



(51) International Patent Classification:
H04W 24/02 (2009.01)

(21) International Application Number:
PCT/EP2022/063572

(22) International Filing Date:
19 May 2022 (19.05.2022)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: NOKIA SOLUTIONS AND NETWORKS OY [FI/FI]; Karakaari 7, 02610 Espoo (FI).

(72) Inventors: LIAO, Qi; Wiederholdstr. 15A, 70174 Stuttgart (DE). K, Senthil Kumaran; No. 55, 10th Cross, Visweswaraya Layout, Akash Nagar, B.Narayanapura Extn, Bangalore, Karnataka 560016 (IN). WELLINGTON, Dan;

1120 Drexel St, Boulder, Colorado 80305 (US). AWADA, Ahmad; Fritz-Bauer-Strasse 13, 81249 Munich (DE).

(74) Agent: PAPULA OY; P.O. Box 981, 00101 Helsinki (FI).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH,

(54) Title: NETWORK OPTIMIZATION BASED ON DISTRIBUTED MULTI-AGENT MACHINE LEARNING WITH MINIMAL INTER-AGENT DEPENDENCY

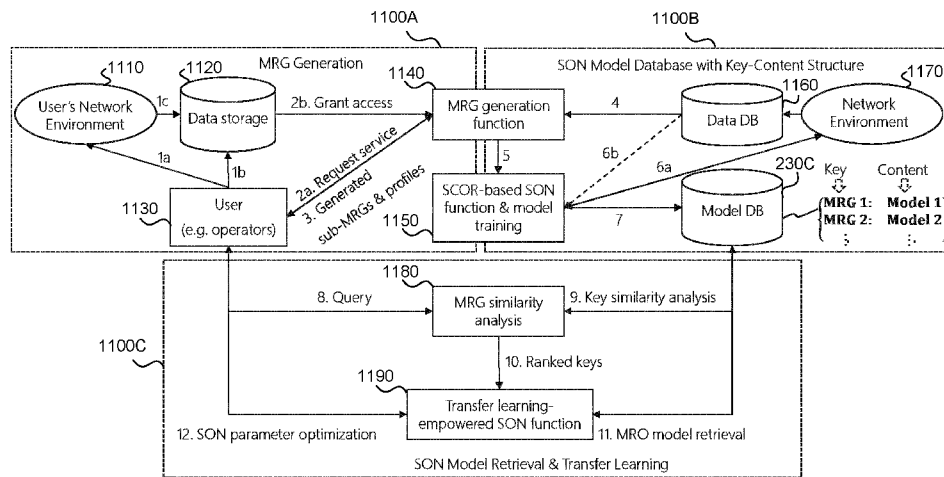


FIG. 11

(57) Abstract: Network optimization based on distributed multi-agent machine learning with minimal inter-agent dependency is disclosed. At least some of the embodiments may allow a distributed multi-agent deep reinforcement learning (DRL) algorithm for a mobility robustness optimization (MRO) problem, where each agent may comprise a varying number of physical or logical network boundaries. At least some of the embodiments may allow minimizing inter-agent dependencies by decomposing a network mobility graph. At least some of the embodiments may allow a transfer learning framework for self-organizing network (SON) model profiling, storage, retrieval, retraining, and management such that one can efficiently retrieve a SON model that was pre-trained in a similar (sub)network environment.



GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ,
TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,
TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,
DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,
LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

**NETWORK OPTIMIZATION BASED ON DISTRIBUTED MULTI-AGENT
MACHINE LEARNING WITH MINIMAL INTER-AGENT DEPENDENCY**

TECHNICAL FIELD

5 The disclosure relates generally to communica-
tions and, more particularly but not exclusively, to
network optimization based on distributed multi-agent
machine learning with minimal inter-agent dependency.

10 **BACKGROUND**

 While fifth generation (5G) mobile networks
have been emphasizing network virtualization, it is ex-
pected that sixth generation (6G) networks will focus
on autonomous intelligence of highly complex network
15 systems consisting of both physical and logical network
entities. For example, the introduction of network slic-
ing into self-organizing network (SON) functionalities
may lead to more complex optimization problems in the
following three aspects: 1) it may increase the dimen-
20 sions of network states by introducing slice-specific
key performance indicators (KPIs), 2) it may increase
the dimensions of optimization variables due to the
slice-specific network configuration parameters, and 3)
it may make the modeling of utility functions more dif-
25 ficult due to highly nonlinear inter-dependencies be-
tween high-dimensional parameters.

 Currently, when using machine learning to solve
network optimization problems, there is tradeoff between
a centralized (single agent) scheme and a distributed
30 (multi-agent) scheme: although training a single agent
in the centralized scheme can capture inter-cell de-
pendencies, it may require an extremely long period of
exploration and cause slow convergence, if it converges
at all, due to an intractably high-dimensional action
35 space. On the other hand, the distributed scheme decom-
poses a system consisting of many network entities into

subsystems, e.g., optimizing on the cell or cell pair basis, which reduces the complexity and accelerates the learning process, but the neglecting of inter-agent dependency may lead to poor performance due to inaccurate modeling based on limited information. Neglecting the inter-agent dependency may also lead to longer convergence times.

Thus, at least in some situations, there may be a need for network optimization based on distributed multi-agent machine learning with minimal inter-agent dependency. Moreover, training many distributed agents faces challenges, such as cost of data collection and storage, learning time, algorithm scalability, and artificial intelligence (AI) / machine learning (ML) model reproducibility. Thus, at least in some situations, there may be a need for an automatic workflow that can detect a similarity between the agents and reuse the knowledge and models in order to avoid having to learn from scratch for a large amount of the distributed agents.

SUMMARY

The scope of protection sought for various example embodiments of the invention is set out by the independent claims. The example embodiments and features, if any, described in this specification that do not fall under the scope of the independent claims are to be interpreted as examples useful for understanding various example embodiments of the invention.

An example embodiment of a communications network device comprises at least one processor, and at least one memory including computer program code. The at least one memory and the computer program code are configured to, with the at least one processor, cause the communications network device at least to decompose a communications network into service level agreement, SLA, coverage overlap regions, SCORs, according to

mobility relations between logical network entity, LNE, pairs within the communication network. The SCOR comprises at least one LNE pair. The at least one memory and the computer program code are further configured to, with the at least one processor, cause the communications network device at least to assign a machine learning agent to at least one of the decomposed SCORs. The machine learning agent is configured to apply a deep reinforcement learning model to solve an optimization problem related to a self-organizing network, SON, function within its assigned SCOR.

In an example embodiment, alternatively or in addition to the above-described example embodiments, LNE pairs in a SCOR comprising at least two LNE pairs are strongly coupled, and dependency between the SCORs is low.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the at least one memory and the computer program code are further configured to, with the at least one processor, cause the communications network device to decompose the communications network into the SCORs by generating a logical network graph corresponding to the communications network and representing the mobility relations between the LNE pairs, and by decomposing the logical network graph into subgraphs. The subgraphs represent SCORs comprising strongly coupled LNE pairs.

In an example embodiment, alternatively or in addition to the above-described example embodiments, vertices of the logical network graph comprise the LNE pairs, and weights of edges of the logical network graph reflect a mobility relationship between two LNE pairs.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the at least one memory and the computer program code are further configured to, with the at least one processor, cause the communications network device at least to

generate a profile for the subgraph. The profile comprises an adjacency matrix or an adjacency list representing the respective subgraph.

5 In an example embodiment, alternatively or in addition to the above-described example embodiments, the profile further comprises at least one of: a number of vertices, a number of edges, a number of involved LNEs, a degree distribution, a distribution of edge weights, a distribution of summed weights of edges incident to a
10 vertex, or at least one LNE specific feature for the respective subgraph including at least one of a deployment type, an LNE type, an associated user mobility distribution, position information, or an LNE load state.

15 In an example embodiment, alternatively or in addition to the above-described example embodiments, the at least one memory and the computer program code are further configured to, with the at least one processor, cause the communications network device at least to
20 obtain the deep reinforcement learning model as pre-trained from a SON node device.

In an example embodiment, alternatively or in addition to the above-described example embodiments, states of the assigned machine learning agent comprise
25 at least one of: LNE -specific metrics, LNE pair -specific metrics, or contextual information for capturing at least one of temporal or spatial correlations.

In an example embodiment, alternatively or in addition to the above-described example embodiments, an
30 action space of the assigned machine learning agent comprises a discrete action space or a continuous action space.

In an example embodiment, alternatively or in addition to the above-described example embodiments,
35 rewards for the assigned machine learning agent are based on at least one of: LNE pair -specific handover

performance metrics, LNE -specific quality of service, QoS, performance metrics, or LNE pair -specific QoS performance metrics.

5 In an example embodiment, alternatively or in addition to the above-described example embodiments, the SON function comprises a mobility robustness optimization, MRO, function, a coverage and capacity optimization function, or a mobility load balancing function.

10 In an example embodiment, alternatively or in addition to the above-described example embodiments, the MRO function comprises optimization of one or more handover parameters.

15 In an example embodiment, alternatively or in addition to the above-described example embodiments, the SCOR further comprises a group of physical cell boundaries, a group of logical cell boundaries, or a group of physical cell boundaries and logical cell boundaries.

20 In an example embodiment, alternatively or in addition to the above-described example embodiments, the LNEs comprise at least one of cells, slices, or QoS flows.

25 In an example embodiment, alternatively or in addition to the above-described example embodiments, the generating of the logical network graph comprises generating the logical network graph based on historical LNE data, statistical mobility data, or an SLA coverage map.

30 An example embodiment of a communications network device comprises means for decomposing a communications network into service level agreement, SLA, coverage overlap regions, SCORs, according to mobility relations between logical network entity, LNE, pairs within the communication network. The SCOR
35 comprises at least one LNE pair. The means are further configured to assign a machine learning agent to at

least one of the decomposed SCORs. The machine learning agent is configured to apply a deep reinforcement learning model to solve an optimization problem related to a self-organizing network, SON, function within its
5 assigned SCOR.

In an example embodiment, alternatively or in addition to the above-described example embodiments, LNE pairs in a SCOR comprising at least two LNE pairs are strongly coupled, and dependency between the SCORs is
10 low.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the means are further configured to decompose the communications network into the SCORs by generating a
15 logical network graph corresponding to the communications network and representing the mobility relations between the LNE pairs, and by decomposing the logical network graph into subgraphs. The subgraphs represent SCORs comprising strongly coupled LNE pairs.

In an example embodiment, alternatively or in addition to the above-described example embodiments, vertices of the logical network graph comprise the LNE pairs, and weights of edges of the logical network graph reflect a mobility relationship between two LNE pairs.
20

In an example embodiment, alternatively or in addition to the above-described example embodiments, the means are further configured to generate a profile for the subgraph. The profile comprises an adjacency matrix or an adjacency list representing the respective
25 subgraph.
30

In an example embodiment, alternatively or in addition to the above-described example embodiments, the profile further comprises at least one of: a number of vertices, a number of edges, a number of involved LNEs,
35 a degree distribution, a distribution of edge weights, a distribution of summed weights of edges incident to a vertex, or at least one LNE specific feature for the

respective subgraph including at least one of a deployment type, an LNE type, an associated user mobility distribution, position information, or an LNE load state.

5 In an example embodiment, alternatively or in addition to the above-described example embodiments, the means are further configured to obtain the deep reinforcement learning model as pretrained from a SON node device.

10 In an example embodiment, alternatively or in addition to the above-described example embodiments, states of the assigned machine learning agent comprise at least one of: LNE -specific metrics, LNE pair -specific metrics, or contextual information for
15 capturing at least one of temporal or spatial correlations.

 In an example embodiment, alternatively or in addition to the above-described example embodiments, an action space of the assigned machine learning agent
20 comprises a discrete action space or a continuous action space.

 In an example embodiment, alternatively or in addition to the above-described example embodiments, rewards for the assigned machine learning agent are
25 based on at least one of: LNE pair -specific handover performance metrics, LNE -specific quality of service, QoS, performance metrics, or LNE pair -specific QoS performance metrics.

 In an example embodiment, alternatively or in addition to the above-described example embodiments, the
30 SON function comprises a mobility robustness optimization, MRO, function, a coverage and capacity optimization function, or a mobility load balancing function.

35 In an example embodiment, alternatively or in addition to the above-described example embodiments, the

MRO function comprises optimization of one or more handover parameters.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the SCOR further comprises a group of physical cell boundaries, a group of logical cell boundaries, or a group of physical cell boundaries and logical cell boundaries.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the LNEs comprise at least one of cells, slices, or QoS flows.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the generating of the logical network graph comprises generating the logical network graph based on historical LNE data, statistical mobility data, or an SLA coverage map.

An example embodiment of a method comprises decomposing, by a communications network device, a communications network into service level agreement, SLA, coverage overlap regions, SCORs, according to mobility relations between logical network entity, LNE, pairs within the communication network. The SCOR comprises at least one LNE pair. The method further comprises assigning, by the communications network device, a machine learning agent to at least one of the decomposed SCORs. The machine learning agent is configured to apply a deep reinforcement learning model to solve an optimization problem related to a self-organizing network, SON, function within its assigned SCOR.

In an example embodiment, alternatively or in addition to the above-described example embodiments, LNE pairs in a SCOR comprising at least two LNE pairs are strongly coupled, and dependency between the SCORs is low.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the method further comprises decomposing the communications network into the SCORs by generating a logical network graph corresponding to the communications network and
5 representing the mobility relations between the LNE pairs, and by decomposing the logical network graph into subgraphs. The subgraphs represent SCORs comprising strongly coupled LNE pairs.

10 In an example embodiment, alternatively or in addition to the above-described example embodiments, vertices of the logical network graph comprise the LNE pairs, and weights of edges of the logical network graph reflect a mobility relationship between two LNE pairs.

15 In an example embodiment, alternatively or in addition to the above-described example embodiments, the method further comprises generating a profile for the subgraph. The profile comprises an adjacency matrix or an adjacency list representing the respective subgraph.

20 In an example embodiment, alternatively or in addition to the above-described example embodiments, the profile further comprises at least one of: a number of vertices, a number of edges, a number of involved LNEs, a degree distribution, a distribution of edge weights,
25 a distribution of summed weights of edges incident to a vertex, or at least one LNE specific feature for the respective subgraph including at least one of a deployment type, an LNE type, an associated user mobility distribution, position information, or an LNE
30 load state.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the method further comprises obtaining the deep reinforcement learning model as pretrained from a SON node
35 device.

In an example embodiment, alternatively or in addition to the above-described example embodiments,

states of the assigned machine learning agent comprise at least one of: LNE -specific metrics, LNE pair -specific metrics, or contextual information for capturing at least one of temporal or spatial correlations.

In an example embodiment, alternatively or in addition to the above-described example embodiments, an action space of the assigned machine learning agent comprises a discrete action space or a continuous action space.

In an example embodiment, alternatively or in addition to the above-described example embodiments, rewards for the assigned machine learning agent are based on at least one of: LNE pair -specific handover performance metrics, LNE -specific quality of service, QoS, performance metrics, or LNE pair -specific QoS performance metrics.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the SON function comprises a mobility robustness optimization, MRO, function, a coverage and capacity optimization function, or a mobility load balancing function.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the MRO function comprises optimization of one or more handover parameters.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the SCOR further comprises a group of physical cell boundaries, a group of logical cell boundaries, or a group of physical cell boundaries and logical cell boundaries.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the LNEs comprise at least one of cells, slices, or QoS flows.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the generating of the logical network graph comprises generating the logical network graph based on historical
5 LNE data, statistical mobility data, or an SLA coverage map.

An example embodiment of a computer program comprises instructions for causing a communications network device to perform at least the following:
10 decomposing a communications network into service level agreement, SLA, coverage overlap regions, SCORs, according to mobility relations between logical network entity, LNE, pairs within the communication network, the SCOR comprising at least one LNE pair; and assigning a
15 machine learning agent to at least one of the decomposed SCORs. The machine learning agent is configured to apply a deep reinforcement learning model to solve an optimization problem related to a self-organizing network, SON, function within its assigned SCOR.

20 An example embodiment of a self-organizing network, SON, node device comprises at least one processor, and at least one memory including computer program code. The at least one memory and the computer program code are configured to, with the at least one
25 processor, cause the SON node device at least to receive from a communications network device a request for a pretrained deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level
30 agreement, SLA, coverage overlap region, SCOR. The request comprises at least one profile of at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs. The SCOR comprises at least one LNE pair and the
35 at least one subgraph corresponds to the SCOR. The at least one memory and the computer program code are further configured to, with the at least one processor,

cause the SON node device at least to determine a pretrained reference deep reinforcement learning model from a model database by a similarity analysis based on the profile of the subgraph. The at least one memory and
5 the computer program code are further configured to, with the at least one processor, cause the SON node device at least to transmit the determined pretrained reference deep reinforcement learning model to the communications network device.

10 In an example embodiment, alternatively or in addition to the above-described example embodiments, the similarity analysis comprises a relational similarity analysis between the at least one subgraph in the received at least one profile and subgraphs associated
15 with the pretrained reference deep reinforcement learning models stored in the model database.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the at least one memory and the computer program code are
20 configured to, with the at least one processor, further cause the SON node device to customize the determined reference deep reinforcement learning model and transmit the customized deep reinforcement learning model to the communications network device.

25 In an example embodiment, alternatively or in addition to the above-described example embodiments, the at least one memory and the computer program code are configured to, with the at least one processor, further cause the SON node device to prescreen the profile of
30 the subgraph based on one or more prescreening parameters to determine one or more candidate pretrained reference deep reinforcement learning models. The determining of the pretrained reference deep reinforcement learning model is performed on the determined one
35 or more candidate pretrained reference deep reinforcement learning models.

An example embodiment of a self-organizing network, SON, node device comprises means for receiving from a communications network device a request for a pretrained deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR. The request comprises at least one profile of at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs. The SCOR comprises at least one LNE pair, and the at least one subgraph corresponds to the SCOR. The means are further configured to determine a pretrained reference deep reinforcement learning model from a model database by a similarity analysis based on the profile of the subgraph. The means are further configured to transmit the determined pretrained reference deep reinforcement learning model to the communications network device.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the similarity analysis comprises a relational similarity analysis between the at least one subgraph in the received at least one profile and subgraphs associated with the pretrained reference deep reinforcement learning models stored in the model database.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the means are further configured to customize the determined reference deep reinforcement learning model and transmit the customized deep reinforcement learning model to the communications network device.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the means are further configured to prescreen the profile of the subgraph based on one or more prescreening parameters to determine one or more candidate pretrained

reference deep reinforcement learning models. The determining of the pretrained reference deep reinforcement learning model is performed on the determined one or more candidate pretrained reference deep reinforcement learning models.

An example embodiment of a method comprises receiving at a self-organizing network, SON, node device from a communications network device a request for a pretrained deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR. The request comprises at least one profile of at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs. The SCOR comprises at least one LNE pair and the at least one subgraph corresponds to the SCOR. The method further comprises determining, by the SON node device, a pretrained reference deep reinforcement learning model from a model database by a similarity analysis based on the profile of the subgraph. The method further comprises transmitting, by the SON node device, the determined pretrained reference deep reinforcement learning model to the communications network device.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the similarity analysis comprises a relational similarity analysis between the at least one subgraph in the received at least one profile and subgraphs associated with the pretrained reference deep reinforcement learning models stored in the model database.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the method further comprises customizing the determined reference deep reinforcement learning model and transmitting the customized deep reinforcement learning model to the communications network device.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the method further comprises prescreening the profile of the subgraph based on one or more prescreening parameters to determine one or more candidate pretrained reference deep reinforcement learning models. The determining of the pretrained reference deep reinforcement learning model is performed on the determined one or more candidate pretrained reference deep reinforcement learning models.

An example embodiment of a computer program comprises instructions for causing a self-organizing network, SON, node device to perform at least the following: receiving from a communications network device a request for a pretrained deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR, the request comprising at least one profile of at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs, with the SCOR comprising at least one LNE pair and the at least one subgraph corresponding to the SCOR; determining a pretrained reference deep reinforcement learning model from a model database by a similarity analysis based on the profile of the subgraph; and transmitting the determined pretrained reference deep reinforcement learning model to the communications network device.

An example embodiment of a network service provider device comprises at least one processor, and at least one memory including computer program code. The at least one memory and the computer program code are configured to, with the at least one processor, cause the network service provider device at least to obtain from at least one communications network device at least one subgraph of a logical network graph representing

mobility relations between logical network entity, LNE, pairs and their associated profiles. The at least one memory and the computer program code are further configured to, with the at least one processor, cause the network service provider device at least to derive at least one pretrained reference deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR. The SCOR comprises at least one LNE pair and the SCOR corresponds to the subgraph. The at least one memory and the computer program code are further configured to, with the at least one processor, cause the network service provider device at least to store the obtained profiles and the derived pretrained reference deep reinforcement learning models in a model database.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the model database comprises an indexed model database such that a same index represents an obtained profile of one subgraph and a corresponding derived pretrained reference deep reinforcement learning model in a SCOR that corresponds to the subgraph.

An example embodiment of a network service provider device comprises means for obtaining from at least one communications network device at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs and their associated profiles. The means are further configured to derive at least one pretrained reference deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR. The SCOR comprises at least one LNE pair and the SCOR corresponds to the subgraph. The means are further configured to store the

obtained profiles and the derived pretrained reference deep reinforcement learning models in a model database.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the model database comprises an indexed model database such that a same index represents an obtained profile of one subgraph and a corresponding derived pretrained reference deep reinforcement learning model in a SCOR that corresponds to the subgraph.

An example embodiment of a method comprises obtaining, by a network service provider device, from at least one communications network device at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs and their associated profiles. The method further comprises deriving, by the network service provider device, at least one pretrained reference deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR. The SCOR comprises at least one LNE pair and the SCOR corresponds to the subgraph. The method further comprises storing, by the network service provider device, the obtained profiles and the derived pretrained reference deep reinforcement learning models in a model database.

In an example embodiment, alternatively or in addition to the above-described example embodiments, the model database comprises an indexed model database such that a same index represents an obtained profile of one subgraph and a corresponding derived pretrained reference deep reinforcement learning model in a SCOR that corresponds to the subgraph.

An example embodiment of a computer program comprises instructions for causing a network service provider device to perform at least the following: obtaining from at least one communications network

device at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs and their associated profiles; deriving at least one pretrained reference deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR, with the SCOR comprising at least one LNE pair and the SCOR corresponding to the subgraph; and storing the obtained profiles and the derived pretrained reference deep reinforcement learning models in a model database.

DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are included to provide a further understanding of the embodiments and constitute a part of this specification, illustrate embodiments and together with the description help to explain the principles of the embodiments. In the drawings:

FIG. 1 shows an example embodiment of the subject matter described herein illustrating an example system, where various embodiments of the present disclosure may be implemented;

FIG. 2A shows an example embodiment of the subject matter described herein illustrating a communications network device;

FIG. 2B shows an example embodiment of the subject matter described herein illustrating a SON node device;

FIG. 2C shows an example embodiment of the subject matter described herein illustrating a network service provider device;

FIG. 3A shows an example embodiment of the subject matter described herein illustrating a SCOR and SCOR-based agents;

FIG. 3B shows an example embodiment of the subject matter described herein illustrating deriving of a logical network graph and SCORs from an LNE log history;

5 **FIG. 4** shows an example embodiment of the subject matter described herein illustrating a communications network partitioned to SCORs and their corresponding subgraphs;

10 **FIG. 5** shows an example embodiment of the subject matter described herein illustrating an example representation of a subgraph with an adjacency matrix or an adjacency list;

FIG. 6 shows another example embodiment of the subject matter described herein illustrating an example of subgraph profiling;

15 **FIG. 7** shows an example embodiment of the subject matter described herein illustrating an example of an indexed profile database and an indexed model database;

20 **FIG. 8** shows an example embodiment of the subject matter described herein illustrating an example of an actor network in a deep reinforcement learning architecture;

25 **FIG. 9** shows an example embodiment of the subject matter described herein illustrating an example of direct application of a source model $M_s(G_s)$ for a target model $M_*(G_*)$;

30 **FIG. 10** shows an example embodiment of the subject matter described herein illustrating an example of knowledge distillation from a source model $M_s(G_s)$ to a target model $M_*(G_*)$;

FIG. 11 shows an example embodiment of the subject matter described herein illustrating a transfer learning framework with logical network graph -based model retrieval;

35 **FIG. 12** shows an example embodiment of the subject matter described herein illustrating a method;

FIG. 13 shows an example embodiment of the subject matter described herein illustrating another method;

FIG. 14 shows an example embodiment of the subject matter described herein illustrating yet another method;

FIG. 15 shows an example embodiment of the subject matter described herein illustrating deriving an MRG and SCORs from an SLA coverage map;

FIG. 16 shows an example embodiment of the subject matter described herein illustrating MRG decomposition when an LNE is defined as a network slice in a physical cell;

FIG. 17 shows an example embodiment of the subject matter described herein illustrating an example of a deep deterministic policy gradient (DDPG) architecture; and

FIG. 18 shows an example embodiment of the subject matter described herein illustrating an example of a deep Q-network (DQN) architecture.

Like reference numerals are used to designate like parts in the accompanying drawings.

DETAILED DESCRIPTION

Reference will now be made in detail to embodiments, examples of which are illustrated in the accompanying drawings. The detailed description provided below in connection with the appended drawings is intended as a description of the present examples and is not intended to represent the only forms in which the present example may be constructed or utilized. The description sets forth the functions of the example and the sequence of steps for constructing and operating the example. However, the same or equivalent functions and sequences may be accomplished by different examples.

Fig. 1 illustrates an example system 100, where various embodiments of the present disclosure may be

implemented. The system 100 may comprise a fifth generation (5G) or a sixth generation (6G) communications network 110. An example representation of the system 100 is shown depicting a communications network device 200,
5 a self-organizing network (SON) node device 210, and a network service provider device 220. At least in some embodiments, the 5G or 6G network may comprise one or more massive machine-to-machine (M2M) network(s), massive machine type communications (mMTC) network(s), internet of things (IoT) network(s), industrial internet-of-things (IIoT) network(s), enhanced mobile broadband (eMBB) network(s), ultra-reliable low-latency communication (URLLC) network(s), and/or the like. In other words, the 5G or 6G network may be configured to serve
10 diverse service types and/or use cases, and it may logically be seen as comprising one or more networks.

The communications network device 200 may comprise an operations, administration, and maintenance (OAM) unit, a network node device, or a client device.
20 The client device may include, e.g., a mobile phone, a smartphone, a tablet computer, a smart watch, or any hand-held, portable and/or wearable device. The client device may also be referred to as a user equipment (UE). The network node device may comprise a base station. The
25 base station may include, e.g., a 5G or 6G base station (gNB) or any such device suitable for providing an air interface for client devices to connect to a wireless network via wireless transmissions.

At least some of the disclosed embodiments may
30 be implemented in an O-RAN architecture. The O-RAN aims for interoperability and standardization of RAN elements including a unified interconnection standard for network functions from different vendors. The O-RAN architecture provides a foundation for building a virtualized RAN on
35 open hardware with an embedded artificial intelligence (AI) -powered radio control.

In the following, handover (HO) process and parameters are briefly discussed.

A condition for sending a measurement report from a client device to a serving cell, which triggers the handover process, is generally based on a reference signal received power (RSRP) and per cell or per cell pair HO parameters. For example, the HO parameters may be defined per network entity or per network entity pair. For example, they may be defined per physical cell or cell pair, or per logical cell or cell pair (e.g., two logical slices from different physical cells), or per a single client device or client device groups. Herein, "cell" and "cell pair" are used as examples, but the disclosure is not limited to physical or logical cells or cell pairs. HO parameters include:

- per cell pair cell individual offset (CIO): $O_{n,m}$ denotes the cell individual offset (CIO) of a neighboring cell m specified in a source cell n , while $O_{m,n}$ denotes the CIO of the cell n specified in the cell m . The cell pair CIOs are not reciprocal, i.e., $O_{n,m}$ and $O_{m,n}$ may be set to different values;

- per cell A3 event offset (OFS) and hysteresis (HYS): the OFS and HYS in the cell n are denoted by OFS_n and H_n , respectively; and

- per cell time-to-trigger (TTT): T_n denotes the TTT defined in the cell n .

A UE k sends a measurement report triggering a handover from the cell n to the cell m if the following condition is fulfilled for a duration of time T_n :

30

$$M_{m,k} + O_{n,m} \geq M_{n,k} + O_{m,n} + OFS_n + H_n \quad (\text{Eq. 1})$$

where $M_{n,k}$ and $M_{m,k}$ are signal measurements (e.g., RSRP) received in the UE k from the source cell n and the neighboring cell m respectively.

35

At least some HO optimization schemes may aim to reduce at least some of the following four HO events:

- a too-late-handover (HOL): this may occur when a user is leaving the coverage area of its serving cell n towards the target cell m but the handover is not triggered which causes a radio link failure (RLF) of the
5 UE;

- a too-early-handover (HOE): this may occur when the HO decision is made too early. The neighboring cell cannot provide a sustainable signal quality to the user and an RLF happens right after the handover. The
10 UE reconnects to the previous source cell of the handover. Random access failures to the target cell (due to too early triggering of handover) may also be considered as a part of the HOE;

- a wrong-cell-handover (HOW): this may occur
15 when a user is handed over from a source cell to a wrong target cell. An RLF is detected shortly after a successful handover to the target cell and then the user is connected to another neighboring cell that is different from the source cell; and

- a ping-pong-handover (HOPP): this may occur
20 when a user is handed over from the cell n to the cell m, but after a short time period the UE is handed over back from the cell m back to the cell n.

In the following, various example embodiments
25 will be discussed. At least some of these example embodiments may allow network optimization based on distributed multi-agent machine learning with minimal inter-agent dependency.

At least some of these example embodiments may
30 allow a distributed multi-agent deep reinforcement learning (DRL) scheme for a service level agreement (SLA) -guaranteed mobility robustness optimization (MRO) in which an agent may be defined based on an SLA coverage overlap region (SCOR for short) to minimize
35 inter-agent dependency. The SLA coverage overlap region may be represented by a group of physical or logical cell boundaries and each agent may jointly optimize the

CIOs and TTTs of the grouped cell boundaries. Herein, "logical" cell boundary refers to the SLA coverage overlap area between two logical network entities, and "cell boundary" refers either to a physical cell boundary or to a logical cell boundary (e.g., between two logical slices from different physical cells).

At least some of these example embodiments may allow grouping the cell boundaries and identifying the SCORs based on a partitioned or decomposed logical network graph or mobility relation graph (MRG). When a customer (e.g., a communication service provider, CSP) makes a query for pretrained models of a similar SCOR area, the profile of an MRG (e.g., an adjacent matrix or index of an MRG class, or other extracted features) may then be used as key to retrieve the SCOR-based artificial intelligence (AI) or machine learning (ML) model in a reference model database (DB).

Herein, the terms "partition" and "decompose" are used interchangeably. Herein, the terms "logical network graph" and "mobility relation graph" and "MRG" are used interchangeably. Herein, the terms "sub-MRG" and "SCOR-based subgraph" are used interchangeably.

Herein, when the term "user" is used to refer to a user mobility pattern or a user mobility distribution, it comprises a user equipment or a client device. When the term "user" is used to refer to a user who sends a query, it comprises a customer, such as a communications service provider (CSP) or an operator, who owns the data and network infrastructure and requests network optimization and transfer learning services in accordance with the disclosure.

At least some of these example embodiments may allow MRG generation in which users (e.g., CSPs) may request an MRG generation service to decompose a large-scale network to a set of subnetworks with minimum inter-subnetwork dependency. The service may provide the users the generated MRG and its sub-MRGs corresponding

to the decomposed subnetworks. Such decomposition may provide a proper granularity of deploying distributed machine learning agents, which may achieve a good tradeoff between low model complexity and good model accuracy.

At least some of these example embodiments may allow building a SCOR-based reference model DB 230C in which by using the collected historical data, it is possible to obtain the SCORs and the SCOR-based sub-graphs, namely, the corresponding sub-MRGs of the SCORs, and pre-train an MRO model for each SCOR. These sub-MRGs and the trained models may be used to construct the reference model DB 230C with a key-content structure. The profiles of the sub-MRGs (e.g., adjacent matrices) may be the keys, and the pre-trained models may be the contents. An alternative may be to build an index-profile DB 230A and an index-model DB 230B (as shown in diagram 700 of Fig. 7).

At least some of these example embodiments may allow ML model retrieval and transfer learning in which, when a user requests the transfer learning -based MRO service, it may first send the profile of the generated sub-MRGs as a query. Then, an MRG similarity analysis may compare the query with the keys in the reference model DB and return highly ranked keys. The transfer learning -empowered SON functions, e.g., an MRO function, may retrieve the highly ranked models, and either directly use the models for the user's network, or further fine-tune them with transfer learning. In this way, it is possible to build reusable and reproducible MRO models for a multi-vendor open environment.

In other words, at least some of these example embodiments may allow distributed multi-agent DRL for an MRO problem in which each agent may comprise a varying number of physical or logical network boundaries. The inter-agent dependencies may be minimized to improve

robustness of the ML model by creating the agents with the decomposed network mobility graph.

Accordingly, at least some of these example embodiments may allow a transfer learning framework for SON functions such as the MRO, such that the ML models may be transferred between similar distributed agents, even when the agents comprise different numbers of grouped network entities (or boundaries between network entities).

Accordingly, at least some of these example embodiments may allow SON model profiling, storage, and management such that one can efficiently retrieve a SON model that was pre-trained in a similar (sub)network environment. The similarity between the relational network graph data may be analyzed. This approach may be beneficial, e.g., for SON use cases because it captures the strong interaction between the network entities within a local agent.

It is to be noted that the disclosure is not restricted to the MRO function only. Rather, it can also be applied to other SCOR-based SON functions, such as coverage and capacity optimization and mobility load balancing. Furthermore, it is to be noted that the reference model DB may be generalized to a reference container DB in which each pretrained SCOR-based SON function may be packaged in a container and saved in the DB. By sending a query including the profile of the sub-MRG, the user may retrieve a packaged pre-trained model, e.g., a Docker image, of a similar mobility and SLA coverage region.

Fig. 2A is a block diagram of the communications network device 200, in accordance with an example embodiment.

The communications network device 200 comprises at least one processor 202 and at least one memory 204 including computer program code. The communications network device 200 may also include other elements, such

as a transceiver configured to enable the communications network device 200 to transmit and/or receive information to/from other devices, as well as other elements not shown in Fig. 2A. In one example, the communications network device 200 may use the transceiver to transmit or receive signaling information and data in accordance with at least one cellular communication protocol. The transceiver may be configured to provide at least one wireless radio connection, such as for example a 3GPP mobile broadband connection (e.g., 5G and/or 6G). The transceiver may be configured to be coupled to at least one antenna to transmit and/or receive radio frequency signals.

Although the communications network device 200 is depicted to include only one processor 202, the communications network device 200 may include more processors. In an embodiment, the memory 204 is capable of storing instructions, such as an operating system and/or various applications. Furthermore, the memory 204 may include a storage that may be used to store, e.g., at least some of the information and data used in the disclosed embodiments.

Furthermore, the processor 202 is capable of executing the stored instructions. In an embodiment, the processor 202 may be embodied as a multi-core processor, a single core processor, or a combination of one or more multi-core processors and one or more single core processors. For example, the processor 202 may be embodied as one or more of various processing devices, such as a coprocessor, a microprocessor, a controller, a digital signal processor (DSP), a processing circuitry with or without an accompanying DSP, or various other processing devices including integrated circuits such as, for example, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a microcontroller unit (MCU), a hardware accelerator, a spe-

cial-purpose computer chip, or the like. In an embodiment, the processor 202 may be configured to execute hard-coded functionality. In an embodiment, the processor 202 is embodied as an executor of software instructions, wherein the instructions may specifically configure the processor 202 to perform the algorithms and/or operations described herein when the instructions are executed.

The memory 204 may be embodied as one or more volatile memory devices, one or more non-volatile memory devices, and/or a combination of one or more volatile memory devices and non-volatile memory devices. For example, the memory 204 may be embodied as semiconductor memories (such as mask ROM, PROM (programmable ROM), EPROM (erasable PROM), flash ROM, RAM (random access memory), etc.).

The communications network device 200 may comprise an operations, administration, and maintenance (OAM) unit, a network node device (such as a base station), or a client device (such as a user equipment). The base station may include, e.g., a fifth-generation or sixth-generation base station (gNB) or any such device providing an air interface for client devices to connect to the wireless network via wireless transmissions.

The at least one memory 204 and the computer program code are configured to, with the at least one processor 202, cause the communications network device 200 at least to decompose or partition the communications network 110 into service level agreement (SLA) coverage overlap regions (SCORs) according to mobility relations between logical network entity (LNE) pairs within the communication network 110. The SCOR comprises at least one LNE pair.

Herein, the logical network entities are called "LNEs" for brevity. A network with a set of N LNEs may be denoted by $\mathcal{N} := \{1, \dots, N\}$. E.g., for a network with C

physical cells and S logical slices, each slice in each cell may be considered as an LNE, thus $N = CS$ if each cell has S slices. It is to be noted that an even finer granularity of the LNEs may be considered, e.g., per QoS flow instead of per slice.

Herein, the term SLA (service level agreement) coverage overlap region (or SCOR for brevity) refers to a group of LNE pairs that is optimized such that the dependency between the SCORs is low. Since the logical network graph is decomposed into a group of SCORs according to the mobility relations between LNE pairs, while the mobility relations between LNE pairs changes from time to time, the numbers of the LNE pairs can be different from one SCOR to another, and the composition of the LNE pairs in a SCOR may be dynamic and adaptable to a changing environment. E.g., a SCOR may include cell boundaries $A \rightarrow B$, $B \rightarrow A$, $B \rightarrow C$, $C \rightarrow B$, while conventional optimizations are performed either per cell base (all one-directional cell boundaries related to a single cell, i.e., if a cell A has neighboring cells B , C , D , then they have a fixed number of boundaries $A \rightarrow B$ and $A \rightarrow C$ and $A \rightarrow D$) or per cell pair -base (bidirectional cell boundaries of a single pair of cells, i.e., $A \rightarrow B$ and $B \rightarrow A$).

For each LNE, a neighboring LNE list may be defined. E.g., if a cell \mathbf{b} is in the neighboring cell list of a cell \mathbf{a} and service for a slice \mathbf{s} may be provided in both cells \mathbf{a} and \mathbf{b} , then the LNE (\mathbf{b}, \mathbf{s}) (representing slice \mathbf{s} in cell \mathbf{b}) may be included in the neighboring LNE list of LNE (\mathbf{a}, \mathbf{s}) . For brevity of notation, a set of all LNE pairs may be defined as $\mathcal{P} := \{(\mathbf{n}, \mathbf{m}) : \mathbf{n}, \mathbf{m} \in \mathcal{N}\}$. It is to be noted that each tuple in \mathcal{P} represents a directional LNE pair, e.g., from LNE $\mathbf{n} := (\mathbf{a}, \mathbf{s})$ to LNE $\mathbf{m} := (\mathbf{b}, \mathbf{s})$. It is to be further noted that the previously mentioned logical cell boundary may be considered as service overlap regions of an LNE pair.

LNE pair-specific HO parameters may be defined. E.g., for each LNE pair $(n,m) \in \mathcal{P}$, the CIO and TTT (denoted by $O_{n,m}$ and $T_{n,m}$, respectively) may be optimized. The HO criterion may remain in the same form as in Eq. (1), except that in replacing the cell pair, (n,m) are the LNE pair.

The following measurements for both LNE-specific and LNE pair-specific performance may be collected:

10 - LNE-specific network states denoted by $s_n, \forall n \in \mathcal{N}$, including but not limited to a per LNE number of users and load;

- LNE-specific SLA-related QoS metrics denoted by $q_n, \forall n \in \mathcal{N}$, including but not limited to a per LNE throughput and delay, and/or LNE pair-specific SLA-related QoS metrics denoted by $q_{n,m}, \forall (n,m) \in \mathcal{P}$; and/or

15 - LNE pair-specific HO-related metrics denoted by $r_{n,m}, \forall (n,m) \in \mathcal{P}$, including but not limited to per LNE pair ratios of HOE, HOL, HOW, and HOPP events.

20 For example, the communications network 110 may comprise a large-scale communications network. For example, at least some of the SCORs may further comprise a group of physical cell boundaries, a group of logical cell boundaries, or a group of physical cell boundaries and logical cell boundaries. For example, the LNEs may comprise cells, slices, and/or quality of service (QoS) flows.

In other words, each LNE may meet a predefined SLA for any mobility status. Thus, an objective may include guaranteeing the SLA services of all the LNEs while minimizing HO failures and unnecessary HOs by optimizing the LNE pair-specific CIOs $o := [o_{n,m}: \forall (n,m) \in \mathcal{P}] \in \Omega$ and TTTs $t := [t_{n,m}: \forall (n,m) \in \mathcal{P}] \in \mathcal{Z}$:

35
$$\begin{aligned} & \min_{o \in \Omega, t \in \mathcal{Z}} C(r(o, t)) && \text{(Problem 1)} \\ & \text{s. t. } U(q(o, t)) \geq \eta \end{aligned}$$

in which $C(\cdot)$ is a defined cost function based on HO-related metrics $r(o, t) := [r_{n,m}(o, t): \forall (n, m) \in \mathcal{P}]$, $U(\cdot)$ is a defined utility function based on QoS-related metrics $q(o, t) := [q_{n,m}(o, t): \forall (n, m) \in \mathcal{P}]$, and η is a threshold for QoS utility.

It is to be noted that the disclosure includes but is not restricted to the optimization of per LNE-pair HO parameters CIO and TTT. The same technique may be applied to optimize a broader class of HO parameters, e.g., HO parameters in idle mode $qOffset$ and $TReselEu$ -tra. Moreover, HO parameters with different granularities may be optimized. For example, the disclosure is also applicable to an MRO scenario where TTT is defined per cell and CIO is defined per cell pair.

In other words, in a distributed multi-agent DRL scheme for MRO an agent may be defined based on the SLA coverage overlap region (SCOR) to minimize the inter-agent dependency.

For a large-scale communications network, a centralized learning approach for MRO may not be practical because of its intractable high-dimensional network state and action spaces. In contrast to the centralized scheme, the distributed scheme decomposes the large-scale network to groups of LNEs and applies a machine learning agent to each of the group. The distributed multi-agent approach may reduce the complexity of the ML model and may speed up the learning speed.

At least in some embodiments, LNE pairs in a SCOR comprising at least two directional LNE pairs may be strongly coupled, and dependency between the SCORs may be low.

Herein, LNE pairs A and B are "strongly coupled" if the HO and QoS performance of A are strongly affected by the HO parameters in B, or the HO and QoS performance of B are strongly affected by the HO parameters in A. Further, herein SCORs X and Y have "low dependency/interaction" if the HO parameters of the LNE

pairs in X have a low effect on the overall HO and QoS performance of the LNE pairs in Y, and the HO parameters in Y have a low effect on the overall HO and QoS performance of the LNE pairs in X.

5 In other words, a distributed multi-agent MRO scheme with minimal inter-agent dependency may be enabled by decomposing the large-scale network into multiple SCORs, such that the LNEs within the same SCOR are strongly coupled, while the interaction between LNEs in
10 the neighboring SCORs is low.

 At least in some embodiments, the at least one memory 204 and the computer program code may be further configured to, with the at least one processor 202, cause the communications network device 200 to decompose
15 the communications network 110 into the SCORs by generating a logical network graph corresponding to the communications network 110 and representing the mobility relations between the LNE pairs, and by decomposing the logical network graph into subgraphs, such that the
20 subgraphs represent SCORs comprising strongly coupled LNE pairs. For example, the generating of the logical network graph may comprise generating the logical network graph based on historical LNE data, statistical mobility data, or an SLA coverage map.

25 At least in some embodiments, vertices of the logical network graph may comprise the LNE pairs, and weights of edges of the logical network graph may reflect a mobility relationship between two LNE pairs.

 Diagram 300B of Fig. 3B illustrates deriving a
30 mobility relation graph (MRG) 308 and SCORs from an LNE log history 306. In other words, to generate an MRG 308 and to decompose it to the SCORs from the LNE log history 306, a logical network graph $G = (V, E)$ may be derived based on mobility relations between LNE pairs. Herein,
35 it is also called the MRG. The vertices of the graph may be the LNE pairs (logical cell boundaries), and the

weights of the edges may reflect the mobility relationship between two LNE pairs, as shown in diagram 300B. From Eq. (1), it is known that $CIO_{n,m}$ and $CIO_{m,n}$ are strongly coupled. Thus, the unidirectional LNE pairs may
5 be used as the vertices of the MRG, i.e., boundary (m,n) and (n,m) are always jointly optimized.

It may be assumed that an LNE log history 306 of all users during a time period can be collected. Then, as shown in Fig. 3B, the steps of generating the
10 MRG 308 and its decompositions may be as follows:

1. from the log history 306, e. g., a sequence of LNE identities (IDs), obtain a sequence 307 of LNE pairs. E.g., the sequential logs of LNEs $a \rightarrow b \rightarrow c$ corresponds to sequential LNE pairs $(a,b) \rightarrow (b,c)$,
- 15 2. use the unique set 307 of LNE pairs as the vertices of the MRG 308,
3. count the number of transitions between each LNE pairs and use the number as weights of the edges of the MRG 308, and
- 20 4. use a graph decomposition/partition scheme (e.g., spectral partitioning) to decompose the MRG 308 to subgraphs 309A, 309B (herein also called sub-MRGs), each representing a SCOR comprising strongly coupled LNE pair(s), as shown in diagram 400 of Fig. 4 (which il-
25 lustrates the communications network 110 partitioned to SCORs 401, 402 and their corresponding subgraphs 309A, 309B). For each SCOR, a machine learning agent may be defined to solve the MRO problem within the SCOR.

Examples of the graph decomposition/partition
30 schemes include but are not limited to spectral partitioning in which a partition is derived from approximate eigenvectors of an adjacency matrix, and spectral clustering which groups graph vertices using an eigen-decomposition of a graph Laplacian matrix.

35 It is to be further noted that in case LNE logs data is not available, it is possible to exploit the HO statistics, adjacent relationship, and a coverage map

to generate an alternative type of MRG, with the LNEs as vertices and LNE pair specifying the edges.

At least in some embodiments, the at least one memory 204 and the computer program code may be further
5 configured to, with the at least one processor 202, cause the communications network device 200 at least to perform generating a profile for at least some of the subgraphs. The profile may comprise an adjacency matrix or an adjacency list representing the respective
10 subgraph.

In other words, a representation of a graph may include an adjacency matrix and/or an adjacency list. The adjacency matrix and list may be built based on the relation between the vertices. That is, the adjacency
15 matrix may indicate the weight of an edge between each pair of vertices, and the adjacency list may collect for each vertex the connected vertices and the edge weights. Since in the disclosure there may be strongly coupled LNE pairs within the agent, the number of edges may be
20 more than the number of vertices. Thus, an adjacent matrix or list may be chosen to represent the sub-MRG. An example is given in diagram 500 of Fig. 5 (which illustrates an example representation of a subgraph 309A with an adjacency matrix 501 or an adjacency list 502).

At least in some embodiments, each profile may further comprise a number of vertices, a number of edges, a number of involved LNEs, a degree distribution, a distribution of edge weights, a distribution of summed weights of edges incident to a vertex, or at least one
30 LNE specific feature for the respective subgraph including at least one of a deployment type, an LNE type, an associated user mobility distribution, position information, and/or an LNE load state.

In other words, for more efficient profile sim-
35 ilarity analysis and model retrieval, prescreening may be performed based on some simple features to reduce the size of the searching set of adjacency matrices/lists

for similarity analysis. Such optional features may include, e.g.:

- a number of vertices,
- a number of edges,
- 5 - a number of involved LNEs,
- a degree distribution: a degree of a vertex is the number of edges that are incident to it. Assuming the maximum number of incident edges of any vertex to be E , the degree distribution is an E -dim vector where the i -th dim is the probability of that vertex having i vertices. As shown in diagram 600 of Fig. 6 (which illustrates an example of sub-MRG profiling), for the upper sub-MRG 309A1, the degrees are 2 for all the three vertices, i.e., with a probability 1 the vertex has
- 10 2 incident edges, and with $E = 5$, the degree distribution is $[0,1,0,0,0]$. For the lower sub-MRG 309A2, the degrees of the vertices (a,b) , (b,c) , (c,a) are 1,2,1, respectively. I.e., there is a $2/3$ probability that a vertex has a degree of 1 and a $1/3$ probability that a vertex has a
- 15 degree of 2. The degree distribution is then $[\frac{2}{3},\frac{1}{3},0,0,0]$,

- other features can be also considered such as:

- a number of involved LNEs: e.g., for both sub-MRGs in Fig. 6, the number of the involved LNEs
- 25 is 3, i.e., LNEs $\{a,b,c\}$,
- a distribution of the edge weights,
- a distribution of summed weights of the edges incident to a vertex.

An example of an indexed sub-MRG profile is shown in the table 601 in Fig. 6.

In case the MRG's vertices are defined by LNEs, while their edges are defined by LNE pairs (LNE boundaries), more LNE-specific features may be included into the profile, e.g.:

- 35 - a deployment type: dense urban, urban, sub-urban and rural,

- an LNE type: e.g., for physical network entities such as macro, micro, pico and femto cells,
- an associated user mobility distribution,
- position information: outdoor, indoor,
- 5 - LNE load conditions: highly utilized, moderately utilized, underutilized.

The at least one memory 204 and the computer program code are further configured to, with the at least one processor 202, cause the communications
10 network device 200 at least to assign a machine learning agent to each of the decomposed SCORs.

Each machine learning agent is configured to apply a deep reinforcement learning model (DRL model) to solve an optimization problem related to a self-
15 organizing network (SON) function within its assigned SCOR. The DRL model may be customizable to each machine learning agent, e.g., based on a similarity analysis between subgraphs of the generated logical network graph. Accordingly, at least in some embodiments, based
20 on the similarity analysis between SCOR-based subgraphs, a pretrained model in one machine learning agent may be retrieved, reused, and/or customized for another agent. For example, a customized DRL model may comprise a transfer learning enhanced DRL model (TL-DRL model).

25 Diagram 300A of Fig. 3A illustrates an example of a SCOR and SCOR-based agents. More specifically, diagram 300A of Fig. 3A illustrates an example of decomposed SCORs and their corresponding agents, with each SCOR comprising a group of strongly coupled LNE pairs.
30 In the example of Fig. 3A, each of Agent #1, Agent #2, Agent #3, Agent #4, and Agent #5 optimizes HO parameters within an SLA coverage overlap region 301, 302, 303, 304, 305, respectively, e.g., for a group of highly dependent cell boundaries of cells 1[A], 1[B], 1[C],
35 2[A], 2[B], 2[C], 3[A], 3[B], 3[C]. In the example of Fig. 3A, there are three sites: Site 1, 2, 3. Each site has three cells, e.g., Site 1 has cells 1[A], 1[B],

1[C], etc. The circles represent coverage areas of the cells. The arrows represent coverage boundaries between two cells. For example, the arrows for Agent #1 represent cell boundaries of cell pairs (1[A], 1[B]), (1[A], 2[A]), (1[B], 2[A]), and the arrows for Agent #2 represent cell boundaries of cell pairs (1[A], 1[C]), (1[A], 3[A]), (1[C], 3[A]).

In other words, with the decomposed SCORs, it is possible to use distributed DRL approaches to optimize, e.g., HO parameters within each SCOR in a distributed manner, i.e., to solve Problem 1 for each SCOR independently. Due to the weak dependencies between the agents, the distributed multi-agent DRL may converge faster, and the learning may be more robust. Given a local agent i optimizing a set of K_i LNE pairs (boundaries), denoted by $\mathcal{P}_i := \{(n_k, m_k) : k = 1, \dots, K_i; (n_i, m_i) \in \mathcal{P}\}$, the involved LNEs may be denoted by \mathcal{N}_i with $|\mathcal{N}_i| = N_i$.

At least in some embodiments, states of the assigned machine learning agent(s) may comprise LNE - specific metrics, LNE pair -specific metrics, and/or contextual information for capturing at least one of temporal or spatial correlations.

In other words, the states may be defined in each local agent, e.g., as follows:

- state \mathbf{s}_i : e.g., the following features may be included in the state:

- LNE-specific metrics: including but not limited to the following per LNE metrics (or the extracted statistics over all involved LNEs): number of UEs, load, distribution of received signal strength-related measurements (e.g., channel quality indicator (CQI), RSRP distribution or reference signal received quality (RSRQ)), QoS-related metrics (e.g., throughputs, delay), and/or other LNE-specific configurations (e.g., if an LNE is defined as a slice in a cell, the slice resource budget may be one of the slice management configurations);

- LNE pair-specific metrics: including but not limited to the following per LNE pair metrics: HO-related metrics computed from counts of HO events, such as HOL, HOE, HOW, HOPP, and HO attempt (HOA), e.g., the ratio of the number of HOL, HOE, HOW, HOPP events to the number of HOAs, and/or LNE pair-specific QoS-related metrics (e.g., per cell boundary averaged throughputs and delay);

- other context information: to capture temporal or spatial correlations, categorical context information may be included, e.g., time index (hour of the day, weekday, or weekend), type of region (e.g., rural or urban). Such categorical data may be included in the state with one-hot encoding or categorical embedding.

At least in some embodiments, an action space of the assigned machine learning agent(s) may comprise a discrete action space or a continuous action space.

In other words, the actions may be defined in each local agent, e.g., as follows:

- action \mathbf{a}_i : e.g., two different models of DRL with different definitions of actions may be considered:

- model I with continuous CIO and TTT values: the policy of DRL directly provides the proto action, i.e., the CIO and TTT values of all LNE pairs in agent i denoted by $\mathbf{a}_i = [(o_{n_k, m_k}, o_{m_k, n_k}, t_{n_k, m_k}, t_{m_k, n_k}) : k = 1, \dots, K_i]$, in a continuous space. However, the CIOs and TTTs may be selected from a defined discrete set, e.g., the pool of CIO values $\mathcal{O} := \{-24, -23, \dots, 23, 24\}$ and the pool of TTT values

$$\mathcal{T} := \left\{ \begin{array}{l} 0.004, 0.064, 0.080, 0.1, 0.128, 0.16, 0.256, 0.320, 0.48, 0.512, 0.640, \\ 1.024, 2.56, 5.12 \end{array} \right\}.$$

This means that when working in discrete space, $o_{n_k, m_k} \in \mathcal{O}$ and $t_{n_k, m_k} \in \mathcal{T}$ for $k = 1, \dots, K_i$, the action space is $(\mathcal{O} \times \mathcal{T})^{2K_i}$. E.g., with only $K_i = 2$ pairs of LNEs in the agent, it is possible to have $(49 \cdot 14)^4$ combinations of

possible discrete actions, which may be intractable with value-based algorithms (such as deep-Q networks (DQN)) in a discrete space. Working in a continuous space may reduce the dimension of the output action and thus re-
 5 duce the model complexity, e.g., by using the policy-based or actor-critic-based DRL algorithms. When interacting with the environment, the proto action may be projected in continuous space to the defined discrete action by finding its nearest neighbor in the discrete
 10 space;

- model II with a discrete step size of CIO and TTT adjustment: to work with the discrete action space the action space may be reduced first. One option is to optimize the step size of the adjustment
 15 of CIO and TTT instead of the actual CIO and TTT values. For example, an action $a_i = [(\Delta o_{n_k, m_k}, \Delta o_{m_k, n_k}, \Delta t_{n_k, m_k}, \Delta t_{m_k, n_k}) : k = 1, \dots, K_i]$ may be defined where the step size for CIO is $\Delta o_{n, m} \in \Lambda^{(\text{CIO})}$ and the step size for TTT is $\Delta t_{n, m} \in \Lambda^{(\text{TTT})}$, for any $(n, m) \in \mathcal{P}_i$. A small step size set, e.g., $\Lambda^{(\text{CIO})} :=$
 20 $\{-2, -1, 0, 1, 2\}$ and $\Lambda^{(\text{TTT})} := \{-1, 0, 1\}$ may be defined, and the discrete action space is $(\Lambda^{(\text{CIO})} \times \Lambda^{(\text{TTT})})^{2K_i}$. With $K_i = 2$ pairs of LNEs in the agent, there are only $(5 \cdot 3)^4$ possible actions. In this way, both the value-based DRL algorithms (e.g., DQN) and the policy- or actor-critic-
 25 based DRL algorithms may be considered.

At least in some embodiments, rewards for the assigned machine learning agent(s) may be based on LNE pair -specific handover performance metrics, LNE -
 30 specific quality of service, QoS, performance metrics, and/or LNE pair -specific QoS performance metrics. Additionally/alternatively, the rewards for the assigned machine learning agents may be based on layer 2 cell based scheduling performance metrics and/or service level agreement success rates.

35 In other words, the rewards may be defined in each local agent, e.g., as follows:

- reward R_i : the SLA-guaranteed MRO may take both HO performance and SLA-related QoS performance into account. Thus, the reward R_i may be computed based on, e.g., the following metrics:

5 - LNE pair-specific HO performance metrics: negative HO cost computed from the HOL, HOE, HOW, and HOPP ratios. One example of the HO cost is the weight sum of the ratios of different HO events:

$$10 \quad c_i^{(HO)} = \frac{1}{K_i} \sum_{(n,m) \in \mathcal{P}_i} w_{n,m}^{(HOL)} r_{n,m}^{(HOL)} + w_{n,m}^{(HOE)} r_{n,m}^{(HOE)} + w_{n,m}^{(HOW)} r_{n,m}^{(HOW)} + w_{n,m}^{(HOPP)} r_{n,m}^{(HOPP)}, \quad (\text{Eq. 2})$$

15 in which $r^{(HOL)}, r^{(HOE)}, r^{(HOW)}, r^{(HOPP)}$ and $w^{(HOL)}, w^{(HOE)}, w^{(HOW)}, w^{(HOPP)}$ are the ratios and weights for HOL, HOE, HOW, and HOPP events, respectively;

- LNE-specific QoS performance metrics: a QoS reward computed from throughput and delay. For example, given the throughput requirement ϕ_n^* and delay requirement d_n^* for each LNE $n \in \mathcal{N}_i$, the QoS reward in agent i may be computed as:

$$20 \quad q_i^{(QoS)} = \sum_{n \in \mathcal{N}_i} w_n^{(QoS)} \min \left\{ \frac{\phi_n}{\phi_n^*}, \frac{d_n^*}{d_n}, 1 \right\} \quad (\text{Eq. 3})$$

25 in which ϕ_n and d_n are the achieved throughput and delay, respectively. $q_i^{(QoS)}$ is a weighted sum of the QoS satisfaction level of all LNEs involved in agent i . It is to be noted that the QoS satisfaction level is upper-bounded by 1, i.e., $\min \left\{ \frac{\phi_n}{\phi_n^*}, \frac{d_n^*}{d_n}, 1 \right\} = 1$ if the throughput is higher than required $\phi_n \geq \phi_n^*$ and the delay is lower than required $d_n \leq d_n^*$. The weight $w_n^{(QoS)}$ reflects the importance/priority of LNE n . It is possible to make $w_n^{(QoS)}$ proportional to the number of UEs/services associated to LNE n , e.g., $w_n^{(QoS)} = \frac{N_n^{(UE)}}{\sum_{l \in \mathcal{N}_i} N_l^{(UE)}}$ such that $q_i^{(QoS)}$ is the averaged QoS satisfaction level over all UEs/services involved in agent i . Other QoS rewards may also be

utilized, such as applying a utility function to address the fairness or priority of different network entities and/or QoS metrics.

The overall reward R_i may be the weight sum of the negative HO cost and the QoS reward:

$$R_i = w^{(QoS)}q_i^{(QoS)} - w^{(HO)}c_i^{(HO)} \quad (\text{Eq. 4})$$

It is to be noted that the disclosed ML model may also be applied to conventional MRO without SLA-awareness, by excluding the SLA-related metrics in the network states and QoS performance metrics in the reward computation.

Various DRL algorithms may be utilized, including deep Q-learning (when discrete action space is relatively small, e.g., with the above disclosed model II where action is defined as a discrete step size of CIO and TTT adjustment), policy-based algorithms such as PPO, and actor-critic-based algorithms such as deep deterministic policy gradient (DDPG) (which may be used for both models I and II).

It is to be noted that $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ for $i \neq j$ because all the LNE pairs are partitioned to separate SCORs. However, an LNE may be involved in several neighboring agents, e.g., an LNE n may have one boundary (n,m) grouped into agent 1 and another boundary (n,l) grouped into agent 2. Thus, in the SLA-guaranteed MRO scheme, when LNE-specific KPIs and QoS metrics are included in the state s_i and the reward R_i , there are still QoS dependencies between the agents, although by MRG decomposition the mobility dependencies may be reduced. Thus, a possible coordination scheme may be designed for multi-agent DRL to ensure the consensus, by adding new features extracted from the neighboring agents' states.

At least in some embodiments, the SON function may comprise a mobility robustness optimization (MRO)

function, a coverage and capacity optimization function, or a mobility load balancing function. For example, the MRO function may comprise optimization of one or more handover parameters.

5 At least in some embodiments, the at least one memory 204 and the computer program code may be further configured to, with the at least one processor 202, cause the communications network device 200 at least to obtain the DRL model as pretrained from a SON node device
10 210.

Fig. 2B is a block diagram of the SON node device 210, in accordance with an example embodiment.

The SON node device 210 comprises at least one processor 212 and at least one memory 214 including
15 computer program code. The SON node device 210 may also include other elements, such as a transceiver configured to enable the SON node device 210 to transmit and/or receive information to/from other devices, as well as other elements not shown in Fig. 2B. In one example, the
20 SON node device 210 may use the transceiver to transmit or receive signaling information and data in accordance with at least one cellular communication protocol. The transceiver may be configured to provide at least one wireless radio connection, such as for example a 3GPP
25 mobile broadband connection (e.g., 5G or 6G). The transceiver may be configured to be coupled to at least one antenna to transmit and/or receive radio frequency signals.

Although the SON node device 210 is depicted
30 to include only one processor 212, the SON node device 210 may include more processors. In an embodiment, the memory 214 is capable of storing instructions, such as an operating system and/or various applications. Furthermore, the memory 214 may include a storage that may
35 be used to store, e.g., at least some of the information and data used in the disclosed embodiments.

Furthermore, the processor 212 is capable of executing the stored instructions. In an embodiment, the processor 212 may be embodied as a multi-core processor, a single core processor, or a combination of one or more multi-core processors and one or more single core processors. For example, the processor 212 may be embodied as one or more of various processing devices, such as a coprocessor, a microprocessor, a controller, a digital signal processor (DSP), a processing circuitry with or without an accompanying DSP, or various other processing devices including integrated circuits such as, for example, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a microcontroller unit (MCU), a hardware accelerator, a special-purpose computer chip, or the like. In an embodiment, the processor 212 may be configured to execute hard-coded functionality. In an embodiment, the processor 212 is embodied as an executor of software instructions, wherein the instructions may specifically configure the processor 212 to perform the algorithms and/or operations described herein when the instructions are executed.

The memory 214 may be embodied as one or more volatile memory devices, one or more non-volatile memory devices, and/or a combination of one or more volatile memory devices and non-volatile memory devices. For example, the memory 214 may be embodied as semiconductor memories (such as mask ROM, PROM (programmable ROM), EPROM (erasable PROM), flash ROM, RAM (random access memory), etc.).

The at least one memory 214 and the computer program code are configured to, with the at least one processor 212, cause the SON node device 210 at least to receive from the communications network device 200 a request for a pretrained DRL model for use in solving an optimization problem related to a SON function within a SCOR. The request comprises at least one profile of

at least one subgraph of a logical network graph representing mobility relations between LNE pairs. As discussed above in more detail, the SCOR comprises at least one LNE pair, and the at least one subgraph
5 corresponds to the SCOR.

The at least one memory 214 and the computer program code are further configured to, with the at least one processor 212, cause the SON node device 210 at least to determine a pretrained reference DRL model
10 from a model database 230A, 230B, 230C by a similarity analysis based on the profile of the subgraph.

The at least one memory 214 and the computer program code are further configured to, with the at least one processor 212, cause the SON node device 210 at least to transmit the determined pretrained reference
15 DRL model to the communications network device 200.

At least in some embodiments, the similarity analysis may comprise a relational similarity analysis between the at least one subgraph in the received at
20 least one profile and subgraphs associated with the pretrained reference DRL models stored in the model database 230A, 230B, 230C. Accordingly, at least in some embodiments, the disclosed transfer learning approaches may preserve relational similarity between SCORs and
25 allow model transfer between SCOR-based agents with a same or different number of nodes and edges.

At least in some embodiments, the at least one memory 214 and the computer program code may be configured to, with the at least one processor 212,
30 further cause the SON node device 210 to customize the determined reference DRL model, and to transmit the customized DRL model to the communications network device 200. For example, the customized DRL model may comprise a TL-DRL model.

35 In other words, it may be assumed that a user sends a query of a sub-MRG $G_* = (V_*, E_*)$ with $N_*^{(V)}$ vertices (i.e., LNE-pairs) and $N_*^{(L)}$ involved LNEs, and requests a

“target” DRL model $M_*(G_*)$ customized to the SCOR characterized by G_* . The network service provider device 220 may identify the sub-MRG $G_s = (V_s, E_s)$ similar to G_t in the index-profile DB, with $N_s^{(V)}$ vertices and $N_s^{(L)}$ involved LNEs, and retrieve G_s 's corresponding pretrained “source” model $M_s(G_s)$ in the index-model DB.

At least in some embodiments, the at least one memory 214 and the computer program code may be configured to, with the at least one processor 212, further cause the SON node device 210 to prescreen the profile(s) of the subgraph(s) based on one or more prescreening parameters to determine one or more candidate pretrained reference DRL models. In this case, the determining of the pretrained reference DRL model may be performed on the determined one or more candidate pretrained reference DRL models.

In other words, for searching and sub-MRG similarity analysis, e.g., at least some of the following steps may be utilized:

- profile pre-screening: because the graph similarity analysis costs more computational effort, simple features (e.g., the number of vertices and edges, number of involved LNEs, and the distribution of degree, as shown in Fig. 6) may first be used to pre-screen the profiles and find a reduced candidate set of “roughly” similar “source” profiles, denoted by $G_s := \{G_{s_i} : i = 1, 2, \dots\}$. Various distance measures may be applied, including, e.g., a Euclidean distance between the numbers of nodes and edges, and/or Kullback-Leibler divergence of the degree distribution. At least in some embodiments, only the candidate sub-MRGs with numbers of the vertices and the involved LNEs no less than (if not the same as) those of the sub-MRG in query, i.e., $N_s^{(V)} \geq N_*^{(V)}$ and $N_s^{(L)} \geq N_*^{(L)}$ may be pre-screened. In this way, even if the sub-MRG sharing the same numbers of LNE-pairs and involved LNEs cannot be found, the retrieved source model $M_s(G_s)$

is given some degree of information redundancy to learn $M_*(G_*)$.

- graph similarity measure: after the pre-screening, a small set of sub-MRGs may be derived to be compared with the sub-MRG sent in the query. Then, e.g., classical graph similarity analysis techniques may be applied to find the most similar one(s). Such techniques may be classified into three categories: edit distance/graph isomorphism, feature exaction, and iterative methods.

If a user sends a query of a set of sub-MRGs generated from its own network environment, it can either apply directly the retrieved pretrained SON models or further request a transfer learning service to customize the retrieved model to its own network environment.

It is to be noted that it may be likely that $N_s^{(V)} \neq N_*^{(V)}$ and $N_s^{(L)} \neq N_*^{(L)}$, i.e., even the most similar sub-MRGs do not necessarily have the same number of vertices and edges as the ones in the query, while a direct application of the ML model prefers the same size of vertices, namely, the same dimension of the LNE pair-specific action. Moreover, the same number of involved LNEs may also be preferred, such that the input network state can have the same dimension. Thus, after retrieving the model, even though G_s is very similar to the G_* in the query, a transfer learning scheme may be needed for better customization.

Diagram 800 of Fig. 8 illustrates an example of actor networks 801A-801D in a DRL architecture in which the source model has higher dimensions of action and state than the target model. $s_{n,m}$ denotes the LNE pair-specific features of the boundary (n,m) , e.g., HO failure ratio between LNE boundary (n,m) , while $[s'_n, s'_m]$ denotes the LNE-specific state of LNE n and LNE m , e.g., s'_n and s'_m may include the load of LNE n and m , respectively. The concatenated $[s'_n, s'_m]$ corresponds to the

LNE boundary (n,m) , providing additional information for HO decision making between n and m .

In the following, state and action dimension matching is discussed. To enable the transfer learning
 5 from one model to another, the features in the state and action in $M_*(G_*)$ may first be ordered such that the dependencies between the features are consistent with the defined state and action in $M_s(G_s)$:

- ordering the LNE within each LNE pair: because the HO parameters are directional, e.g., $CIO_{n,m} \neq CIO_{m,n}$, the LNEs in each LNE pair may be ordered such that the order of the action outputs are consistent in both the source and target model. The ordering may be based on summed edge weights related to the LNE (an edge
 10 reflects an intermediate LNE between two neighboring LNE pairs). E.g., in Fig. 8, in the source graph G_s , 309C within the LNE pair $(2,1)$, LNE 2 may be put first because the edge $(2,1) - (2,3)$ shares the LNE 2 and has a higher weight 3, while the edge $(2,1) - (1,3)$ shares the LNE 1 and has a lower weight 2. Alternative schemes to order the
 15 LNEs in an LNE pair may include ranking them based on selected LNE features, e.g., per-LNE average load or average received signal strength.

- finding matching LNE pairs between G_* and G_s :
 25 for each vertex (LNE pair) in the target sub-MRG G_* , denoted by $v \in \mathcal{V}(G_*)$, where $\mathcal{V}(\cdot)$ denotes the set of all vertices in a graph, its matching vertex in the source sub-MRG G_s , denoted by $u \in \mathcal{V}(G_s)$ needs to be found. Thus, the problem is to find:

30

$$\forall v \in \mathcal{V}(G_*), u^*(v) = \operatorname{argmin}_{u \in \mathcal{V}(G_s)} d(F(v), F(u))$$

in which $d(F(v), F(u))$ is the distance measure between the extracted features, denoted by $F(v)$ and $F(u)$,
 35 of the vertices u and v .

The rationale behind this is that the two vertices in two graphs are similar if their neighbors and

the edges to the neighbors are similar. Thus, the distance measure of the extracted features of the neighboring vertices and edges may be computed to find the most similar vertex in G_s to a given vertex in G_* .

5 Table I below gives an example of the matched order of LNE pairs, as well as the LNEs within each LNE pair, between the source 309C and target 309A sub-MGRs in Fig. 8.

10 Table I: Matched order of LNE pairs and LNEs

Matched order of LNE pairs & matched order of LNEs within each pair
(a, b) - (2, 1)
(b, c) - (2, 3)
(a, c) - (1, 3)
None - (3, 4)

In the following, transfer learning from a source model to a target model is discussed. With the matched order of LNE pairs and LNEs within each pair,
 15 the order of state and action features may be sorted such that the inter-feature dependencies in the source and target model are consistent, as shown in Fig. 8. It is to be noted that in Fig. 8 the actor network 801A-801D is used as an example for the actor-critic DRL
 20 algorithms, but the same approach may be used in critic network and other DRL architectures, such as DQN.

In the following, examples of transfer learning schemes are provided:

- if $N_t^{(V)} = N_s^{(V)}$ and $N_t^{(L)} = N_s^{(L)}$, the architecture of
 25 $M_t(G_t)$ is the same as $M_s(G_s)$ because they have the same dimension of state s and action a ,

- direct application: The user may simply retrieve and apply the source model, i.e.,
 $M_t(G_t) = M_s(G_s)$,

30 - finetuning: if the user requests transfer learning service, standard transfer learning

in diagram 1000 of Fig. 10 (which illustrates an example of knowledge distillation from $M_s(G_s)$ 1010A to $M^*(G^*)$ 1010B, with actor network 1001A, 1001B, respectively).

Further features of the SON node device 210 directly result from the functionalities and parameters of the communications network device 200 and thus are not repeated here.

Fig. 2C is a block diagram of the network service provider device 220, in accordance with an example embodiment.

The network service provider device 220 comprises at least one processor 222 and at least one memory 224 including computer program code. The network service provider device 220 may also include other elements, such as a transceiver configured to enable the network service provider device 220 to transmit and/or receive information to/from other devices, as well as other elements not shown in Fig. 2C.

Although the network service provider device 220 is depicted to include only one processor 222, the network service provider device 220 may include more processors. In an embodiment, the memory 224 is capable of storing instructions, such as an operating system and/or various applications. Furthermore, the memory 224 may include a storage that may be used to store, e.g., at least some of the information and data used in the disclosed embodiments.

Furthermore, the processor 222 is capable of executing the stored instructions. In an embodiment, the processor 222 may be embodied as a multi-core processor, a single core processor, or a combination of one or more multi-core processors and one or more single core processors. For example, the processor 222 may be embodied as one or more of various processing devices, such as a coprocessor, a microprocessor, a controller, a digital signal processor (DSP), a processing circuitry with or without an accompanying DSP, or various other processing

devices including integrated circuits such as, for example, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a microcontroller unit (MCU), a hardware accelerator, a special-purpose computer chip, or the like. In an embodiment, the processor 222 may be configured to execute hard-coded functionality. In an embodiment, the processor 222 is embodied as an executor of software instructions, wherein the instructions may specifically configure the processor 222 to perform the algorithms and/or operations described herein when the instructions are executed.

The memory 224 may be embodied as one or more volatile memory devices, one or more non-volatile memory devices, and/or a combination of one or more volatile memory devices and non-volatile memory devices. For example, the memory 224 may be embodied as semiconductor memories (such as mask ROM, PROM (programmable ROM), EPROM (erasable PROM), flash ROM, RAM (random access memory), etc.).

The at least one memory 224 and the computer program code are configured to, with the at least one processor 222, cause the network service provider device 220 at least to obtain from at least one communications network device 200 at least one subgraph of a logical network graph representing mobility relations between LNE pairs and their associated profiles.

The at least one memory 224 and the computer program code are further configured to, with the at least one processor 222, cause the network service provider device 220 at least to derive at least one pretrained reference DRL model for use in solving an optimization problem related to a SON function within a SCOR. As discussed in more detail above, the SCOR comprises at least one LNE pair and the SCOR corresponds to the subgraph.

The at least one memory 224 and the computer program code are further configured to, with the at least one processor 222, cause the network service provider device 220 at least to store the obtained profiles and the derived pretrained reference DRL models in the model database 230A, 230B, 230C.

In other words, the network service provider device 220 may obtain the sub-MRGs and their profiles with its MRG generation function. Then, the network service provider device 220 may apply its SCOR-based SON functions online in its network environment and/or offline on its collected dataset and derive the pretrained reference DRL models. The profiles and the models may be stored in the SON model DB 230 as reference models to be retrieved by users or other service providers.

At least in some embodiments, the model database 230A, 230B, 230C may comprise an indexed model database such that a same index represents an obtained profile of one subgraph and a corresponding derived pretrained reference DRL model in a SCOR that corresponds to the subgraph.

In other words, e.g., for more efficient query an index-profile DB 230A and/or an index-model DB 230B may be built, such that the sub-MRG's corresponding profile and its pre-trained model may be accessed independently with the same index. For the sub-MRG similarity analysis, the index-profile DB 230A may be accessed to find the sub-MRG profile with the highest similarity. Then, the index of the found sub-MRG may be used to retrieve the model in the index-model DB 230B. In this case, the SON model DB 230C in Fig. 11 may be replaced by the index-profile DB 230A and an index-model DB 230B shown in diagram 700 of Fig. 7 (which illustrates an example of an indexed profile database 230A and an indexed model database 230B where sub-MRG 1 309A is associated to the profile 1 701 and the pre-trained model 1

703, while sub-MRG 2 309B is associated to the profile 2 702 and the pre-trained model 2 704).

Further features of the network service provider device 220 directly result from the functionalities and parameters of the communications network device 200 and the SON node device 210, and thus are not repeated here.

Diagram 1100 of Fig. 11 illustrates an overview of the disclosed embodiments. More specifically, diagram 1100 of Fig. 11 illustrates a transfer learning framework with logical network graph -based model retrieval, MRG generation, decomposition of the MRG to subgraphs and their corresponding SCORs, subgraph profiling, and DRL model retrieval based on a subgraph similarity analysis. The transfer learning framework of diagram 1100 of Fig. 11 may be implemented with any combination of the communications network device 200, the SON node device 210 and the network service provider device 220.

The main functional blocks in Fig. 11 include:

- 20 - MRG generation, partitioning, and profiling 1100A: for partitioning a large-scale network graph to sub-MRGs with minimum dependencies. A sub-MRG may represent a SCOR, i.e., a strongly coupled SLA coverage overlap and mobility dependent region;
- 25 - building a SCOR-based reference model DB with key-content data structure 1100B:
 - keys: profiles of the sub-MRGs, and
 - contents: pre-trained SCOR-based distributed DRL models.
- 30 - MRG similarity analysis, model retrieval, and transfer learning 1100C: the profile of the sub-MRG may be used for similarity analysis and model retrieval. The retrieved model may be reproduced and fine-tuned to a customized model with, e.g., transfer learning.

35 The workflow in Fig. 11 may include, e.g., at least some of the following:

- steps 1a, 1b, 1c: a user (e.g., an operator) 1130 saves network measurements in its network environment 1110 in a data storage 1120. The data storage 1120 may be, e.g., in operations and maintenance (O&M) or other cloud data servers,
5
- steps 2a, 2b: the user 1130 requests MRG generation service and grants the service function 1140 access to process its data,
- step 3: upon the request of the user 1130,
10 the MRG generation service 1140 generates a networkwide MRG and partitions it to sub-MRGs. It also profiles the sub-MRGs and sends back the profiles,
- step 4: the service provider (e.g., the network service provider device 220) applies the MRG generation service function 1140 to its collected data
15 1160, obtains a set of reference sub-MRGs and an offline dataset in the corresponding SCOR for training a local DRL agent,
- step 5: the SON function module 1150 (e.g.,
20 the SON node device 210) obtains the reference SCORs and the corresponding sub-MRGs,
- steps 6a, 6b: the service provider (e.g., the network service provider device 220) applies SCOR-based SON functions (e.g., a distributed multi-agent MRO algorithm) online in its network environment 1170 and/or
25 offline on its collected dataset 1160, and derives the pre-trained reference DRL models,
- step 7: the service provider (e.g., the service provider 220) saves the pretrained SCOR-based DRL
30 models and their corresponding sub-MRG profiles in the SON model DB 230C,
- step 8: the user 1130 makes a query and sends its sub-MRG profiles to a MRG similarity analysis service 1180,
- 35 - step 9: the MRG similarity analysis function 1180 finds matching keys,

- step 10: the MRG similarity analysis function 1180 reports the matching key to a transfer learning -empowered SON function 1190,

5 - step 11: based on the matching key, the transfer learning -empowered SON function 1190 retrieves the pretrained model upon the request of the user 1130, and/or

10 - step 12: the transfer learning -empowered SON function 1190 applies transfer learning techniques to fine-tune the retrieved DRL models on the environment 1110 of the user 1130 and/or the dataset 1120 of the user 1130 and customizes the models for the user 1130.

15 The disclosed transfer learning framework for distributed multi-agent SON functions may be added into an O-RAN AI/ML workflow.

In the following, alternatives for the MRG generation are briefly discussed.

20 Alternative 1 - generating the MRG and its decomposition to SCORs from HO statistics: in case the LNE logs data is not available, the HO statistics and adjacent relationship may be exploited to generate the MRG, e.g., using neighboring cell list and HO metrics, such as the number of HO success between neighboring nodes. Also, a mobility graph may be created from the statistical data of previous handovers. However, unlike in the
25 MRG generation described above, here, a vertex defines an LNE and an edge defines an LNE pair. Thus, when decomposing the graph, the boundaries between the subgraphs are not included in any of the subgraphs. Because
30 the aim is to optimize per LNE pair HO parameters, the edges need to be grouped, but not the vertices. To this end, a heuristic step may be applied to add the edge (LNE pair) between two subgraphs to the subgraph with less members.

35 Alternative 2 - generating the MRG and its decomposition to SCORs from an SLA coverage map: in case neither the LNE logs data nor the mobility statistics

are available, e.g., for a network in planning or a newly deployed network system, sufficient historical user logs or mobility statistics are not available. Then, either a simulated (e.g., from a simulator or a digital twin) or measured SLA coverage map (e.g., from a driving test) may be used to generate the MRG.

An MRG $G = (V, E)$ may be derived from the SLA coverage map, as shown in Fig. 15. Diagram 1500 of Fig. 15 illustrates deriving an MRG 1501 and SCORs 1503A, 1503B from an SLA coverage map 1504.

Here, the vertices are the LNEs instead of LNE pairs. Since the CIOs and TTTs are defined per LNE pairs, the SCOR is further defined based on the edges of the MRG (an edge indicates the mobility interaction between two LNEs). The steps may include, e.g., at least some of the following:

1. the mobility playground may be gridded spatially. Each grid is marked by a set of LNE indices: if the SLA coverage of an LNE (partially) covers a grid, the index of the LNE is included. It is to be noted that for different geographical areas the size of the grid may be different. E.g., a rural area may have a larger size and urban area may have a smaller size. The sizes of the grids may be dynamically configured depending on the UE or service density,

2. each vertex of the MRG 1501 defines an LNE. The weight of an edge between two LNEs may be computed by counting the number of the coverage overlapped grids of two LNEs if at least one LNE is in the neighboring LNE list of another,

3. classical graph decomposition/partition schemes may be used to decompose the MRG 1501 to sub-graphs 1502A, 1502B,

4. because the CIOs and TTTs are defined per LNE pair, groups of LNE pairs may need to be obtained from the decomposed MRG. The vertices of the MRG in this scheme are the LNEs, while the edges are the LNE pairs.

Same heuristics may be used to add the edges between the subgraphs to the one with less members to balance the agent scale. One SCOR 1503A (optimized by a first agent) may comprise, e.g., LNE pairs $\{(a, b), (b, c), (a, c)\}$, while another SCOR 1593B (optimized by a second agent) may comprise LNE pairs $\{(d, e), (c, d), (c, e)\}$.

Diagram 1600 of Fig. 16 illustrates MRG decomposition when an LNE is defined as a network slice in a physical cell. The MRG 1602A, 1602B, 1602C, 1602D are decomposed to four agents optimizing four SCORs, SCOR_1= $\{(a, b), (b, c), (a, c)\}$ 1603A, SCOR_2= $\{(d, e), (c, d), (c, e)\}$ 1603B, SCOR_3= $\{(g, f), (f, h)\}$ 1603C, SCOR_4= $\{(h, i), (I, j), (h, j)\}$ 1603D, respectively. In the case where an LNE is a logical slice in a physical cell, there are SLA coverage maps 1604A, 1604B of each slice. Fig. 16 is an example of how the decomposed SCORs 1603A, 1603B, 1603C, 1603D may be derived from the two SLA coverage maps 1604A, 1604B corresponding to two slices. Without inter-slice HO, the LNEs in Slice 2 are not included in the neighboring LNE lists of the LNEs in Slice 1. Thus, there are no edges between the subgraphs generated by different slice coverage maps, and four decomposed SCORs 1603A, 1603B, 1603C, 1603D may be obtained, each optimized by a distributed agent. Furthermore, the disclosure may be extended to a scenario with inter-slice HO. In that case, edges with higher weights may exist between LNEs connected to different slices, and a SCOR may comprise LNE pairs associated with different slices.

Diagram 1700 of Fig. 17 shows an example embodiment of the subject matter described herein illustrating an example of a deep deterministic policy gradient (DDPG) architecture of a local agent i with actions of CIO and TTT values in a continuous space. The example DDPG architecture comprises a network environment 1701, an experience replay memory 1702, a mini-

batch 1703, an actor network 1704, a target actor network 1705, a critic network 1706, a target critic network 1707, a policy gradient 1708, and a loss computing block 1709.

5 In other words, diagram 1700 of Fig. 17 is an example of the above-discussed model I with continuous CIO and TTT values (defined in a continuous action space) of the DRL model for an agent with an actor-critic architecture. The elements of state, action and
10 reward may include those discussed above. The actor network 1704 may include a policy network that takes the state as input and outputs the exact continuous action instead of a probability distribution over actions. The critic network 1706 may include a Q-value network that
15 takes in state and action as input and outputs the Q-value. Experience replay memory 1702 may include a replay memory technique used in reinforcement learning where the agent's experiences are stored at each time-step. The target critic network 1707 may include a copy
20 of an action-value function (or Q-function) that is held constant to serve as a stable target for learning for a fixed number of timesteps. The policy gradient 1708 may include a computation of a gradient descent to optimize the parameters of the neural network based on an expected
25 return of a long-term cumulative reward. The loss computing block 1709 may compute a loss function of the networks based on but not limited to the computed reward.

Diagram 1800 of Fig. 18 shows an example embodiment of the subject matter described herein illustrating an example of a deep Q-network (DQN) architecture 1801 of a local agent i with actions of step sizes of CIO and TTT adjustments in a discrete space. As discussed above in more detail, inputs s_i may include, e.g.,
35 LNE-specific metrics (e.g., a number of UEs, load, a

distribution of CQIs, QoS-related metrics, other configurations) and/or LNE-pair specific metrics (e.g., HO-related metrics, other context information).

In other words, diagram 1800 of Fig. 18 is an example of the above-discussed model II with a discrete step size of CIO and TTT adjustment (having a discrete action space) of the DRL model for an agent with a DQN architecture. The elements of state, action and reward may include those discussed above.

Fig. 12 illustrates an example flow chart of a method 1200, in accordance with an example embodiment.

At operation 1201, the communications network device 200 decomposes the communications network 110 into the SCORs according to mobility relations between LNE pairs within the communication network 110. As discussed in more detail above, the SCOR comprises at least one LNE pair.

At optional operation 1202, the communications network device 200 may generate a profile for the subgraph(s), the profile(s) comprising an adjacency matrix or an adjacency list representing the respective subgraph.

At optional operation 1203, the communications network device 200 may obtain a DRL model as pretrained from the SON node device 210.

At operation 1204, the communications network device 200 assigns a machine learning agent (e.g., the pretrained one obtained in operation 1203) to at least one of the decomposed SCORs. The machine learning agent is configured to apply a DRL model to solve an optimization problem related to a SON function within its assigned SCOR.

The method 1200 may be performed by the communications network device 200 of Fig. 2A. The operations 1201-1204 can, for example, be performed by the at least one processor 202 and the at least one memory 204. Further features of the method 1200 directly result

from the functionalities and parameters of the communications network device 200, and thus are not repeated here. The method 1200 can be performed by computer program(s).

5 Fig. 13 illustrates an example flow chart of a method 1300, in accordance with an example embodiment.

At operation 1301, the SON node device 210 receives from the communications network device 200 a request for a pretrained DRL model for use in solving
10 an optimization problem related to a SON function within a SCOR. The request comprises at least one profile of at least one subgraph of a logical network graph representing mobility relations between LNE pairs, with the SCOR comprising at least one LNE pair, and the at
15 least one subgraph corresponding to the SCOR.

At optional operation 1302, the SON node device 210 may prescreen the profile of the subgraph based on one or more prescreening parameters to determine one or more candidate pretrained reference DRL models.

20 At operation 1303, the SON node device 210 determines a pretrained reference DRL model from the model database 230A, 230B, 230C by a similarity analysis based on the profile of the subgraph. When the prescreening of operation 1302 is performed, the
25 determining of the pretrained reference DRL model may be performed on the determined one or more candidate pretrained reference DRL models.

At optional operation 1304, the SON node device 210 may customize the determined pretrained reference
30 DRL model, e.g., for at least one machine learning agent of the communications network device 200. For example, the customized DRL model may comprise a TL-DRL model.

At operation 1305, the SON node device 210 transmits the (determined/customized) pretrained
35 reference DRL model (e.g., a TL-DRL model) to the communications network device 200.

The method 1300 may be performed by the SON node device 210 of Fig. 2B. The operations 1301-1305 can, for example, be performed by the at least one processor 212 and the at least one memory 214. Further features of the method 1300 directly result from the functionalities and parameters of the SON node device 210, and thus are not repeated here. The method 1300 can be performed by computer program(s).

Fig. 14 illustrates an example flow chart of a method 1400, in accordance with an example embodiment.

At operation 1401, the network service provider device 220 obtains from at least one communications network device 200 at least one subgraph of a logical network graph representing mobility relations between LNE pairs and their associated profiles.

At operation 1402, the network service provider device 220 derives at least one pretrained reference DRL model for use in solving an optimization problem related to a SON function within a SCOR, with the SCOR comprising at least one LNE pair, and the SCOR corresponding to the subgraph.

At operation 1403, the network service provider device 220 stores the obtained profiles and the derived pretrained reference DRL models (e.g., TL-DRL models) in the model database 230A, 230B, 230C.

The method 1400 may be performed by the network service provider device 220 of Fig. 2C. The operations 1401-1403 can, for example, be performed by the at least one processor 222 and the at least one memory 224. Further features of the method 1400 directly result from the functionalities and parameters of the network service provider device 220, and thus are not repeated here. The method 1400 can be performed by computer program(s).

At least some of the embodiments may allow a distributed multi-agent DRL algorithm for an MRO problem, where each agent may comprise a varying number of

physical or logical network boundaries. At least some of the embodiments may allow minimizing the inter-agent dependencies by decomposing the network mobility graph. Such decomposition may provide proper granularity to
5 deploy distributed learning agents, which may achieve a good tradeoff between low model complexity and good model accuracy.

At least some of the embodiments may allow a transfer learning framework for SON model profiling,
10 storage, retrieval, retraining, and management such that one can efficiently retrieve the SON model that was pretrained in a similar (sub)network environment. At least some of the embodiments may allow analyzing the similarity between the relational network graph data.
15 This approach may be especially beneficial for SON use cases because it captures the strong interaction between the network entities within a local agent.

At least some of the embodiments may allow SON service providers and their customers to easily produce,
20 archive, share, and reproduce SON models in an open, multi-vendor, and multi-stakeholder platform.

At least some of the embodiments may allow either online training (e.g., using the live network and its data to train) or offline training to generate the
25 transfer model or a combination of online and offline training. The convergence times may be offset by more real time data.

At least some of the embodiments may allow SON and O-RAN / virtualized RAN (VRAN) network optimization.

30 The communications network device 200 may comprise means for performing at least one method described herein. In an example, the means may comprise the at least one processor 202, and the at least one memory 204 including program code configured to, when executed by
35 the at least one processor 202, cause the communications network device 200 to perform the method.

The SON node device 210 may comprise means for performing at least one method described herein. In an example, the means may comprise the at least one processor 212, and the at least one memory 214 including program code configured to, when executed by the at least one processor 212, cause the SON node device 210 to perform the method. The SON node device 210 may comprise a group of virtualized network optimization functions which may be implemented as cloud computing service(s).

The network service provider device 220 may comprise means for performing at least one method described herein. In an example, the means may comprise the at least one processor 222, and the at least one memory 224 including program code configured to, when executed by the at least one processor 222, cause the network service provider device 220 to perform the method.

The communications network device 200, the SON node device 210, and/or the network service provider device 220 may be implemented as separate physical devices. Alternatively, any combination of the communications network device 200, the SON node device 210, and the network service provider device 220 may be implemented as a single physical device.

The functionality described herein can be performed, at least in part, by one or more computer program product components such as software components. According to an embodiment, the communications network device 200, the SON node device 210 and/or the network service provider device 220 may comprise a processor or processor circuitry, such as for example a microcontroller, configured by the program code when executed to execute the embodiments of the operations and functionality described. Alternatively, or in addition, the functionality described herein can be performed, at least in part, by one or more hardware logic components. For example,

and without limitation, illustrative types of hardware logic components that can be used include Field-programmable Gate Arrays (FPGAs), Program-specific Integrated Circuits (ASICs), Program-specific Standard Products (ASSPs), System-on-a-chip systems (SOCs), Complex Programmable Logic Devices (CPLDs), and Graphics Processing Units (GPUs).

Any range or device value given herein may be extended or altered without losing the effect sought. Also, any embodiment may be combined with another embodiment unless explicitly disallowed.

Although the subject matter has been described in language specific to structural features and/or acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as examples of implementing the claims and other equivalent features and acts are intended to be within the scope of the claims.

It will be understood that the benefits and advantages described above may relate to one embodiment or may relate to several embodiments. The embodiments are not limited to those that solve any or all of the stated problems or those that have any or all of the stated benefits and advantages. It will further be understood that reference to 'an' item may refer to one or more of those items.

The steps of the methods described herein may be carried out in any suitable order, or simultaneously where appropriate. Additionally, individual blocks may be deleted from any of the methods without departing from the spirit and scope of the subject matter described herein. Aspects of any of the embodiments described above may be combined with aspects of any of the other embodiments described to form further embodiments without losing the effect sought.

The term 'comprising' is used herein to mean including the method, blocks or elements identified, but that such blocks or elements do not comprise an exclusive list and a method or apparatus may contain additional blocks or elements.

It will be understood that the above description is given by way of example only and that various modifications may be made by those skilled in the art. The above specification, examples and data provide a complete description of the structure and use of exemplary embodiments. Although various embodiments have been described above with a certain degree of particularity, or with reference to one or more individual embodiments, those skilled in the art could make numerous alterations to the disclosed embodiments without departing from the spirit or scope of this specification.

CLAIMS:

1. A communications network device (200),
comprising:
at least one processor (202); and
5 at least one memory (204) including computer
program code;
the at least one memory (204) and the computer
program code configured to, with the at least one
processor (202), cause the communications network device
10 (200) at least to:
decompose a communications network (110) into
service level agreement, SLA, coverage overlap regions,
SCORs, according to mobility relations between logical
network entity, LNE, pairs within the communication
15 network (110), said SCOR comprising at least one LNE
pair; and
assign a machine learning agent to at least one
of the decomposed SCORs,
wherein said machine learning agent is
20 configured to apply a deep reinforcement learning model
to solve an optimization problem related to a self-
organizing network, SON, function within its assigned
SCOR.
- 25 2. The communications network device (200)
according to claim 1, wherein LNE pairs in a SCOR
comprising at least two LNE pairs are strongly coupled,
and dependency between the SCORs is low.
- 30 3. The communications network device (200)
according to claim 2, wherein the at least one memory
(204) and the computer program code are further
configured to, with the at least one processor (202),
cause the communications network device (200) to
35 decompose the communications network (110) into the
SCORs by:

generating a logical network graph corresponding to the communications network (110) and representing the mobility relations between the LNE pairs; and

5 decomposing the logical network graph into subgraphs, said subgraphs representing SCORs comprising strongly coupled LNE pairs.

4. The communications network device (200)
10 according to claim 3, wherein vertices of the logical network graph comprise the LNE pairs, and weights of edges of the logical network graph reflect a mobility relationship between two LNE pairs.

15 5. The communications network device (200) according to any of claims 1 to 4, wherein the at least one memory (204) and the computer program code are further configured to, with the at least one processor (202), cause the communications network device (200) at
20 least to generate a profile for said subgraph, said profile comprising an adjacency matrix or an adjacency list representing the respective subgraph.

6. The communications network device (200)
25 according to claim 5, wherein said profile further comprises at least one of: a number of vertices, a number of edges, a number of involved LNEs, a degree distribution, a distribution of edge weights, a distribution of summed weights of edges incident to a
30 vertex, or at least one LNE specific feature for the respective subgraph including at least one of a deployment type, an LNE type, an associated user mobility distribution, position information, or an LNE load state.

35

7. The communications network device (200) according to any of claims 1 to 6, wherein the at least

one memory (204) and the computer program code are further configured to, with the at least one processor (202), cause the communications network device (200) at least to obtain the deep reinforcement learning model as pretrained from a SON node device (210).
5

8. The communications network device (200) according to any of claims 1 to 7, wherein states of said assigned machine learning agent comprise at least one of: LNE -specific metrics, LNE pair -specific metrics, or contextual information for capturing at least one of temporal or spatial correlations.
10

9. The communications network device (200) according to any of claims 1 to 8, wherein an action space of said assigned machine learning agent comprises a discrete action space or a continuous action space.
15

10. The communications network device (200) according to any of claims 1 to 9, wherein rewards for said assigned machine learning agent are based on at least one of: LNE pair -specific handover performance metrics, LNE -specific quality of service, QoS, performance metrics, or LNE pair -specific QoS performance metrics.
20
25

11. The communications network device (200) according to any of claims 1 to 10, wherein the SON function comprises a mobility robustness optimization, MRO, function, a coverage and capacity optimization function, or a mobility load balancing function.
30

12. The communications network device (200) according to claim 11, wherein the MRO function comprises optimization of one or more handover parameters.
35

13. The communications network device (200) according to any of claims 1 to 12, wherein said SCOR further comprises a group of physical cell boundaries, a group of logical cell boundaries, or a group of physical cell boundaries and logical cell boundaries.

14. The communications network device (200) according to any of claims 1 to 13, wherein the LNEs comprise at least one of cells, slices, or QoS flows.

15. The communications network device (200) according to any of claims 3 to 14, wherein the generating of the logical network graph comprises generating the logical network graph based on historical LNE data, statistical mobility data, or an SLA coverage map.

16. A communications network device (200), comprising means (202, 204) for:

decomposing a communications network (110) into service level agreement, SLA, coverage overlap regions, SCORs, according to mobility relations between logical network entity, LNE, pairs within the communication network (110), said SCOR comprising at least one LNE pair; and

assigning a machine learning agent to at least one of the decomposed SCORs,

wherein said machine learning agent is configured to apply a deep reinforcement learning model to solve an optimization problem related to a self-organizing network, SON, function within its assigned SCOR.

17. A method (1200), comprising:

decomposing (1201), by a communications network device, a communications network into service level agreement, SLA, coverage overlap regions, SCORs,

according to mobility relations between logical network entity, LNE, pairs within the communication network, said SCOR comprising at least one LNE pair; and

5 assigning (1204), by the communications network device, a machine learning agent to at least one of the decomposed SCORs,

wherein said machine learning agent is configured to apply a deep reinforcement learning model to solve an optimization problem related to a self-organizing network, SON, function within its assigned SCOR.

18. A computer program comprising instructions for causing a communications network device to perform at least the following:

15 decomposing a communications network into service level agreement, SLA, coverage overlap regions, SCORs, according to mobility relations between logical network entity, LNE, pairs within the communication network, said SCOR comprising at least one LNE pair; and

20 assigning a machine learning agent to at least one of the decomposed SCORs,

wherein said machine learning agent is configured to apply a deep reinforcement learning model to solve an optimization problem related to a self-organizing network, SON, function within its assigned SCOR.

19. A self-organizing network, SON, node device (210), comprising:

at least one processor (212); and

at least one memory (214) including computer program code;

the at least one memory (214) and the computer program code configured to, with the at least one processor (212), cause the SON node device (210) at least to:

receive from a communications network device (200) a request for a pretrained deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR, the request comprising at least one profile of at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs, with said SCOR comprising at least one LNE pair and said at least one subgraph corresponding to said SCOR;

determine a pretrained reference deep reinforcement learning model from a model database (230A, 230B, 230C) by a similarity analysis based on said profile of said subgraph; and

transmit the determined pretrained reference deep reinforcement learning model to the communications network device (200).

20. The SON node device (210) according to claim 19, wherein the similarity analysis comprises a relational similarity analysis between the at least one subgraph in the received at least one profile and subgraphs associated with the pretrained reference deep reinforcement learning models stored in the model database (230A, 230B, 230C).

21. The SON node device (210) according to claim 19 or 20, wherein the at least one memory (214) and the computer program code are configured to, with the at least one processor (212), further cause the SON node device (210) to:

customize the determined reference deep reinforcement learning model; and

transmit the customized deep reinforcement learning model to the communications network device (200).

22. The SON node device (210) according to any of claims 19 to 21, wherein the at least one memory (214) and the computer program code are configured to, with the at least one processor (212), further cause the SON node device (210) to:

prescreen said profile of said subgraph based on one or more prescreening parameters to determine one or more candidate pretrained reference deep reinforcement learning models,

wherein the determining of the pretrained reference deep reinforcement learning model is performed on the determined one or more candidate pretrained reference deep reinforcement learning models.

15

23. A self-organizing network, SON, node device (210), comprising means (212, 214) for:

receiving from a communications network device (200) a request for a pretrained deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR, the request comprising at least one profile of at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs, with said SCOR comprising at least one LNE pair and said at least one subgraph corresponding to said SCOR;

determining a pretrained reference deep reinforcement learning model from a model database (230A, 230B, 230C) by a similarity analysis based on said profile of said subgraph; and

transmitting the determined pretrained reference deep reinforcement learning model to the communications network device (200).

35

24. A method (1300), comprising:

receiving (1301) at a self-organizing network, SON, node device from a communications network device a request for a pretrained deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR, the request comprising at least one profile of at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs, with said SCOR comprising at least one LNE pair and said at least one subgraph corresponding to said SCOR;

determining (1303), by the SON node device, a pretrained reference deep reinforcement learning model from a model database by a similarity analysis based on said profile of said subgraph; and

transmitting (1305), by the SON node device, the determined pretrained reference deep reinforcement learning model to the communications network device.

20

25. A computer program comprising instructions for causing a self-organizing network, SON, node device to perform at least the following:

receiving from a communications network device a request for a pretrained deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR, the request comprising at least one profile of at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs, with said SCOR comprising at least one LNE pair and said at least one subgraph corresponding to said SCOR;

determining a pretrained reference deep reinforcement learning model from a model database by a

35

similarity analysis based on said profile of said subgraph; and

transmitting the determined pretrained reference deep reinforcement learning model to the communications network device.

26. A network service provider device (220), comprising:

at least one processor (222); and
at least one memory (224) including computer program code;

the at least one memory (224) and the computer program code configured to, with the at least one processor (222), cause the network service provider device (220) at least to:

obtain from at least one communications network device (200) at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs and their associated profiles;

derive at least one pretrained reference deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR, with said SCOR comprising at least one LNE pair and said SCOR corresponding to said subgraph; and

store the obtained profiles and the derived pretrained reference deep reinforcement learning models in a model database (230A, 230B, 230C).

27. The network service provider device (220) according to claim 26, wherein the model database (230A, 230B, 230C) comprises an indexed model database such that a same index represents an obtained profile of one subgraph and a corresponding derived pretrained

reference deep reinforcement learning model in a SCOR that corresponds to said subgraph.

28. A network service provider device (220),
5 comprising means (222, 224) for:

obtaining from at least one communications network device (200) at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs and their associated
10 profiles;

deriving at least one pretrained reference deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR, with said SCOR
15 comprising at least one LNE pair and said SCOR corresponding to said subgraph; and

storing the obtained profiles and the derived pretrained reference deep reinforcement learning models
20 in a model database (230A, 230B, 230C).

29. A method (1400), comprising:

obtaining (1401), by a network service provider device, from at least one communications network device
25 at least one subgraph of a logical network graph representing mobility relations between logical network entity, LNE, pairs and their associated profiles;

deriving (1402), by the network service provider device, at least one pretrained reference deep reinforcement learning model for use in solving an optimization problem related to a self-organizing network, SON, function within a service level agreement, SLA, coverage overlap region, SCOR, with said SCOR
30 comprising at least one LNE pair and said SCOR corresponding to said subgraph; and

35 storing (1403), by the network service provider device, the obtained profiles and the derived pretrained

reference deep reinforcement learning models in a model database.

30. A computer program comprising instructions
5 for causing a network service provider device to perform
at least the following:

obtaining from at least one communications
network device at least one subgraph of a logical
network graph representing mobility relations between
10 logical network entity, LNE, pairs and their associated
profiles;

deriving at least one pretrained reference deep
reinforcement learning model for use in solving an
optimization problem related to a self-organizing
15 network, SON, function within a service level agreement,
SLA, coverage overlap region, SCOR, with said SCOR
comprising at least one LNE pair and said SCOR
corresponding to said subgraph; and

storing the obtained profiles and the derived
20 pretrained reference deep reinforcement learning models
in a model database.

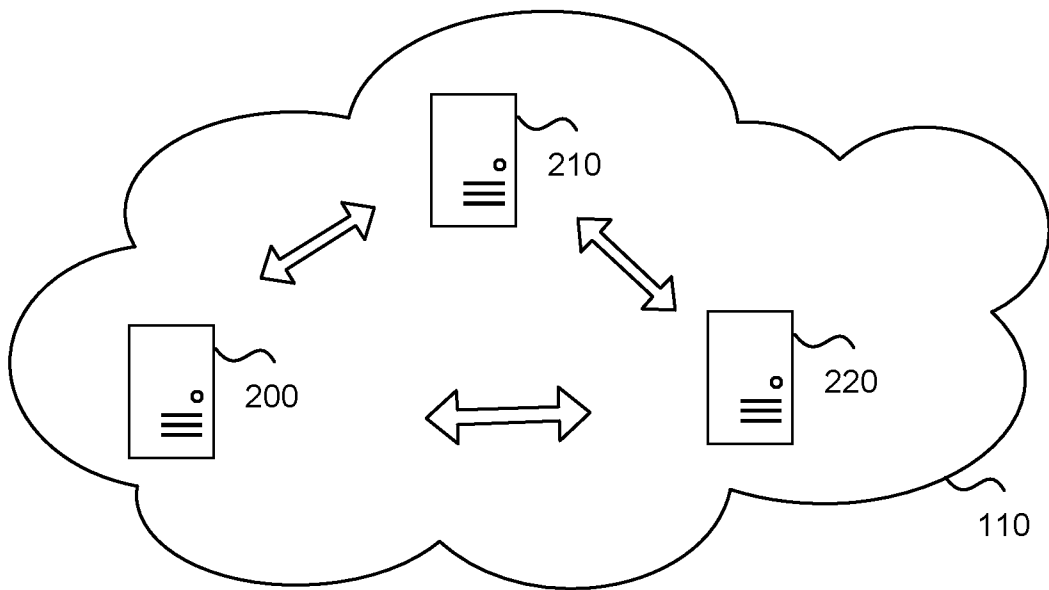


FIG. 1

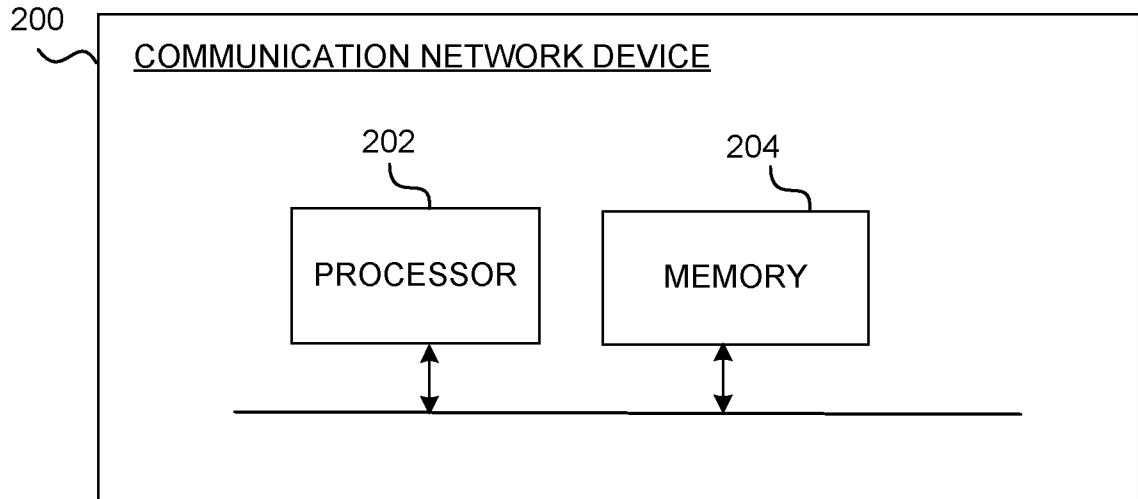


FIG. 2A

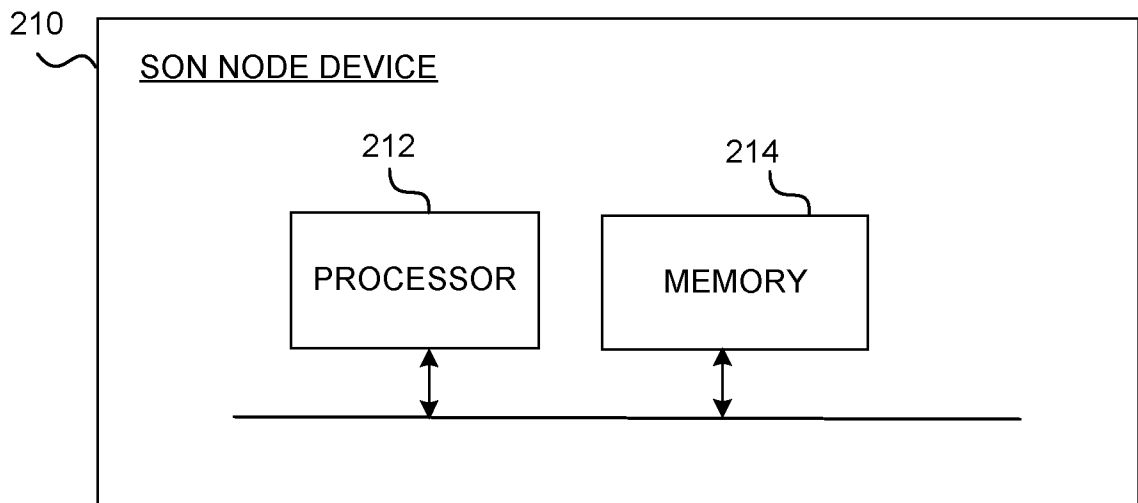


FIG. 2B

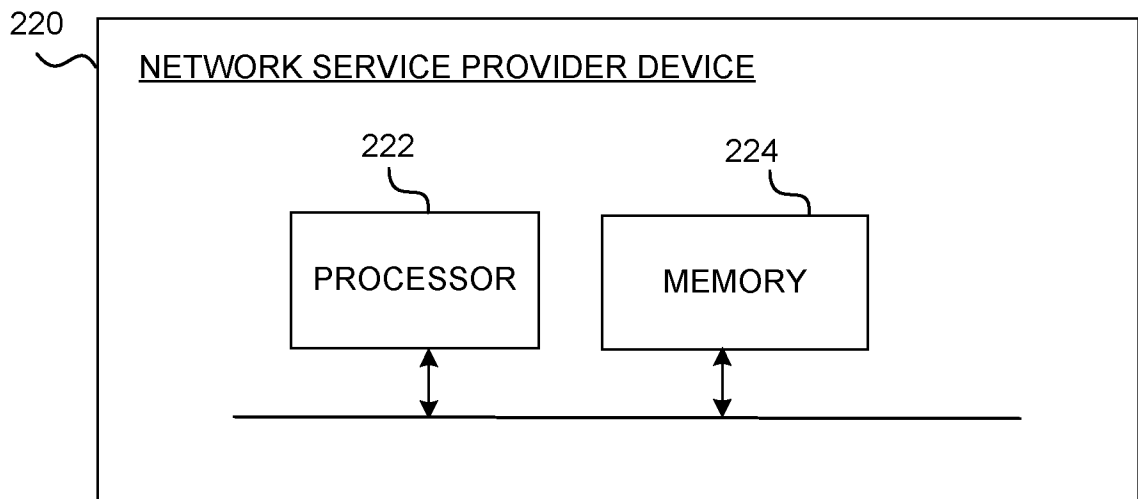


FIG. 2C

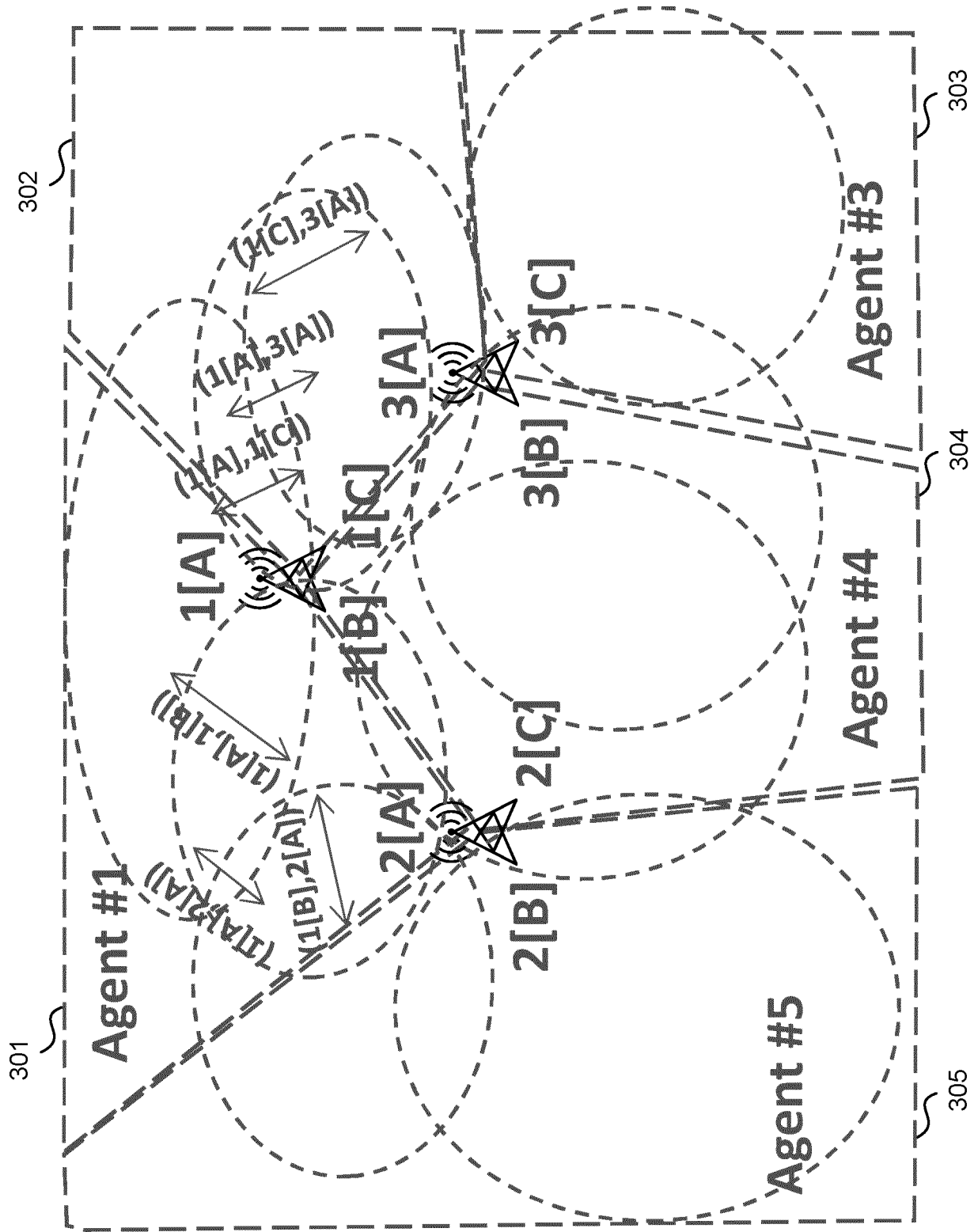


FIG. 3A

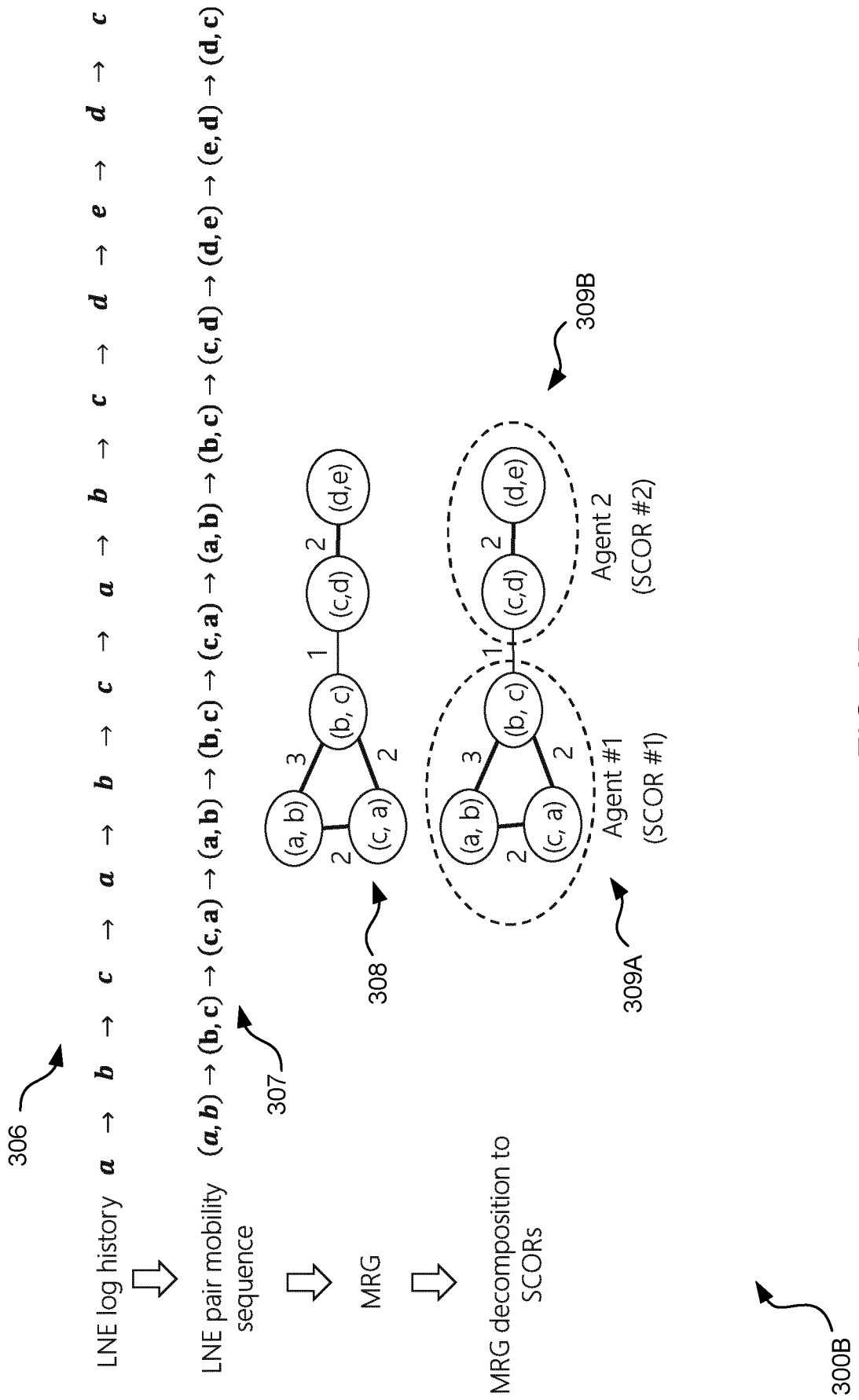


FIG. 3B

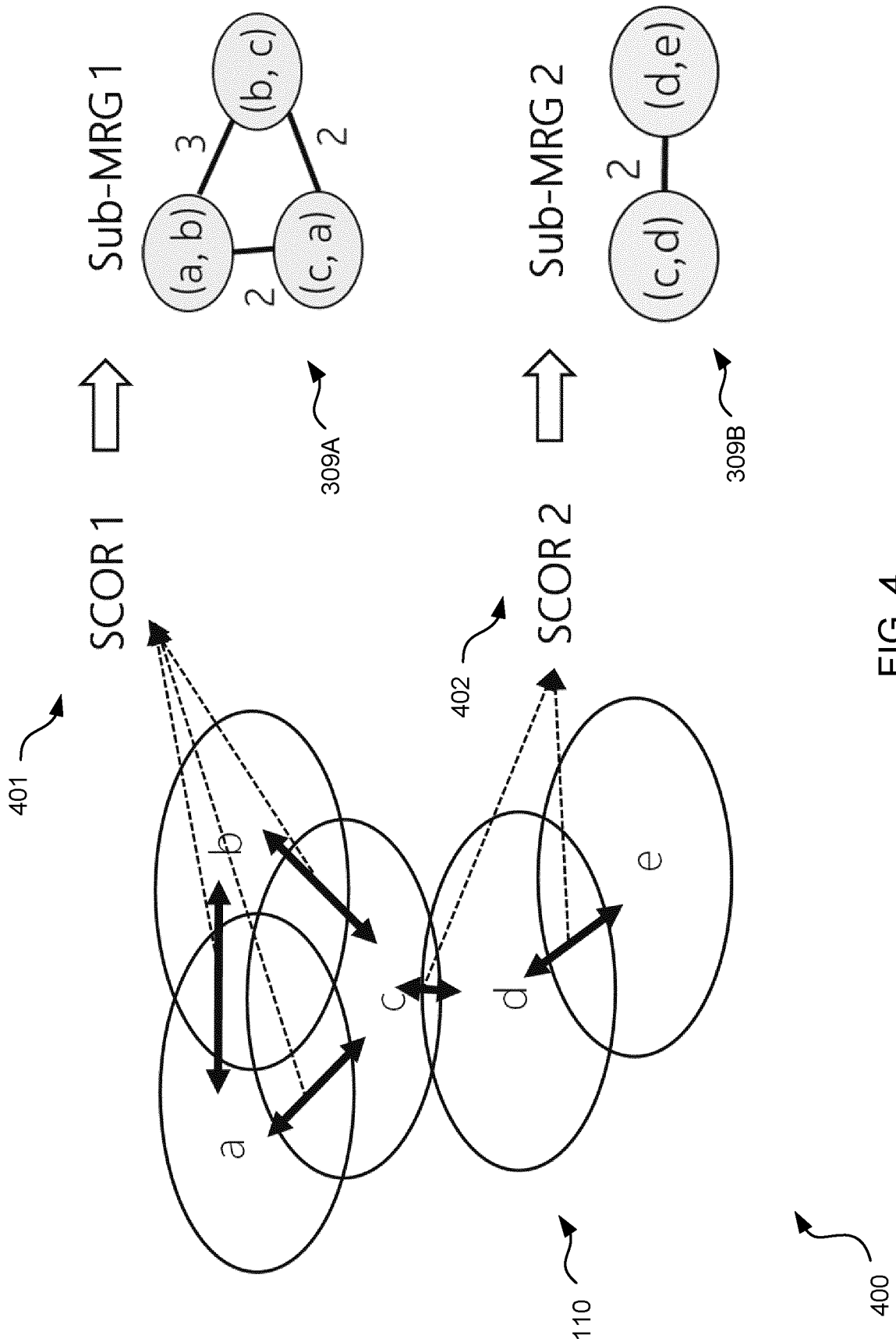


FIG. 4

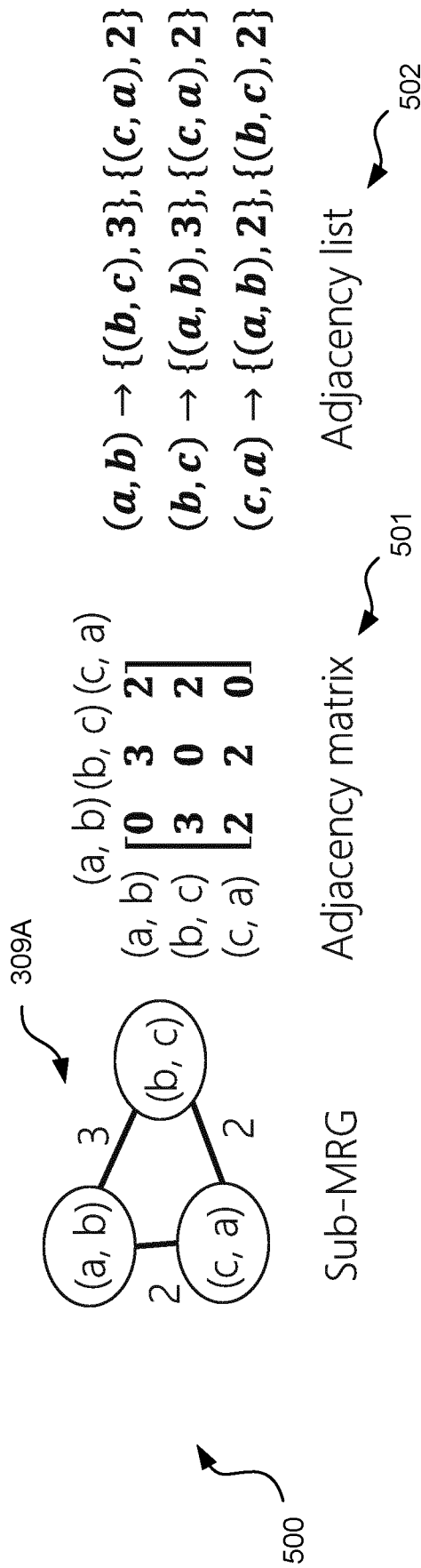


FIG. 5

Graph index	Number of vertices	Number of edges	Number of LNEs	Degree distribution	Adjacency list of the weight graph
0	3	3	3	$[0, 1, 0, 0, 0]$	$(a, b) \rightarrow \{(b, c), 3\}, \{(c, a), 2\}$ $(b, c) \rightarrow \{(a, b), 3\}, \{(c, a), 2\}$ $(c, a) \rightarrow \{(a, b), 2\}, \{(b, c), 2\}$
1	3	2	3	$\left[\frac{2}{3}, \frac{1}{3}, 0, 0, 0 \right]$	$(a, b) \rightarrow \{(b, c), 3\}$ $(b, c) \rightarrow \{(a, b), 3\}, \{(c, a), 2\}$ $(c, a) \rightarrow \{(b, c), 2\}$

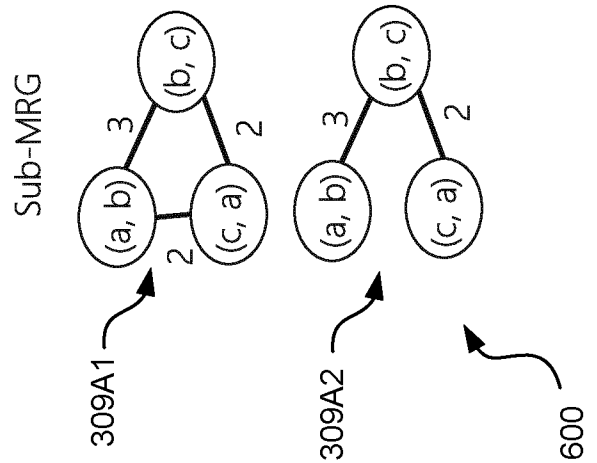


FIG. 6

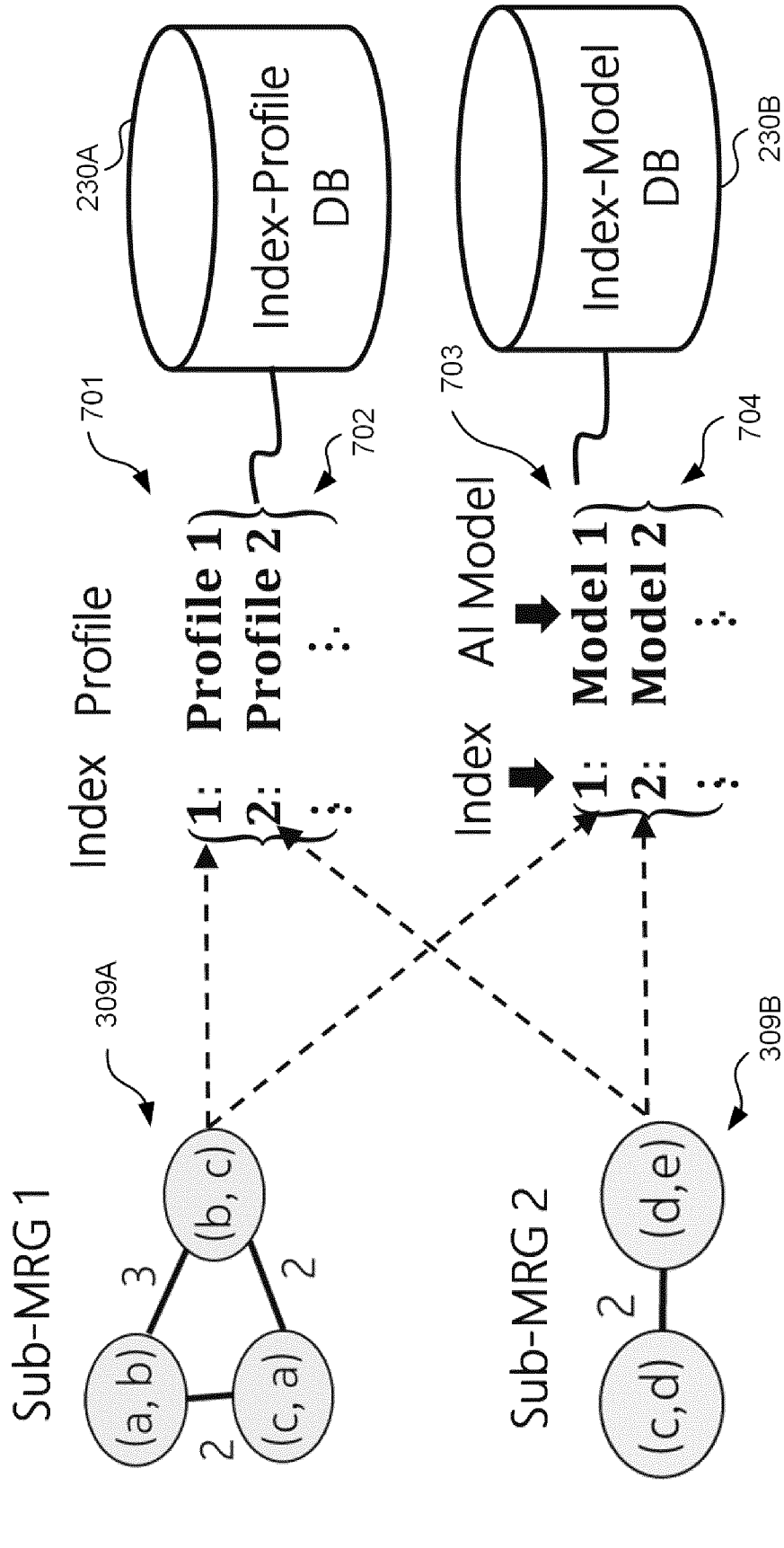
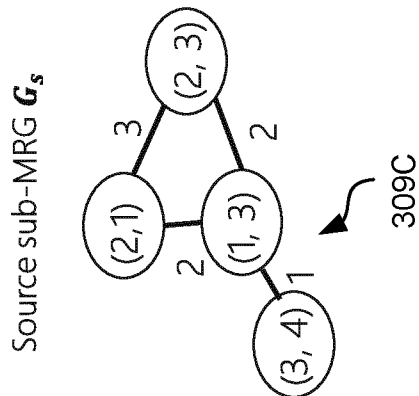
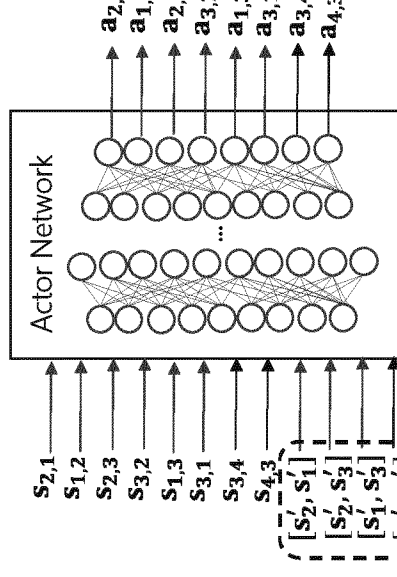
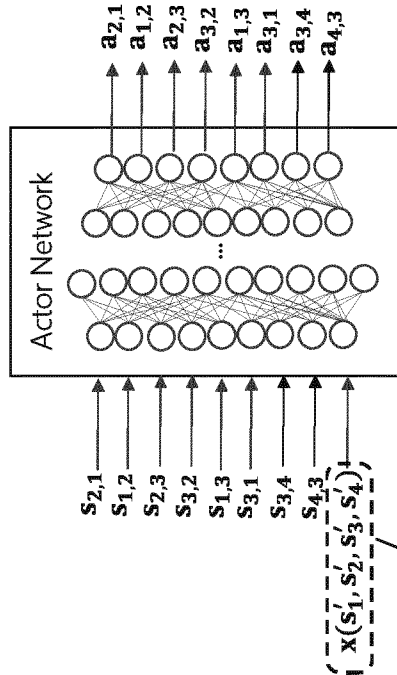


FIG. 7

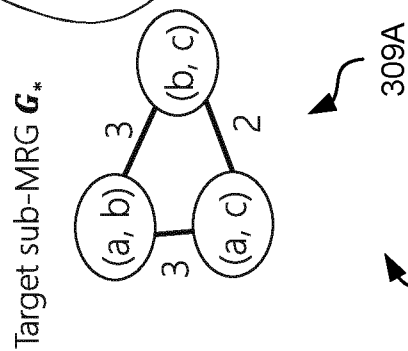
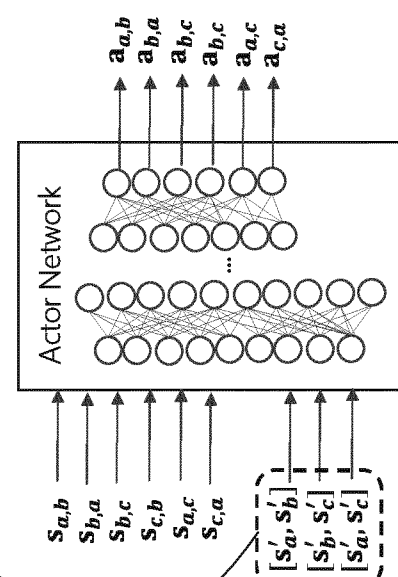
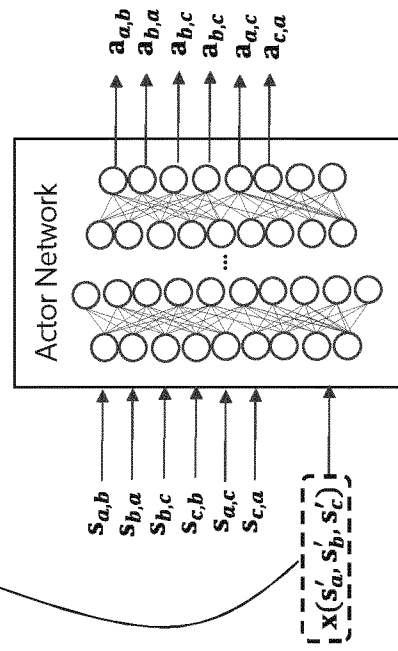
State includes extracted overall LNE features

State includes LNE-specific features



Extracted overall LNE features

LNE-specific features



801C

801A

801D

801B

309A

800

FIG. 8

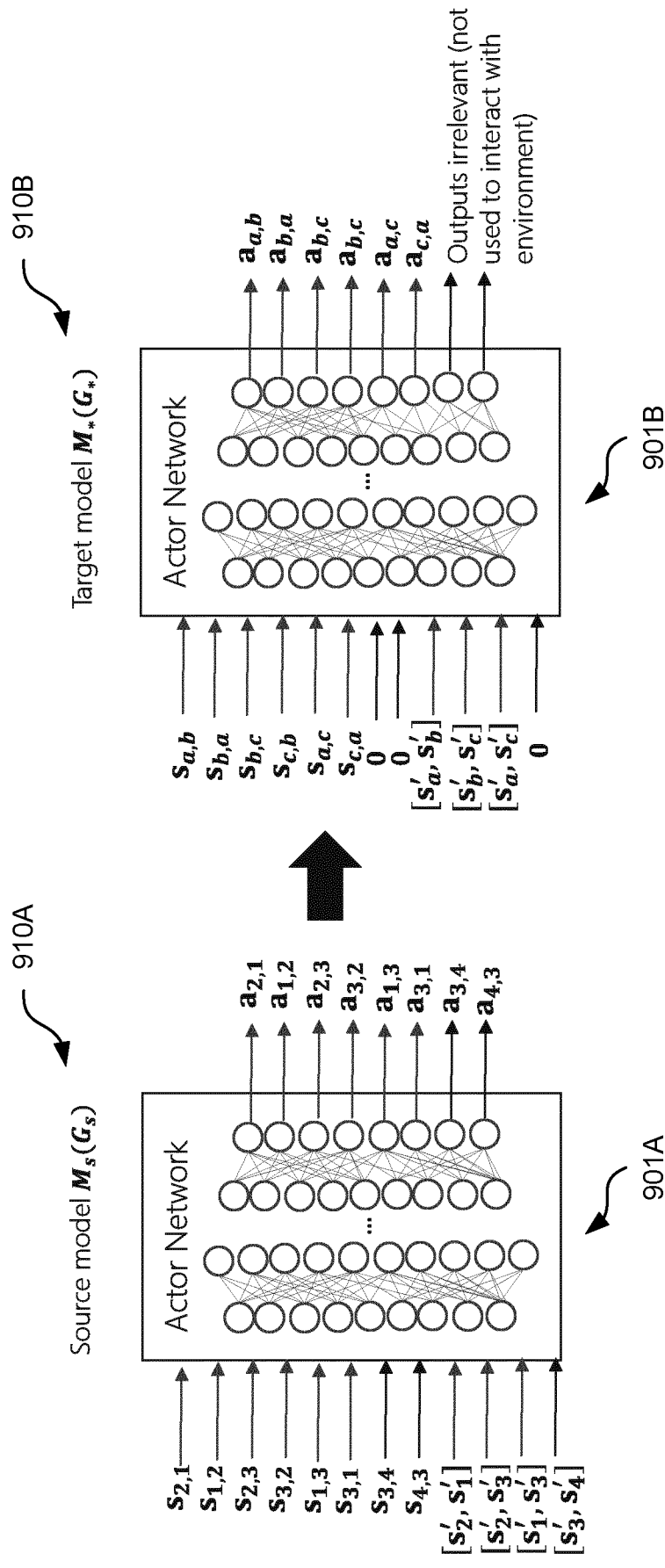


FIG. 9

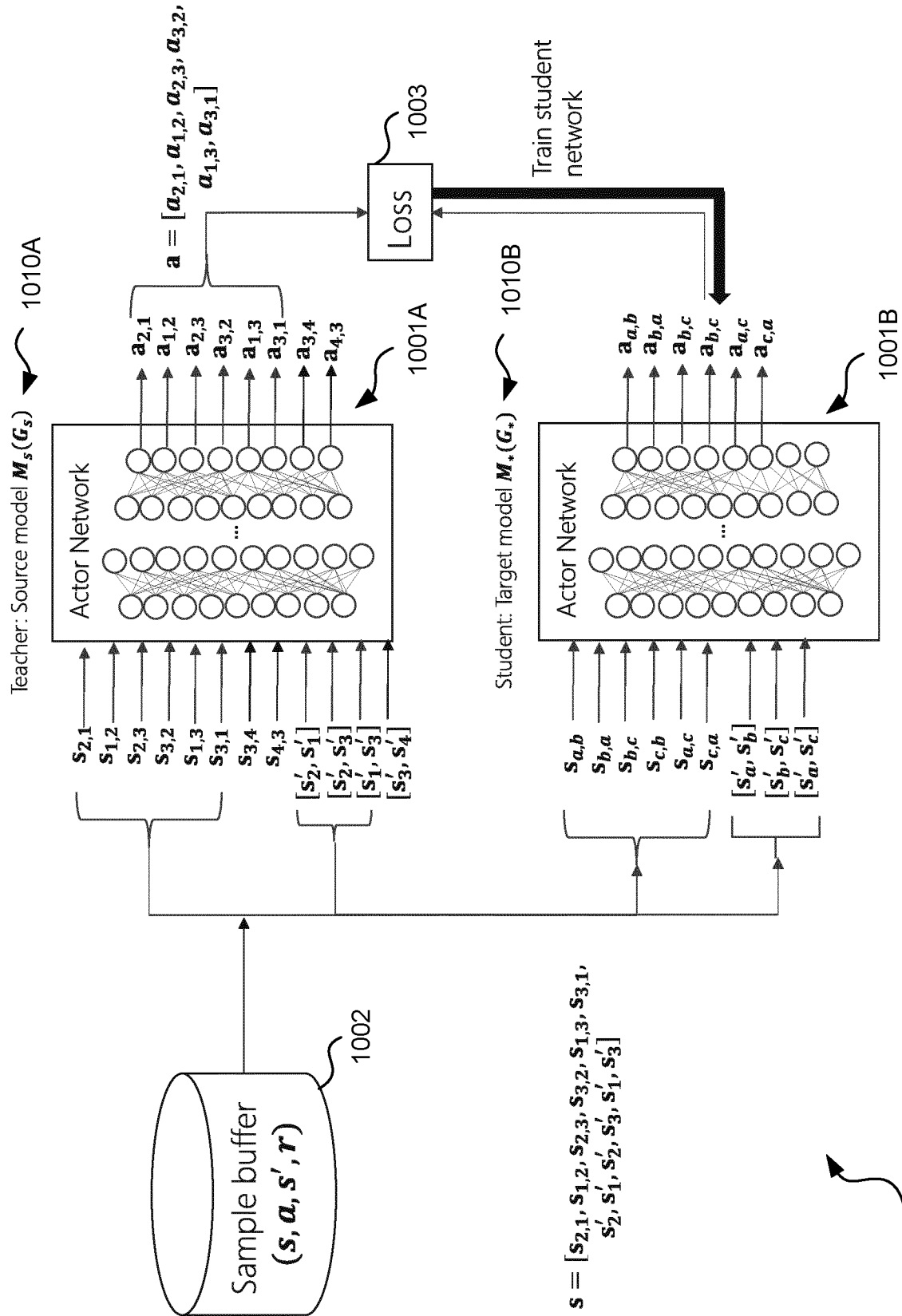


FIG. 10

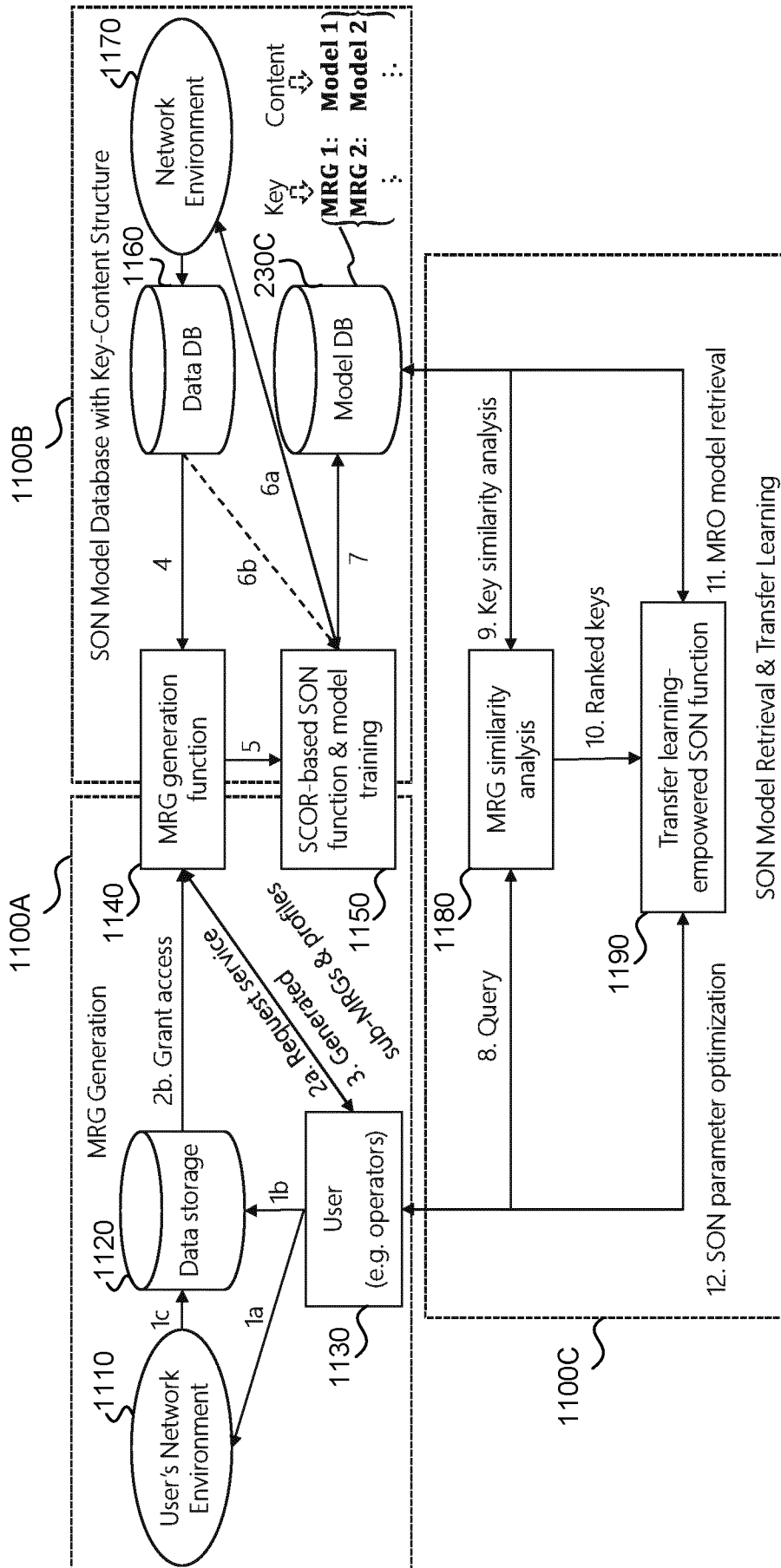
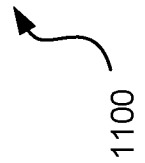


FIG. 11



1100

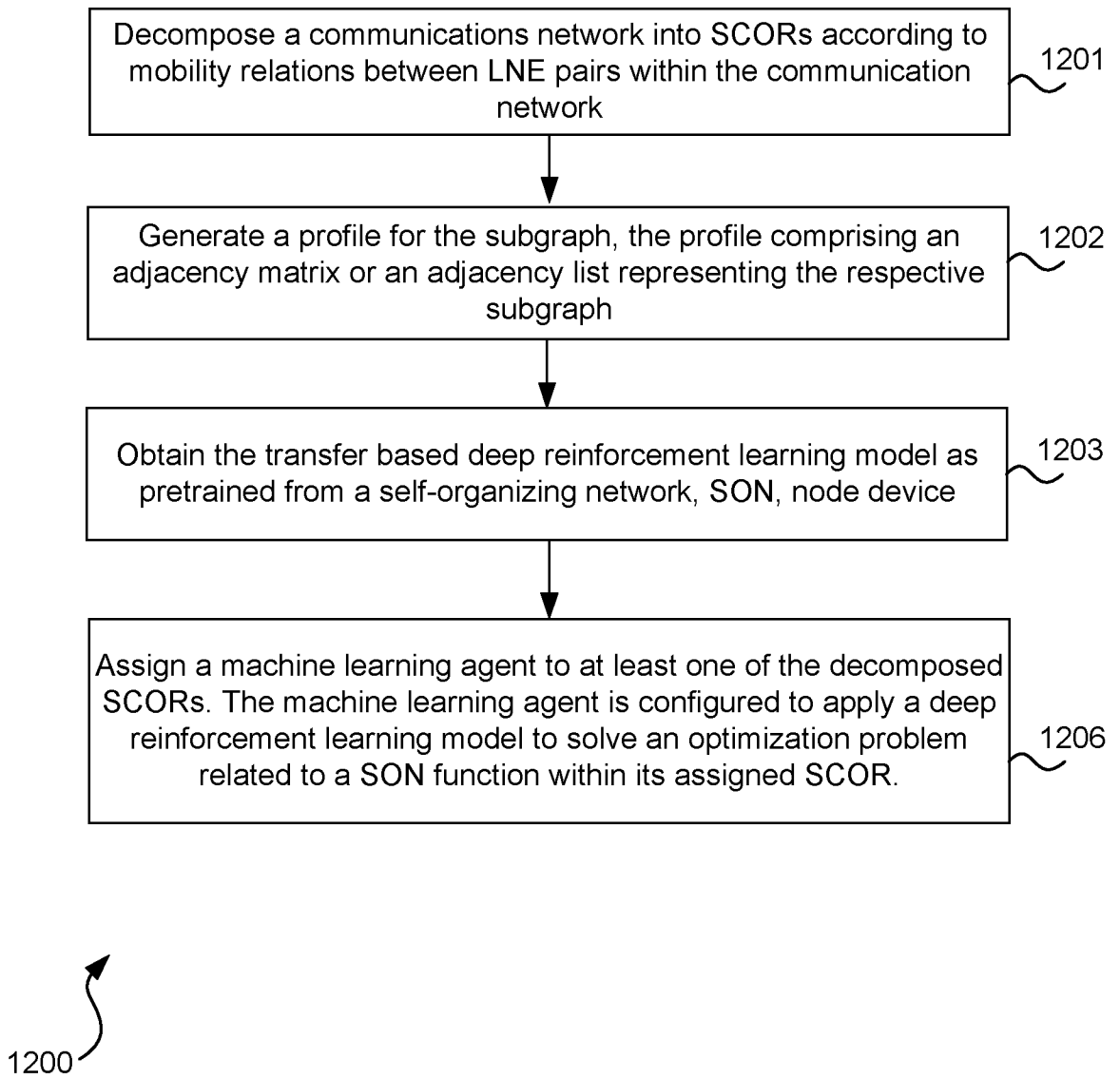


FIG. 12

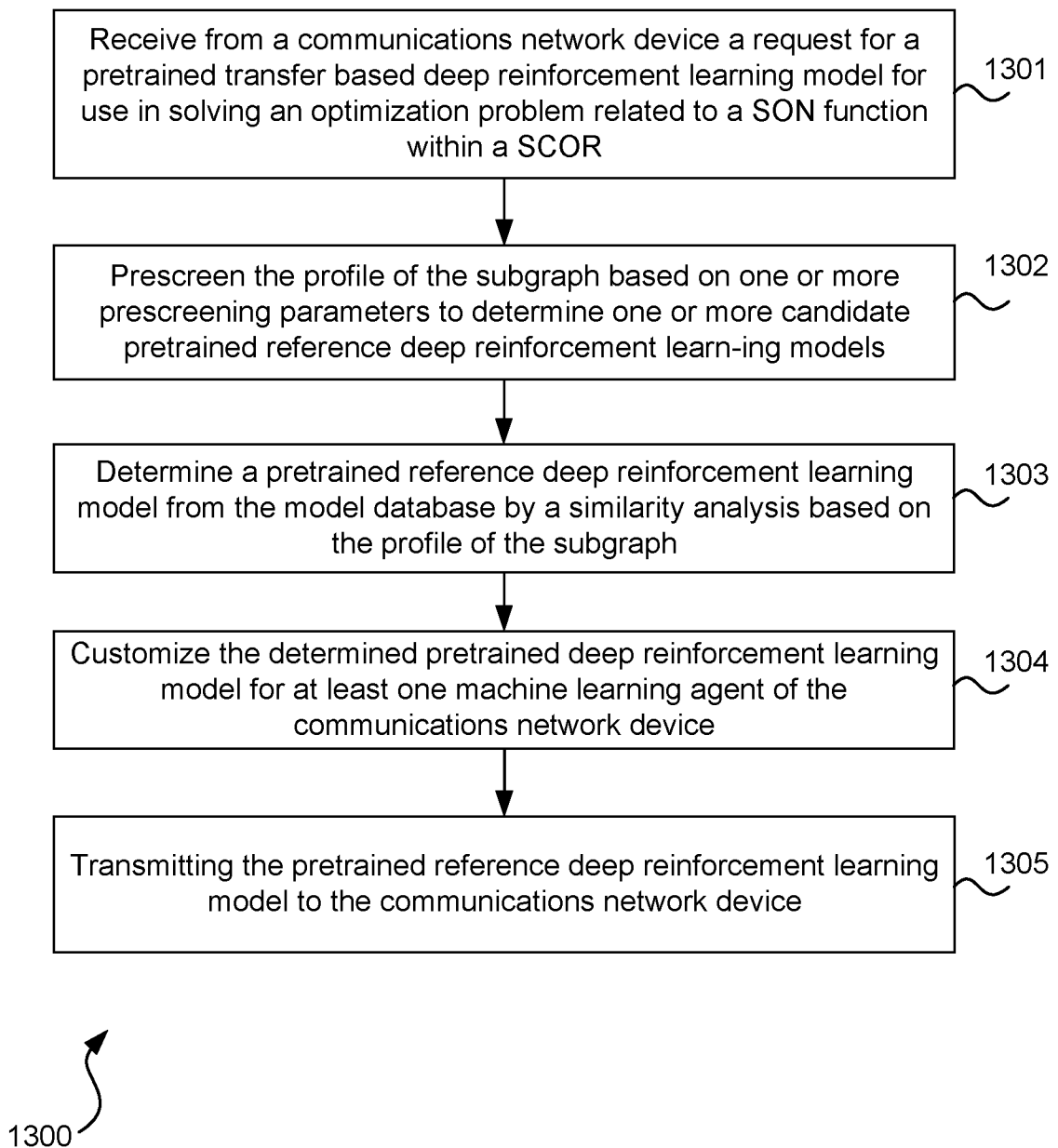


FIG. 13

14 / 18

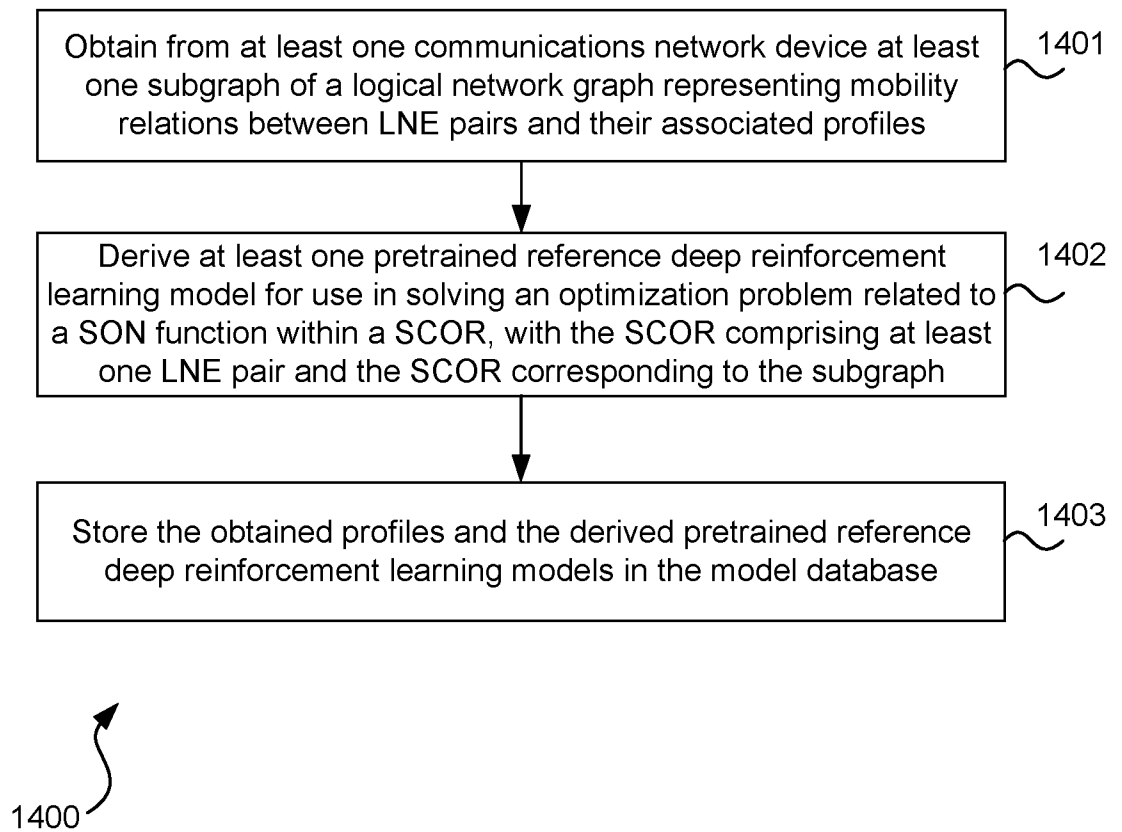
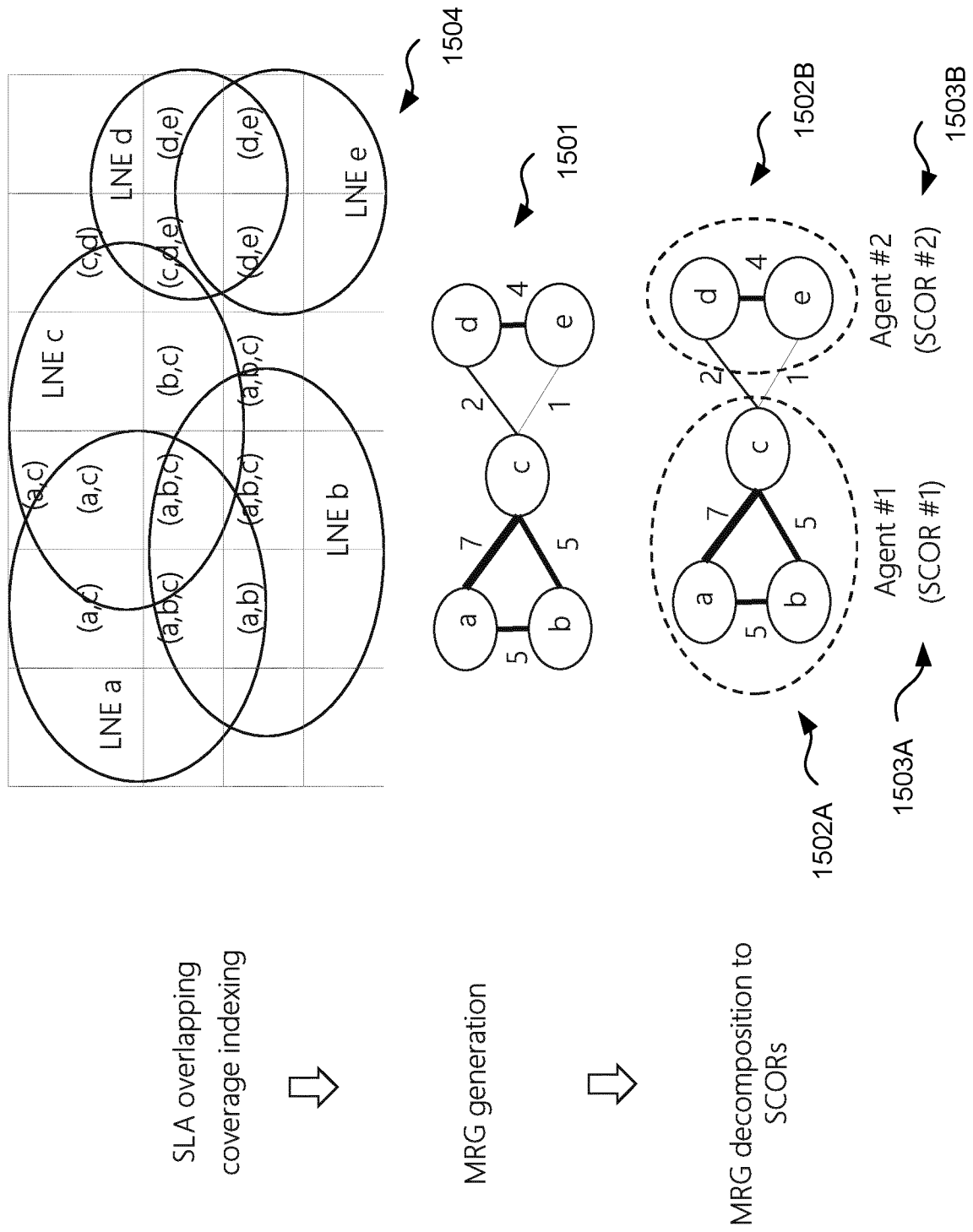


FIG. 14



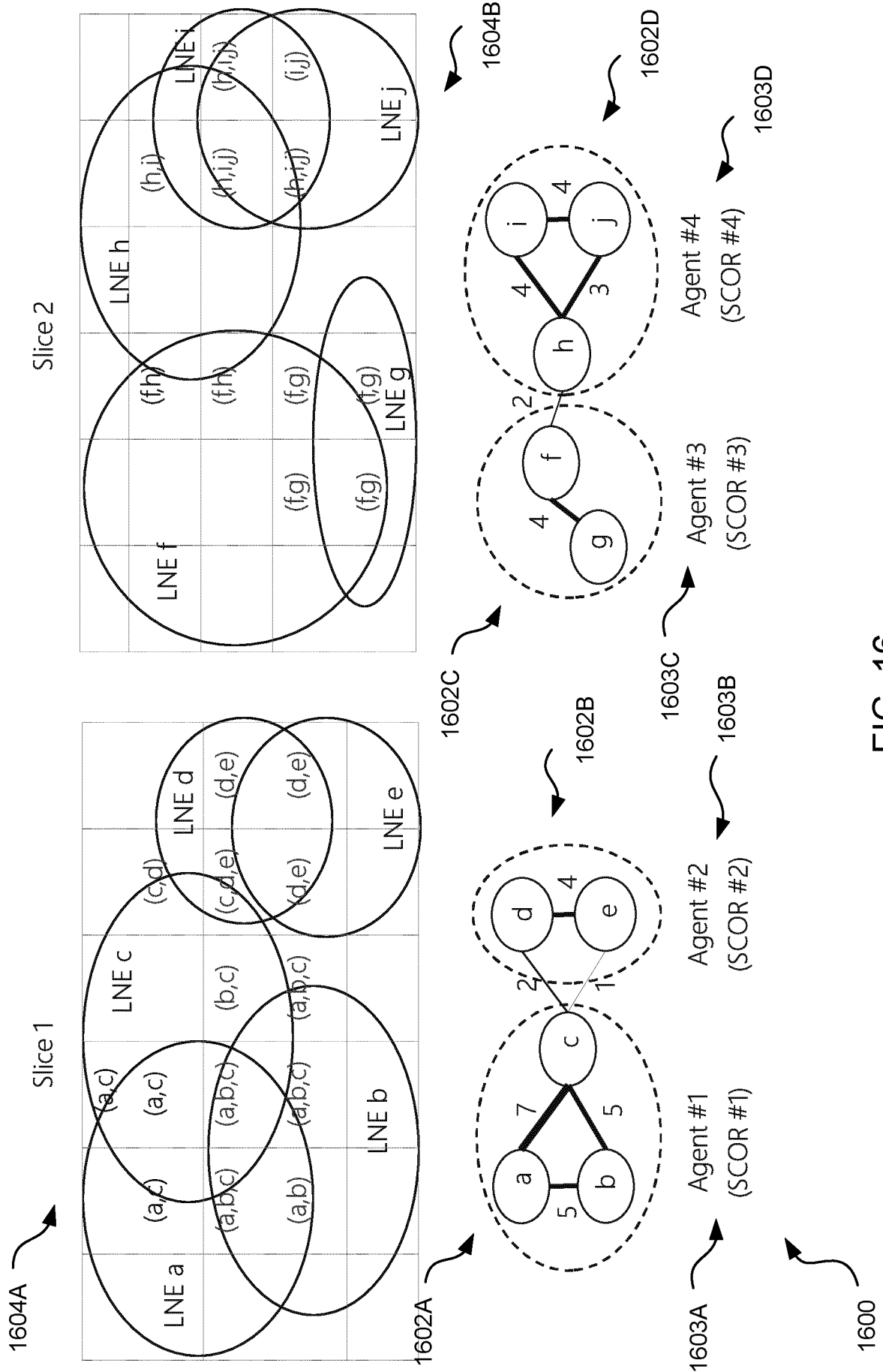


FIG. 16

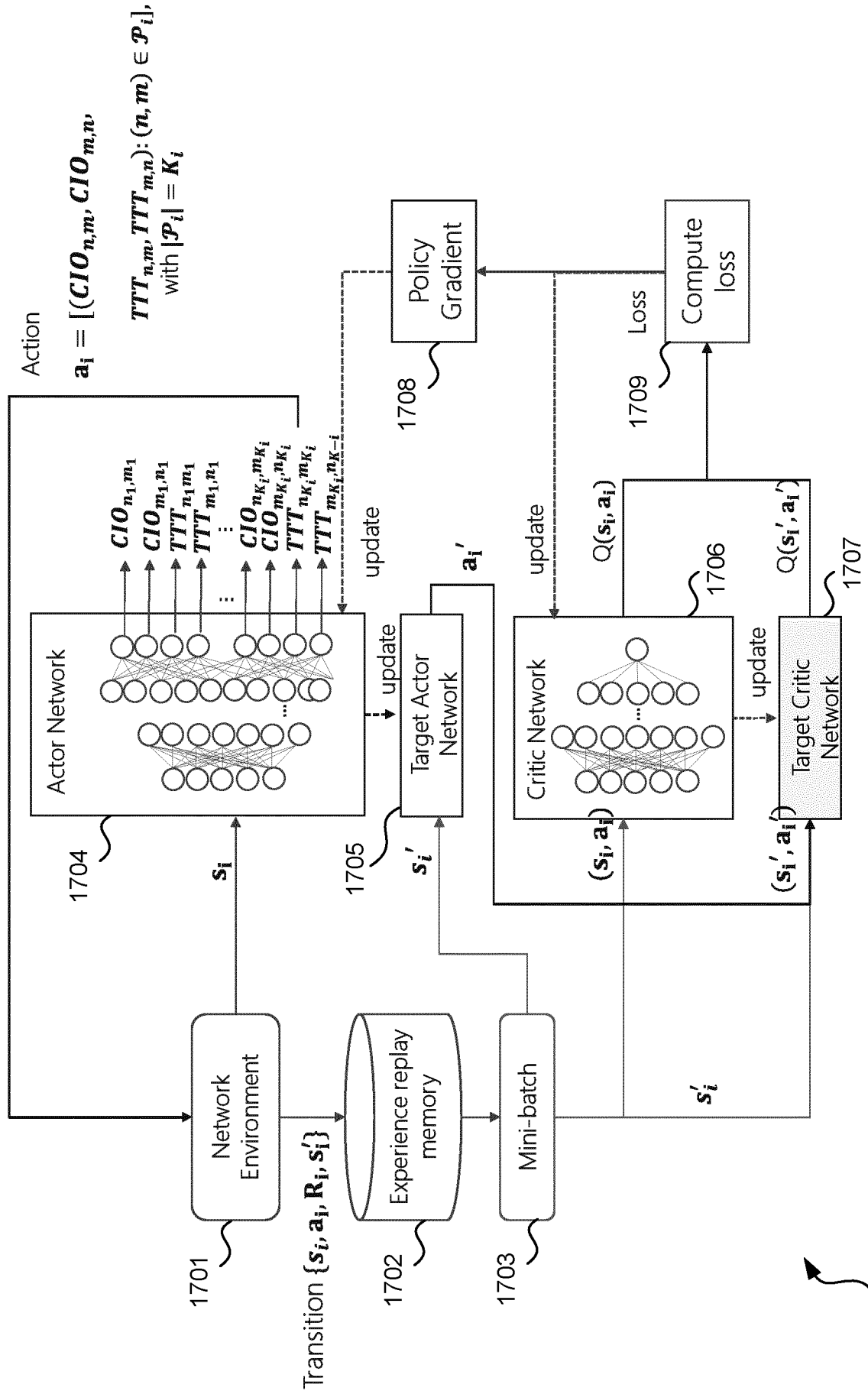
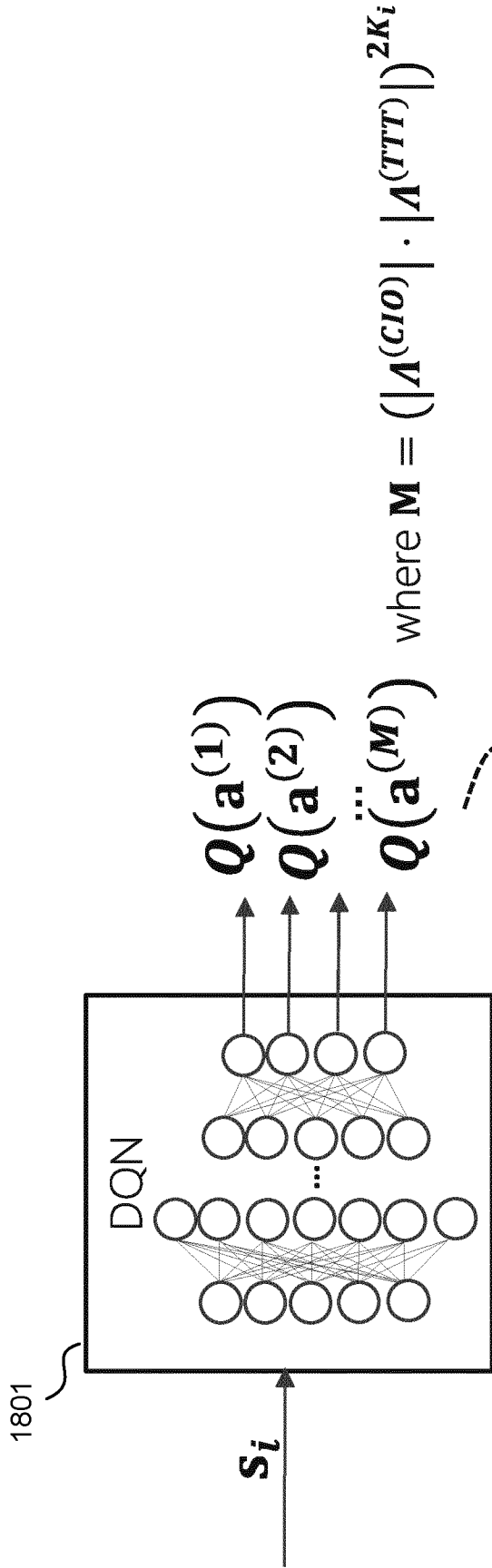


FIG. 17



where $\mathbf{M} = (|\Lambda^{(CIO)}| \cdot |\Lambda^{(TTT)}|)^{2K_i}$

$a^{(j)}$ is the j -th action of the whole action space of $[(\Delta CIO_{n,m}, \Delta TTT_{m,n}): (n, m) \in \mathcal{P}_i]$

1800 ↗

FIG. 18

INTERNATIONAL SEARCH REPORT

International application No.
PCT/EP2022/063572

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims;; it is covered by claims Nos.:

1-18

Remark on Protest

- The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2022/063572

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04W24/02
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2021/244765 A1 (ERICSSON TELEFON AB L M [SE]) 9 December 2021 (2021-12-09)	1, 2, 8-10, 13-18
Y	page 37, line 6 - page 40, line 24; figures 8, 9 page 25, line 20 - page 33, line 30; figure 6 page 22	3-7, 11, 12
Y	----- WO 2020/254376 A1 (KONINKLIJKE PHILIPS NV [NL]) 24 December 2020 (2020-12-24) paragraph [0034]; figure 2 -----	3-6
	-/--	

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier application or patent but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search

Date of mailing of the international search report

11 January 2023

14/03/2023

Name and mailing address of the ISA/
 European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040,
 Fax: (+31-70) 340-3016

Authorized officer

Cremer, Jan

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2022/063572

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	<p>NOKIA GERMANY ET AL: "ZSM005 MoA Transfer Learning", ETSI DRAFT; ZSM(19)000269R3, EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE (ETSI), 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS ; FRANCE</p> <p>, vol. ISG ZSM Zero touch network and Service Management 16 July 2019 (2019-07-16), pages 1-8, XP014348869, Retrieved from the Internet: URL:docbox.etsi.org/ISG/ZSM/05-Contributions/2019/ZSM(19)000269r3_ZSM005_MoA_Transfer_Learning.docx [retrieved on 2019-07-16] paragraph [0X.3]</p> <p>-----</p>	7, 11, 12
A	<p>MOHAJER AMIN ET AL: "Mobility-aware load Balancing for Reliable Self-Organization Networks: Multi-agent Deep Reinforcement Learning", RELIABILITY ENGINEERING AND SYSTEM SAFETY., vol. 202, 1 October 2020 (2020-10-01), page 107056, XP093012787, GB ISSN: 0951-8320, DOI: 10.1016/j.ress.2020.107056 Retrieved from the Internet: URL:https://www.sciencedirect.com/science/article/pii/S0951832020305573/pdf?md5=07d2d8690cb3454f4d851548ed4f7688&pid=1-s2.0-S0951832020305573-main.pdf> abstract</p> <p>-----</p>	1-18

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2022/063572

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2021244765	A1	09-12-2021	NONE

WO 2020254376	A1	24-12-2020	CN 114009088 A 01-02-2022
		EP 3987843 A1	27-04-2022
		US 2022263723 A1	18-08-2022
		WO 2020254376 A1	24-12-2020

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-18

Communications network device which is adapted to decompose a network into regions and assigns a machine learning agent to at least one region.

2. claims: 19-25

Self-organizing network node which is adapted to apply similarity analysis to retrieve a learning model from a database.

3. claims: 26-30

Network service provider device which is adapted to perform subgraph profiling.
