Colegio de Postgraduados

Effectiveness study of learning algorithms in supply chains of a dairy business

Arellano-Cruz, Erika P.[1]; Martínez-Sibaja, Albino[1*]; Rodríguez-Jarquin, José P.[1]; Posada-Gómez, Rubén[1]; Bello-Ramírez, Angélica M.[1]; Núñez-Dorantes, Juan C.[2]

[1] Tecnológico Nacional de México - Instituto Tecnológico de Orizaba. Avenida Oriente 9 No. 852. Col. Emiliano Zapata. Orizaba, Veracruz, México. C.P. 94320.
[2] Universidad Tecnológica del Centro de Veracruz. Av. Universidad No 350, carretera federal Cuitlahuac-La tinaja, Loc. Dos caminos. Cuitlahuac, Veracruz, México. C.P. 94910.
* Correspondence: albino.ms@orizaba.tecnm.mx

ABSTRACT

Objective: To develop a control system to prevent over-response of the supply chain of a dairy business.

Methodology: The following methods were used: DQN, Double DQN, Dueling DQN, and Dueling Double DQN to determine the distribution of demand: normal and uniform.

Results: Results were calculated based on stability in learning (the last 10,000 episodes). It was observed that the means of DQN and DDQN were very similar. To validate whether the performance of the Dueling DQN algorithm is better than that of the DQN algorithm, a non-parametric test was performed to compare the mean rank of two related samples and to determine if there are differences between them. The p values were 5.83e−38 and 0.000 for the Normal and Uniform distributions, respectively.

Conclusions: The algorithm with the best results is Dueling DQN, with an average total cost of 151.27 units for the demand with a normal distribution and an average of 155.3 units for the demand with a uniform distribution. This method has less variability once convergence is achieved.

Keywords: Reinforcement learning; Deep learning; supply chain; inventory control.

Citation: Arellano-Cruz, E. P., Martínez-Sibaja, A., Rodríguez-Jarquin, J. P., Posada-Gómez, R., Bello-Ramírez, A. M., & Núñez-Dorantes, J. C. (2024). Effectiveness study of learning algorithms in supply chains of a dairy business. *Agro Productividad*. https://doi.org/10.32854/agrop.v17i10.2978

Academic Editor: Jorge Cadena Iñiguez
Associate Editor: Dra. Lucero del Mar Ruiz Posadas
Guest Editor: Daniel Alejandro Cadena Zamudio

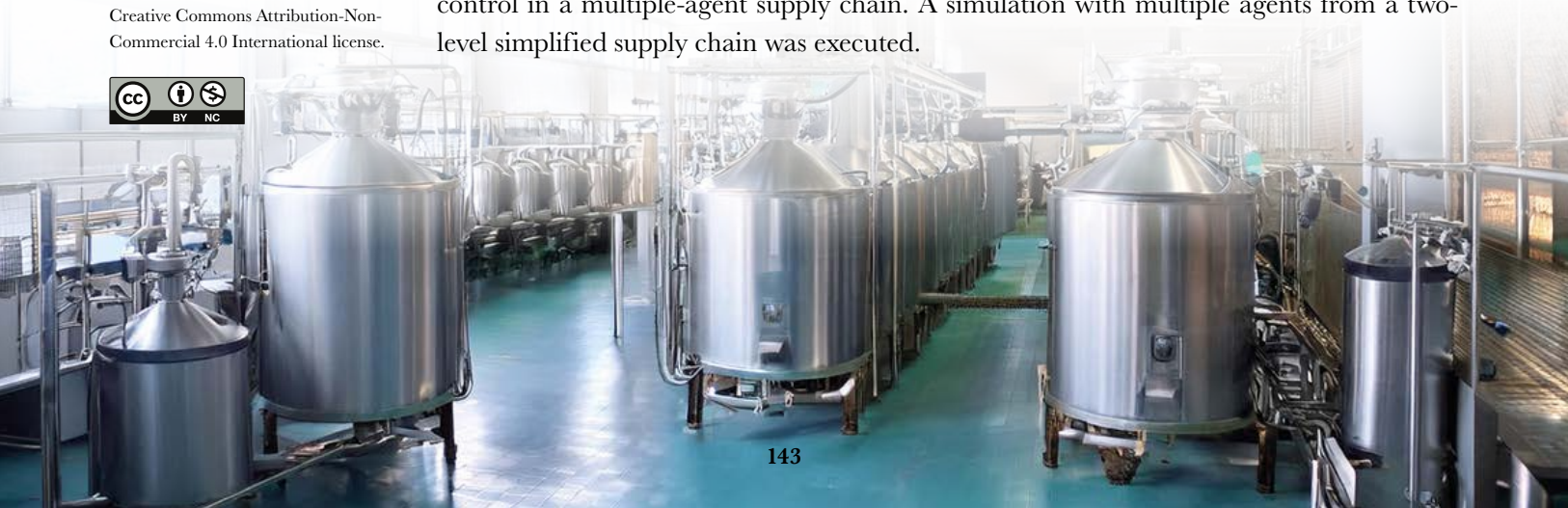
Received: July 16, 2024.
Accepted: September 13, 2024.
Published on-line: November 08, 2024.

Agro Productividad, 17(10). October. 2024. pp: 143-153.

INTRODUCTION

A policy for inventory control to integrally manage the decisions taken in every stage of the supply chain is presented in [9]. The inventory problem is modelled as a Markov Decision Problem (MDP) and is resolved using an algorithm of Reinforcement Learning (RL) to determine a nearly optimal balance policy under a criterion of average reward.

In a dynamic environment, traditional policies for placing orders, based on time and events, can become inefficient, leading to an excess or shortage of inventory. A learning algorithm by case reinforcement learning (CRL) is presented in [9], for dynamic inventory control in a multiple-agent supply chain. A simulation with multiple agents from a two-level simplified supply chain was executed.



An approach is presented in [2] to minimize the inventory costs through the determination of integral purchase order policies for the members of the supply chain. The management of orders is considered as a system of multiple agents which generates a RL model. Therefore, a Q-learning algorithm is proposed to solve the RL model. The results show that the Reinforcement Learning Order Mechanism (RLOM) obtained good results compared to other algorithms, such as the algorithm based on GA.

Optimal base inventory levels are presented in [3], [4] and [5] when there are no costs from lack of stock in the non-retail stages. However, no algorithm was found in the literature to find the optimal levels of base inventory for general cost structures of stock shortage. Recently, an algorithm based on Deep Q Networks (DQN) with experience repetition is proposed in [16] to solve the base inventory policy outlining the use of a RL algorithm.

This article presents a study of a dairy business that has the objective of minimizing the increase in inventory management costs. The business sends the product to a central distributor, which supplies different retailers. In this case, the farmer acts as the agent. There is no feedback between the farmer, the distributor or the retailer. The information available for the agent is the orders made by the distributor to the business. The objective of the agent is to determine the size of the production order to minimize the total level of inventory of the supply chain.

This study is divided into six sections: section two presents a review of the materials and methods used. Section three describes the application of the RL algorithm to the CS, and the configuration of the hyperparameters to achieve convergence. Section four describes the results obtained. Section five proposes the discussion, where the points proposed are defended, and finally, section six presents a conclusion of this study.

MATERIALS AND METHODS

The algorithm used to simulate the supply chain was adapted from the one proposed in the article, "A Deep Q-Network for the Beer Game: Deep Reinforcement Learning for Inventory Optimization". However, another algorithm may be used to represent the environment of the supply chain. The codes used to perform the experiments are available under request to the author through email.

Deep Q-Network (DQN) in the supply chain

In each t period, the agent observes the current state of the environment, where is the set of every state possible. In this case, the agent will be the farmer and the environment contains all the information about the levels of inventory and the costs. In function of the data from the state, the agent selects an action $a_t \in A(s_t)$, where $A(s_t)$ is the set of possible actions when the system is in state s_t. The agent (the farmer) performs an action, that is, issues a production order based on the information of the levels of inventory and the costs. The agent receives a reward $R_t \in \mathbb{R}$. The reward consists in the total accumulated cost until period t. Then the system makes the transition to the state $s_{t+1} \in S$.

The base of this experiment is Q-learning, a temporal difference learning algorithm (TD learning). The algorithm updates the estimation of the agent of the value function in each step of time during the episode.

$$V(s_t) = V(s_t) + \alpha\left[R_{t+1} + \gamma V(s_{t+1}) + V(s_t)\right] \quad (1)$$

Given that the equation has terms that are proportional to the estimation of time t and time $t+1$, that is, an estimation to update another estimation, it means that Q-Learning is what is known as a booting algorithm. In this case, $V(s_t)$ represents the prior state. R_{t+1} is the reward in $t+1$, $\gamma V(s_{t+1})$ is the updated value in the next step, and the sum of both terms represents the objective of temporal difference (Temporal Difference TD).

Case study

The study was conducted in a condensed milk manufacturing business that is facing challenges related to the excess of inventories. Currently, the business maintains high inventory levels, which results in high costs of storage and risk of obsolescence. In addition, the lack of effective planning of the demand and a limited capacity to predict the sales patterns, due to the scarce information, has led to lost sales when the inventory does not match the demand.

The intention is to find an optimal policy, known as π*, which could lead to the best expected cumulative reward (the least total cost). There are two main types of RL methods used to find this semi-optimal policy:

The methods based on policies directly train the policy to determine the appropriate action to take given a specific state. In this case, the action is calculated in function of factors such as the size of the shipment to the distributor, the level of inventory, the demand, and the order placed by the distributor.

Methods based on the value: These methods train a value function to determine the relative value of different states and to use this information to make decisions. In our case, the most valuable state is the level of inventory that guarantees an efficient delivery and minimizes the storage costs. The optimal order size (action) to maintain this level of inventory is calculated.

There are two different methods to find the value function.

$$v_\pi(s) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s\right] \quad (2)$$

Functions based on policies: The policy is trained directly to select an action given certain state. In this case, we do not have a value function defined by hand that determines the behavior of our policy. The training will define it.

Functions based on value: The value function is trained indirectly to map the value of a state or a state-action pair. With this value function, our policy can take actions. However, given that we do not train directly our policy, we need to specify its behavior. In our case, for example, given that our objective is to minimize the total cost, we use a policy that selects actions that consistently result in the lowest value, given the value function (Equation 2, greedy policy).

$$\pi^*(S) = \arg\min_a Q^*(s,a) \quad (3)$$

Given a state, our function of action value (which we trained) calculates the size of the order in this state. Then, our greedy policy (which we defined) selects the size of the order that incurs in the lowest storage cost and minimizes lost sales.

We have two types of functions based on the value:

State-Value Function: The state-value function (Equation 3) generates the expected yield for each state, that is, for each level of inventory, if the agent begins in the state and the policy continues until the game ends. However, this does not make sense in a supply chain (SC) since the states are repetitive. This situation could lead to divergence in the learning process of the RL algorithm.

$$v_{\pi}(s) = \mathbb{E}_{\pi}\left[G_{\pi} \middle| S_t = s\right] \tag{4}$$

Value-Action Function: The value-action function (Equation 4) returns the expected yield for each state-action pair when the agent begins in this state, performs this action, and then follows the policy. In the case of the supply chain (SC), this type of function is more appropriate because the agent observes its level of inventory, executes several actions (placing different orders with different order sizes), and then evaluates the cost of each action.

$$Q(s,a) = \mathbb{E}_{\pi}\left[G_{\pi} \middle| S_t = s, A_t = a\right] \tag{5}$$

The difference lies in that, in the value-state function, we calculated the value of a state, while in the value-action function we calculated the value of the state-action pair. This means that we determined the value of taking a particular action in a specific state.

Demand distribution

The demand data has two types of behavior depending on the sales season. The parameters for the normal distribution are N(5,1), while the parameters for the uniform distribution are U(0,10). The action space for the agent considers values between 0 and 10 units. The experiments were conducted with both distributions to evaluate the robustness of the model.

Configuration of hyperparameters

Each experiment in this study uses the same hyperparameters to achieve a valid comparison (see Table 1). The design of the neural network has three layers: 128, 64 and 32 neurons. The training of each method lasted 40,000 episodes.

A greedy strategy ε was used to balance the exploration and the exploitation of the agents. This strategy allows the agent to explore the environment before deciding on an exploitation strategy. This process of exploration and exploitation helps the agent to refine its model of the environment and to gradually approach a value function close to optimal every time the agent tries the state and receives a reward. The maximum value of epsilon, $\varepsilon max=1$, decreases linearly to its minimum value, $\varepsilon min=0.1$, during the first 10,000 episodes of training.

The initial values of the hyperparameters (Table 1) are proposed in [7] and [16]. Initially, the learning rate was 0.00025 and the discount factor was 0.99, although these configurations did not allow for the algorithm to reach convergence. The algorithm could not learn because a discount factor of 0.99 prioritizes the future reward, which does not make sense in an infinite game as in this case. In addition, when there is a delay between the action and the effect of the environment, a high learning rate confounds the learning process of the value function.

Initially, a neural network was configured with a single hidden layer of 10 neurons. However, because of the high non-linearity of the system, it did not produce significant results.

RESULTS AND DISCUSSION

DQN with experience repetition

The results of the DQN method with experience repetition can be seen in Figure 1c (the results from the DQN method with experience repetition show the normal distribution with the blue line, and the uniform distribution with the red line). The average cost is higher with a uniform distribution compared to a normal distribution. However, the agent's actions were similar for both types of demand distribution. It is important to highlight that, due to the nature of the uniform distribution, the cost of lost sales increases because of variability in the demand, which results in an average negative inventory. The behavior of the state variables is presented in Figure 1a, 1b, 1c, and 1d.

In the uniform distribution, the agent learned a policy of not having inventory to minimize the total cost. This policy is similar to the current policy established by several businesses. They prefer to lose sales instead of maintaining a high level of inventory because of the uncertainty in the demand. On the other hand, in the normal distribution, the agent learned a policy based on maintaining a minimum level of inventory to minimize lost sales, since the behavior of sales in this type of distribution is predictable. In the normal distribution, the total cost is made up mainly by the storage cost, while in the uniform distribution the total cost is composed mainly by the cost of lost sales.

Table 1. Configuration of hyperparameters.

Hyperparameters	Values
Gamma	0.9
Learning rate	0.00001
Agent history (m)	3
Number of neurons per layer	[128, 64, 32]
Activation function	[RELU, RELU, RELU, LogSigmoide]
Loss function	MSE
mini batch size	64
Optimization algorithm	Adam

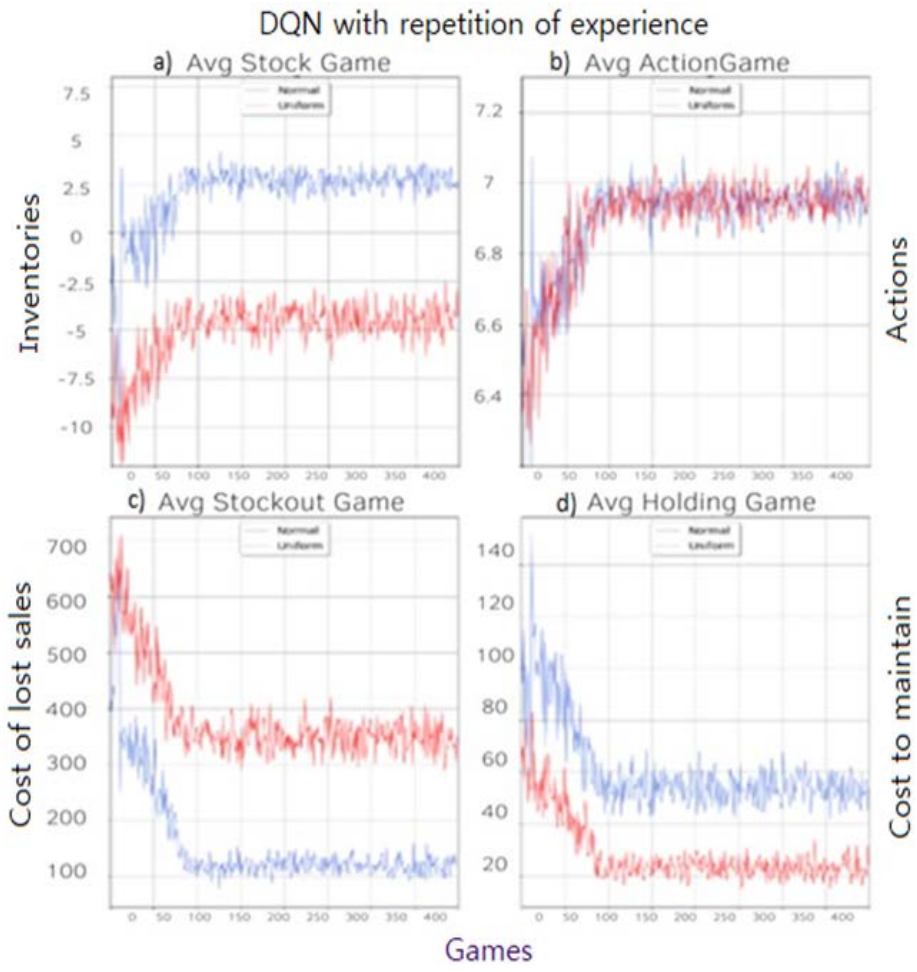


Figure 1. Evolution of the state variables during training.

Double DQN

Figure 2 shows the results of the training used in the DDQN method. Due to the similar behavior between the DQN and Double DQN algorithms, as presented in Figure 3, a Wilcoxon test is carried out to determine whether there is a significant difference between the methods. In the first place, a test is conducted to compare the scores of the algorithms under a normal distribution of the demand, resulting in a p value of 0.741039. Therefore, the differences in the results are not statistically significant. In other words, the yield of both algorithms under a demand with normal distribution is similar. The second Wilcoxon test examines the yield of both algorithms under a demand with uniform distribution, resulting in a p value of 0.8664. Therefore, the differences in the results are also not statistically significant in this case.

Given that the results of the DQN and DDQN methods do not show statistical significance, the DQN algorithm was selected to compare its yield with the Dueling DQN method.

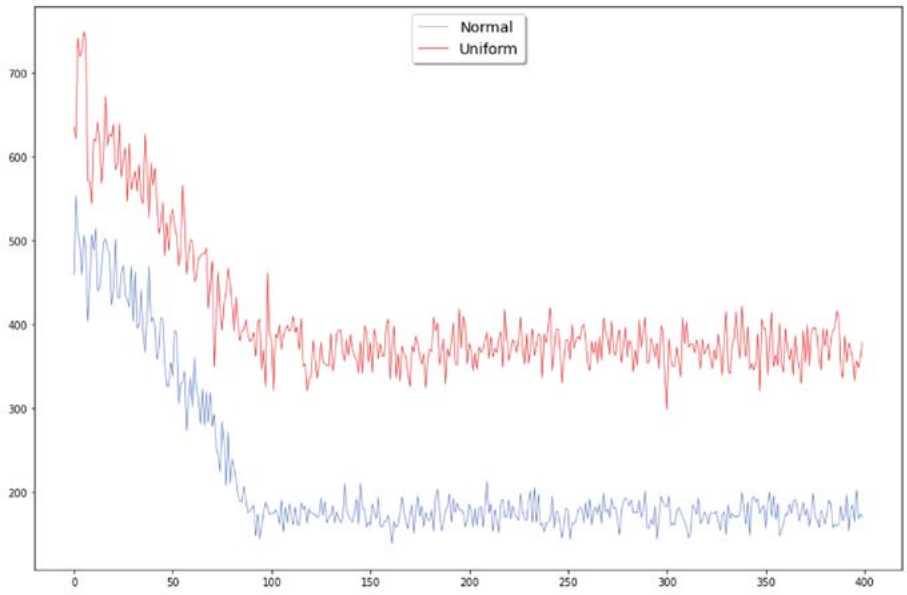


Figure 2. Results from the Double DQN method. Blue line: normal distribution; red line: uniform distribution.

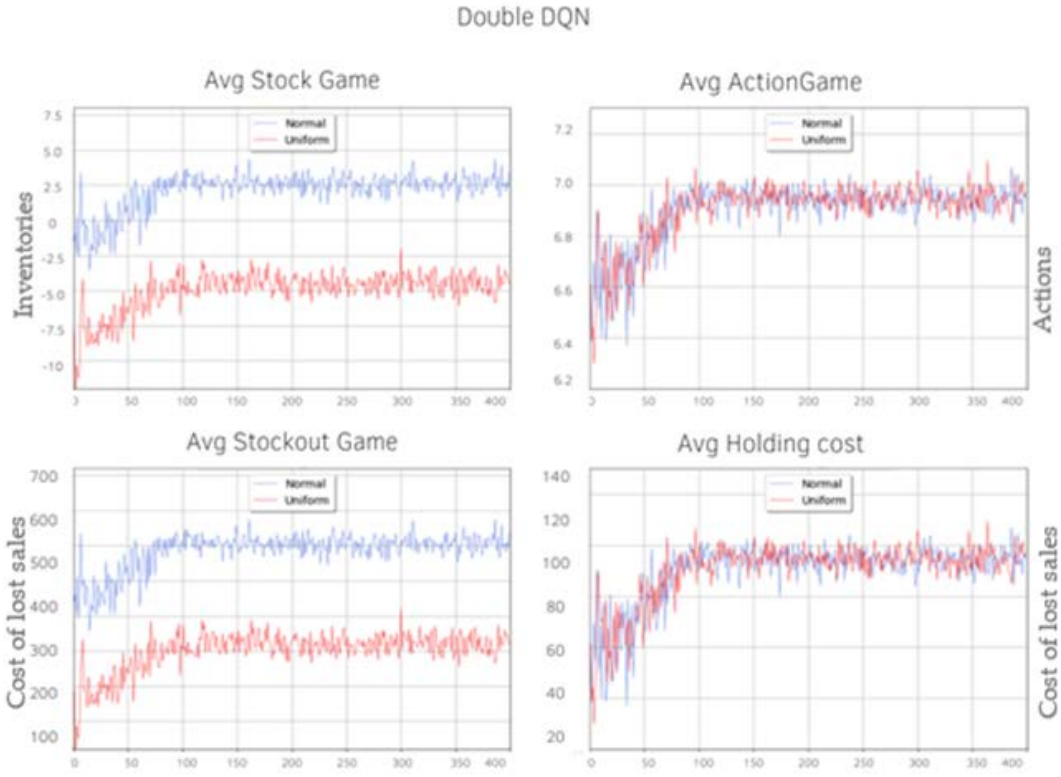


Figure 3. Evolution of the state variables during the training.

Dueling DQN

Figure 4c shows the results of the Dueling DQN method. With blue line: normal distribution; with red line: uniform distribution; and Figure 4a, 4b, 4c, and 4d shows the training results using the Dueling DQN method. Figure 5 shows how, after 30,000

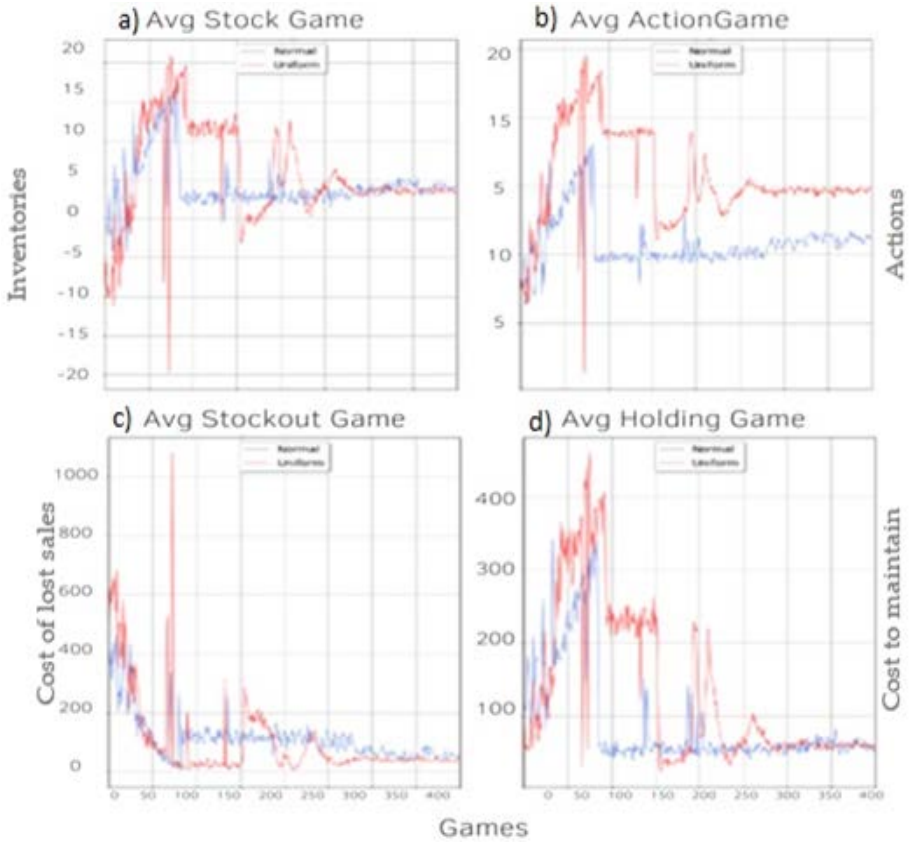


Figure 4. Evolution of the state variables during training.

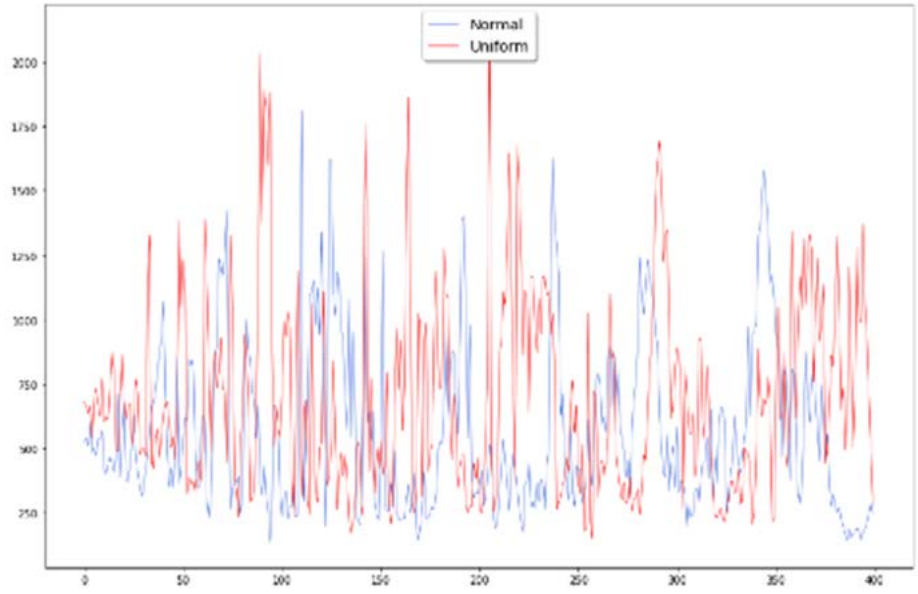


Figure 5. Results from the Dueling Double DQN method. Blue line: normal distribution; red line: uniform distribution.

episodes, the algorithm attains stability in learning and reaches the minimum values of the total cost. It is important to mention that this method achieved a lower total cost than the DQN method for both types of demand distribution. Although this method took more episodes to converge, after completing the 40,000 training episodes, the agent learned a better policy for both types of demand. The policy implies maintaining a minimum level of inventory for both behaviors of demand, to minimize the cost from lost sales. However, if the distribution is uniform, the average level of actions is higher than in a normal distribution to compensate the peaks.

Dueling Double DQN

The Dueling Double DQN method did not attain convergence in any type of demand, as can be seen in Figure 5. The divergence takes place during the training process because it avoids excessive responses, and the separate evaluation of state-action pairs.

The summary of the results obtained from the training of each method is presented in Table 2. These results were calculated based on the stability in learning (the last 10,000 episodes). As can be seen, the means of DQN and DDQN were very similar, which agrees with the previously applied Wilcoxon test [27]. To validate whether the yield of the Dueling DQN algorithm is better than the DQN algorithm, a non-parametric test was conducted to compare the mean range of two related samples and to determine if there are differences between them. The p values were 5.83e−38 and 0.000 for the Normal and Uniform distributions, respectively. Therefore, the conclusion can be reached that the Dueling DQN is statistically better than DQN.

Although the convergence of the DQN algorithm is more stable than the Dueling DQN algorithm, the final policy obtained by Dueling DQN attained a better total cost. In practice, a more efficient method has the objective of reducing storage costs by minimizing the level of inventory without incurring in a high level of lost sales [12].

Among all the hyperparameters, it is interesting to mention that the discount factor had a significant impact on the learning process, since it is not possible to evaluate the long-term reward in a supply chain. The results reached by the Dueling DQN algorithm were lower than those of other methods. In addition, the behavior of the demand was not significant due to the robustness of the method. Therefore, in practice, the use of the Dueling DQN algorithm to determine the size of the order allows reducing inventory costs and lost sales.

Table 2. Scores of reinforcements learning methods.

Method	Score with DN (mean)	Score with DU (mean)
DQN with repetition of experience	173.96	371.53
Double DQN	173.94	371.61
Dueling	151.27	155.30
Double Dueling DQN	571.39	682.39

CONCLUSIONS

Four reinforcement learning (RL) methods were compared to solve the problem of determining the optimal size of the purchase order that a farmer must meet to minimize the total cost of the supply chain. The algorithm with the best results is the Dueling DQN, with an average total cost of 151.27 units for the demand with a normal distribution and an average of 155.3 units for the demand with a uniform distribution. This method presents lower variability once convergence is reached. The policy proposed by the Dueling DQN is applied to the case study, since a perishable product must maintain a minimal inventory to avoid the risk of expired products. As future work, this method will be applied to the entire supply chain, including the retailer, in an environment of multiple products.

REFERENCES

1. Avances en informática y ciencia de datos. Springer Berlín Heidelberg. 2018
2. Chaharsooghi, SK, Heydari, J. y Zegordi, SH. Un modelo de aprendizaje reforzado para la gestión de pedidos de la cadena de suministro: una aplicación al juego de la cerveza. *Decision Support Systems, 45*(4), 949-959. https://doi.org/10.1016/j.dss.2008.03.007. 2008
3. Chen, F. y Zheng, Y.-S. Límites inferiores para sistemas de inventario estocástico de varios niveles. *Ciencias de la administración, 40*(11), 1426-1443. https://doi.org/10.1287/mnsc.40.11.1426. 1994.
4. Clark, AJ y Bufanda, H. Políticas óptimas para un problema de inventario de varios niveles. *Management Science, 50*(12_suplemento), 1782-1790. https://doi.org/10.1287/mnsc.1040.0265. 2004
5. Gallego, G. y Zipkin, P. Posicionamiento de Stock y Estimación de Rendimiento en Sistemas de Producción-Transporte en Serie. *Gestión de operaciones de fabricación y servicios, 1*(1), 77-88. https://doi.org/10.1287/msom.1.1.77. 1999.
6. Giannoccaro, I. y Pontrandolfo, P. Gestión de inventario en las cadenas de suministro: un enfoque de aprendizaje por refuerzo. *Revista Internacional de Economía de la Producción, 78*(2), 153-161. https://doi.org/10.1016/S0925-5273(00)00156-0. 2002.
7. Gijsbrechts, J., Boute, RN, Van Mieghem, JA y Zhang, D. ¿Puede el aprendizaje de refuerzo profundo mejorar la gestión de inventario? Desempeño e implementación de problemas de modo dual de abastecimiento. Revista Electrónica SSRN. 2018.
8. Hasselt, HV. Doble Q-learning. In Advances in Neural Information Processing Systems (NeurIPS), 2613-2621. 2010
9. Jiang, C. y Sheng, Z. Aprendizaje de refuerzo basado en casos para el control de inventario dinámico en un sistema de cadena de suministro de múltiples agentes. *Sistemas expertos con aplicaciones, 36*(3), 6520-6526. 2009.
10. Kwon, I., Kim, C., Jun, J. y Lee, J. Aprendizaje de refuerzo miope basado en casos para satisfacer el nivel de servicio objetivo en la cadena de suministro. *Sistemas expertos con aplicaciones, 35*(1-2), 389-397. 2008.
11. Li, D., Fast-Berglund, Å. y Paulin, D. Capacidades actuales y futuras de la Industria 4.0 para el intercambio de información y conocimientos: caso de dos pymes suecas. *Revista internacional de tecnología de fabricación avanzada, 105*(9), 3951-3963. 2019.
12. Marún, EA. Desafíos en la previsión de la demanda de productos inciertos en la cadena de suministro: una revisión sistemática de la literatura. . . Tecnología, 29. 2018.
13. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, AA, Veness, J., Bellemare, MG, Graves, A., Riedmiller, M., Fidjeland, AK, Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. y Hassabis, D. Control a nivel humano a través del aprendizaje de refuerzo profundo. *Naturaleza, 518*(7540), 529-533. 2015
14. Musavi, M. y Bozorgi-Amiri, A. Un problema de programación de ubicación de centro sostenible de objetivos múltiples para la cadena de suministro de alimentos perecederos. *Informática e ingeniería industrial*, 113, 766-778. 2017.
15. Ni, D., Xiao, Z. y Lim, MK. Una revisión sistemática de las tendencias de investigación del aprendizaje automático en la gestión de la cadena de suministro. *Revista internacional de aprendizaje automático y cibernética*. 2019.
16. Oroojlooy, A., Nazari, M., Snyder, L. y Takac, M. Una red Q profunda para el juego de la cerveza: uso del aprendizaje automático para resolver problemas de optimización de inventario. 2017.

17. Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J. y Harnisch, M. Industria 4.0: El futuro de la productividad y el crecimiento en las industrias manufactureras. 14. 2015.
18. Schuh, G., Anderl, R., Gausemeier, J. y Wahlster, W. Índice de madurez de Industria 4.0.60. 2018
19. Schwartz, JD y Rivera, DE. Un enfoque de control de procesos para la gestión táctica de inventario en sistemas de producción-inventario. *Revista Internacional de Economía de la Producción, 125*(1), 111-124. 2010.
20. Shaikh, AA, Das, SC, Bhunia, AK, Panda, GC y Al-Amin Khan, Md. Un modelo EOQ de dos almacenes con costo de inventario valorado por intervalos y pago anticipado por artículo deteriorado bajo optimización de enjambre de partículas. *Informática blanda, 23*(24), 13531-13546. 2019.
21. Stockheim, T., Schwind, M. y Koenig, W. Un enfoque de aprendizaje por refuerzo para la gestión de la cadena de suministro, 14. 2003
22. Sui, Z., Gosavi, A. y Lin, L. Un enfoque de aprendizaje por refuerzo para el reabastecimiento de inventario en sistemas de inventario administrados por proveedores con inventario en consignación. *Revista de gestión de ingeniería, 22*(4), 44-53. 2010.
23. Sutton, RS y Barto, AG. Aprendizaje por refuerzo: una introducción. 352. 1992.
24. Van Hasselt, H., Guez, A. y Silver, D. Aprendizaje de refuerzo profundo con doble Q-learning. 14. 2015
25. Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M. y de Freitas, N. Duelo de arquitecturas de red para el aprendizaje por refuerzo profundo. 2016.
26. Weng, T., Liu, W. y Xiao, J. Pronóstico de ventas de la cadena de suministro basado en el modelo combinado lightGBM y LSTM. *Gestión industrial y sistemas de datos, 120*(2), 265-279. 2019.
27. Zheng, W., Lei, Y. y Chang, Q. Estudio comparativo de dos políticas de control en tiempo real basadas en aprendizaje de refuerzo para un sistema de producción de dos máquinas y un búfer. 13.ª Conferencia IEEE sobre ciencia e ingeniería de automatización (CASE) 2017.