Implementation of intelligent forecasting subsystem of real-time

Alexander Eremeev, Alexander Kozhukhov
Institute of Automatics and Computer Engineering
Moscow Power Engineering Institute
Moscow, Russia
eremeev@appmat.ru, saaanchezzz@yandex.ru

Natalia Guliakina
Belarussian State University
Informatics and Radioelectronics
Minsk, Belarus
guliakina@bsuir.by

***Abstract*—The paper describes architecture of intelligent forecasting subsystem of real-time based on multi-agent temporal differences reinforcement learning, statistical module and monitoring module with milestone anytime algorithm. Analysis of anytime algorithms were made in terms of using into the forecasting subsystem type of intelligent decision support system of real-time for improving performance and reducing response and execution time. The considered tools can be used to implement the possibility of self-learning and adaptation both in the intelligent systems of real-time created on their basis, and in the actual tools for creating such systems. The work was supported by BRFFR projects 17-07-00553 a, 18-51-00007 Bel a, F16R-102.**

***Keywords*—artificial intelligence, intelligent system, real time, reinforcement learning, forecasting, decision support, anytime algorithm.**

I. Introduction

Reinforcement learning (RL) methods [1], based on the using large amount of in-formation for learning in arbitrary environment, are the most rapidly developing areas of artificial intelligence, related with the development of advanced intelligent systems of real-time (IS RT) typical example of which is an intelligent decision support system of real-time (IDSS RT) [2].

One of the most promising in terms of use in IDSS RT and central in RL is Temporal Difference (TD) learning [1]. TD-learning process is based directly on experience with TD-error, bootstrapping, in a model-free, online, and fully incremental way. Therefore, process do not require knowledge of the environment model with its rewards and the probability distribution of the next states. The fact that TD-methods are adaptive is very important for the IS of semiotic type able to adapt to changes in the controlled object and environment [3].

Using the multi-agent approach contains of groups of autonomous interacting entities (agents) having a common integration environment and capable to receive, store, process and transmit information in order to address their own and corporate (common to the group of agents) analysis tasks and synthesis information is the fastest growing and promising approach for dynamic distributed control systems and data mining systems, including IDSS RT. Multi-agent systems could be characterized by the possibility of parallel computing, exchange of experience between the agents, resiliency, scalability, etc. [4].

Usually data encountered by an online RL-agent is non-stationary, and online RL updates are strongly correlated. Deep reinforcement learning (DRL) approach provide rich representations that can enable RL-algorithms to perform effectively and enables automatic feature engineering and end-to-end learning through gradient descent, so that reliance on environment is significantly reduced or even removed. Common idea of DRL is storing the agent's data in an experience replay memory where the data can be batched or randomly sampled from different time-steps. Aggregating over memory in this way reduces non-stationarity and decorrelates updates, but at the same time limits the methods to off-policy RL-algorithms [5, 6, 7].

When modern IDSS RT are developing, important consideration should be given to means of forecasting the situation at the object, consequences of decisions, expert methods and learning tools [8]. In addition, attention should be given to optimal using of system available resources and ability to work in the environment with restrictions in time. These resources are necessary for modification and adaptation of IDSS RT regarding changes in object and external environment and for enhancing the application field and improving system performance.

For solving these problems, developed forecasting subsystem using parallel algorithms for deep reinforcement learning, statistical methods and monitoring submodule using milestone anytime algorithm that can receive acceptable information within the resources and time constraints were considered.

II. Analysis of anytime algorithms

When agent interacting in complex dynamic real-time system, where available time for planning is highly limited, generating optimal solutions can be infeasible. In such situations, the agent must be satisfied with the best solution that can be generate within the available computation time and within the range of tolerance of error. A useful set of algorithms for generating such solutions are known as anytime algorithms. Typically, these start out by computing an initial, potentially highly sub-optimal solution and then improve this solution as time allows [9].

Anytime algorithms allow making a tradeoff between computation time and solution quality, making it possible to compute approximate solutions to complex problems under time constraints. They also need to have settings that can adjust the flexibility of finding a tradeoff. They can be represented as sampling rates or iterative improvement functions that affect the quality in terms of accuracy, coverage, certainty and level of detail. A tradeoff between quality and time can be achieve by several methods:

- milestone method that executed in the minimum period of time and made subsequent evaluation of progress at control points. Based on the remaining time, the algorithm can decide to perform both mandatory and optional operations or simply mandatory operations;
- sieve functions that allows to skip the calculation steps. So, the minimum useful selection can be reached in a shorter period of time;
- multiple versions of the same algorithm in which intensive calculations can be replaced with faster but less accurate versions of the same algorithm;

For each of the above methods, it is necessary that the different implementation approaches have the ability to measure the quality of explicit metrics in the current state. At any time, the way to execute the algorithm depends on several factors such as: the quality of the available solution, the prospects for further improving the quality of the solution, current time, cost of delaying the actions taken, current state of the environment and prospects for further environmental change that could be deter-mined only in runtime. Thus, it is necessary to determine the path that provides the most optimal result relative to the current state of the environment.

Anytime algorithms should be able to be interrupt at any time or at predetermined control points, to output an intermediate result and be able to continue working using intermediate and incomplete results. Anytime algorithms are increasingly used in a number of practical areas including: planning, deep confidence networks, evaluation of impact diagrams, processing queries to databases, monitoring and collecting information, etc. This approach can make of decisive importance for complex IDSS RT, with a large number of sensors capable of analysis, and large numerical complexity of the scheduling algorithms to obtain optimal solutions in a limited time and can significantly improve the system's productivity and efficiency [10].

III. IMPLEMENTATION OF INTELLIGENT FORECASTING SUBSYSTEM OF REAL-TIME

A. Sub-module of prediction

On the basis of statistical and expert methods of forecasting was suggested combined (integrated) prediction method [11], which contains of an averaging the results obtained on the basis of the moving average method and the Bayesian approach, based on weighting coefficients. Then, resulting prediction corrected by values of series obtained by the method of exponential smoothing. After that, forecast adjusted by results of the expert methods: ranking and direct evaluations. The probability of each outcome acquired by statistical methods, increased or decreased depending on the expert assessment values for these outcomes.

The forecasting sub-module is based on the methods and algorithms described above.

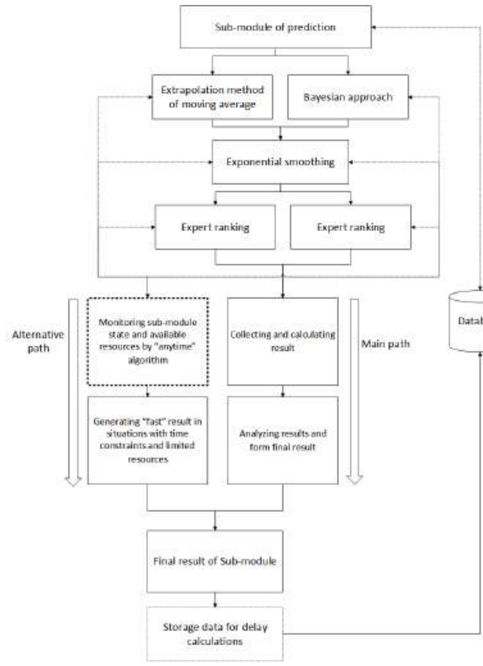

Figure 1. Architecture of sub-module of prediction

The sub-module has two paths: the main and the alternative. Under normal system conditions, sequential results obtained by each algorithm are collected and calculated. After that, the analysis and the formation of the final result of the calculation is performed.

Throughout the entire calculation process, the status of the system, the available resources and the presence of time constraints are monitored, during which it is necessary to form the result immediately, using the anytime algorithm. In such situations, the process proceeds along an alternative path when the calculations are transferred to the background (if possible) or stopped and a quick result is generated based on the performed calculations by parts of the system at the current time.

As a result, each of the ways forms the final result: "full" in the case of normal operation of the system and "approximate" in the presence of instabilities. Finally, the results are stored in a database for use in the next iterations of the subsystem.

B. Sub-module of deep reinforcement learning

Reinforcement learning sub-module consist of the group of independent agents that learning on the basis of a developed TD-methods (TD (0), TD (λ), SARSA, Q-learning).

Sub-module represents of a multi-agent network, that learning in parallel by various algorithms are divided into two networks also learning in parallel - one determines the behavior and second the objective function. Each agent consist of several additional intermediate hidden learning layers created between the input and output layer. Also each agent storing separate data in experience replay memory where the data can be batched or randomly sampled from different time-steps [12, 13]. The sub-module also has two paths: the main and the alternative. Under normal system conditions, agents are learning in parallel. After the end of episode data is collecting and analyzing, the gradient descent is calculating. Network become completely updated, formation of the final result of the calculation is performed.

Under conditions of severe time or resources constraints, system switches to alternative path: the milestone method is apply to the system. In this method algorithm chooses which of the paths is the most promising, relative to the accuracy of the forecast and the execution time, and calculates the result only by methods capable of obtaining the necessary optimal results at the current moment. In this case, all other steps can also be executed in the background and could be included in the analysis in the next steps As a result, each of the ways forms the final result: "full" in the case of normal operation of the system with completely updated network and "approximate" in the presence of instabilities. Finally, the results are stored in a database for use in the next episodes of the subsystem.

C. Architecture of prediction subsystem

Proposed architecture (Fig. 3) of prediction subsystem includes:

- emulator, which simulates the state of the environment with using of various system parameters change algorithms (linear and random) in the online database. Emulator capable simulate different constraints for the system such as time and resource constraints;
- prediction sub-module based on statistical methods (extrapolation method of moving average, exponential smoothing and the Bayesian approach) and forecasting expert methods (ranking and direct evaluation). Sub-module also contains monitoring sub-module, that capable of generating fast results and analyzing sub-module;
- multi-agent RL-learning module consist of the group of independent agents that learning on the basis of a developed TD-methods (TD (0), TD (λ), SARSA, Q-learning) divided into two networks learning in parallel. Sub-module also contains monitoring sub-module, that of generating fast results and analyzing sub-module;
- decision-making module designed for the data analysis coming from the prediction module and multi-agent RL-learning module, making decisions on follow-up actions and adjusting management strategies;
- module that collecting and analyzing statistic for valuation the effectiveness and performance of the system;
- monitoring sub-module based on milestone anytime algorithm, that analyze system state and could initialize getting fast result from all sub-modules;

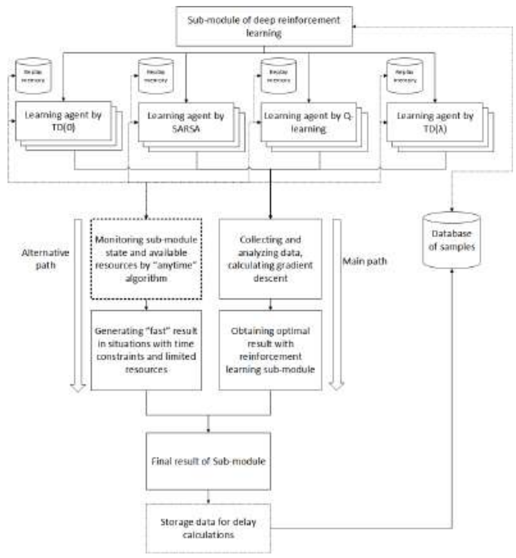

Figure 2. Architecture of sub-module of deep reinforcement learning

IV. CONCLUSION

In the paper, the basic idea and main methods for anytime algorithms, capable of finding a compromise between the computation time and the quality of the solution, that allows calculating approximate solutions of complex problems under time constraints and their basic methods in terms of use in IS RT (the systems of type IDSS RT) were described [14].

Architecture of intelligent forecasting system of real-time [15], consist of sub-module of prediction, sub-module of reinforcement learning and main decision-making and monitoring module were proposed.

The sub-module of prediction contains of statistical, expert methods and monitoring module.

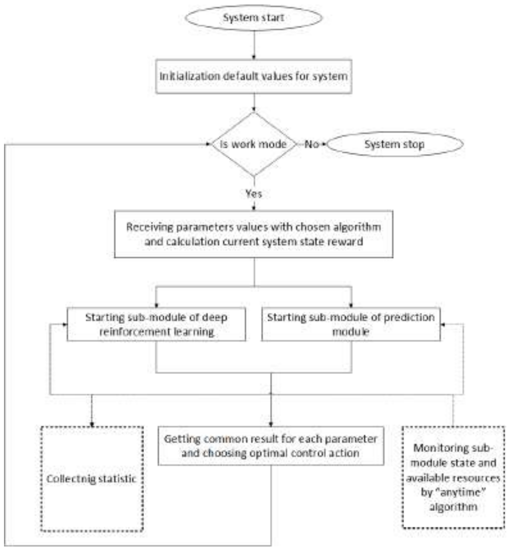

Figure 3. Architecture of forecasting subsystem

The multi-agent RL-module, contains of a group of independent agents, each of that is learning in parallel on the basis of one of the developed TD-methods (TD(0), TD(λ), SARSA, Q-learning) as well as used for the accumulation of knowledge about the environment and capable of adaptation, modification and accumulation of knowledge. Also each agent has hidden layers that created between the input and output layer, separate agent's storing data in an experience replay memory and after the end of the episode, the gradient descent is calculated.

The decision-making module is designed to analyze the data coming from the forecasting and RL modules, making decisions on follow-up actions and methods to adjust management strategies.

Monitoring module based on milestone anytime algorithm, could obtain approximate fast results in the situations with time and resource constraints.

The approach based on the integration of learning, decision-making and monitoring modules was applied in the development of a emulator prototype of IDSS RT for monitoring and control of one of the subsystems of the nuclear power plant unit. The considered tools can be used to implement the possibility of self-learning and adaptation both in the intelligent systems of real-time created on their basis, and in the actual tools for creating such systems. It is planned to include a module that takes into account temporal reasoning formalization and use the OSTIS-technology for development of the intelligent systems based on knowledge is given [16-18]. The performed studies are based on the results of previous studies of the authors, including the implementation of an earlier joint project on "The formalization of temporal reasoning in intelligent systems", supported by RFBR and BRFBR [16-17].

References

[1] R.S. Sutton, A.G.Barto. Reinforcement Learning. – London, The MIT Press, 2012, - 320 p. (Russ. Ed. Moscow: BINOM., 2011).

[2] Vagin V.N., Yeremeyev A.P. Some Basic Principles of Design of Intelligent Systems for Supporting Real-Time Decision Making // Journal of Computer and Systems Sciences International. - 2001. - No 6. Pp. 953-961.

[3] Osipov G.S. Methods of artificial intelligence. – 2nd edition. – Moscow.: FIZMATLIT, 2015 (in Russian).

[4] L. Busoniu, R. Babuska, and B. De Schutter, «Multi-agent reinforcement learning: An overview» Chapter 7 in Innovations in Multi-Agent Systems and Applications – 1 (D. Srinivasan and L.C. Jain, eds.), vol. 310 of Studies in Computational Intelligence, Berlin, Germany: Springer, 2010. - Pp. 183–221.

[5] Mnih V., Badia A.P., Mirza M., Graves A., Harley T., et al.: Asynchronous Methods for Deep Reinforcement Learning. In: Proceedings of The 33rd International Conference on Machine Learning (PMLR 48), 1928-1937 (2016).

[6] Nikolenko S., Kadurin A., Archangelskaya E.: Deep Learning. Immersion in the world of neural networks (in Russian). St. Petersburg: PITER (2017).

[7] Guo H. Generating Text with Deep Reinforcement Learning, arXiv, 2015. http:// arxiv.org/abs/1510.09292.

[8] Eremeev A.P., Kozhukhov A.A.: About implementation of machine learning tools in real-time intelligent systems (in Russian). Journal of Software and Systems, 2, 239-245 (2018).

[9] Hansen E. A., Zilberstein S.: Monitoring and control of anytime algorithms: A dynamic pro-gramming approach. Journal of Artificial Intelligence, 126, 139–157 (2001).

[10] Mangharam R., Saba A.: Anytime Algorithms for GPU Architectures. In: IEEE Real-Time Systems Symposium (2011).

[11] Sort J., Singh S., Lewis R.L.: Variance-based rewards for approximate Bayesian reinforce-ment learning. In: Proceedings of Uncertainty in Artificial Intelligence, 564-571 (2010).

[12] Mnih V., Kavukcuoglu K., Silver D., Rusu A.A., Veness J., Bellemare M.G., et al.: Human-level control through deep reinforcement learning. Nature 518, 529–533 (2015).

[13] Li Y.: Deep Reinforcement Learning: An Overview, arXiv, 2017. http://arxiv.org/abs/ 1701.07274.

[14] Likhachev M., Ferguson D., Gordon G., Stentz A., Thrun S.: Anytime Dynamic A*: An Anytime, Replanning Algorithm. In: Proceedings of the International Conference on Auto-mated Planning and Scheduling (ICAPS), 262-271 (2005).

[15] Eremeev A.P., Kozhukhov A.A.: About the integration of learning and decision-making models in intelligent systems of real-time. In: Pro-ceedings of the Second International Scientific Conference "Intelligent Information Technol-ogies for Industry" (IITI'18). Springer. Volume 2. Pp. 181-189 (2018).

[16] Vagin V.N., Eremeev A.P., Guliakina N.A.: About the temporal reasoning formalization in the intelligent systems: In: Proceedings of the International Conference on Open Semantic Technologies for Intelligent Systems (OSTIS-2016). Minsk. Pp. 275-282 (2016).

[17] Golenkov V.V., Gulyakina N.A., Grakova N.V., Nikulenka V.Y., Eremeev A. P., Tarasov V.B.: From training intelligent systems to training their development means: In: Proceedings of the International Conference on Open Semantic Technologies for Intelligent Systems (OSTIS-2018). Minsk. Vol. 2, N 8. Pp. 81-99 (2018).

[18] Eremeev A.P., Kozhukhov A.A., Golenkov V.V., Guliakina N.A.: On the implementation of the machine learning tools in intelligent systems of real-time // Journal of Software and Systems. - 2018. – Vol. 31, No. 2. Pp. 81-99 (in Russian).

РЕАЛИЗАЦИЯ ИНТЕЛЛЕКТУАЛЬНОЙ ПОДСИСТЕМЫ ПРОГНОЗИРОВАНИЯ РЕАЛЬНОГО ВРЕМЕНИ

А. Еремеев, А. Кожухов, Н. Гулякина

В статье описывается архитектура интеллектуальной подсистемы прогнозирования реального времени, основанная на мультиагентном обучении с подкреплением на основе временных различий, статистическом модуле и модуле мониторинга на основе гибких алгоритмов. Анализ гибких алгоритмов проводился с точки зрения использования в подсистеме прогнозирования типа интеллектуальных систем поддержки принятия решений реального времени, для повышения производительности и уменьшения времени отклика и выполнения. Предложенные инструменты могут быть использованы для реализации возможности самообучения и адаптации как в интеллектуальных системах реального времени, созданных на их основе, так и в реальных инструментах для создания таких систем. Работа выполнена при поддержке проектов РФФИ 17-07-00553 а, 18-51-00007 Бел-а.

Received 24.12.18