Multidiszciplináris tudományok, 10. kötet. (2020) 4 sz. pp. 194-205 **https://doi.org/10.35925/j.multi.2020.4.23**

SURVEY ON ARTIFICIAL INTELLIGENCE ALGORITHMS USED IN INDUSTRIAL ROBOTICS

Rabab Benotsmane
PhD student, University of Miskolc, Institute of Information Science
3515 Miskolc, Miskolc-Egyetemváros, e-mail: iitrabab@uni-miskolc.hu

László Dudás
associate professor, University of Miskolc, Institute of Information Science
3515 Miskolc, Miskolc-Egyetemváros, e-mail: iitdl@uni-miskolc.hu

György Kovács
associate professor, University of Miskolc, Institute of Logistics
3515 Miskolc, Miskolc-Egyetemváros, e-mail: altkovac@uni-miskolc.hu

Abstract

Recently, the elements of Industry 4.0 as the Artificial Intelligence (AI) and the Machine Learning (ML) are implemented at the manufacturing companies in order to increase their competitiveness. These elements have important role in robotics technology where the main aim is the efficiency improvement, which requires the improvement on the rigid, inflexible capabilities of industrial robots. The most important and commonly used AI algorithms applied in industrial robotics are presented in this article. At first the overview of the Machine Learning algorithms used by industrial robots will be discussed. In the second part of the study the most important AI algorithms used to optimize and improve the trajectory of robotic arms will be introduced.

Keywords: *Industry 4.0, AI, machine learning, industrial robots, robotic arm.*

1. Introduction

Artificial Intelligence (AI) is known in the science field as one of the most significant tools for solving problems that can be caused in machine, where it suggests different solutions presented in a logical structure using mathematical techniques. AI is developing fast and applied to different areas like modeling, controlling, and optimizing in different engineering sectors [1]. The appearance of AI in the science field pushed many researchers to suggest different views and developed new techniques proposed in literature reviews. It is a branch of science aimed to deal with machine and develop its capacity of performing functions to make it smarter in perception, learning and planning, where a computer can think like a human operator [2-3]. Nowadays, thanks to AI many techniques used in multiple segments of the modern industries are developed where the important role is integrating people, information, and products through the networked enterprise area, as well as collecting data of manufacturing, logistics, warehousing, and distribution of goods [4]. Nowadays, the industrial sector is exposing to the fourth industrial revolution (Industry 4.0) where AI and Machine Learning (ML) capabilities are taking a large part in the development of Industry 4.0 that requires smart agents and robotic systems endowed with AI and ML technologies. The significant role of using intelligent machines reside in the purpose of creating a smart and flexible industry that reduce the lead time achieving higher productivity with improving the product quality [5-6]. AI field includes different subsets as ML and the newest Deep Learning techniques where algorithms learn from data without the need of programming. Traditional robots in the industry usually have the ability to pick and place objects in a predetermined trajectory as long as the process scenario is set in the robot controller, on the other side and with the use of AI tools, modern robots fitted with smart sensors are programmable via AI to detect a specific object location in the working space [7]. Robots using ML can teach themselves within a short period how to react and execute a task that has not executed before [8]. The algorithms employed in machine learning are improving better repeating the task executed by the robots several times. It is also well known that the main effective target of AI in industrial robotic is the training, where the conventional process would require training the robotic automation system using many rules to distinguish parts to pick up, this needs running many iterations and experiencing many times trial and error method. It is clear that the conventional process is human time and effort-consuming, therefore using AI tools making robots easier to train, especially where industrial robots are newly installed in the workshop. Where it is harder to find engineers and skilled human operators to train the new robots [9], AI tools helps to simplify the training process. The emerging of AI in industrial robotics has real benefits to create a smart factory where a human can focus on what needs to be done, not how the robot needs to do it [10]. Industrial robotics introduced AI applications on automatic systems installed in the production line, where the building of a robust manufacturing technology is always based on the deployment of industrial robots, that can achieve the speed, accuracy and safety of production [11]. The emerging of AI makes this robust manufacturing industry smarter. The most common industrial robots used today are welding robots, working in automotive and metal production, due to the precision and uniformity of results as well as worker safety by reducing exposure to fires and risks. Assembly robots used on automated assembly lines including automotive chains, electronics equipment production using computer vision and sensors to identify, grasp and manipulate parts and to assemble such parts using feedback to evaluate the level of force applied and if the parts are well assembled. The accurate manipulation speeded up production times dramatically. Picking and packing robots are involved in taking the products off a production line and packaging them which require flexibility and a range of motions, as well as we can cite other manufacturing task robots including cutting, drilling, milling and painting robots. The major features of modern robots guarantee the safety in working area in close distance to operators and the ability to reconfigure themselves in malfunction cases which need intelligent adaptation [11]. Figure 1 describes the subsets that regroup AI science.

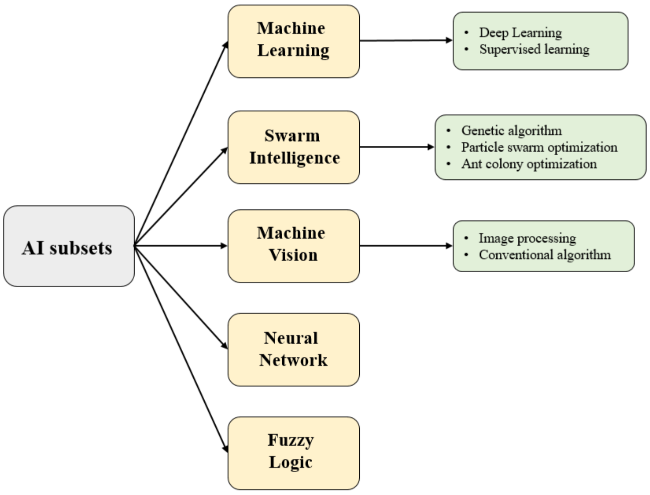

Figure 1. *Subset fields of AI science [own compilation]*

In this area, we highlight AI-powered computer vision systems that present one of the most important avenues of research in robotics, where AI technology can give robots the power to see and to identify and interact with what it is they are seeing [12].

This article presents a contribution to the emerging of AI and ML techniques in industrial robots to create a flexible process. The first part introduces an overview of the AI used in robotics and the benefit of ML on this area, the second part shows to the reader the most often used AI algorithms applied by industrial robots to achieve a predefined task and optimize their control function.

2. The emerging of AI and ML in industrial robotics applications

Since 2016 many leading robot manufacturers as Fanuc and KUKA have success to apply AI and Internet of Things (IoT) technology in different automation systems [13]. Application of AI and ML in collaborating robots has an essential target resides in the capability of many robots not only working together but also working in safely alongside human operators, where they can easily reprogram themselves for new tasks unlike traditional robots, that depend always on the human decision, where increasing the lead time and achieving the higher productivity are the greatest benefits that can be guaranteed with the use of AI by today's manufacturers. AI science is based on theories of mathematics and optimization techniques used to solve complex problems in different sectors. ML is a subset of AI which allows machine to automatically learn from historic data without programming, where the main purpose of ML is to allow robots to learn from data to accommodate to the needs of environment. Otherwise the goal of AI is to make a smart computer system enabling stimulation as humans to solve complex problems [14]. The emergence of AI and ML sciences in robotics field has create a new concept known as robotic learning [15] based on the concept of cooperation between machines and industrial robots, by making decision as operators and learn from previous experiences.

2.1. Modern Machine Learning applications in industrial robotics

Robotic learning highlights five key areas where machine learning has had a significant impact on industrial robotic technologies. The five keys are presented in the next sub-sections [16].

2.1.1. Computer vision

Computer vision is a technology that aims to create machines that can detect and classify objects regarding the shape, the color, and their movement by receiving information from the series of images usually using a camera [17]. The domain of computer vision can be characterized as developing, diverse and dynamic area where planning or decision-making in systems which can perform mechanical actions, such as moving a robot through certain environment, is usual. This type of processing usually needs the input data given by the computer vision system, which provides high-level information about the robot's environment, related sensors and devices or video devices. The computer vision expands the number of solved problems in robotics.

2.1.2. Imitation learning

Imitation learning is closely related to observational learning, a behavior exhibited by infants and toddlers [16]. Imitation learning is a category for reinforcement learning which will be introduced in the following. The goal is to make a robot to act in the environment by maximizing its rewards. Bayesian or probabilistic models are a common feature of this approach. Imitation learning has become an integral part of robotics field to solve many problems including inverse optimal control methods, or "programming by demonstration".

2.1.3. Self – supervised learning

Self-supervised learning approaches enable robots to generate their own training samples in order to improve performance or testing results, using a tested training and data captured close range to interpret long-range ambiguous sensor data where it is integrated into robots and optical devices to detect objects [18].

2.1.4. Assistive technologies

Assistive technology presents a process to sense the information and performs actions, as driver assistance tools and movement therapy robots that provide a diagnostic or therapeutic benefit.

2.1.5. Multi-agent learning

Coordination and negotiation are the essential keys of multi-agent learning, that combine machine learning-based agents which can be presented as robots, that are able to adapt to a new environment of other agents and find equilibrium strategies for this shift, multi-agent learning approaches include no-regret learning tools, which involve weighted algorithms that boost learning outcomes in multi-agent planning, and learning in distributed control systems [16-19].

3. AI algorithms specified for industrial robots

In the following of this article we would like to show to the reader the main algorithms and models of AI that can be used in the research to developing the functioning of an industrial robot in controlling task and optimization of trajectories.

3.1. Reinforcement Learning algorithms

Reinforcement Learning (RL) is a branch of ML where an agent learns to interact in its workspace by performing actions and seeing the results [20]. The concept of the algorithm has been used in different areas, where it was not very adaptable and was incapable to execute continuous tasks. Nowadays, RL is used e.g. for calculating inverse kinematic problem of a manipulator arm which is a complicated task based on calculating the angles needed for each joint to reach the end effector, the process inputs are a set of equations that present the homogeneous transformation matrixes between the links, joints and the position and orientation vectors.

The algorithm looks at a robot arm trying to maximize its cumulative reward given the state it is in by taking a sequence of actions as presented in Figure 2 [21].

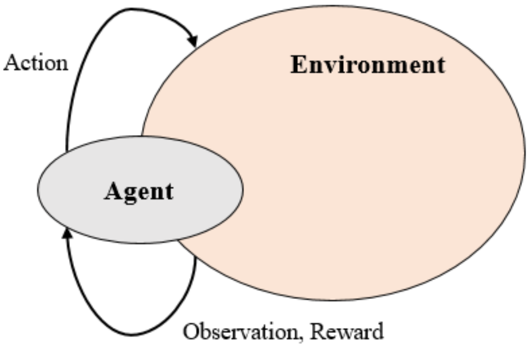

Figure 2. *Reinforcement Learning algorithm methodology [own compilation]*

The robot arm interacts with its environment by performing an action in turn, returns a reward and the new state the robot finds itself in, where:

- The **environment** consists of the arm links and joints in space.
- The **reward** is distance between the finger and the goal position.
- The **actions** are based on rotational motion executed by each joint.

3.2. Search algorithms

Search techniques play an important role in AI science where they are universal problem-solving methods for robots to solve a specific problem and provide the best result [22]. Search algorithms based on the following components:

- **Search Space**: Represents a set of possible solutions, which a robot may execute.
- **Start State**: It is a state where a robot begins the actions.
- **Goal test**: It is a function which observes the current state and returns whether the goal state is achieved or not.

Search problem is represented as a tree; the root of the search tree is the root node which is corresponding to the initial state or the starting step. Figure 3 presents the search tree, where it is composed of the following:

- **Actions:** It gives the description of all the available actions of the robot.
- **Transition model:** A description of the dynamics of each action.
- **Path Cost:** It is a function that deals with a numeric cost of each path.
- **Solution:** It is an action sequence that leads from the start node to the goal node.
- **Optimal Solution:** A solution having the lowest cost of all solutions.

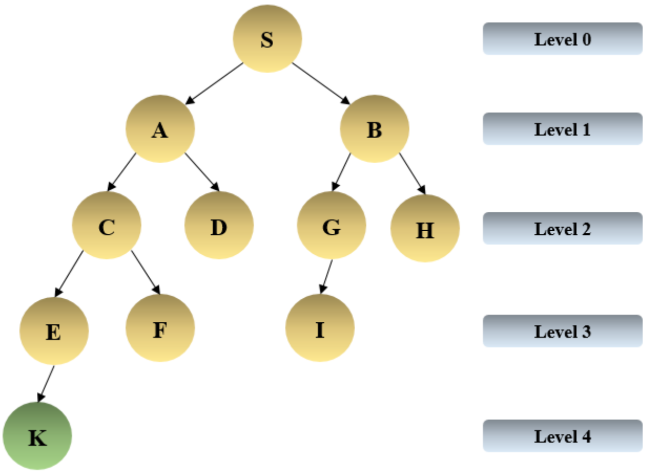

Figure 3. *Search tree scheme [own compilation]*

Search algorithms are divided in two main categories: 1) - Blind search algorithms where the search algorithm starts to examine each node of the tree without any information about the search space until achieve the goal node. 2) - Informed search algorithms where a problem information is known that helps the search to find a solution more efficiently than a blind search strategy, it is called also a Heuristic search. A heuristic is a strategy presented with a cost function that do not guarantee best solution but guarantees to find a good solution in reasonable time. It is used as an additive technique to different algorithms like genetic algorithm to find the optimal solution.

The efficiency of these algorithms approved by comparing the four essential properties:

- **Completeness:** A search algorithm is complete if at least one solution exists for any random input.
- **Optimality:** Optimal solution where the solution found is guaranteed to be the best solution (lowest path cost).
- **Time Complexity:** Time complexity is a measure of algorithmic steps for an algorithm to complete its task.
- **Space Complexity:** It is the maximum storage space required at any point during the search, as the complexity of the problem.

In industrial robotics the application of search algorithms can be a good strategy to find the optimal solution by scheduling the steps of any process. From literature review, we can cite different search algorithms that deal with the control and the optimization of a robotic arm.

3.2.1. Tabu Search algorithm

Tabu Search (TS) algorithm was first proposed by Fred Glover in 1986 [23]. This approach is a combination of optimization and search methods based on converging to the best available neighbourhood solution point [24]. In other words, the principle idea is to move to the best point of local search space in term of cost function, even though it is worse than the current best solution point. TS method is a local search approach that includes two memories, short and long-term memory. The short-term memory prevents the reversal of the recent moves and steps of a robotic arm, otherwise the long-term memory reinforces attractive components, forcing the algorithm to move towards more preferable solutions, the tabu list helps the algorithm to move out from local optimum and to reach the global one.

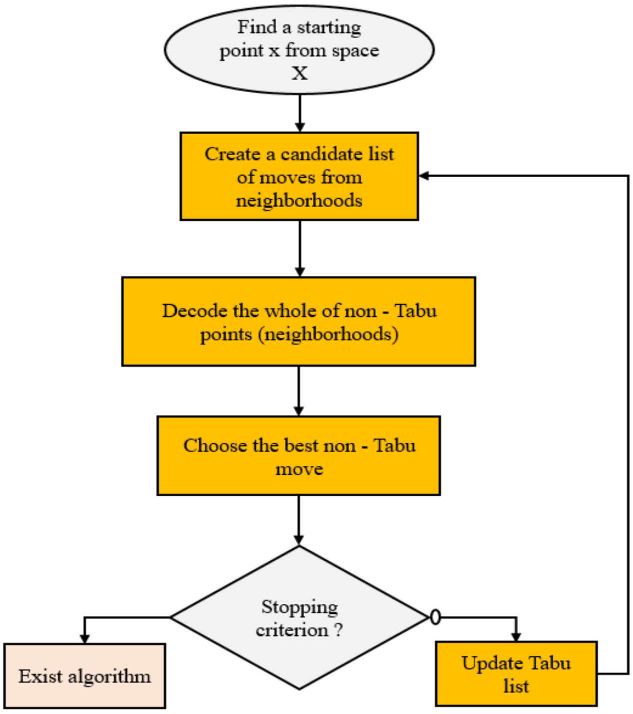
Find a starting point x from space X
Create a candidate list of moves from neighborhoods
Decode the whole of non - Tabu points (neighborhoods)
Choose the best non - Tabu move
Stopping criterion ?
Exist algorithm
0
Update Tabu list

Figure 4. *Tabu Search flow chart [[24], own compilation]*

The flowchart in Figure 4 describes TS algorithm in five steps. Assume that *X* is a total search space and *x* is a solution point sample, and *f(x)* is cost function. First, we choose *x* from space *X* to start the process, then a candidate list of non-Tabu moves in neighborhood is created *N*. All solution points are decoded and then we find the best local *xwinner* from *N*. In the last step, the algorithm exits if the stopping criterion is satisfied. If not, *x* = *xwinner*, the Tabu List will be updated, and it returns to the first step of algorithm.

Usually it is used for task scheduling application for pick and place or disassembling processes, as well as to execute the optimal trajectory for the manipulator arm.

3.2.2. Decision Tree Search algorithm

Decision Tree Search Algorithm (DTS) is a flowchart structure in which each internal node represents a "test" on an attribute, each branch represents the output of the test, a class label which describes the decision taken after testing is known as a leaf node, where classification rules are presented as the paths from the root to leaf [25-26]. Mostly used in operations research, specifically in decision analysis, to help identify a strategy, most likely to reach a goal and to execute a trajectory or schedule a steps to pick and place, also in the cooperative robotic systems, where DTS can be used to determine which robot is suitable for a specified action. Figure 5 illustrates the global scheme of DTS including rules and actions of the process where each test has a gain and using this gain the cost function can be calculated to determine which path from the tree is optimal.

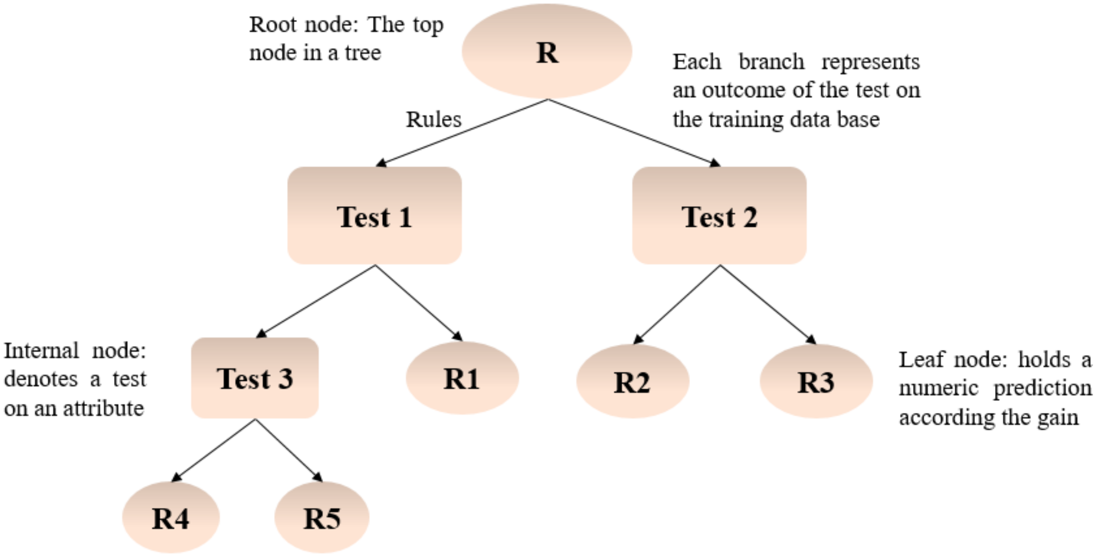
Root node: The top node in a tree
R
Each branch represents an outcome of the test on the training data base
Rules
Test 1
Test 2
Internal node: denotes a test on an attribute
Test 3
R1
R2
R3
Leaf node: holds a numeric prediction according the gain
R4
R5

Figure 5. *Decision Tree Search scheme [own compilation]*

3.3. Fuzzy Logic control systems

The approach of fuzzy logic (FL) works on the levels of human decision making which based on a range of possibilities between YES and NO [27], where the conventional logic block that a computer can understand takes precise input and produces a definite output as TRUE or FALSE. The idea was proposed by Lotfi Zadeh [28] based on the knowledge of any systems using a series of IF/THEN rules. Fuzzy logic system is one of the most important results of research in the area of artificial intelligence, otherwise the problem arises when systems have many inputs and outputs. It has been applied in voice and image recognition and automobile transmissions, where the researchers justify the use of fuzzy logic for two reasons - ease of design and the ability of a fuzzy system to work with inaccurate sensors. Its effect in industrial robotics can be appear in the training of a robotic arm to plan its movements to avoid a collision with obstacles in its workspace, or in collaborative robot system where executing a parallel task, the use of fuzzy logic can facilitate which robot has more ability to achieve better the task.

FL is a method of distinguishing information as fuzzy sets and rules where a fuzzy system has one or more inputs that are fuzzified, a rule set which is evaluated according to the inputs, and one or more outputs that are defuzzyfied into real-world values. Fuzzy system structure includes the following:

- **Fuzzification**: is to transform real-world numbers into fuzzy space, by applying fuzzy propositions and identify linguistic variables and their membership functions.
- **Fuzzy processing**: is to generate IF-THEN fuzzy rules and implement them in order to generate a solution space for the given problem. Generally, membership functions are laid on top of each other, and solution space is defined according to AND / OR operators of the rules as in traditional logic.
- **De-fuzzification**: is the process of finding an output real-world value for the solution, by transforming the fuzzy set obtained by the fuzzy processing into a crisp value.

3.4. Swarm optimization algorithms

Swarm Intelligence (SI) has taken a large interest of researchers for many years as seen in the literature reviews. SI presents the collective intelligence that regroups many agents, it is the collective intelligence behavior of self-organized and decentralized systems taking the idea from the behavior of insects in nature [29]. SI is based on two fundamental concepts 1) self-organization and 2) division of labor. Self-organization is defined as the capability of a system to evolve its agents or components into a suitable form without any external help. Division step is defined as the simultaneous execution of various simple and feasible tasks by individuals. This division allows the swarm to deal with complex problems requiring individuals for collective work. With the development of technologies many algorithms based on SI has been discovered, in the following part we focus on describing Swarm optimization algorithms that most used in industrial robotics.

3.4.1. Genetic Algorithm

Genetic Algorithm (GA) is a search heuristic inspired by Charles Darwin's theory of natural evolution [30]. The algorithm applies the natural selection process where the fittest individuals will be selected for reproduction and produce the next generation, the algorithm can be useful for solving the point-to-point trajectory planning for redundant robot arm, and to deal with robotic arm problems. Usually the algorithm uses different fitness criteria to reach an optimal result as inverse kinematic problem, the minimization of energy consumption, minimization of operation time as well as the criterion of minimal total angular changes of the robotic arm.

Figure 6 presents a flow chart of the main structure of GA, where five phases are considered [31]. 1) Creating initial population: The process starts with a set of individuals called population or strings, the set is generated randomly at the first time then the evolution of the population elements is non-generational, meaning that the new replaces the worst elements. Each individual is a solution to the problem, in inverse kinematic problem the set of the genes of individuals are joints angles for the robotic arm. 2) Fitness function determines the fitting capacity of an individual with others providing a fitness score for this individual, the fitness score define the probability to select the same individual for the next generation, in our case it is described as an equation aims to calculate such gain as maximum torques for set of joints angles or an objective function of collision avoidance, the total joint traveling distance of the manipulator criteria. A new population has been got after a defined tournament step that accomplishes a reproduction process including: 3) selection where the successive generations of new strings are selected based on their fitness values; 4) crossover, with a given crossover probability. The crossover operator works with e.g. the single point technique, where the crossover point is only allowed between genes. 5) The mutation operator replaces one gene value with another one generated randomly with a specified range by a given mutation probability.

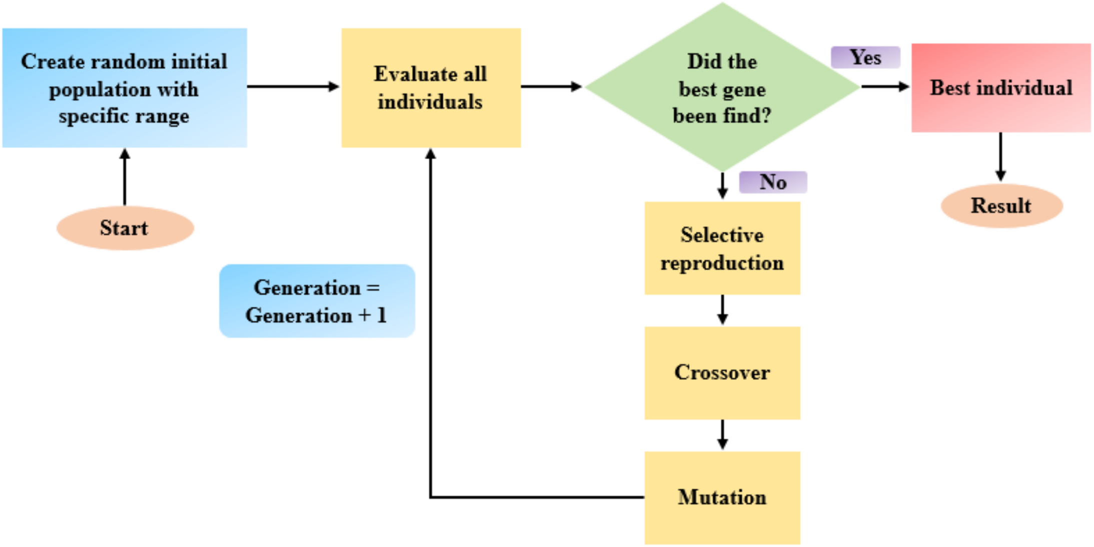
Create random initial population with specific range
Evaluate all individuals
Did the best gene been find?
Yes
Best individual
Result
No
Start
Selective reproduction
Generation = Generation + 1
Crossover
Mutation

Figure 6. *Genetic Algorithm flow chart structure [[31], own compilation]*

3.4.2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a stochastic optimization technique based on the intelligence and movement of birds swarms [29]. Inspired by the behavior of nature social and dynamic motions with communications of fish schooling, insects, or birds flocking.

The algorithm is based on the following scenario: a group of birds is randomly searching for food in a defined space [32]. There is only one piece of food in the whole space. All the birds do not know the place of food, but in each iteration, they can know how far the food is. Therefore, the best strategy to find the food is to follow the bird which is nearest to the food. In PSO the optimization problem is solving as follows:

- Every single solution is a "bird" in the search space, called "particle".
- All the particles have fitness values that are evaluated by the fitness function to be optimized and have velocities that direct the flying of the particles.
- The particles fly around the problem space by checking the current optimum particle. PSO technique principle based on generating a defined number of samples in random positions in a defined area, where the velocity of those samples is nominated randomly, each particle has a memory that stores all the best position have been checked before, in addition to the fitness in that position which has been improved over time [33].

Considering a swarm with P particles, there is a position vector X and a velocity vector V at a t iteration for each one of the i particle that composes it. These vectors are updated through the dimension *j* according to lead the particle *i* toward either the personal best vector *Pbes*t or the swarm's best vector *Gbest* using the following equations:

$$X_{i,j}^{t+1} = X_{i,j}^{t} + V_{i,j}^{t+1} \tag{1}$$

$$V_{i,j}^{t+1} = \omega \times V_{i,j}^{t+1} + c_1 \times r_1^t (Pbest_{i,j} - X_{i,j}^{t}) + c_2 \times r_2^t (Gbest_{i,j} - X_{i,j}^{t}) \tag{2}$$

Where: $i = 1,\dots, P$ and $j = 1,\dots, n$

Where Eq (1) calculates the position of each particle which is updated using the new velocity for that particle, the new velocity is calculated according to Eq (2), where c_1 and c_2 are the cognitive coefficients ($c_1 + c_2 \leq 4$), and r_1 and r_2, ω are random real numbers between [0,1]. The inertia weight ω controls the particle momentum [34].

The value of *V* is clamped to the range [-*Vmax*, *Vmax*] to reduce the probability of leaving the search space by the particle. If the search space is defined by the bounds [-*Xmax, Xmax*] then the value of *Vmax* is typically set so that $Vmax = k \times Xmax$, where $0.1 \leq k \leq 1.0$.

A large inertia weight (ω) eases the local search when its value is large and eases the global search when its value is small. Summarizing PSO technique in steps it consists of just three steps [35]:

- Calculate fitness for all particles.
- Update local and global bests.
- Update velocity and position for all particles.

4. Summary

Recently, many scientists deal with science of Artificial Intelligence as an important topic due to the potential benefits, which can be achieved using the tools of it. Artificial Intelligence offers a new paradigm to transform the industrial environment into a smart platform, based on collecting data and decision making of machines and automatic systems. In the first part of the article an overview of the Artificial Intelligence algorithms used in industrial robotics and the benefits of Machine Learning were presented. In the second part of the study the most often used Artificial Intelligence algorithms to optimize and improve the trajectory of robotic arms were discussed.

Acknowledgement

The described article was carried out as part of the EFOP-3.6.1-16-2016-00011 "Younger and Renewing University – Innovative Knowledge City – institutional development of the University of Miskolc aiming at intelligent specialization" project implemented in the framework of the Szechenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

References

[1] Benyoucef, L., Grabot, B.: *Artificial Intelligence techniques for networked manufacturing Enterprises Management*, Springer-Verlag, London, 2010. ISBN 978-1-84996-119-6. **https://doi.org/10.1007/978-1-84996-119-6**

[2] Perez, J.A., Deligianni, F., Ravi, D., Yang, G.Z.: *Artificial Intelligence and robotics*, UKRAS. Org, 2018. ISSN 2398-4422. **https://doi.org/10.31256/WP2017.1**

[3] Tzafestas, S.G, Verbruggen, H.B.: *Artificial Intelligence in industrial decision making, control, and automation: An introduction*, Springer nature, 1995. ISBN 9789401041348. **https://doi.org/10.1007/978-94-011-0305-3**

[4] Thomassey, S., Zeng, X.: *Artificial Intelligence for fashion industry in the Big Data Era*, Springer Singapore, 2018. ISBN 978-981-13-0080-6. **https://doi.org/10.1007/978-981-13-0080-6_1**

[5] Benotsmane, R., Dudás, L., Kovács, G.: *Collaborating robots in Industry 4.0 conception*, 2018 IOP Conference Series: Materials Science and Engineering, Kecskemét, Hungary, pp. 12-23. ISBN 978-963-358-162-9. **https://doi.org/10.1088/1757-899X/448/1/012023**

[6] Benotsmane, R., Kovács, G., Dudás, L.: *Economic, social impacts and operation of smart factories in Industry 4.0 focusing on simulation and artificial intelligence of collaborating robots*, Social Sciences, Vol. 143: No.8 (2019) pp. 1-20. **https://doi.org/10.3390/socsci8050143**

[7] JavaTpoint, https://www.javatpoint.com/difference-between-artificial-intelligence-and-machine-learning, Last accessed 10.08.2020.

[8] Wisskirchen, G., Thibault, B., Bormann, B. U., *Artificial Intelligence and robotics and their impact on the workplace*, IBA Global Employment Institute, London, UK, 2017.

[9] Robotic industries association, https://www.robotics.org/blog-article.cfm/Training-an-Industrial-Robot-Using-AI/, Last accessed 08.08.2020.

[10] Robotics and Automation, https://roboticsandautomationnews.com/2020/03/10/how-ai-affects-the-robotic-industry-and-what-the-future-holds/, Last accessed 10.08.2020.

[11] Khan, Z. H., Azfar, K., Jamshed, I.: *Towards realizing robotic potential in future intelligent food manufacturing systems*, Innovative Food Science & Emerging Technologies, 48 (2018) pp. 11-24. **https://doi.org/10.1016/j.ifset.2018.05.011**

[12] Riabchenko, V., Nikitin, A., Khomenko, V., Melnyk, A.: *Application of the computer vision technology to control of robot manipulators*, Theoretical and Applied Aspects of Cybernetics, Ukraine, 2012. ISBN 978-966-399-310-2.

[13] Wang, W., Siau, Keng., *Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: A review and research agenda*, Journal of Database Management. Vol. 30 (2016) pp 61-79. **https://doi.org/10.4018/JDM.2019010104**

[14] Awad M., Khanna R., *Machine learning. In: Efficient learning machines*. Apress, Berkeley, CA, 2015. ISBN 978-1-4302-5990-9. **https://doi.org/10.1007/978-1-4302-5990-9_1**

[15] Peters J., Lee D.D., Kober J., Nguyen-Tuong D., Bagnell J.A., Schaal S.: *Robot learning*. In: Siciliano B., Khatib O. (eds) Springer Handbook of Robotics. Springer Handbooks. Springer, Cham, (2008). ISBN 978-3-319-32550-7.

[16] Emerj, https://emerj.com/ai-sector-overviews/machine-learning-in-robotics/, Last accessed: 10.08.2020.

[17] Ruiz-del-Solar, J., Loncomilla, P, Soto, N.: *A survey on Deep learning methods for Robot vision*, 2018 Advanced Mining Technology Center & Department of Electrical Engineering Universidad de Chile, Chile, pp. 1-43.

[18] Liu, Q., Wu, Y.: *Supervised learning*, In: Seel N.M. (eds) Encyclopedia of the Sciences of Learning. Springer, Boston, MA, 2012. ISBN 978-1-4419-1427-9.

[19] Wang, S., Wan, J., Zhang, D., Li, D., Zhang, C.: *Towards smart factory for Industry 4.0: a self-organized multi-agent system with big data-based feedback and coordination*, Computer Networks, 101 (2016) pp. 158-168. **https://doi.org/10.1016/j.comnet.2015.12.017**

[20] Vincent, F., Peter, H., Riashat, I., Bellemare, M., Pineau, J., *An introduction to deep reinforcement learning*, Foundations and Trends in Machine Learning: Vol. 11: No. 3-4 (2018) pp 219-354. **https://doi.org/10.1561/2200000071**

[21] Medium, https://medium.com/datadriveninvestor/training-a-robotic-arm-to-do-human-like-tasks-using-rl-8d3106c87aaf, Last accessed: 06.08.2020.

[22] Salander, E. C.: *Computer search algorithms*, Nova Science Publishers, UK. 2011. ISBN 978-1611225273.

[23] Glover, F.: *Tabu search-part I*, ORSA Journal on Computing, Vol. 1 (1989) pp. 190-206. **https://doi.org/10.1287/ijoc.1.3.190**

[24] Alshibli, M., El Sayed, A., Kongar, E., *Disassembly Sequencing Using Tabu Search*, Journal Intell Robot System 82 (2016), pp.69-79. **https://doi.org/10.1007/s10846-015-0289-9**

[25] Hofer, L.: *Decision-making algorithms for autonomous robots*, Robotics [cs.RO]. Université de Bordeaux, 2017. NNT: 2017BORD0770f.

[26] Torres, F., Puente, S., Díaz, C., *Automatic cooperative disassembly robotic system: Task planner to distribute tasks among robots*, Control Engineering Practice, Vol. 17: No.1 (2009), pp.112-121. **https://doi.org/10.1016/j.conengprac.2008.05.013**

[27] Rashidifar, M., Rashidifar, A., Ahmadi, H., *Modeling and Control of 5DOF robot arm using fuzzy logic supervisory control*, IAES International Journal of Robotics and Automation, 1 (2013) pp.56-68. **https://doi.org/10.11591/ijra.v2i2.2974**

[28] Ashfaq, M., *A Tribute to father of fuzzy set theory and fuzzy logic*, International Journal of Swarm Intelligence and Evolutionary Computation. Vol. 7: No.2 (2018), pp.1-5. **https://doi.org/10.4172/2090-4908.1000170**

[29] Ab Wahab, M. N., Nefti-Meziani, S., Atyabi, A.: *A Comprehensive Review of Swarm Optimization Algorithm,* PLOS ONE, Vol. 10: No.5 (2015), pp. 1-5. **https://doi.org/10.1371/journal.pone.0122827**

[30] Bjorlykhaug, E., Egeland, O., *Mechanical Design Optimization of a 6DOF Serial Manipulator Using Genetic Algorithm*, IEEE Access, 6 (2018) pp. 59087-59095. **https://doi.org/10.1109/ACCESS.2018.2875272**

[31] Ibraheem, B., Kazem, Mahdi, B., Talib, A *Motion planning for a robot Arm by Using Genetic Algorithm*, Jordan Journal of Mechanical and Industrial Engineering, Vol. 2: No.3 (2008), pp.131-136.

[32] PSO tutorial, http://www.swarmintelligence.org/tutorials.php, Last accessed: 11.08.2020.

[33] Firas, R., Umniah, I.: *Heuristic D* algorithm based on Particle Swarm Optimization for Path Planning of Two-Link Robot Arm in Dynamic Environment*, Al-Khwarizmi Engineering Journal, Vol. 15: No.2 (2019) pp.108-123. **https://doi.org/10.22153/kej.2019.01.004**

[34] Chaturvedi, K.T., et al.: *Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch*, International Journal of Electrical Power & Energy Systems, Vol. 31 (2009) pp.249- 257. **https://doi.org/10.1109/ICCCNT.2012.6396022**

[35] Seixas, B., De Almeida, G., Coppo Leite, V.: *Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems*, Intechopen, London, 2019. ISBN: 978-1-78984-537-2. **https://doi.org/10.5772/intechopen.89633**