# Intelligent virtual agent, learning how to reach a goal by making the least number of compromises

# Intelligent virtual agent, learning how to reach a goal by making the least number of compromises

**Dilyana Budakova, Veselka Petrova-Dimitrova, Lyudmil Dakovski**

Technical University of Sofia, Plovdiv Branch, Bulgaria
Technical University of Sofia, Plovdiv Branch, Bulgaria
European Polytechnical University, Pernik, Bulgaria

dilyana_budakova@yahoo.com, vesi_s_petrova@yahoo.com, l.dakovski@gmail.com,

**Abstract.** The learning process in the Q-learning algorithm is characterized by maximizing a single, numerical reward signal. However, there are tasks for which the requirements toward the way to reach a goal are complex. This paper proposes a modification to the Q-learning algorithm. In order to make the Q-learning agent find the optimal path to the goal by meeting particular complex criteria, the use of measures model (a model of environment criteria), represented as a new memory matrix, is introduced. If the goal cannot be reached by following the pre-set criteria, the learning agent can compromise a given criterion. The agent makes the least possible number of tradeoffs in order to reach the goal. If the criteria are arranged by their level of importance, then the agent can choose more in number and more acceptable compromises. The aim of the modification is to empower the learning agent to control the way of reaching a goal. The modified algorithm has been applied to training smart shopping-cart learning agents. The tests show improvement in their behaviour.

**Keywords:** Intelligent system, Reinforcement learning, Smart shopping-cart learning agents

## 1. Introduction

Q-learning is a model-free reinforcement learning algorithm [1][2][3]. The learning process is characterized by maximizing a single, numerical reward signal and by interaction with an unknown environment. The teacher does not point at the actions to be undertaken. Instead, the trainee has to find out those, leading to the greatest reward and then to try to realize them. In the most interesting and challenging cases, not only the immediate reward could be taken into account when choosing an action, but also the further situations and the future rewards.

There are very many ways to improve the reinforcement learning algorithms [4-8]. It is known, for example, that Imitation Learning is a way for their optimization. [4][5][6]. In [7][8] the efforts of the authors are addressed to considering real-life problems that involve multiple objectives, often giving rise to a conflict of interests. The authors propose multi-objective reinforcement learning (MORL) algorithms that provide one or more Pareto optimal balances of the problem's original objectives. In the case the decision maker's preferences are clear and known a priori, single-policy techniques such as secularization functions can be employed to guide the search toward a particular compromise solution [7]. In case the preference of the decision maker is unclear before the optimization process takes place,

it might be appropriate to provide a set of Pareto optimal compromise solutions to the decision maker, each comprising a different balance of objectives [7]. A more advanced idea is to simultaneously learn a set of compromise solutions [7]. In other words, criteria are to be set, characterizing a goal, and solutions proposed that lead to another tradeoff goal, optimally balancing between the pre-set requirements.

This paper considers objectives, for which the requirements toward the way of reaching a goal are far too complex to be described by a single, numerical reward criterion. In order to make the Q-learning agent find the optimal path by meeting specific complex criteria, the measures model (a model of environmental criteria), represented as a new memory matrix, is introduced. When it is impossible to reach the goal by following the given criteria, then the agent arrives at compromising a criterion. One option for the agent in the proposed algorithm modification is to make the least possible number of compromise solutions in order to reach the goal. This option has been studied in the present paper. There is another option suitable for the case when there are a lot of requirements toward the way of reaching the goal. The criteria can be arranged by their level of importance then. In such a situation the agent can make more in number and more acceptable tradeoffs. The aim of the proposed modification to the Q-learning algorithm is to empower the agent to control the way of reaching a given goal.

An intelligent system for smart shopping modeling allows for applying and improving the learning algorithms. Therefore the proposed algorithm is implemented to training Smart Shopping Cart Learning Agents (SSCLA). The design of the appropriate intelligent system for smart shopping is described in [9]. The proposed prototype of the system incorporates SSCLA, beacon-based technology, holographic technology, picture exchange communication system, text-to-speech and speech recognizing technology, face recognizing and machine learning techniques. It is envisaged that concrete implementation of the shopping agents will be running on each shopping cart in the shopping centers or on holographic displays [10][11]. The design of the smart shopping cart can be different [10][12]. The k-d decision tree, the best identification tree, and the reinforcement-learning algorithm are used for training the agents. The performance measures, to which both the intelligent system for smart shopping and the shopping agents aspire, include: getting to the correct shop in the shopping mall; getting to the new promotion in the shopping mall; minimizing the path when going through the shops from the shopping list; maximizing passenger comfort; maximizing purchases; and enabling people with different communication possibilities to navigate and shop in the big shopping centers [9-11].

The studies show improvement in SSCLA behavior in result of applying the proposed modified Q-learning algorithm.

The rest of the paper is structured as it follows: in Section II the realization of a Goal-Based Reinforcement Learning Agent is described; in Section III the Q-learning algorithm modification realization is explained; a survey of the performance of the Q-learning algorithm with an introduced measures model (a model of environmental criteria), presented as memory Matrix K is presented in Section IV; the section for discussion and future work is Section V; in the VI-th Section a number of conclusions are drawn.

## 2. Goal-Based Learning Agent

All reinforcement learning agents have explicit goals, can sense aspects in their environment and choose actions to influence it accordingly. The agent is realized by a program, matching the way the agent perceives reality and the actions it undertakes.

A reinforcement learning algorithms is used for the Goal-based learning agent in [10][11]. The agent receives the shopping list from the customer (this is what the agent perceives) and informs the customer about the sequence of shops he/she can visit in order to buy all the goods needed (these are the actions the agent undertakes). The shortest possible route is suggested, in accordance with the particular shopping list.

Since the goal is to visit all the shops from the shopping list, the particular shopping list can be regarded as a plan or a sequence of goals to achieve in order to fulfill the task completely.

The environment model is a graph (Figure 2) of the different environment conditions. The nodes in the graph are the shops in the exemplary shopping mall. The edges point at the shops, between which there is a transition. Then, this graph is presented by a reward matrix (Figure 1). The number of rows and columns in this matrix is equal to the number of shops in the mall. Zero is put in the matrix in a place where there is a connection between the number of a shop, set by a number of a row, and the number of a shop, given by a number of a column. Values of -1 are placed in the other positions of the reward matrix. The rewards model is needed to set a goal for the agent. Reaching every shop from the customer's shopping list is such a goal. Since the agent is a goal-based one, its behavior can be changed by just setting a new goal, changing the rewards model [1]. A reward is only given when the agent gets to a particular shop.

The agent's memory is modeled by presenting it with the help of an M-matrix (Memory of the agent). The rows in the M-matrix represent the current location of the customer, while the columns are the shops, where he/she can go. It is assumed at the beginning that the agent does not have any knowledge and therefore all elements in the M-matrix are zeros.

The rule for calculating the current location of the customer at the moment of choosing the next shop to visit is as it follows: M (current location of the customer, chosen shop to visit next) = R(current location of the customer, next shop) + $\Upsilon$. Max[M(next shop, all possible shops where the customer could go from the next shop)].

The following is taken into account in the above formula: The immediate reward, obtained when the customer decides from the current location to go to a next shop: R(current position, chosen shop to go next); the biggest possible future reward - this is the biggest reward, chosen from among all rewards, which would have been obtained when the customer goes out of the next shop and enters any possible other shop: Max[M(next shop, all the shops where it is possible to go from the next shop)]. The value of the learning parameter $\Upsilon$ defines the extent, to which the agent will take into account the value of the future reward. The value of the learning parameter $\Upsilon$ is within 0 to 1 ($0 \leq \Upsilon < 1$). If $\Upsilon$ is closer to zero, then the agent will prefer to consider only the immediate reward. Experiments have shown that in this case it is impossible to teach the agent to achieve the goal. If $\Upsilon$ is closer to one, then the agent will consider the future reward to a greater extent. This is the better option for successful training of the agent. The value of the learning parameter was experimentally chosen to be $\Upsilon$=0.8 [11]. One of all shops is chosen, where it is possible to go from the current position. The shop, to which the customer would go next is considered. For this next position now all the shops, to which it is possible to go further are considered. The value of the highest reward is taken. The next position is then set as a current one.

## 3. Q-learning algorithm modification realization

The task considered here is related to smart shopping realization and it does not allow a teacher to show how to reach the goal in order to achieve better results. Therefore Imitation Learning cannot be applied because of the availability of lots of ways for achieving a particular goal. Besides, the goal is different every time. The shops and stands in the Shopping Center that customers want to reach are different. Some of them look for promotional goods; others need artwork. Some customers use the shopping process as a therapy and want to reach the most frequently visited shops and go through the busiest lobbies and hallways; others want to avoid the crowded zones. So it is important in this task not only to reach the goal. The way of reaching it, together with the criteria, which a certain path meets, are of the same level of importance.

In order to make the Q-learning agent find the optimal sequence of lobbies or hallways by meeting specific criteria, the use of measures model represented as memory K matrix is introduced.

For the purposes of the experiment the Shopping Center is represented by a graph with 17 nodes and 36 edges between them as shown in Figure 2. Every shop in the considered Shopping Center is represented as a node. Every lobby or a hallway, connecting the shops, is represented by an edge. The busiest and most wanted to go through lobbies or hallways are marked in orange (Figure 2) and have a measure of 1 in the K matrix (Figure 1). The secondary, distant, non-desired pathways are marked in blue (Figure 2) and have a measure of 2 in the K matrix (Figure 1). The measures model presented by

the K matrix K is similar to the environment reward model, presented by the R matrix (Figure 1). The values of a given criterion, to which each edge in the graph corresponds, are kept in the K matrix. Minus one (-1) in the R matrix and in the K matrix says that there is no edge in this place of the graph. The learning algorithm is changed now. The agent has to go only through those edges in the graph, which have a specific measure value in the K matrix. In other words, an edge to a given node could exist according to the reward model, but this edge should have a precisely defined measure value in the K matrix as well. It means that some nodes remain inaccessible to the agent because the edges that lead to them do not meet the specified criteria in the K matrix.

In some cases it may be found that there is no path to a goal that meets the pre-set criteria. Then the maximum number of compromises the agent can make by selecting edges that have a different measure value from the required one, is set. The rule allows compromises that are less than or equal to (<=) the maximum pre-set number of tradeoffs. As a result, the agent finds the optimal path by making the least number of compromise solutions. When the maximum number of compromises is made but the goal is not reached yet, the agent moves on to searching for another way to reach it.

The software implementation introduces the measures model presented by the K matrix and changes the source code of the function, determining all those possible shops, where the customer could decide to go from the next visited shop. The rule for calculating the current location of the customer at the moment when he/she chooses the next shop to visit does not change.

## 4. A survey of the Q-learning algorithm performance with an introduced measures model (a model of environment criteria), presented as memory Matrix K

The experiment is conducted in the following way: the goal to reach node 15 in the graph is set in front of the agent (Figure 2); a reward of 100 for going through the edge, connecting nodes 11 and 15 is announced as well in the reward matrix R (Figure 1). The other edges have zero reward points (Figure 1). The black dot line denotes the optimal found path from node 0 to node 15.

**First stage.** The specific measure values in the K matrix are not used. The desired sequence of edges to reaching the goal is defined based on the rewards model. The optimal found path from node 0 to the goal is given in Figure 3. It can be seen that the path goes through edges with a different value of the criterion, set in the K matrix.

**Second stage**. The agent is required to reach the goal by going only through edges with a measure value of 1. The optimal found path from node 0 to the goal is given in Figure 4. As it can be seen, the path goes only through edges with a measure value of 1 for the criterion, set in the K matrix.

**Third stage**. The agent has to reach the goal by going only through edges, having a measure value of 2. The optimal path for this case, starting from node 0 and reaching the goal, is shown in Figure 5, from which it becomes clear that the path goes only through edges with a measure value of 2 for the criterion, set in the K matrix.

In the example under consideration, there are primary and secondary paths that connect all locations in the exemplary Shopping Center. There might be a situation, in which a primary or a secondary path to a given place is missing. Then the algorithm can be modified by allowing the agent to go through a certain number of edges, which do not correspond to the value of the criterion “measure” in the K matrix.

```
//R - reward matrix
public static int [,] R = new int[,] {
{ -1,  0,  0,  0,  0,  0, -1, -1,  -1, -1, -1, -1, -1, -1, -1, -1, -1 },
{  0, -1,  -1, 0,  0, -1, -1, -1,  -1, -1, -1, -1, -1, -1, -1, -1, -1 },
{  0, -1, -1, -1,  0,  0, -1, -1,  -1, -1, -1, -1, -1, -1, -1, -1, -1 },
{  0,  0, -1, -1,  0, -1,  0,  0,  -1, -1, 0, -1, -1, -1, -1, -1, -1 },
{  0,  0,  0,  0, -1,  0, -1,  0,   0, -1, -1, -1, -1, -1, -1, -1, -1 },
{  0, -1,  0, -1,  0, -1, -1, -1,   0,  0, -1, -1, -1, -1,  0, -1, -1 },
{ -1, -1, -1,  0, -1, -1, -1, -1,  -1, -1,  0,  0, -1, -1, -1, -1, -1 },
{ -1, -1, -1,  0,  0, -1, -1, -1,   0, -1, -1,  0, -1, -1, -1, -1, -1 },
{ -1, -1, -1, -1,  0,  0, -1,  0,  -1,  0, -1, -1,  0,  0, -1, -1, -1 },
{ -1, -1, -1, -1, -1,  0, -1, -1,   0, -1, -1, -1, -1,  0,  0, -1, -1 },
{ -1, -1, -1,  0, -1, -1,  0, -1,  -1,  0, -1, -1, -1, -1, 0, -1, -1 },
{ -1, -1, -1, -1, -1, -1, -1,  0,   0, -1, -1,  0, -1,  0, -1, -1,  100, -1 },
{ -1, -1, -1, -1, -1, -1, -1, -1,   0, -1, -1,  0, -1,  0, -1,  0,  0 },
{ -1, -1, -1, -1, -1, -1, -1, -1,  -1,  0,  0, -1, -1,  0, -1,  0, -1,  0 },
{ -1, -1, -1, -1, -1,  0, -1, -1,  -1,  0, -1, -1, -1,  0, -1, -1,  0 },
{ -1, -1, -1, -1, -1, -1, -1, -1,  -1, -1, -1,  0,  0,  0, -1, -1, -1,  0 },
{ -1, -1, -1, -1, -1, -1, -1, -1,  -1, -1, -1, -1, -1,  0,  0,  0,  0, -1 }};
```

```
//K- measure matrix
public static int[,] K = new int[,] {
{ -1,  2,  2,  2,  1,  2, -1, -1,  -1, -1, -1, -1, -1, -1, -1, -1, -1 },
{  2, -1,  -1, 2,  1, -1, -1, -1,  -1, -1, -1, -1, -1, -1, -1, -1, -1 },
{  2, -1, -1, -1,  1,  2, -1, -1,  -1, -1, -1, -1, -1, -1, -1, -1, -1 },
{  2,  2, -1, -1,  1, -1,  2,  2,  -1, -1,  2, -1, -1, -1, -1, -1, -1 },
{  1,  1,  1,  1, -1,  2, -1,  1,   1, -1, -1, -1, -1, -1, -1, -1, -1 },
{  2, -1,  2, -1,  2, -1, -1, -1,   1,  2, -1, -1, -1, -1,  2, -1, -1 },
{ -1, -1, -1,  2, -1, -1, -1, -1,  -1, -1,  2,  1, -1, -1, -1, -1, -1 },
{ -1, -1, -1,  2,  1, -1, -1, -1,   1, -1, -1,  1, -1, -1, -1, -1, -1 },
{ -1, -1, -1, -1,  1,  1, -1,  1,  -1,  2, -1, -1,  1,  1, -1, -1, -1 },
{ -1, -1, -1, -1, -1,  2, -1, -1,   2, -1, -1, -1, -1,  1,  1, -1, -1 },
{ -1, -1, -1,  2, -1, -1,  2, -1,  -1, -1, -1, -1,  1, -1, -1, -1,  2, -1 },
{ -1, -1, -1, -1, -1, -1, -1,  1,   1, -1, -1,  1, -1,  1, -1, -1,  2, -1 },
{ -1, -1, -1, -1, -1, -1, -1, -1,   1, -1, -1, -1, -1,  1, -1,  1,  2 },
{ -1, -1, -1, -1, -1, -1, -1, -1,  -1,  1,  1, -1, -1,  1, -1,  2, -1,  1 },
{ -1, -1, -1, -1, -1,  2, -1, -1,  -1,  1, -1, -1, -1,  2, -1, -1,  2 },
{ -1, -1, -1, -1, -1, -1, -1, -1,  -1, -1, -1,  2,  2,  1, -1, -1, -1,  2 },
{ -1, -1, -1, -1, -1, -1, -1, -1,  -1, -1, -1, -1, -1,  2,  1,  2,  2, -1 }};
```

**Figure 1.** Reward matrix R and Measure matrix K.

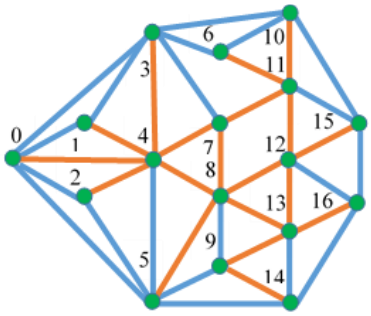


**Figure 2.** A Shopping Centre with 17 shops and 36 lobbies or hallways between them, presented by an undirected graph.

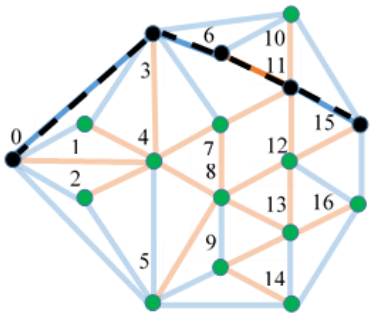


**Figure 3.** Optimal path from node 0 to node 15. Requirement for measure value not set.

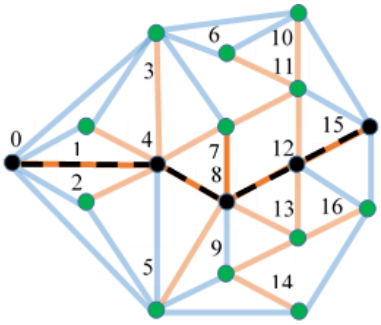


**Figure 4.** Optimal path from node 0 to node 15. Requirement for measure value: equal to 1.

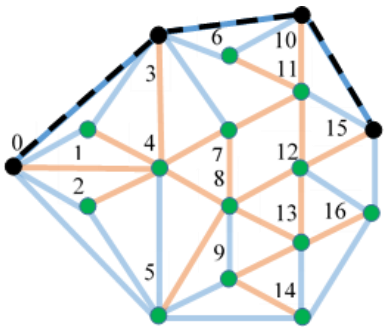


**Figure 5.** Optimal path from node 0 to node 15. Requirement for measure value: equal to 2.

**Forth stage.**
For the purposes of the next stages of the experiment the measure values modification of the graph edges is made (Figure 6). There is no path from node 0 to node 15, going only through edges with a measure value of 1 for the criterion, set in the K matrix.

**Fifth stage.** The agent has to reach the goal by going only through edges, having a measure value of 1. Two tradeoffs at the maximum are allowed, i.e., to achieve the goal the agent can choose two edges at the maximum, having a measure value of 2 as set in the K matrix. Fig.7 shows the found path. It can be seen that the agent has made exactly two tradeoffs on his way to the goal.

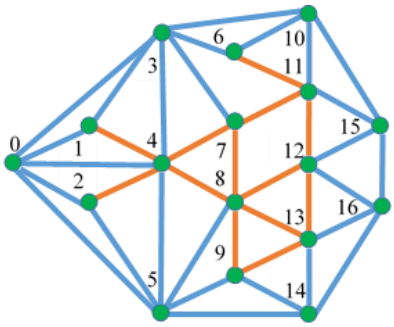


**Figure 6.** A Shopping Centre with 17 shops and 36 lobbies or hallways between them, presented by an undirected graph. Measure value modification of the graph edges is made.

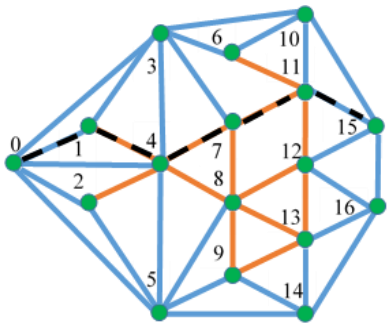


**Figure 7.** An optimal main path from the start node 0 to the end node 15 is sought, with a maximum of two compromises allowed. It is seen that the agent has chosen an edge with measure value 2 set in the K matrix twice.

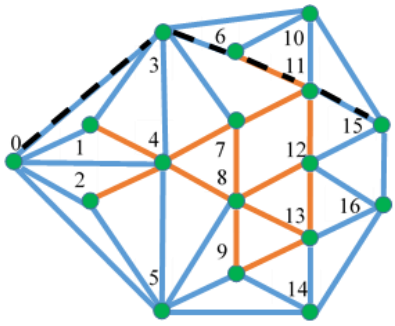


**Figure 8.** An optimal main path from the start node 0 to the end node 15 is sought, with a maximum of four compromises allowed. It is seen that the agent has chosen an edge with measure value 2 set in the K matrix twice. The agent finds the optimal path to the goal by making only three compromise solutions.

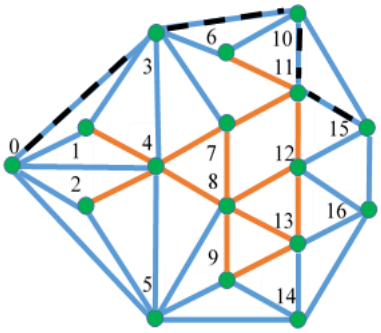


**Figure 9.** An optimal secondary path from the start node 0 to the end node 15 is sought. No compromises allowed (not allowed to go along main paths to reach the goal). The found path is shown.

**Sixth stage.** The agent has to reach the goal by going only through edges, having a measure value of 1. Four tradeoffs are allowed at the maximum, i.e., the agent can choose only four edges with a measure value of 2 set in the K matrix to reach the goal. Fig. 8 shows that the agent chooses the same optimal path to the goal as in Fig. 3. Three compromise solutions have only been made to reach the goal, meaning that the agent has made no unnecessary compromise solutions.
**Seventh stage.** The agent has to reach the goal by going only through edges, having a measure value of 2. No compromises allowed. The optimal secondary path, found by the agent, is given in Figure 9.

## 5. Discussion and future work

Many problems remain to be solved. In the first place, the work on the development of the Reinforcement learning algorithm will continue.

There are several options when setting requirements for reaching a goal. If possible, the agent will find the optimal path to the goal by meeting all the criteria simultaneously. However, there may not be a path, meeting all the criteria. In addition, some compromises may be more acceptable than others.

One of the solutions is to require from the agent to reach the goal by making as few compromises as possible. Another solution is to allow the agent to choose the compromises to make. If they are graded, the agent can be required to choose to make more in number and more acceptable compromise solutions rather than make fewer but unacceptable ones. These options for modification of the training algorithm under consideration will be useful in more complex social scenarios implementation.

An advantage of the proposed modification of the Q-learning algorithm is that it allows the agent to give explanation of the reasons behind the choice of a given path to a goal. In addition, the proposed modification allows to introduce various criteria for choosing a particular path. If the criteria from Maslow's theory of personality motivation are used, a model of a system of values could be developed using different scenarios.

Opportunities for modeling the training agent's value system will be looked for; efforts will be put to modeling a system for generating explanations by the learning agent. Using a holographic computer, it is possible to model and visualize a virtual advertising agent. It is assumed that the communication with such an agent will be engaging and helpful to customers. As mentioned in [11], there is a lot of interest in modeling a robotic shopping cart to follow the consumer. Efforts will therefore be made to this end. For example, it is important to combine and share intelligent behaviors such as: wander behavior; path following; collision avoidance; obstacle and wall avoidance; patrol between a set of points; flee behavior.

## 6. Conclusion

The paper describes a modification of the Q-learning algorithm. In order to make the-Q learning agent find the optimal path to the goal by following specific complex criteria, the use of measures model (a model of environment criteria), represented as a new memory matrix, is introduced. If the goal cannot be reached by meeting the set criteria, the agent could just ignore a given criterion and make a compromise solution. In this case, the agent is required to make as few compromises as possible in order to reach the set goal. Experiments have been conducted, illustrating the performance of the modified algorithm. If having the criteria graded by importance, the agent can make more in number and more acceptable compromise choices. This would be useful when developing complex social scenarios. The aim of the modification is to empower the learning agents to: control the way of reaching a goal; better understand the customers; be able to justify their decisions. The modified algorithm has been applied to training smart shopping-cart learning agents. The studies show improvement in their behavior.

## Acknowledgments

The authors gratefully acknowledge the financial support provided within the Technical University of Sofia, Research and Development Sector, Project for PhD student helping N202PD0007-19 "Intelligent Cognitive Agent behavior modeling and researching".

**References**

[1] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, The MIT Press, Cambridge, London, England, 2014. [Online]. Available from: http://incompleteideas.net/book/ebook/the-book.html, [retrieved: 12, 2019].

[2] A. Gosavi, "Reinforcement Learning: A Tutorial Survey and Recent Advances," INFORMS Journal on Computing, Vol. 21 No.2, pp. 178-192, 2008.

[3] R. R. Torrado, P. Bontrager, J. Togelius, J. Liu, D. Perez-Liebana, "Deep Reinforcement Learning for General Video Game AI," IEEE Conference on Computatonal Intelligence and Games, CIG. 2018-August, 10.1109/CIG.2018.8490422

[4] B. Argall, "Learning Mobile Robot Motion Control from Demonstration and Corrective Feedback", Robotics Institute Carnegie Mellon University Pittsburgh, PA 15213, March 2009.

[5] H. B. Amor, D. Vogt, M. Ewerton, E. Berger, B. Jung, J. Peters, "Learning Responsive Robot Behavior by Imitation," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013) IEEE, Tokyo, Japan, November 3-7, 2013, pp. 3257-3264.

[6] K. Takahashi, K. Kim, T. Ogata, S. Sugano, "Tool-body assimilation model considering grasping motion through deep learning," Robotics and Autonomous Systems, Elsevier, Volume 91, pp. 115–127, 2017.

[7] K. V. Moffaert, Multi-Criteria Reinforcement Learning for Sequential Decision Making Problems, Dissertation for the degree of Doctor of Science: Computer Science, Brussels University Press, ISBN 978 90 5718 094 1, 2016.

[8] S. Natarajan, P. Tadepalli, Dinamic Preferences in Multi-Criteria Reinforcement Learning, 22nd International Conference on Macine Learning, Bonn, Germany, 2005.

[9] Dilyana Budakova, Lyudmil Dakovski, Smart shopping system, TECHSYS 2019, Plovdiv, 16-18 May 2019, doi:10.1088/issn.1757- 899X; Online ISSN: 1757-899X; Print ISSN: 1757-8981, Scopus.

[10] Dilyana Budakova, L.Dakovski, Veselka Petrova-Dimitrova, Smart Shopping Cart Learning Agents Development, TECIS 2019, 19th IFAC-PapersOnLine, Conference on International Stability, Technology and Culture,Volume 52, Issue 25, 26-28 September, 2019, pp. 64-69, Sozopol, Bulgaria, Web of Science, Elsevier ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2019.12.447

[11] Dilyana Budakova, Lyudmil Dakovski, Veselka Petrova-Dimitrova, Smart Shopping Cart Learning Agents, International journal on Advances in internet technology, IARIA, pages: 109 – 121, issn: 1942-2652, Vol. 12, nr 3&4. 2019

[12] N. G. Shakev, S. A. Ahmed, A. V. Topalov, V. L. Popov, and K. B. Shiev, "Autonomous Flight Control and Precise Gestural Positioning of a Small Quadrotor," Learning Systems: From Theory to Practice, Springer, pp. 179-197, 2018.