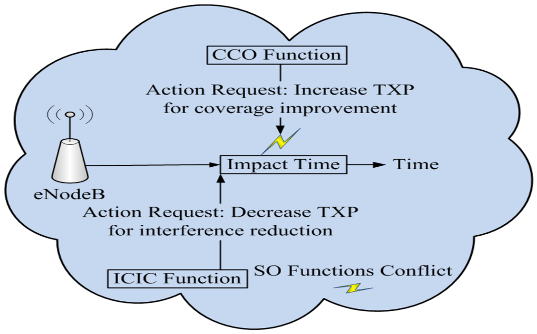

Figure 1. Conflict between CCO & ICIC SO functions.

Capacity Optimisation, CIO: Cell Individual Offset, COC: Cell Outage Compensation, EE: Energy Efficiency, eNB: Evolved Node B, HeNB: Home eNB, ICIC: Inter-Cell Interference Coordination, KPI: Key Performance Indicator, MLB: Mobility Load Balancing, MRO: Mobility Robustness Optimisation, NTM: Network Topology Mutation, PCI: Physical Cell Identity, RAT: Radio Access Technology, RET: Remote Electrical Tilt, RLF: Radio Link Failure, RRC: Radio Resource Control, TCA: Trigger-Condition-Action, TTT: Time To Trigger, TXP: Transmission Power, and UE: User Equipment.

II. Classification of Self Organising Function Conflicts

In this section, we present a comprehensive classification of SO function conflicts. More specifically, we firstly identify and annotate the potential conflicts among state of the art SO functions. Secondly, we categorise these conflicts into the following five primary categories: (*A*) Key Performance Indicator Conflict; (*B*) Parameter Conflict; (*C*) Network Topology Mutation (NTM) Conflict; (*D*) Logical Dependency Conflict; and (*E*) Measurement Conflict. The classification of novel forefront and precedent (previously identified in the literature) SO functions conflicts is provided in Table I and Table II respectively. Moreover, we present novel Network Topology Mutation Conflicts in Table III. In the subsequent sections, we elaborate on SO function conflicts, and provide a representative example for each of these cases.

A. Key Performance Indicator Conflict

A KPI is a quantifiable measurement, which reflects the critical success factor of the network. Representative examples are the Reference Signal Received Quality (RSRQ), or the Cell Aggregate Throughput, etc. KPI conflicts may occur when different SO function actions alter the same KPI of a cell while adjusting different parameters of that cell. An example of a KPI conflict is given below.

CCO Remote Electrical Tilt (RET) and CCO Transmission Power (TXP) Conflict: The alteration of both, downlink TXP and RET, influences the coverage area of a cell and can cause a KPI conflict.

B. Parameter Conflict

Parameter conflicts arise from un-desirable changes to network parameters by SO functions. SO functions act upon a single or group of Network Elements (NEs), which create the function area of these SO functions. SO functions may also take into account some additional network elements characteristics outside its functional area. The function area combined with additional NEs is defined as the impact area. A SO function is taken into account for possible conflict from the moment it has started until the end of its actions impact upon other SO functions, which is defined as impact time. Parameter conflicts are sub-categorised into the following categories.

1) Output Parameter Conflict: Output parameter conflicts may occur when two or more SO functions try to change the same network configuration parameter with intersecting impact time and impact area. The examples of different output parameter conflict cases are explained in the following.

CCO and Inter-Cell Interference (ICIC) Functions Conflict: ICIC may decrease TXP for interference reduction, while CCO may increase TXP for coverage improvement which can cause output parameter conflict, as shown in Fig. 1.

CCO and Energy Efficiency (EE) Functions Conflict: EE function may try to reduce Evolved Node B (eNB) TXP or try to activate sleep mode at eNB for energy saving, while CCO function may try to increase TXP for better coverage. Hence, both EE & CCO try to set different output values for TXP of eNB and subsequently cause output parameter conflict.

Output parameter conflict can be sub-categorised into output parameter direction or magnitude conflict as explained below.

Output Parameter Direction Conflict: This may occur when two SO functions aim at modifying the same parameter in opposite directions. An example of this conflict is given below.

CCO and Cell Outage Compensation (COC) Functions Conflict: COC function may increase TXP or change RET of the neighbouring eNB for outage compensation, while CCO may decrease TXP or RET in the opposite direction for coverage optimisation.

Output Parameter Magnitude Conflict: This may occur when one SO function aims at a large increment or decrement in a network parameter, while the second SO function wants to allow only small increment or decrement for the same network parameter. An example of this conflict is given below.

MRO and MLB Functions Conflict: MLB function may want a large decrement in the handover offset in order to balance the load distribution between neighbour cells, while MRO function may want only a small decrement in the handover hysteresis in order to reduce ping-pong effects.

2) Input Parameter Conflict: This may occur when a SO function is triggered by an input network parameter whose value is dependent upon some other network parameters. An example of input parameter conflict is given below.

Conflict between two Physical Cell Identity (PCI) Functions: In order to assign PCI to the target cell, the PCI configurations of neighbouring cells are gathered as an input to PCI function. However, if there are two PCI functions executing with intersecting impact area and time, then the PCI configurations collected by the first PCI function could be changed by the second PCI function while the first PCI function is assigning PCI to the target cell. Hence, the new PCI allocation by the first PCI function could be erroneous due to outdated input neighbouring configurations.

Table I
CLASSIFICATION OF FOREFRONT SO FUNCTIONS CONFLICTS

Sr. No.	Forefront Conflict Scenario	Conflict Category
1	CCO and ICIC	Output Parameter Conflict
2	CCO and COC	Output Parameter Conflict
3	CCO and EE	Output Parameter Conflict
4	MRO and COC	Measurement Conflict
5	MLB and COC	Logical Dependency Conflict
6	MLB and EE	Logical Dependency Conflict
7	MRO and PCI	Logical Dependency Conflict
8	MLB and PCI	Logical Dependency Conflict

Table II
CLASSIFICATION OF PRECEDENT SO FUNCTIONS CONFLICTS

Sr. No.	Precedent Conflict Scenario	Conflict Category
1	MRO and MLB	Output Parameter Conflict
2	Two PCI Instances	Input Parameter Conflict
3	MRO and CCO	Measurement Conflict
4	CCO (RET & TXP)	KPI Conflict
5	MLB and CCO	Logical Dependency Conflict
6	CCO and PCI	Logical Dependency Conflict
7	COC and PCI	Logical Dependency Conflict

C. Network Topology Mutation Conflict

Network topology mutation conflict may occur due to the change in network conditions by the addition/ removal of eNB, Home eNB (HeNB) or Relay. The details of NTM conflicts are provided in the following examples.

New eNB/ HeNB/ Relay and CCO: CCO function may configure optimum settings of TXP and RET for coverage improvement. However, the addition of new eNB/ HeNB/ Relay will have an impact on the coverage area and, as such, CCO function may need to readjust the optimum settings for coverage area. Moreover, HeNBs are frequently switched on/off or relocated, which will continuously disturb the optimum configuration of CCO function. Similarly, mobile relays can affect coverage area in a random pattern and will cause a NTM conflict for CCO function.

New eNB/ HeNB/ Relay and EE: EE function may calculate optimum settings for activating/de-activating sleep mode at specific eNBs. However, addition/removal of eNB/ HeNB/ Relay will affect the base station power model, TXP and coverage area. As a result, it will alter those assumptions under which optimum EE setting were calculated.

New eNB/ HeNB/ Relay and MRO: MRO function may calculate optimum settings of Cell Individual Offset (CIO) and hysteresis for efficient handover. However, addition/removal of eNB/ HeNB/ Relay will affect neighbouring cells relation and coverage area, which will modify those assumptions under which optimum settings for MRO were calculated.

Table III
Classification Of Conflicts Between SO Functions And New eNB/ HeNB/ Relay

Sr. No.	Conflict Scenario	Conflict Category
1	New eNB/ HeNB/ Relay and CCO	NTM Conflict
2	New eNB/ HeNB/ Relay and EE	NTM Conflict
3	New eNB/ HeNB/ Relay and MRO	NTM Conflict
4	New eNB/ HeNB/ Relay and MLB	NTM Conflict
5	New eNB/ HeNB/ Relay and ANR	NTM Conflict

New eNB/ HeNB/ Relay and MLB: MLB function may calculate optimum settings of cell reselection threshold and priorities for load balancing. However, addition/removal of eNB/ HeNB/ Relay will affect neighbouring cells relation and coverage area, which will modify those assumptions under which optimum settings for MLB were calculated.

New eNB/ HeNB/ Relay and Automatic Neighbour Relation (ANR): Addition/removal of eNB/ HeNB/ Relay will modify the neighbouring relations and ANR function needs to re-calculate new neighbour relations due to change in network conditions.

D. Logical Dependency Conflict

This may occur if there is a logical dependency between the objectives of SO functions. The details of logical dependency conflicts are provided in the following examples.

MLB and COC Functions Conflict: MLB function may change handover hysteresis in order to achieve efficient load distribution among neighbouring cells. Meanwhile, a COC function can change TXP or RET for outage compensation. However, the new handover hysteresis settings might be potentially incorrect for load balancing, due to the change in coverage overlap area between neighboring cells, as a result of COC function execution.

MLB and EE Functions Conflict: EE function may change TXP or RET in order to improve energy efficiency. However, these changes will modify the cell size and, as a result the hysteresis threshold calculated by MLB might be erroneous.

COC and PCI Functions Conflict: COC can change the cell size for outage compensation. However, it will have an impact on the assumptions under which PCI of the cell under COC functionality and neighbouring cells have been computed.

MRO and PCI Functions Conflict: MRO function can try to modify handover region for efficient user handover which will have an impact on the cell boundary. Meanwhile, if a PCI allocation function is running, then the new PCI allocation could be erroneous due to a change in cell boundary, as a result of MRO function execution.

MLB and PCI Functions Conflict: MLB function can try to modify the handover region for load balancing which will have an impact on the cell boundary. Meanwhile, if a PCI allocation function instance is running, then the new PCI allocation could be erroneous due to a change in the cell boundary, as a result of MLB function execution.

E. Measurement Conflict

This may occur if a SON function is either triggered or computes new parameter values based on outdated measurements. An example of measurement conflict is given below.

MRO and COC Functions Conflict: COC function can modify RET in order to compensate for cell outage, which will have impact on the cell size. Meanwhile, if a MRO function is triggered based on measurements collected before the change in cell size, then the MRO function could be using outdated measurements for calculating new handover settings.

Table IV
Self-Coordination Mechanisms

Self-Coordination Mechanisms	Mechanism Approach
Policy Functions	Policies are derived from operator requirements and consist of a set of constraints on the network behaviour.
Workflows	Workflows consist of a set of activities to accomplish SO goals, according to a set of procedural roles.
Decision Tree Logic	Decision trees provide a sequence of conditions that need to be evaluated in order to take a coordination decision in response to a SO function execution request.
Autognostics Function	Autognostics function collects and processes performance, fault and configuration data as input to the SON system.
Alignment Function	The Alignment function monitors output parameter configuration requests from SO functions. It rejects or reschedules the requests in case of conflict between SO function goals or parameter values.
Trigger-Condition-Action (TCA) Policy	TCA policy monitors possible SO function conflicting conditions and requests for necessary actions to be executed in order to avoid SO function conflicts.
Co-Design	Co-design function combines the goals of multiple SO functions into a single optimisation function that optimizes multiple parameters simultaneously.
Guard Function	The Guard function detects extreme or undesirable network behaviours and triggers countermeasures.
Parameter Locks	A SO function configuring a parameter may lock this parameter for a certain period of time in order to prevent other SO functions modifying this parameter negatively.
Algorithm Coordination	The algorithm coordination function operates on the algorithm execution request, which allows it to take coordination decisions before the algorithm execution.
Action Coordination	In action coordination, SO functions send requests to Self-Coordinator in order to execute its action. Only in case of an acknowledgment the actions are triggered.

III. Self-Coordination Framework

A. Self-Coordination Mechanisms In Literature

In this subsection, we present a comprehensive description of Self-Coordination mechanisms, such as Policy functions, Workflows, Decision Tree Logic, Autognostics Function, Alignment Function, TCA policy, Co-Design, Guard Function, Parameter locks, Algorithm Coordination and Action Coordination in Table IV. These Self-Coordination mechanisms are summarised in Table IV and the interested reader may refer to [7]-[9] for further details.

B. Possible Evolution of Self-Coordination Mechanisms

In this subsection, we propose novel TCA policies for MRO and MLB functions. These cutting-edge TCA policies for case study of MRO and MLB functions provide a foundation for possible evolution of Self-Coordination mechanisms. More specifically, the newly proposed TCA policies, as shown in Fig. 2 and Fig. 3, are the keys to designing conflict-free Decision Tree Logic or joint optimisation algorithms for MRO and MLB functions.

The TCA policy for MRO function starts off by scanning for possible input triggering conditions (e.g. call drop, Radio Link Failure (RLF), ping pong etc.) as shown in Fig.2. These triggering conditions are recorded via both UE measurement reports and handover reports communicated between eNBs over X2 interface [Section 5.1, 7]. Once triggering conditions for MRO are fulfilled, then the TCA MRO policy investigates, using the communication between neighbouring eNBs over X2 interface, whether there are any other conflicting SO functions active with intersecting impact area and impact time. If any MLB function is active, then MRO function sets the limits on the handover region, in which MLB function can operate without creating handover issues for MRO [10] and restarts the MRO TCA policy function. If there is any CCO function active, then its execution can have long visibility delay upon cell KPIs, such as coverage area. Therefore, updated coverage measurements are requested from neighbouring cells in order to obtain accurate characteristics of neighbouring cells for MRO calculation. If there is any new HeNB or relay inserted within the impact area of MRO function, then new ANR is calculated before MRO execution, as shown in Fig. 2. However, if there are no SO functions conflicts with MRO, then MRO TCA policy proceeds to check for handover optimisation related conditions, such as intra-frequency, inter-frequency or inter-Radio Access Technology (RAT) (from LTE to Third Generation (3G) or Second Generation (2G)) handover in both connected mode and idle mode, as shown in Fig. 2. The details of these connected and idle mode handover conditions (A3, A5, B2 etc.) can be found in [Table 5.1, 7] and [Table 5.2, 7] respectively. MRO function possible actions in connected mode (e.g. adjusting Time To Trigger (TTT) and CIO for resolving ping pong and too early handover problems) and idle mode (e.g. adjusting cell reselection thresholds and priorities for handover optimisation) are presented in Fig. 2. Finally, MRO TCA policy action requests are allowed/re-scheduled or rejected by Self-Coordinating Alignment Function for conflict-free operation of MRO function as shown in Fig. 2. It must be mentioned here that most of the previous research has focused on intra-frequency connected mode handover condition for MRO. However, optimum configuration of CIO, TTT, filter co-efficient, cell reselection thresholds and priorities for inter-frequency, inter-RAT handover in both connected and idle mode need to be identified. It is evident from Fig. 2 that TCA MRO policy is quite useful for designing conflict-free MRO algorithms as it provides a holistic picture of all the possible triggers, SO function conflicts with MRO, handover conditions to be checked and the corresponding actions to be executed.

The TCA MLB Policy begins with the examination of input triggering conditions for MLB function (e.g. blocked calls, overloaded cell etc.) as shown in Fig. 3. MLB function can also be triggered due to high energy consumption and handover problems. More specifically, handover problems could be created due to the concurrent operation of both MLB & MRO functions on handover thresholds [10]. The identification of this handover problem can trigger MLB function to limit its operation within restricted HO region. Moreover, high energy consumption can also trigger MLB function in order to switch off particular cells after transferring the load from light loaded

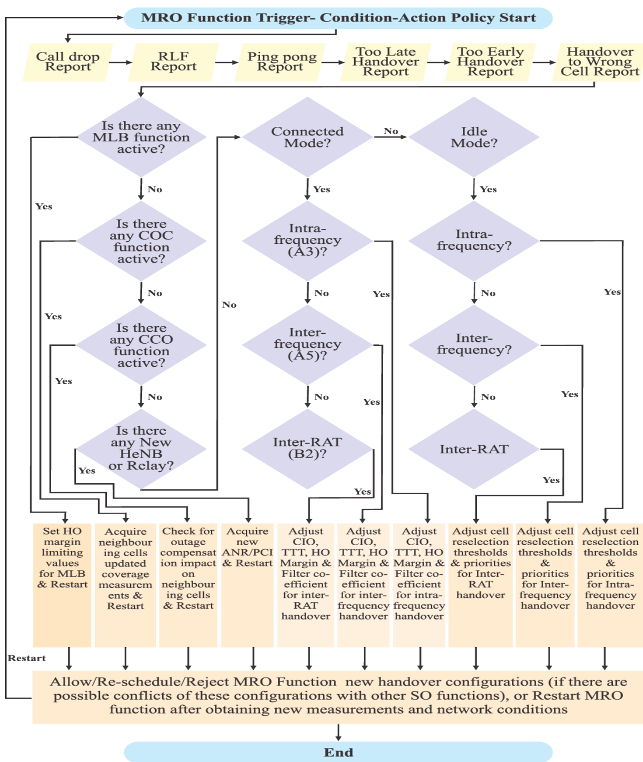

Figure 2. MRO Function Trigger-Condition-Action Policy

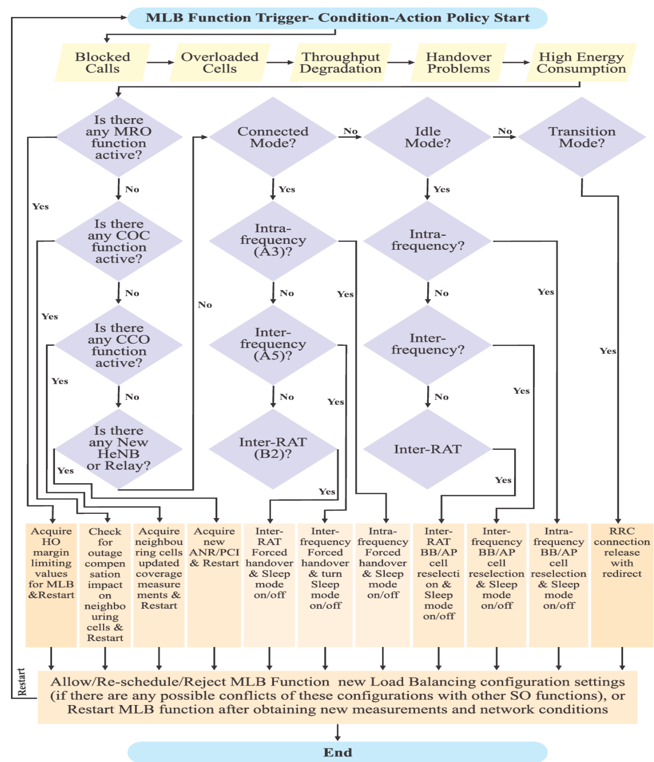

Figure 3. MLB Function Trigger-Condition-Action Policy

cells to neighbouring cells [11]. Once triggering conditions for MLB are fulfilled, then the TCA MRO policy investigates, using the communication between neighbouring eNBs over X2 interface, whether there are any other conflicting SO functions active with intersecting impact area and time. If the conflicting conditions are detected, then corresponding precautionary actions are executed in order to avoid those conflicts as shown in Fig. 3. However, if there are no SO function conflicts with MLB, then MLB TCA policy proceeds to check for mobility load balancing related conditions, such as intra-frequency, inter-frequency or inter-RAT handover in connected mode, idle mode and transition mode (from connected to idle mode and vice versa) as shown in Fig. 3. MLB function also checks for the following conditions: (a) what is the status of the neighbouring cells load?; (b) are there any requests received from neighbouring cells [11] for load transfer?; (c) can the energy saving mode be activated by MLB function due to light load [11] conditions? As shown in Fig. 3, MLB function possible actions for load balancing in connected, idle and transition mode are elaborated as follows: (a) Basic Biasing (BB) cell reselection method adjusts the cell-pair offset in such a way that it increases/decreases the cell boundary within which it can be selected by a UE for camping in idle mode; (b) Absolute Priorities (AP) cell reselection method modifies the cell priorities in order to increase/decrease the probability of a cell being selected for camping by UE in idle mode; (c) Radio Resource Control (RRC) connection release with re-directs method rejects the connection establishment request during idle to connected mode transition and provides the redirection information to UE for load balancing; (d) Forced handover method adjusts the CIO, TTT and filter co-efficient in such a way that handovers for boundary User Equipments (UEs) are forced towards lightly loaded cells; (e) Sleep mode activation/de-activation method switches on/off base stations according to load conditions in order to save energy. Finally, MLB TCA based policy action requests are allowed/re-scheduled or rejected by Self-Coordinating Alignment Function for conflict free operation of MLB function as shown in Fig. 3. It must be mentioned here that most of the previous research has focused on intra-frequency connected mode mobility load balancing. However, optimum configuration of CIO, TTT, filter co-efficient, BB thresholds and Absolute Priorities for inter-frequency, inter-RAT mobility load conditions in connected, idle and transition mode need to be identified.

IV. Optimum Interactions between Self-Optimisation & Self-Coordination

A. Motivation

If Self-Optimisation and Self-Coordination functions are executed independently, then the algorithm part of Self-Optimisation functions will always be executed. This is irrespective of the subsequent acknowledgment or rejection of the Self-Optimisation action request by the Self-Coordination function. Moreover, based on the rejected action requests, numerous Self-Optimisation algorithms might have been executed without any performance gains. The above mentioned facts indicate the need to find optimum interactions between Self-Optimisation and Self-Coordination functions.

Table V
Classification of Coordination Type in Each Conflict Category

Conflict Category	Preferred Coordination Type
Parameter Conflict	TCA policy or Decision Tree Logic or Co-Design or Algorithm Coordination
NTM Conflict, KPI Conflict or Logical Dependency Conflict	Alignment Function or Action Coordination or Algorithm & Action Coordination
Measurement Conflict	Algorithm Coordination

B. Classification of Optimum Interactions between Self-Optimisation & Self-Coordination

The classification of optimum interactions between Self-Optimisation and Self-Coordination for different conflict categories is proposed in Table V. The rationale behind the proposal of using Co-Design, TCA policy, Decision Tree Logic or Algorithm Coordination mechanism in case of parameter conflict is as follows: (a) TCA policy or Decision Tree Logic can avoid SO function conflicts by investigating the possible conflict conditions and executing necessary actions (e.g. MRO and MLB TCA policies shown in Fig 2 and Fig. 3); (b) Co-Design can reduce the shared parameters by pre-defining clear responsibilities for SO function parameters [7]; (b) Co-Design can combine the goals of multiple SO functions into a single optimisation function; (c) Co-Design can pre-assign priority to certain SO functions over others, depending upon the operator policy and customer demands; (d) The algorithm coordination function governs the algorithm execution request. As such, it can take coordination decisions before the algorithm execution and reduces computational burden. An example of MRO & MLB Co-Design is given below.

MRO & MLB Co-Design: In this case, after the Co-Design MRO will set the limiting values on HO region within which MLB can operate [10].

The rationale behind the proposal of using Action Coordination, combined Algorithm & Action Coordination, or Alignment function for NTM, KPI or logical dependency conflicts is as follows: (a) Alignment function can allow, reschedule or reject SO function execution requests at run time, based upon the susceptibility of NTM, KPI or Logical Dependency conflict; (b) Action Coordination can deal with NTM, KPI and logical dependency conflicts at run time by providing means for run-time control of SO functions behaviour according to operator policies; (c) Combined Algorithm & Action Coordination can not only stop the false triggering of Self-Optimisation functions by providing stable KPI measurements, but can also provide run time control over conflicting SO functions behaviour. However, the computational complexity of combined Algorithm and Action Coordination is higher than only Action Coordination [7]. The fact behind the proposal of using Algorithm Coordination for Measurement Conflict is that Algorithm Coordination does not acknowledge the SO functions execution request until stable measurement values are available.

V. Conclusion & Future Work

In this paper, we have identified and annotated a number of possible conflicts among the range of SO functions anticipated to be implemented in LTE and LTE-Advanced. We also present a comprehensive taxonomy of these conflicts, which classifies them on the basis of network topology mutation, KPI, output parameters direction/magnitude, measurement and logical dependency conflicts. This classification can ultimately pave the way for the identification of the suitable conflict avoidance mechanisms. More specifically, we have defined twenty SO function possible conflicts, including thirteen new SO conflict cases, described in Table I and Table III, which have not been identified before in the literature. While these conflicts vary in the degree of difficulty of identification, the KPI, logical dependency and measurement characteristic conflicts are the most difficult conflict categories to be detected and resolved. We have also provided a comprehensive description of possible Self-Coordination approaches that can be exploited to avoid the various kinds of identified conflicts. For possible evolution of the Self-Coordination paradigm, we propose a TCA policy framework that provides the basis for designing conflict free Decision Tree Logic or Joint Optimisation Algorithm for MRO and MLB functions. Building on the study of the conflict types and potential Self-Coordination schemes available to address these conflicts, we identify the suitable Self-Coordination mechanisms for each conflict category. The analysis presented in this paper will assist further research for delivering conflict-free SON pragmatic solutions. It will also shift the focus from designing standalone SON algorithms towards more holistic Co-Design based approaches or SON solutions that feature the essentially needed, but often overlooked, explicit or implicit Self-Coordination compatibility.

References

[1] 3GPP TS 32.500, "Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements," v. 10.0.0, June 2010.

[2] NGMN, "Recommendation on SON and O&M Requirements," Requirement Specification, www.ngmn.org, version 1.23, Dec. 2008.

[3] NGMN, "Use cases related to Self Organising Network, Overall Description," White Paper, www.ngmn.org, version 2.02, Dec. 2008.

[4] 3GPP, "TR 36.902 - Self-configuring and self-optimizing network (son) use cases and solutions,". Technical Specification, 2010.

[5] M. M S Marwangi, N. Fisal, S. K S Yusof, R.A. Rashid, A.S.A. Ghafar, F.A. Saparudin, and N. Katiran. Challenges and practical implementation of self-organizing networks in lte/lte-advanced systems. In Proceedings of Information Technology and Multimedia (ICIM' 2011), pages 1–5, 2011.

[6] Jansen, M. Amirijoo, U. Turke, L. Jorguseski, K. Zetterberg, R. Nascimento, L.C. Schmelz, J. Turk, and I. Balan. Embedding multiple self organisation functionalities in future radio access networks. In Proceedings of IEEE Vehicular Technology Conference (VTC Spring 2009), pages 1 –5, April 2009.

[7] Seppo Hämäläinen Cinzia Sartori, Henning Sanneck. "LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency". WILEY, 2011.

[8] T. Bandh, R. Romeikat, H. Sanneck, and Haitao Tang. Policy-based coordination and management of son functions. In Proceedings of IEEE Integrated Network Management (IM' 2011), pages 827–840, May 2011.

[9] L.C. Schmelz, M. Amirijoo, A. Eisenblaetter, R. Litjens, M. Neuland, and J. Turk. A coordination framework for self-organisation in lte networks. In Proceedings of IEEE Integrated Network Management (IM' 2011), pages 193 –200, May 2011.

[10] Li Yun, Li Man, Cao Bin, Wang Yong, and Liu Wenjing. Dynamic optimization of handover parameters adjustment for conflict avoidance in long term evolution. Communications, China, vol. 10, issue no. 1, pages. 56–71, Jan. 2013.

[11] Jinlin Peng, Peilin Hong, Kaiping Xue, and Hao Tang. A dynamic energy savings scheme based on enhanced mobility load balancing. In Proceedings of IEEE Vehicular Technology Conference (VTC Fall 2012), pages 1–5, 2012.