Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

REINFORCEMENT LEARNING WITH DISCRETE EVENT SIMULATION: THE PREMISE, REALITY, AND PROMISE

Sahil Belsare
Emily Diaz Badilla
Mohammad Dehghanimohammadabadi

Mechanical and Industrial Engineering Department
Northeastern University
360 Huntington Ave
Boston, MA 02115, USA

ABSTRACT

Several studies have shown the success of Reinforcement Learning (RL) for solving sequential decision-making problems in domains like robotics, autonomous vehicles, manufacturing, supply chain, and health care. For such applications, uncertainty in real-life environments presents a significant challenge in training an RL agent. RL requires a large number of trials (training examples) to learn a good policy. One of the approaches to tackle these obstacles is augmenting RL with a Discrete Event Simulation (DES) model. Learning from a simulated environment, makes the training process of the RL agent more efficient, faster, and even safer by alleviating the need for expensive real-world trials. Therefore, integrating RL algorithms with simulation environments has inspired many researchers in recent years. In this paper, we analyze the existing literature on RL models using DES to put forward the benefits, application areas, challenges, and scope for future work in developing such models for industrial use cases.

1 INTRODUCTION

Artificial Intelligence (AI), due to its proven benefits, is being utilized in various industries such as engineering, transportation, healthcare, energy, finance, and e-commerce (Powell 2019). AI's ability of 'self-learning algorithms' has become an attractive solution approach for industries to solve a vast range of sequential decision-making problems. Due to the diversity of applications, several communities have emerged to address the problem of making decisions over time to optimize some metrics (Powell 2019). One such influential approach in AI is Machine Learning (ML), which can be further branched into Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, and Reinforcement Learning (RL).

Recently, optimization theorists are focusing on RL due to its remarkable success in the operations management domain (Gosavi 2009). This is evident by the growing body of literature that is applying RL techniques to existing Operations Research (OR) problems. Among the numerous approaches to solve sequential decision-making problems, two that stand out are optimal control (which laid the foundation for stochastic optimal control), and Markov Decision Processes (MDP), which provides the analytical foundation for reinforcement learning (Powell 2019). There is a growing sense of belief that RL is a better alternative than classical Dynamic Programming (DP) to optimize MDPs and Semi-MDPs (S-MDP) (Sutton and Barto 2018). This can be attributed to RL's prominent feature of optimizing stochastic systems by tackling the curse of dimensionality and the curse of modeling (Idrees et al. 2006). By pre-computing optimal policy, RL overcomes the issue of long online computation when compared with traditional mathematical control methods.

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

In RL, an agent learns to map an optimal policy for a model by multiple trial-and-error searches guided by a scalar reward. The agent's actions affect the immediate rewards and the future rewards as well. These two features – *guided trial-and-error search* and *delayed feedback*, distinguish RL from all other topics of ML, to push the current boundaries of knowledge (Sutton and Barto 2018). However, these features introduce a unique challenge of a trade-off between exploration and exploitation. Thus, to map a near-optimal policy, an RL agent must undergo exploration of state spaces and later exploit the available knowledge. For complex optimization, offline training of the RL agent may require hundreds of thousands of steps to achieve even a near-optimal policy (Nian et al. 2020). The other challenge is to explore in a model-free MDP, where the transition probabilities are estimated based on the sample interaction with the environment.

A large number of unknown possible states in the model makes it more difficult to solve the MDP using the direct approaches than using a simulation-based RL approach (Shitole et al. 2019). Use of simulation to create multiple scenarios and optimize the process gave importance to the field of simulation-optimization. Similarly, computer simulations prove to be an impressive alternative to overcome the challenges for RL modeling to optimizing a variety of real-life problems. Simulation proves to be a great resource to:

- build a virtual environment when training data set is impossible or infeasible to obtain from the real world.
- facilitate an agent to interact with the environment in a digital world and gain its intelligence while saving time and money.
- allow an RL agent to map various stochastic states, otherwise unable to predict - as is the case with most industrial problems.

Therefore, augmenting RL algorithms with simulated environments is becoming a growing field of research. The noticeable success in implementing an integrated framework calls for a need to analyze it's vast potential. This paper aims to contribute to this line of research by:

- collecting, reviewing and analysing the published studies using an integrated RL agent with simulated environments, with focus on discrete event simulation (DES) models.
- categorizing the related studies based on multiple factors such yearly trends, journals, and applied DES models.
- discussing the benefits, application areas, challenges and provide insights for the future works.

The rest of this paper is organized as follows: Section 2 and 3 provide an overview of RL and its application for sequential decision-making, respectively; Section 4 discusses the the need of an integrated RL and DES framework; Section 5 reviews a summary of existing papers; Sections 6 addresses discussions, future works, and a conclusion.

2 AN OVERVIEW OF REINFORCEMENT LEARNING

An emerging period for RL was in the 1990's with its foundation on Markov Decision Process (introduced in the 1950's). Later, RL grew to prominence in 2016 when it was credited with solving the game of Go using AlphaGo (Powell 2019). As seen in the Figure 1, an RL model consists of an agent, environment, its states, sets of possible actions, a scalar reward, and transition probabilities (Powell 2019).

MDPs can be further classified as Fully observable MDPs, Partially observable MDPs, Semi-MDPs (Nian et al. 2020). These categories gives a broad classification for RL models – Model-based RL and Model-Free RL. In Model-based RL we are aware of the transition probabilities and sample them from the probability distribution. Whereas in Model-free RL, we estimate the transition probabilities by simply observing the state transitions (Powell 2019). In an RL model, an agent is a decision-maker and an environment is everything that influences the agent's decision. An RL agent tries to learn a near-optimal policy by interacting with the environment with improved trial-and-error episodes. This process repeats itself trying to maximize the overall reward value (Sutton and Barto 2018).

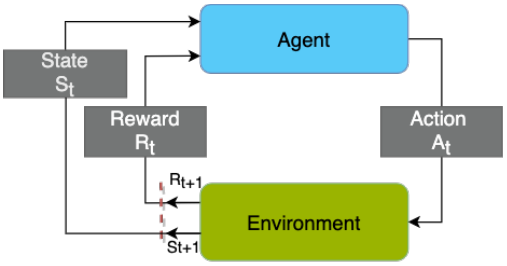

Figure 1: The agent-environment interaction.

3 REINFORCEMENT LEARNING FOR SEQUENTIAL DECISION MAKING

Tremendous research has been published on optimal control focusing on deterministic and stochastic control with a strong mathematical background. These traditional optimal control methods have demonstrated success in solving multistage optimal control problems. However, they account for limited online computation capabilities for industrial optimization problems with Multiple-Input and Multiple-Output (MIMO) (Nian et al. 2020). Along with that, Dynamic Programming proves ineffective for large scale MDPs due to the curse of dimensionality and curse of modeling. It is on such a large scale and complex dynamic optimization, in particular, usage of RL becomes more obvious (Nian et al. 2020). RL models are formulated to account stochasticity within the system. And it is due to the recent success in solving some complex problems that RL has attracted significant attention from the optimization community (Nian et al. 2020).

Lately, RL has been applied in a variety of complex optimization problems in domains like supply chain, autonomous vehicles, drones, manufacturing, finance, and health science. A stochastic scheduling using dynamic RL policy outperforms various cyclic policies and meets the production constraints in a manufacturing industry. While on the other end, human-generated static policies proved less optimal when compared with RL developed dynamic response policies to minimize infectious disease outbreaks (Farhan et al. 2020). Deep Reinforcement Learning (DRL) has been applied successfully to scheduling problems, such as resource management or global production scheduling (Gros et al. 2020).

Current RL libraries, such as OpenAI Gym, have many interesting problems, but they are not directly relevant to industrial use (Hubbs, C D and Perez, H D and Sarwar, O and Sahinidis, N V and Grossmann, I E and Wassick, J M 2020). Recently, there are notable efforts to showcase its industry potential like being used to solve classical optimization problems like VRP, TSP, Knapsack, and much more Hubbs, C D and Perez, H D and Sarwar, O and Sahinidis, N V and Grossmann, I E and Wassick, J M (2020). Another example is OR-Gym, an open-source Python based library of reinforcement learning environments consisting of classic operations research and optimization problems. Hubbs, C D and Perez, H D and Sarwar, O and Sahinidis, N V and Grossmann, I E and Wassick, J M (2020) showed that RL approach outperforms the benchmark in many of the more complex environments where uncertainty plays a significant role. The paper also suggests that such platforms would ensure accessibility of RL to solve industrial optimization problems where models do not exist or are too expensive to compute online.

4 NEED OF SIMULATION FOR REINFORCEMENT LEARNING

The characteristic of RL is learning through continuous interaction with the environment. For a gigantic model, training an agent offline may need a lot of steps to achieve a near optimal policy. Thus making it infeasible for live operations. To overcome this issue, the agent may be first trained in a simulated environment to obtain general knowledge of the process.

Applications like supply chain, manufacturing resource allocation, and industrial robotics are good examples of large problems with limited visibility of all possible states. This makes it extremely difficult, if not impossible, to write a program that could effectively manage every possible combination of circumstances occurring in real-life scenarios. For practical optimization purposes, the list of uncertainty is countless and hence the transition probabilities. Such complex models with large states can be augmented with RL in a simulated environment provided we know the distribution. Another hindrance is finding a readily available training data set. Sometimes a publicly available dataset cannot exactly recreate the training scenarios of the desired model. In many optimization problems, obtaining a real-world data can be expensive due to the cost of operation.

In response to these challenges, researchers have harnessed the power of simulation tools to generate limitless training data set. These tools have become essential in the development of algorithms, particularly in the fields of Robotics and Deep Reinforcement Learning. Simulation enables rapid prototyping by easily customizing new tasks or scenarios. It also provides a low-risk benefit in training costly application like robotics or a million dollar industrial setup.

The above-mentioned systems have a high demand for flexibility- whether to minimize cost, maximize production, or synchronize actuators, and to account for the complexity and uncertainty, factory control systems can leverage AI models such as RL algorithms. In these cases, simulation is highly relevant for the RL model as it can generate a large number of scenarios closer to the complexity of real-scale systems that would be difficult to be ensured otherwise (Zielinski et al. 2021).

There are primarily three types of system simulation methods: Discrete-Event Simulation (DES), Agent Based Modeling (ABM), and System Dynamics (SD). DES is a powerful tool to perform 'What If analysis' of a system, an essential characteristic to optimize sequential decision-making processes (Arulkumaran et al. 2017). Limited research is available regarding the combination of RL with DES. However, there exists a growing body of literature that is applying RL techniques to existing OR problems. The paper takes an effort to cover some of such cases involving simulation and RL. Section 5, elaborates on techniques used, simulation setup, Agents used, Architecture, and findings from the reviewed paper. Below we summarize the existing literature of RL models using DES for various applications.

5 RL WITH DES AS OPTIMIZATION TECHNIQUE

5.1 Literature Selection Process and Overview

This section illustrates the literature using RL with DES for optimization purposes.

The search for papers included keywords like Reinforcement Learning, RL, Simulation, Discrete Event Simulation, and DES. These words were used in different permutations to obtain varied results. The papers were selected based on the inclusion criteria decided that is to have usage of RL with DES for specific applications. Figure 2 shows stages of the review process and number of studies included in each.

The papers searched were obtained from the database of Winter Simulation Conference (WSC), Elsevier, SIMULTECH, IEEE, and arXiv. Figure 4 shows the trend by top 5 publishers for the studies that were selected by the inclusion criteria, with Winter Simulation Conference accounting for almost half of the papers and having a significantly higher increase of papers in this topic in the past years compared to other top publishers.

As seen in Figure 3, comparatively limited research has been published on the application of RL with DES for optimization. While there were instances of publication in early 2000s, the trend seems to be growing since 2017.

5.2 Selected Work on RL with DES Deep Dive

Given the diversity of simulation environment tools available, Figure 5 shows the distribution of the software or programming language used by the selected works analyzed. This results shows many of these research is conducted in Python, both for simulation modeling, and the RL algorithm design. Python is an open source

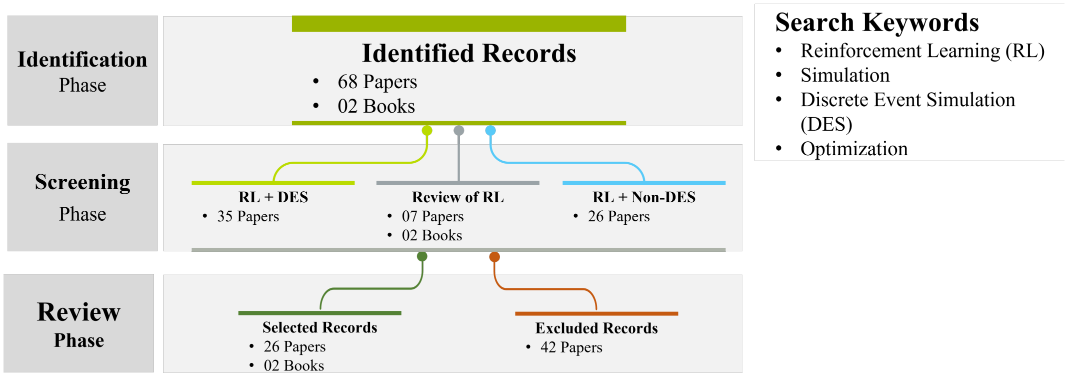

Figure 2: The literature profile methodology.

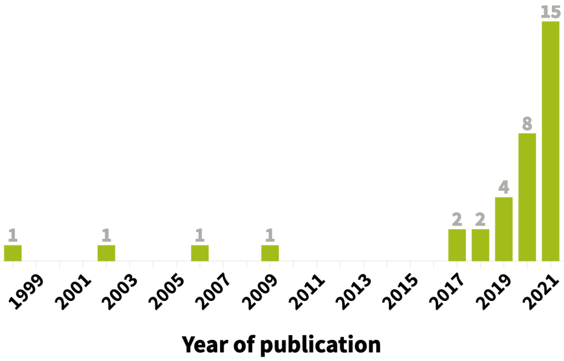

Figure 3: Trend of selected publications over time.

programming language with multiple Machine Learning and Simulation modeling libraries. Therefore, it become a favorable platform to design, develop, and implement simulation-based RL models. Users can develop the simulation model in Python packages such as SimPy, ORGym, etc., and integrate them with their own design RL agents or pre-stabled RL model.

In addition to the open-source programming language, some papers used commercial simulation packages such as FlexSim, Simio, and AnyLogic. These packages are perfect tools to mirror the real-life systems in details and create a virtual representation for them in a simulated environments. However, developing RL agent for these model requires extensive user-defined setting, or API connection.

From the application stand-point view, majority of the research are conducted in the domain of manufacturing, followed by the supply-chain industry. This popularity arises since both industries are highly dependent on the policy in-effect. Thus, it makes sense that optimization researchers tend to apply RL in these domains. The oldest paper in the following survey dates to 2002. Creighton and Nahavandi (2002) have studied the behavior of an RL agent interacting with the DES model for training and developing a robust policy. The DES model consists of a stochastic serial line production facility with breakdowns that could be repaired online or offline. The interaction between a MATLAB RL agent and simulation software was facilitated by 'Quest', a visual basic server. The whole architecture was collectively termed as 'Reinforcement Agent Simulation Environment' (RASE). RASE was successful in training an agent to achieve an average lower cost of action and effectively identify an optimal operating policies of real production facilities. The RASE architecture allowed interfacing between the RL agents and commercial simulation products - a popular approach of integrating RL with DES. The paper provides valuable insight on building such architecture which can be similarly used for other problem cases.

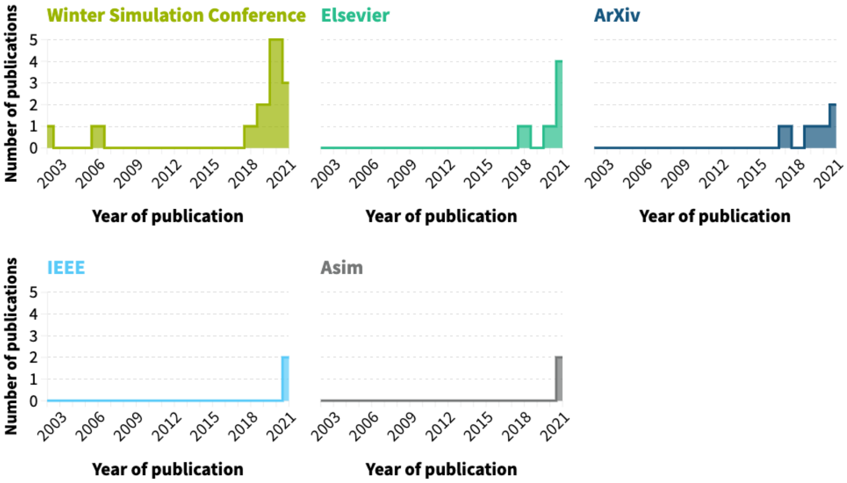

Figure 4: Trend of selected articles by publisher.

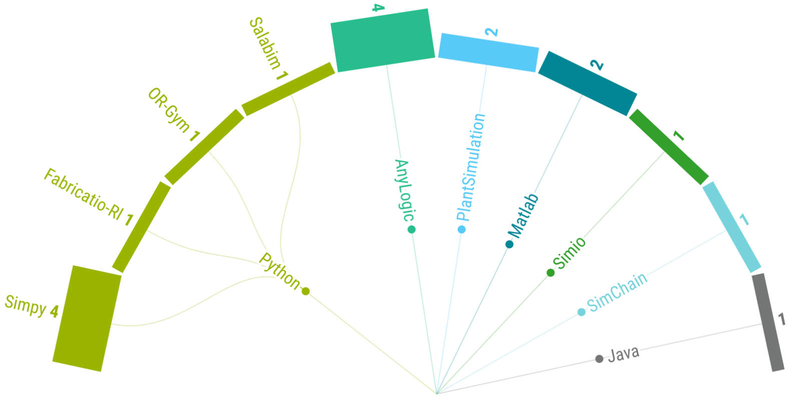

Figure 5: Breakdown of selected articles by simulation environment.

An important characteristic of a simulated environment is that it randomly samples new states from the simulation model. This enables the RL+DES model to map stochasticity of the real-life environment. A similar approach was tested using DES and RL framework by Dadone et al. (1998) to simulate operations of a manufacturing plant. The proposed model entirely depended on the simulation developed and assumes no analytical background of the plant. Furthermore, two different approaches of offline and online adaptation were compared as well. Standard simulation optimization techniques like finite differences and response surface methodologies were used as off-line approaches. Whereas, the on-line reinforcement learning used the approximation paradigms of Fuzzy Logic Systems (FLS) and Artificial Neural Networks (ANN) to provide adaptation. Both approaches, FLS and ANN, showed good approximation and adaptation properties. The neural adaptation also showed better extrapolation capabilities. The author states that proposed method is still in its early phase and needs further study and experimentation to prove its general validity and applicability. The paper introduces a noticeable experimentation where simulation can be used to RL to work in a nearly unknown environment.

On narrowing down to a specific application in manufacturing domain, the next popular problem for RL+OR community seems to be job shop scheduling. The Job shop problem operates in a highly uncertain environment and adapting improved policies with time is crucial. Idrees et al. (2006) worked on scheduling arriving jobs to a single machine with an objective to minimize the mean tardiness. A DES model using JAVA programming was used to generate multiple scenarios of arriving jobs. The RL agent was trained to choose actions from a combination of the dispatching rules and number of workers. This paper displays a unique framework of optimizing traditional mathematical methods by effectively employing them based on the current state of model. The results obtained showed significant savings in the overall system's cost and was able to find optimal policy of hiring and dispatching rules once trained using the integrated environment.

Taking a step further by including integrated process planning, Lang et al. (2020) experimented with Deep Q-Network (DQN) and DES for solving a flexible job shop problem. The authors integrated two DQN agents with a DES model wherein one agent is responsible for the selection of operation sequences, while the other allocates jobs to machines. Instead of using off-the shelf simulation software, the simulation training environment was implemented in Python library Salabim. The authors perform a thorough investigation on DQN's performance with a focus on its ability to transfer learning, a highly researched RL characteristic. A couple of notable outcomes from the paper are DQN's ability to always find a better solution than the GRASP algorithm for every problem, once trained. The other being, efficient transfer of knowledge as the prediction and evaluation of newly introduced production schedules requires less than 0.2 seconds.

Several other papers have displayed positive results using DES model to train DQN RL agents. Zhang et al. (2018) showed that the algorithm interacts with the simulation and learns the batching knowledge gradually. The experimental results validate that this approach performs better than conventional decision rules for real-time batching job shops. These papers have successfully highlighted the combined use of DES and RL to solve a job-shop scheduling problems while integrating dispatching techniques.

Expanding the scope from Job-shop allocation, several efforts have been published to model DES environment to train and test RL's performance for global production scheduling. Waschneck et al. (2018) advocate RL's importance in implementing Industry 4.0 across the various industries. The author states that breaking down and balancing global goals to local Key Performance Indicators (KPIs) is challenging in complex job shop environments. To solve this, the authors present a different approach for using a DES model with the Google DeepMind's DQN agent. The factory simulation was implemented in MathWorks MATLAB to work with recent machine learning algorithms. Later, the MATLAB API for Python was used to implement an OpenAI Gym Interface. The proposed system automatically develops a scheduling solution, which is on a par with the expert benchmark, without human intervention or any prior expert knowledge. Similar research has been carried out by Gros et al. (2020) to assert that DQN learning performs extremely well, and the resulting strategies provide near-optimal decisions in real-time, while alternative approaches are either slow or give strategies of poor quality. Both the papers conclude with the proposed future work on optimizing a balanced or weighted cost function.

One of the benefits of using simulation for queuing models is the identification of bottlenecks and the comparison of what-if scenarios. This feature offers an RL agent to explore random extracts the new samples states from the dataset (Farhan et al. 2020). To test this, Farhan et al. (2020) simulated an imaginary coffee shop to map the consequence of actions taken by the RL agent on the operational cost. The paper lays out the feasibility of using an off-the shelf simulation software with its inbuilt function of integrating an RL model. The simulation was carried out on AnyLogic 8 Professional 8.5 software with a two-agent system – Customer and Server following DES and ABS modeling, respectively. The observations provided by the model cater to the necessary information needed about the environment and impact the actions to be taken (Farhan et al. 2020). The RL model's ability to make better and faster decisions than rule-based heuristics can be attributed to Population Based Training (PBT) pioneered by Google's DeepMind that was applied to train the model. The objective of PBT is to automatically discover the best set of hyperparameters that encourage the learning agent to find the best performing policy as quickly as possible (Farhan et al. 2020). On successful training, this method was seamlessly replicated to represent similar industrial-scale models of queuing theory. The author affirms that the simplicity of using the Pathmind cloud-based platform is due the fact that it does not require prior knowledge of RL algorithms.

In the manufacturing job scheduling field, it is of great relevant to have real-time multi-objective rescheduling methods with the ability of handling random disturbances such as dynamic demands, machine breakdowns and uncertain processing times in real time while simultaneously considering different objectives such as makespan and total tardiness (Luo et al. 2021). Most of scheduling problems of this nature, can be regarded as dynamic multi-objective flexible job shop scheduling problem (DMOFJSP). DMOFJSP can be modeled as a Markov decision process (MDP) where the decision maker should successively determine the right actions, i.e., the feasible dispatching rules based on the production status of different rescheduling points so as to optimize the predefined long-term objectives. The work by (Luo et al. 2021) proposes the combination of of Deep Reinforcement Learning (DRL) and Hierarchical Reinforcement Learning (HRL) to solve the DMOFJSP, with a two-hierarchy deep Q network that uses DQN based agents (THDQN). Similarly to the already mentioned works on this field, the results of their experiments have confirmed both the effectiveness and generality of the proposed THDQN compared to different composite dispatching rules and other RL methods. (Luo et al. 2021)

In semiconductor manufacturing, photolithography is a key process which transfers geometric patterns to the semiconductor. This part can be expensive to increase resources and therefore, a good production scheduling for it increases the throughput and efficiency of the entire fab (Kim et al. 2021). On their work, (Kim et al. 2021) propose to use DQN with Action Filter (AF) to solve large-scale photolithography scheduling problem modeled by simulation with DES. Experiments demonstrated improved performance compared to typical rule-based strategies, weighted shortest processing time (WSPT) and apparent tardiness cost with setups (ATCS) rules perform 28% and 32% worse for weighted tardiness, respectively (Kim et al. 2021).

Shipbuilding industry has recently leveraged RL for scheduling jobs. This industry deals with intrinsic variation to plans and a high complexity given the multiple factors that play a role such as resources, space, workforce, among others (Woo et al. 2021). (Woo et al. 2021) developed a DDQN for an efficient schedule system, trained over a DES model of their production system implemented with SimPy. The objective is to minimize total lead time and it was found that the RL-based scheduling algorithm showed satisfactory performance in terms of shortening lead time (Woo et al. 2021).

In the mining industry, Rl developments has also been shown satisfactory results on truck-dispatching rules. The pursue for this kind of methods comes from the necessity on generating real-time truck dispatching responses that adapts given different mining complex configurations in order to deliver supply material extracted by the shovels to the processors (de Carvalho and Dimitrakopoulos 2021). The method aims to improve adherence to the operational plan and fleet utilization in a mining complex context by using a multi-agent modeled, using DDQN algorithm for each agent and trained on an environment defined as a DES simulator that emulates the interaction arising from mining operations. The approach was tested at a copper–gold mining complex and the results showed improvements in terms of production targets, metal production, and fleet management (de Carvalho and Dimitrakopoulos 2021).

RL has shown promise in multiple experiments on production scheduling, however they are difficult to reproduce and hence to validate because of the implementation overhead associated with writing a simulation and the lack of controlled stochasticity (Rinciog and Meyer 2021). With this in mind, (Rinciog and Meyer 2021) have developed FabricationRL, RL compatible, customizable and extensible benchmarking simulation framework available in python. It comes to bridge the validation gap that RL in concept-phase for settings such as scheduling in manufacturing are still needing to robustly embed them (Rinciog and Meyer 2021). FabricationRL is a production scheduling simulation framework implementing the OpenAI Gym interface, the framework covers and has examples of different types of scheduling problems and can be controlled using pre-computed plans as well Reinforcement Learning agents.

The next popular topic of interest for the RL and the optimization community seems to be policy development for autonomous vehicles. The same problem becomes a combinatorial optimization problem when one works with a fleet of vehicles or robots. Li et al. (2019) addressed the dispatching and routing problems for autonomous mobile robots using a DQN model. A DQN model was trained to dispatch a fleet of robots used for material handling tasks. The model was then tested on a DES environment. With the help of DES, author models real life constraints like narrow warehouse aisles, multiple pick up and drop off locations, the possibility of collision, and restricted traffic routing. The paper sheds some light on usage of simulation environment for training and testing an RL agent. When benchmarked with the shortest travel distance rule, the DQN model reduces the occurrences of traffic congestion but increases the travel distance. In contrast, the net effect reduces the makespan to complete a fixed number of delivery tasks and reduces the mean flow time. On similar lines, Feldkamp et al. (2020) used RL in combination with DES to control Automated Guided Vehicle (AGV) in a modular production systems. The author makes use of an off-the shelf DES software ‘Siemens Plant Simulation’ to create the training and testing DES environment. The DQL-algorithm was implemented in Python’s Tensorflow/Keras library and it was integrated with Plant Simulation via a TCP/IP-based interface. The author puts forward a detailed explanation of the training environment and its ability to create multiple stochastic scenarios. After considerable training, the agent was successful in obtaining a policy which was able to beat a heuristic decision rules. The author then goes ahead to discuss challenges and possible future scope of combining DES with DQN, which would be discussed in the later part of the paper. Taking a step further, Shitole et al. (2019) used a combination of model-free DQN, DES, and weighted neural network to optimize allocation of costly Earth Moving Machinery in construction and mining industries. The author compares simulation based RL approach for solving the MDP over non-simulation based methods such as mathematical programming or analytical fleet balancing. Different RL methods like Q-learning, actor-critic, and trust region policy optimization obtained significantly better policies than human-designed heuristics. Apart from successful optimization the author worked on a crucial topic of transfer-learning. The author asserts that transfer-learning is an essential research area for the deployment the proposed methodology, since it involves transferring the policies learnt over the simulator to the real world environment. Further discussion on this topic can be found in the later section of this paper.

Another area of application with growing importance is in the supply chain and logistics domain. Rabe and Dross (2015) points out that the usage of RL techniques in the inventory management and logistics domain date back to 2002. Following on the same, the authors presented the architecture and working principles of a Decision Support System (DSS) for logistics networks. The proposed system uses a DES model developed on SimChain to predict the consequences of possible changes in the logistics network. The author asserts that RL techniques seem to be a suitable approach to construct the internal principles of the proposed DSS for logistics networks. The discussion is moved on to the important topic of the ability of the developed model to provide useful action recommendations for states in the real logistics system.

To add, mastering end-to-end supply chain is inevitable for achieving a competitive edge. In order to enable the collaborating partners to track and optimize their performance, seamless evaluation is required (Afridi et al. 2020). To achieve that, the authors used RL to solve a Vendor Managed Inventory problem – a mainstream supply chain collaboration model. The aim was to find an optimal policy that would ensure desired inventory levels with suitable product life cycles. Using a simulation model, different demand scenarios were generated based on real data and later compared based on KPIs. The author laid out previous foundation noting success of DRL in various supply chain applications while asserting that no literature was found on simulating and finding optimal replenishment policy including split responsibility to the concept of VMI (Afridi et al. 2020). The model consists of an DQN RL agent implemented in Java (RL4J) library with AnyLogic simulation as a training environment and an external integrated development environment (IDE) called IntelliJ IDEA. The agent was imported back into AnyLogic model as a testbed, where the extended model was used to teach the learning agent on taking appropriate actions to achieve a desired state (Afridi et al. 2020). Upon training and simulating the environment for various demand scenarios it was observed that the percentage no-violation inventory status improved from 43% to 95% and 99% for both scenarios respectively. The agent was also able to maintain higher KPI service levels. The author urges to work on investigating additional demand scenarios and conduct a full design of experiment considering the volatility in demand and demand levels.

6 DISCUSSION AND FUTURE WORK

Integrating simulation models with RL is promising, and recently has received attention by the community. However, implementing and using these models are at the early-stages and there are lots of constraints and challenges. A significant barrier in expanding research on RL with DES model is that the practitioners in each area have backgrounds in using different tools/software. Upon exploring various options for integrating RL with DES, it was concluded that there is a requirement to learn skill sets from both areas (Greasley 2020). Another solution to this could be developing an interface between the RL agent and DES software – an approach that was evident in many studied papers.

Greasley (2020) provides an extensive survey on papers that used an external interface to combine simulation packages and RL agents. The author stresses exploring architectures to enable smooth execution of such integrated model. To add, Microsoft's Project Bonsai gives an option to custom teach AI agents and solve real-world problems with little or no background in coding. Upon a cursory glance, off-the-shelf simulation packages like Simio are working on integrating simulation with RL to introduce improved simulation optimization techniques. Along similar lines, Pathmind, a SaaS platform enables simulation and RL integration to solve industrial problems. It makes use of simulation to create a digital twin to train and test RL agents and obtain optimal policies.

In terms of coding-based solutions, more packages have appeared on recent years bringing new and powerful RL models. As an example, the RL Glue project allowed RL practitioners to connect agents, environments, and experiment programs, even if they are written in different languages, enabling leveraging multiple platforms and homologation of models easily. Following that, some papers made use of Python-based simulation environment because that enabled a smooth integration of the DES model with the RL agent. SimPy and Salabim and newly developed RL simulation libraries were observed in the papers reviewed.

The other scope of future work is to test the scalability of the DES + RL model. Most of the authors conclude their research with the stress on the said topic. These limitations exist because RL algorithms share the same complex issues as other algorithms: computational complexity, sample complexity, and memory complexity (Creighton and Nahavandi 2002). The solution to computational complexity could be using a Deep Reinforcement Learning (DRL) algorithms. DRL relies on the property of automatically finding compact low-dimensional representations (features) of high-dimensional data (Arulkumaran et al. 2017). Although deep neural networks can make reasonable predictions in simulated environments over hundreds of time-steps, they typically require many samples to tune the large number of parameters they contain (Feldkamp et al. 2020). This opens a vast area of research - the sensitivity of the DQN model to changes in any of its parameters, a research interest noted in most of the reviewed papers. It is followed by the challenge significantly present in the machine learning field - developing a standardized way to evaluate and benchmark RL techniques against traditional known methods.

The solution to sample complexity is to simulate digital twins as training and testing environments. One of the main reasons for simulating a probabilistic digital twin is to enable the added capability of proper treatment of uncertainties associated, without the need to interfere with the physical system itself (Agrell, Christian and Dahl, Kristina Rognlien and Hafver, Andreas 2021). The probabilistic digital twin also caters to the inherent challenge of an RL algorithm – exploration v/s exploitation. Having said this, utmost care should be taken while simulating a training environment. The discrepancy between the real-world constraints and the simulated environment might cause RL agents to behave in unpredictable ways. The success of using a simulation environment to train and test the RL models has intrigued the optimization community to work on the challenging task of 'transferring learning' from a simulated to a real environment. Arulkumaran et al. (2017) reviewed a significant amount of work to facilitate the transfer of learning. To make DRL more efficient, one can make use of previously acquired knowledge in the form of transfer learning, multitask learning, and curriculum learning (Arulkumaran et al. 2017). Combining simulation and RL opens another field of learning - known as behavioral cloning in traditional RL literature.

To sum up, RL has demonstrated its potential in surpassing or achieving comparable results in many complex tasks when benchmarked against traditional techniques. A steady progress in the optimization community to solve various industrial problems using RL is evident by reviewing the published papers. A compelling case to combine simulation and RL to overcome the uncertainties in the industrial application was made by the reviewed studies. Even though limited research has been published on simulated reinforcement learning, the trend seems to be increasing. Perhaps, we are not too far from making simulation an integral part of training and testing a Reinforcement Learning agent. Most of the reviewed studies call for exhaustive work in the coming years. Along with highlighting the success and opportunities, the parallel goal of this paper is to encourage researchers to work on the challenges addressed. This needs to be accomplished by a combined effort from the simulation, reinforcement learning, and operations research community.

REFERENCES

Afridi, M. T., S. Nieto-Isaza, H. Ehm, T. Ponsignon, and A. Hamed. 2020. "A Deep Reinforcement Learning Approach for Optimal Replenishment Policy in A Vendor Managed Inventory Setting For Semiconductors". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 1753–1764. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Agrell, Christian and Dahl, Kristina Rognlien and Hafver, Andreas 2021. "Optimal Sequential Decision Making with Probabilistic Digital Twins". https://arxiv.org/abs/2103.07405. accessed 10th October 2022.

Arulkumaran, K., M. P. Deisenroth, M. Brundage, and A. A. Bharath. 2017. "Deep Reinforcement Learning: A Brief Survey". *IEEE Signal Processing Magazine* 34(6):26–38.

Creighton, D., and S. Nahavandi. 2002. "Optimising Discrete Event Simulation Models Using a Reinforcement Learning Agent". In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 1945–1950. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Dadone, P., H. VanLandingham, and B. Maione. 1998. "Modeling and Control of Discrete Event Dynamic Systems: a Simulator-Based Reinforcement-Learning Paradigm". *International Journal of Intelligent Control and Systems* 2(4):609–631.

de Carvalho, J. P., and R. Dimitrakopoulos. 2021. "Integrating Production Planning with Truck-Dispatching Decisions through Reinforcement Learning While Managing Uncertainty". *Minerals* 11(6).

Farhan, M., B. Göhre, and E. Junprung. 2020. "Reinforcement Learning in Anylogic Simulation Models: A Guiding Example Using Pathmind". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 3212–3223. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Feldkamp, N., S. Bergmann, and S. Strassburger. 2020. "Simulation-Based Deep Reinforcement Learning For Modular Production Systems". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 1596–1607. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gosavi, A. 2009. "Reinforcement Learning: A Tutorial Survey and Recent Advances". *INFORMS Journal on Computing* 21(2):178–192.

Greasley, A. 2020. "Architectures for Combining Discrete-event Simulation and Machine Learning". In *Proceedings of the 10th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, edited by F. De Rango, T. Oren, M. Obaidat, M. Obaida, and M. Obaidat, 47–58. Lieusaint - Paris, France: SCITEPRESS - Science and Technology Publications.

Gros, T. P., J. Groß, and V. Wolf. 2020. "Real-Time Decision Making for a Car Manufacturing Process Using Deep Reinforcement Learning". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 3032–3044. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hubbs, C D and Perez, H D and Sarwar, O and Sahinidis, N V and Grossmann, I E and Wassick, J M 2020. "OR-Gym: A Reinforcement Learning Library for Operations Research Problem". https://arxiv.org/abs/2008.06319. accessed 10th October 2022.

Idrees, H. D., M. O. Sinnokrot, and S. Al-Shihabi. 2006. "A Reinforcement Learning Algorithm to Minimize the Mean Tardiness of a Single Machine with Controlled Capacity". In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1765–1769. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kim, T., H. Kim, T.-e. Lee, J. R. Morrison, and E. Kim. 2021. "On Scheduling a Photolithograhy Toolset Based on a Deep Reinforcement Learning Approach with Action Filter". In *2021 Winter Simulation Conference (WSC)*, edited by S. Kim,

B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1–10. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lang, S., F. Behrendt, N. Lanzerath, T. Reggelin, and M. Müller. 2020. "Integration of Deep Reinforcement Learning and Discrete-Event Simulation for Real-Time Scheduling of a Flexible Job Shop Production". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 3057–3068. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Li, M. P., P. Sankaran, M. E. Kuhl, R. Ptucha, A. Ganguly, and A. Kwasinski. 2019. "Task Selection by Autonomous Mobile Robots in a Warehouse Using Deep Reinforcement Learning". In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H. Bae, S. Lazarova-Molnar, M. Rabe, P. H. C. Szabo, and Y.-J. Son, 680–689. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Luo, S., L. Zhang, and Y. Fan. 2021, sep. "Dynamic Multi-Objective Scheduling for Flexible Job Shop by Deep Reinforcement Learning". *Computers and Industrial Engineering* 159(C).

Nian, R., J. Liu, and B. Huang. 2020. "A Review on Reinforcement Learning: Introduction and Applications in Industrial Process Control". *Computers and Chemical Engineering* 139:106886.

Powell, W B 2019. "From Reinforcement Learning to Optimal Control: A Unified Framework for Sequential Decisions". https://arxiv.org/abs/1912.03513. accessed 10th October 2022.

Rabe, M., and F. Dross. 2015. "A Reinforcement Learning Approach for a Decision Support System for Logistics Networks". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 2020–2032. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Rinciog, A., and A. Meyer. 2021. "Fabricatio-Rl: A Reinforcement Learning Simulation Framework for Production Scheduling". In *2021 Winter Simulation Conference (WSC)*, edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1–12. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Shitole, V., J. Louis, and P. Tadepalli. 2019. "Optimizing Earth Moving Operations Via Reinforcement Learning". In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H. Bae, S. Lazarova-Molnar, M. Rabe, P. H. C. Szabo, and Y.-J. Son, 2954–2965. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sutton, R. S., and A. G. Barto. 2018. *Reinforcement Learning: An Introduction*. Second ed. Cambridge, Massachusetts: The MIT Press.

Waschneck, B., A. Reichstaller, L. Belzner, T. Altenmüller, T. Bauernhansl, A. Knapp, and A. Kyek. 2018. "Optimization of Global Production Scheduling with Deep Reinforcement Learning". *Procedia CIRP* 72:1264–1269.

Woo, J. H., Y. I. Cho, S. H. Nam, and J.-H. Nam. 2021. "Development of a Reinforcement Learning-Based Adaptive Scheduling Algorithm for Block Assembly Production Line". In *2021 Winter Simulation Conference (WSC)*, edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1–12. Piscataway New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhang, T., S. Xie, and O. Rose. 2018. "Real-time Batching in Job Shops Based on Simulation and Reinforcement Learning". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. Juan, N. Mustafee, S. J. A. Skoogh, and B. Johansson, 3331–3339. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zielinski, K. M., L. V. Hendges, J. B. Florindo, Y. K. Lopes, R. Ribeiro, M. Teixeira, and D. Casanova. 2021. "Flexible Control of Discrete Event Systems Using Environment Simulation and Reinforcement Learning". *Applied Soft Computing* 111:107714.

AUTHOR BIOGRAPHIES

SAHIL BELSARE completed his MS in Industrial Engineering at Northeastern University, Boston, MA, USA. His research includes using simulation combined with various optimization techniques. Currently, he is working on developing a framework for Reinforcement Learning Algorithms with Discrete Event Simulation models for industrial optimization. His e-mail address is belsare.s@northeastern.edu

EMILY DIAZ BADILLA is a graduate student pursing MS in Data Analytics Engineering at Northeastern University, Boston, MA, USA. She has 5 years of working experience as Senior Data Scientist in Management Consulting industry. Her current research includes applied Reinforcement Learning and Deep Learning algorithms. Her email address is diazbadilla.e@northeastern.edu.

MOHAMMAD DEHGHANIMOHAMMADABADI Associate Teaching Professor of Mechanical and Industrial engineering, Northeastern University, Boston, MA, USA. His research is mainly focused on developing and generalizing simulation and optimization frameworks in different disciplines. This article is part of his initiatives to develop a framework to integrate Discrete Event Simulation with Reinforcement Learning. His e-mail address is m.dehghani@northeastern.edu