
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Tang, Haitao; Stenberg, Kaj; Apajalahti, Kasper; Niiranen, Juha ; Räisänen, Vilho
Automatic Definition and Application of Similarity Measures for Self-Operation of Network

Published in:
Mobile Networks and Management

DOI:
[10.1007/978-3-319-52712-3_15](https://doi.org/10.1007/978-3-319-52712-3_15)

Published: 01/01/2017

Document Version
Peer reviewed version

Please cite the original version:
Tang, H., Stenberg, K., Apajalahti, K., Niiranen, J., & Räisänen, V. (2017). Automatic Definition and Application of Similarity Measures for Self-Operation of Network. In *Mobile Networks and Management: 8th International Conference, MONAMI 2016, Abu Dhabi, United Arab Emirates, October 23-24, 2016, Proceedings* (pp. 206-219). (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Vol. 191). https://doi.org/10.1007/978-3-319-52712-3_15

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Automatic Definition and Application of Similarity Measures for Self-Operation of Network

Haitao Tang¹, Kaj Stenberg², Kasper Apajalahti³, Juha Niiranen⁴, and Vilho Räsänen¹

¹Bell Labs
Nokia
Finland

²Mobile Network
Products
Nokia
Finland

³Semantic Compu-
ting Research Group
Aalto University
Finland

⁴Dept. of Mathemat-
ics and Statistics
University of Hel-
sinki, Finland

haitao.tang@nokia.com

Keywords: similarity measure, context aware, network operations, self-operation, operation experience, OSS, cellular network, case based reasoning.

Abstract. Self-operation concept is proposed to learn the past experiences of network operations and apply the learned operation experiences to solve new but similar problems. It works based upon the observation that actions appropriate for achieving an objective resemble each other in similar network contexts. Plenty of such similarities exist at the level of network elements, functions, and their relations. Similarity measure definition and application are essential components for the self-operation to apply the learned operation experiences. This paper provides a solution for self-operation to define and apply two types of similarity measures for two self-operation use cases. The first use case answers how to select a best suitable function to achieve any given objective. The second use case tells how the selected function should be configured with the most optimal parameter values so that the given objective could be achieved. This solution is realized on a demonstrator implementing the self-operation concept. Corresponding experiments are made with the demonstrator. The experimental results show that the self-operation solution works well.

1 Introduction

The network environments of multi-RAT, multi-access, and multi-vendor have added significant complexity to the network operations. Self-x functions (e.g., SON and traffic steering functions [1-3]) have become an essential part of the 3~4G networks and their management. These self-x functions have reduced a clear part of manual work related to operations that would be needed otherwise for the 3~4G networks. This effectively reduces the operational complexity perceived by human operators as well. The coming 5G systems (i.e., their networks and management systems [4]) are expected to have a much wider scope and, a larger number and variety, of self-x functions and multi-x network environments. In addition, one of the 5G goals is to minimize the need of human involvement in their operations.

These fundamental developments have created the industry-wide determination to gradually evolve towards cognitive network management systems. In such systems, relevant past experiences could be used to predict the future status of a network. The corresponding decisions could thus be made to improve either network performance or subscriber perceived experience. In such systems, the minimal but still critically needed role of human operator can be seamlessly integrated to observe the systems and instruct them when the non-human parts of systems have no knowledge to deal with certain situations or are otherwise incapable of drawing conclusions or making decisions by their own based on their predefined logics. Such systems can prevent the functions (self-x or not) of the systems from executing the operations that may cause (and are known to have caused already earlier) degradation in network performance metrics or unfavorable user experience. Such systems can also make an operation of a function favorable if its execution is expected to induce improvement in network performance or customer experience.

Self-operation [5] proposes a solution to realize such an aforementioned cognitive network management system. It creates a self-operation case for each relevant event, learns every corresponding operation and outcome of the system, and stores the learned experience to the self-operation case. The outcome consists of the performance metrics and customer experience, etc. The self-operation case also stores the learned context data such as system conditions and other relevant circumstances (e.g. cell configuration, location, and traffic profile) that may have impacted the triggering of the event. All the data relevant to a corresponding operation execution are learned and collected in the data elements of the self-operation case, and thus inherently linked into a piece of useful corresponding experience, which can thus be applied on the fly. The availability of such experience is very important. As of today, these data (if any) are quite scattered and distributed in a system. Some data elements are stored in different locations. Other necessary data elements may not yet be stored at all. In such a situation, data mining cannot help to find the experience. In addition, data mining is usually time / resource consuming, and cannot therefore meet a request for such experience in timely fashion.

The self-operation solution in [5] does not answer how to define a similarity measure to find the corresponding operation experiences from the potentially large number of learned but different self-operation cases. The similarity measure definition is an essential and critical component of the self-operability and, it is specific to the given use case of self-operation. It determines if the operation experience cases (i.e., knowledge) learned from the earlier executed operations can be applied to future operations of a given use case.

As a major use case of self-operation, an operator may want to request the self-operation for guidance on how to achieve given objective(s) for network performance or service in a certain area (i.e. scope of the network). The objectives are usually related to improvement of certain Key Performance Indicators (KPI) regarding for example coverage, traffic, mobility, or quality. Different functions may however cause impacts on many of those KPIs at the same time. It is thus difficult for the operator to select the best function (out of several candidate functions) to achieve the objective. This is where the definition and application of the corresponding similarity measure

can help. The relevant self-operation cases can be matched from the knowledge database, with the corresponding objective-specific similarity measure. The information about the best suitable function can then be extracted from the relevant self-operation cases.

As another major use case of self-operation, the operator can have difficulty to determine the (best) suitable configuration (e.g., SCV - SON function Configuration parameter Value) for the selected function to achieve the result expected by the objective. The suitable configuration of the function depends on the corresponding conditions (network configurations, status, traffic, etc.) of the managed objects (MOs) where the function is planned to be executed. This is again where the definition and application of the corresponding similarity measure can help. The relevant self-operation cases could be matched from the knowledge database, with the corresponding function-specific similarity measure. The information of the (best) suitable function configuration can then be extracted from the relevant self-operation cases.

The function-specific similarity measure could also be used to find the corresponding operation experience case(s) and extract the knowledge concerning another related major use case of self-operation that answers the question: "Can an action request for the function be executed or not?" Thus, the function-specific similarity measure enables both the operation to select the corresponding configuration for the chosen function, and the decision on an action of the function.

The motivation of this paper is therefore to design a complete solution by solving the following problems: (1) how to define an objective-specific similarity measure to match an objective to its corresponding function; (2) how to automatically match the objective and corresponding rule given by an operator to the best suitable function for achieving the given objective; (3) how to automatically define the corresponding function-specific similarity measure; (4) how to automatically apply the function-specific similarity measure for a function-specific operation. For example, the corresponding operation information and the configuration(s) can be found from the matching operation experience cases.

The sections of this paper are organized as follows. In Section 2, the self-operation architecture to define and apply a similarity measure is described. In Section 3, the approach to define an objective-specific similarity measure is introduced. In Section 4, it is explained how an objective-specific similarity measure is used to find the relevant function. In Section 5, the approach to define a function-specific similarity measure is presented. Section 6 depicts how a function-specific similarity measure is used to find the proper configuration value for the selected function. Section 7 introduces the example implementation of a self-operation system and its experiment results. Section 8 summarizes and discusses the major finding of the current work.

2 Self-Operation Architecture for Similarity Definition and Application

Figure 1 shows the architecture of defining and applying similarity measures based on stored operation experience cases. The arrows are logical and can be implemented by direct or indirect connections between the entities in the real implementations.

The definition of a similarity measure is started when a request message for action recommendation (i.e., Message 1, A, or I) is received. These messages serve as the triggers to define relevant similarity measures and use them to find the matching self-operation cases with the principle of case based reasoning [6]. The corresponding experiences in the matching self-operation cases are replied back to the requesting entities.

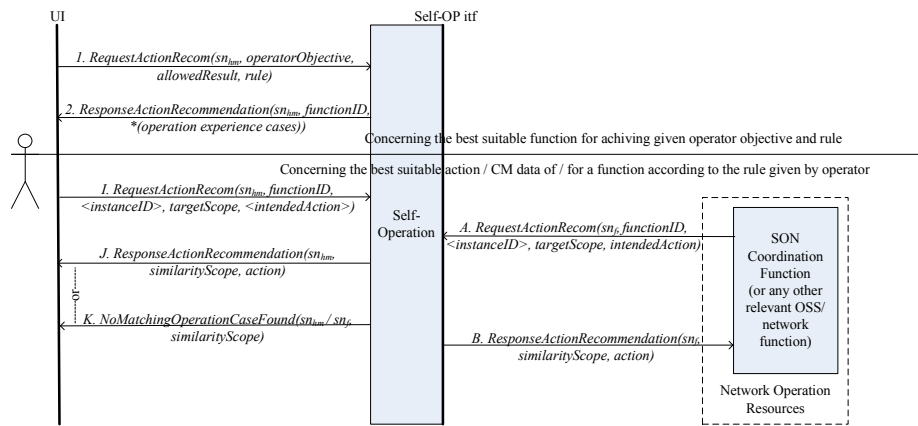


Fig. 1. The architecture of defining and applying similarity measures based on stored operation experience cases.

There are different types of similarity measures, which are usually specific to their actual applications (i.e., use cases) [7-9]. The similarity measures of this paper belongs to the family of semantic similarity measures. The similarity measures of this paper are used to find the exactly matching self-operation cases (if any) stored in the database of the self-operation entity. Their specific definitions and applications of the similarity measures are given in the following sections.

3 Definition of Similarity Measure for a Given Operator Objective and Rule

The operational objectives and rules of a network (e.g., [10]) are usually defined with a set of high level KPIs (e.g., [11, 12]) for the network operations. When an operator needs to achieve a specific objective for network performance under certain rules (i.e., constrains and options), the operator sends a request (Message 1 in the Figure 1) to

self-operation function. This message carries the information of the operation objective including the target scope (i.e., the targeted MOs), allowed result, and rules. The rules can be created either by the system vendor or by the operator via means of Rule Editor, which is a specialized tool for the creation and maintenance of the rules.

The self-operation function uses the received information to define the corresponding objective-specific similarity measure, which can be simply in the form of a text string carrying the provided information elements.

4 Selection of a Suitable Function for the Given Operator Objective and Rule

The self-operation function uses the objective-specific measure (defined in Sec. 3) to find all the matching operation experience cases and their functions. For demonstration purposes, we present two Capacity and Coverage Optimization (CCO) functions, CCO-SURROUNDED (optimizing a cell surrounded by its first-tier neighbor cells) and CCO-HOTSPOT (optimizing a hotspot source cell). For example, these functions (CCO-SURROUNDED and CCO-HOTSPOT, shown in Figure 6) have caused similar operation experiences (in areas containing both surrounded and hotspot cells) in the past.

According to the rule, the self-operation function selects the best suitable experience cases from all the matching operation experience cases. For example, *CCO-SURROUNDED* is the function that has achieved the optimization objective in most of the matching cases (96 % of all the matching cases). *CCO-SURROUNDED* function is thus selected automatically as the best suitable function to achieve the intended operation. The decision for the selection can be made based on several different criteria such as the highest probability to achieve successful results, operations' priorities or operator's preferences and policies. The criteria is actually defined by the rules provided by the operator. The general procedure of objective specific similarity definition and function selection is described with the diagram shown in Figure 2.

5 Definition of Similarity Measure for the Selected Function

The common information elements needed by a function-specific similarity measure instance are defined as a set of general similarity attributes and function-specific attributes, as shown in Table 1. The function-specific attributes are always explicitly defined for the specific function selected. A function-specific similarity measure instance is always operation specific.

After the best suitable function is found by self-operation, the self-operation is invoked to define its function specific similarity measure, which consists of two parts. With the information of the function (e.g., *CCO-SURROUNDED*) and the objective-specific similarity measure (e.g., coverage-related optimization of the cells with ID 1-5), the self-operation function defines the first part of the corresponding function specific similarity measure. Here, the information of any function in the network is pre-

defined and made available in the form of function metadata [13] by the operator or its vendor. The function information also defines the impacting scopes [14] of the function.

The first part of the function-specific similarity measure is the static information of the function and the MOs that are either pre-defined or available beforehand. This “static” part is defined by extracting the information of the function, the corresponding cells, and the relevant rule. For example, the first part of *CCO-SURROUNDED* - specific similarity measure can consist of the information elements (and their values) of such as CCO ID, cell technology, cell type, and antenna mode. For simplicity, we assume the target scope consists of only one similarity scope in this example. In reality, if multiple similarity scopes exist in a given target scope (as often the case), their corresponding function-specific similarity measures are defined one by one with the same approach shown in this example.

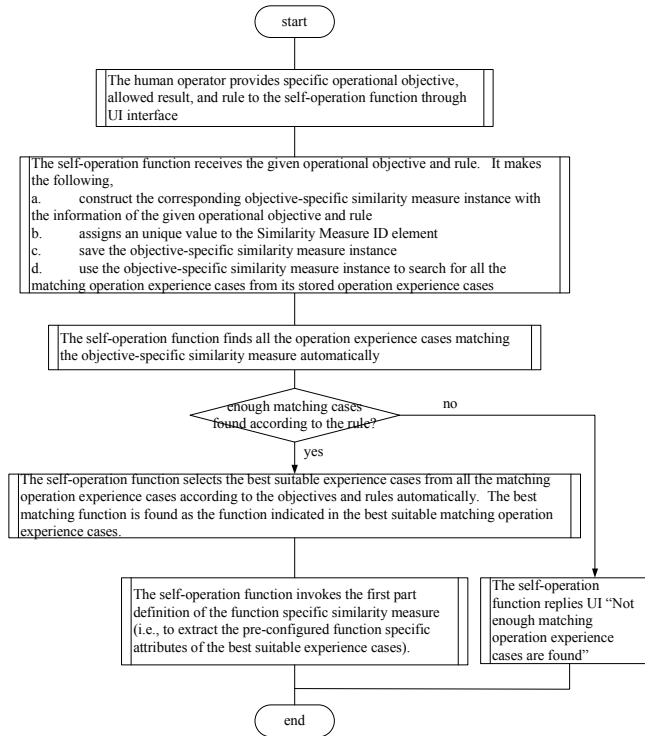


Fig. 2. Procedure to find the best matching function and its corresponding operation experience cases that can achieve a given operator objective.

The self-operation function then uses the defined first part of the function-specific similarity measure to further select the matching operation experience cases from all the cases still fulfilling the search criteria. For example, there are 51 self-operation cases found under the selected *CCO-SURROUNDED* function. 25 self-operation cases match the defined first part of the *CCO*-specific similarity measure.

The self-operation function extracts the information of the 25 cases. What to extract depends on the given rule or otherwise a default configuration. For example, an extraction can be done from all those performance metrics information elements and their value ranges shared by some or all of the 25 cases. These performance metrics are, for example, the impacting and impacted metrics of *RLF INPUT* and *RLF OUTPUT*. The extracted result serves and becomes the remaining part of the similarity measure definition. Now, the complete function-specific similarity measure has been defined.

Table 1. The common information elements of a function-specific similarity measure instance.

	Element Name	Definition
General Attributes	Function-specific Similarity Measure ID (F SM ID)	A character string that uniquely identifies this similarity measure instance. It helps the further process and application of this similarity measure instance.
	Similarity Scope	The type of the managed objects (MOs) relate to this similarity measure instance. For example, A similarity scope can be one type of <i>{individual cell, cell pair, first-tier neighbor cells, second-tier neighbor cells, subnetwork, network, etc.}</i> . A similarity scope is usually specific to a function. For example, a <i>CCO-SURROUNDED</i> function is optimizing the coverage performance of the given cells. Thus, the similarity scope for this function is the given cell and its 1 st -tier neighbors of the same type.
Function-Specific Attributes	Function ID	The unique ID of a function (e.g., <i>CCO-SURROUNDED</i>) that is selected as the most relevant function to pursue the requested operation.
	Function Specific Attribute ₁	The first feature specific attribute and value that is automatically extracted from the selected experience cases. Note: an attribute and its value are extracted only when this attribute is impacting the function or is impacted by the output of the function. The attribute is identified according to the impacting scopes [14] of the function.
	...	
	Function Specific Attribute _n	The n th feature specific attribute and value that is automatically extracted from the selected experience cases, where $n \geq 0$ and, $n=0$ means there is no feature specific attribute for the specific similarity measure instance.

6 Selection of Suitable Configuration for the Selected Function

With the defined function-specific similarity measure, self-operation function finds, e.g., 9 self-operation cases (out of the 25 cases) matching the similarity measure exactly. The configuration values (SCVs) of the 9 self-operation cases are collected into a configuration set called “*CCO-SURROUNDED Config Set*”. The self-operation

function then uses the extracted configuration value to configure the *CCO-SURROUNDED* function and activate it to achieve the given objective.

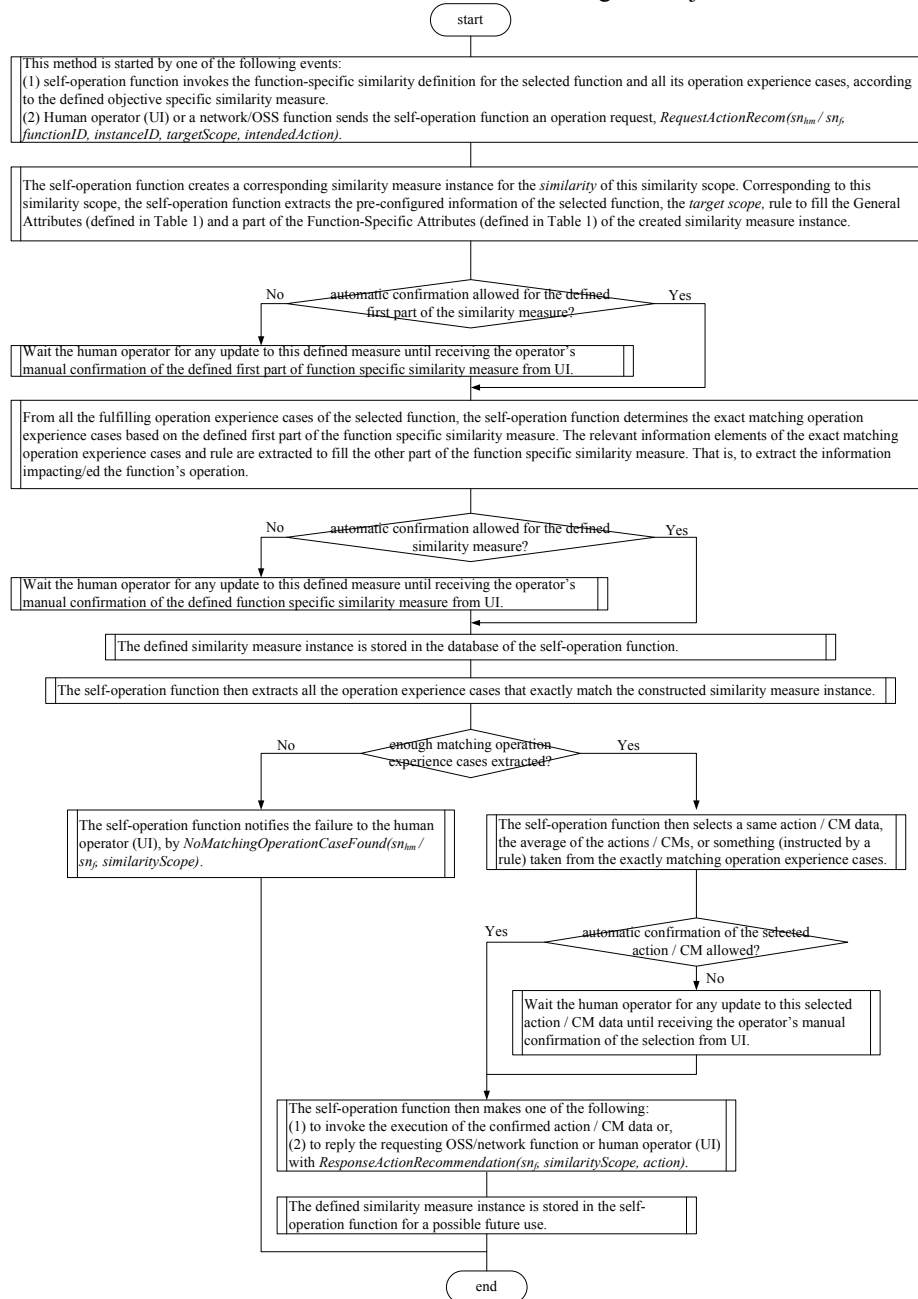


Fig. 3. Procedure for the automatic function-specific similarity measure construction and application.

The general procedure of the definition and application of function specific similarity measure is described with the diagram shown in Figure 3. In this procedure, the intervention of human operator is supported in the otherwise automatic definition and application of a function-specific similarity measure. For example, the operator may need to update or confirm a selection.

7 Experiments on Determining Suitable Function and Configuration Automatically

In this section, we describe a demonstrator for self-operation and show its experimental results. It currently realizes two use cases: 1) finding the suitable function to achieve a given (high level) objective and 2) finding the corresponding configuration for the selected function so that the objective can be achieved. The details concerning these two use cases have been presented in Sec. 3-6.

7.1 Demonstrator Description

The demonstrator set-up for learning operation experiences is shown in Figure 4. It also supports the applications of operation experiences for the self-operation use cases that receive their configuration or instruction from self-operation with means not shown in Figure 4.

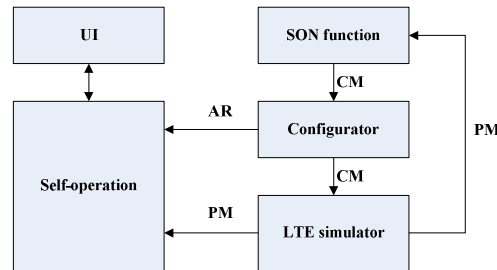


Fig. 4. Architecture of self-operation demonstrator for learning operation experiences, where AR = action request, CM = configuration management data, PM = performance management data, and UI = user interface.

An LTE simulator (Nokia internal tool and its main principle introduced in [15]) acts as a source of PM data, which are sent via REST (Representational State Transfer) interfaces to a SON function and the self-operation control logic. Two groups of SON functions are used in the demonstrator, i.e., RET (Remote Electrical Tilt)-based CCO and energy saving (ES). Configurator adjusts cell and other (e.g., function) parameters on the one hand, and amends CM data with metadata to create Action Requests (ARs) for self-operation on the other. The UI can be used for similarity definitions and related operation case searches. Direct configuration from the UI is not implemented at the moment.

PM data are stored into self-operation internal database as they arrive. ARs are received by the self-operation system in such a way that each received AR triggers the creation of a corresponding operation case. A MongoDB NoSQL database is used for storing PM data, operation cases, and function profiles. The REST interface is implemented with Java, the self-operation control logic with Clojure, and the user interface with HTML 5 / JavaScript. SON functions, self-operation, and UI JavaScript back-end are run on Ubuntu desktop machine (Intel Core2 2.5 GHz, 2GB memory, 64-bit Ubuntu). UI front-end is run on browser (Chrome) over a Windows 7 laptop (8 GB).

7.2 Experiment and Result

This section shows an example experiment and results in which a human operator provides high-level operation objective and then finds a best matching function and its configuration set in a desired context. The experiment is done in three phases, i.e., defining goal and preconditions, retrieving relevant search results based on the automatically defined similarity measure, and, if needed, sharpening the results by adjusting similarity measure.

7.2.1 Defining Goal and Precondition for the Objective-Specific Similarity Measure

Figure 5 depicts the first phase in the view. Through UI, the user can define a goal and its context by Wizard 1 of Figure 5 and set numerical boundaries (or other rules) for the search results by Wizard 2 of Figure 5.

Target Setting for the Operation

1 2

Select goals

Goals	Cell selection	Time of day
optimize_coverage	1, 2, 3, 4, 5	Morning (06-10)

Add goal
Remove goals

Selected goal(s)

Goal: optimize_coverage
In: [1,2,3,4,5]
Time: 06-10

Remove goal

Define similarity measure and find function

Define rules for the operation

More cases than	1
Confidence level [0-1] higher than	0.01
Automatic confirmation	<input type="checkbox"/>

Fig. 5. A snapshot of an example to define objective specific similarity measure via specifying corresponding goal, network context, and numerical boundaries.

In this use case, the user wants to find suitable operations for optimizing coverage for an area including, e.g., five cells (1, 2, 3, 4, and 5), and with a time range from, e.g., 6AM to 10AM. In the rule definition form (Wizard 2), numerical boundaries can be selected to further exclude irrelevant search results. For example, the minimum

amount of cases per retrieved SON function is set to 1. The minimum confidence level (success ratio to achieve a goal) is set to 0.01. Here, the boundaries have low values in order to maximize the amount of operation cases in the search results.

The above information is used to define the objective-specific similarity measure. With this similarity measure, Wizard 3 of Figure 6 searches and shows the result of the self-operation cases matching the objective-specific similarity measure. The self-operation analysed approximately 500 cases in several seconds. Two SON functions are then identified in these self-operation cases. *CCO-SURROUNDED* is a CCO function instance that optimizes a source cell surrounded completely by its 1st-tier neighbor cells. *CCO-HOTSPOT* is a CCO function instance that optimizes a hotspot source cell. The columns in Wizard 3 describe the name of the function (*function_name*), the total amount of self-operation cases matching the target that the SON function has been involved with (*matching_cases*), the amount of successful cases achieved the target (*successful_cases*), the success ratio of the matching cases (*confidence*), and the proportion of the number of matching cases of the function to the total number of the matching cases of all functions (*proportion*). The function (*CCO-SURROUNDED*) and its 51 self-operation cases in the first row are selected as this function has the highest proportion value.

7.2.2 Function-Specific Similarity Definition and Search for Configuration

For the selected function *CCO-SURROUNDED*, the available function-specific similarity attributes are antenna elevation, antenna type, and cell type of the source cells, as well as their value ranges for RLF. The value ranges indicate the values before (*RLF INPUT*) and after (*RLF OUTPUT*) the operation case has been executed.

Matching operation case

function_name	matching_cases	successful_cases	confidence	proportion
CCO-SURROUNDED	51	25	0.49	0.96
CCO-HOTSPOT	1	1	1.00	0.02

Select configuration set (CCO-SURROUNDED)

instance_name	matching_cases	successful_cases	confidence	proportion
v1	18	9	0.50	0.35
v2	33	16	0.48	0.65

Function-specific similarity measure (CCO-SURROUNDED)

Attributes for the TARGET cells

ret: moderate
 antenna.location.elevation: 17, 18, 19, 20
 antenna_type: Nothing selected
 cell_type: surrounded

Expected KPIs for the TARGET cells

RLF INPUT: 91-383
 RLF OUTPUT: 68-420

Thresholds:
 0.63 < CQI < 1.35
 25 < CUE < 60
 12 < RLF < 80
 -107.09 < RSRP < -102.03

Confirm

Fig. 6. A snapshot of the objective-based function selection, the function-specific similarity definition, and the function configuration selections.

Wizard 4 of Figure 6 collects the relevant attributes of the source cells to define the first part of the function-specific similarity measure for the selected function. With this first part of the similarity measure, the 25 matching self-operation cases (found by Wizards 3) are further filtered. The remaining part of the function-specific similarity measure is then defined by extracting the relevant KPI information of the further matched self-operation cases. With this fully defined function-specific similarity measure, the first configuration set ($v1$) of the 9 self-operation cases is selected, as shown in Wizard 5. The actual configuration values of the selected configuration set are presented in the configuration list shown in Wizard 6.

The selected configuration can now be confirmed (automatically or manually) and configured to the selected function so that the selected function can make its decision (e.g., CM output) accordingly. If not, the function-specific similarity measure can be updated to find another configuration set, or another function can be re-selected and the process is repeated from Wizard 3.

7.2.3 Refining Function Specific Similarity Definition and its Match Results

In addition to the automatic confirmation of a matching result, a user can also take the manual control of the confirmation when needed. In this mode, the user can explore the current search results to see if more accurate results for the context are needed. Figure 7 demonstrates such a situation, in which the user has decided to refine the results by reducing the value range of the *RLF OUTPUT* shown in Entry A and the elevation value shown in Entry B.



Fig. 7. A snapshot of the match result and configuration via refining the values of the function specific similarity measure.

Figure 7 shows how the amount of cases has reduced (with respect to Figure 6.) for the *CCO-SURROUNDED* and for its configurations $v1$ and $v2$, shown in Entry C and

Entry D. The confidence levels of these elements have increased. Configuration set v_2 is now automatically selected as the preferred configuration for the user to confirm. The user could even pick and confirm the configuration set v_1 , if the user would prefer so instead.

8 Conclusion and Discussion

This work provides a solution to define and apply two types of similarity measures for two self-operation use cases. The use cases are as follows: Self-operation uses its learned operation experiences to answer the question “Given any objective and its corresponding network context, what function should be used to achieve it?” Self-operation uses its learned operation experiences to answer the question “Given any objective and network context, what should be the suitable configuration for that function so that it could achieve the objective?”

The solution consists of the self-operation architecture for similarity measure definition and application, the data elements needed by the two types of similarity measures, and their definition and application procedures.

This work also describes a demonstrator implementation of self-operation, which learns operation experiences into self-operation cases and applies operation experiences for certain self-operation use cases. These use cases receive their configuration or instruction from self-operation with the means not shown in the demo architecture. The demonstrator uses an LTE simulator and SON function instances as a source of data, where the LTE simulator simulates a whole LTE network. In the experiments, the demonstrator defines an objective-specific similarity measure based on the given network context, objective, and rule. It then matches the corresponding self-operation cases with the defined objective-specific similarity measure. The best suitable function is extracted from the matching self-operation cases by the demonstrator. The demonstrator then automatically defines a function-specific similarity measure based on the selected function and the given network context, objective, and rule. The more relevant self-operation cases are further matched with the defined function-specific similarity measure by the demonstrator. The (best) suitable function configuration is extracted from the further matched self-operation cases by the demonstrator. The function can then be configured with the suitable configuration and activated to achieve the given objective.

The experimental results of the demonstrator (including the implemented solution, proposed by this paper) show the concept of self-operation (including the solution) work well as expected. This self-operation scales well (with respect to use of a distributed database, MongoDB) and works automatically while being able to interact with human operator through UI during the network operations.

The network operations of 5G networks are expected to have much more automation capabilities when compared with the current network operations. The proposed solution for self-operation by this paper serves naturally as an important part of the 5G network operations.

As the future work, the current demonstrator is expected to be enhanced to support the direct configuration of the functions and the network from UI. In addition, a machine to machine interface is expected to be added to the demonstrator so that self-operation can directly control and configure the functions and the network.

9 References

1. 3GPP, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuration and self-optimizing network use cases and solutions (Release 8)," 3GPP TR 36.902 V1.0.1, Sept. 2008.
2. NGMN, "NGMN Use Cases related to Self Organising Network, Overall Description," Deliverable (<https://www.ngmn.org/uploads/media/>), NGMN Alliance, Dec. 2008.
3. Nokia, "Business aware traffic steering," White Paper of Nokia Networks, http://networks.nokia.com/sites/default/files/document/nokia_traffic_steering_white_paper.pdf, 2013.
4. 5G-PPP, "5G Empowering Vertical Industries," Brochure (https://5g-ppp.eu/wp-content/uploads/2016/02/BROCHURE_5PPP_BAT2_PL.pdf), February 2016.
5. Haitao Tang and Kaj Stenberg, "Self-Operation of a Network," Proceedings of IEEE DataCom 2016, pp. 647-653, Aug. 2016.
6. Agnar Aamodt and Enric Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches," *Artificial Intelligence Communications* 7: 1, pp. 39-52, 1994.
7. Ahmad Rawashdeh and Anca L. Ralescu, "Similarity Measure for Social Networks– A Brief Survey," Proceedings of Modern AI and Cognitive Science Conference (MAICS), pp. 153-159, April 2015.
8. Djamel Guessoum, Moeiz Miraoui, and Chakib Tadj, "Survey of Semantic Similarity Measures in Pervasive Computing," *International Journal of Smart Sensing and Intelligent Systems*, VOL. 8, NO. 1, pp.125-158, March 2015.
9. Wael H Gomaa and Aly A Fahmy, "A Survey of Text Similarity Approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13-18, April 2013.
10. TRAI, "The Standards of Quality of Service for Wireless Data Services (Amendment) Regulations," Regulation (http://www.trai.gov.in/Content/Regulation/1_0_RegulationUser.aspx), Telecom Regulatory Authority of India, New Delhi, India, July 2014.
11. 3GPP, "Key Performance Indicators (KPI) for Evolved Universal Terrestrial Radio Access Network (E-UTRAN): Definitions, v13.0.0," 3GPP TS32.450, Jan 2016.
12. TRAI, "The Indian Telecom Services Performance Indicators, July - September, 2015," Indicator Report (http://www.trai.gov.in/WriteReadData/PIRReport/Documents/Indicator_Reports.pdf) Telecom Regulatory Authority of India, New Delhi, India, pp.61-84, Feb. 2016.
13. NGMN, "NGMN Informative List of SON Use Cases," An Annex Deliverable (https://www.ngmn.org/uploads/media/NGMN_Informative_List_of_SON_Use_Cases.pdf), NGMN Alliance, pp.6-47, April 2007.
14. Tobias Bandh, Haitao Tang, Henning Sanneck, and Christoph Schmelz, "SON Operation," Chapter 9 (pp.322-356), *LTE Self-Organizing Networks (SON)*, WILEY, ISBN 978-1-119-97067-5, 2012.
15. Ingo Viering, Martin Döttling, and Andreas Lobinger, "A mathematical perspective of self-optimizing wireless networks," Proc. ICC'09, p.1 ff., Dresden, Germany, June 2009.