

PAPER

Sistema Multi-agente como Tomador de Decisiones sobre la Planeación de una Línea de Producción con Variabilidad Realista

Jose Castro,^{1, *} David Jiménez,^{2, *} Rachel Cisneros^{3, *} and Natalia Molina^{4, *}

^{*}Escuela de Ingeniería Industrial, Universidad de Costa Rica, Av. 7, 11501, San Jose, Costa Rica

¹Corresponding author. email:cm.jose18@gmail.com²Corresponding author. david@ucr.ac.cr³Corresponding author. reysheill.cisneros@ucr.ac.cr⁴Corresponding author. natalia.molinaquesada@ucr.ac.cr

Abstract

El artículo tiene como objetivo realizar una prueba sobre la capacidad de los sistemas multi-agente de adoptar un rol de tomadoras de decisiones, trascendiendo su actual rol de apoyo para un humano tomador de decisiones, asumiendo riesgos en un marco realista de una línea de producción y simulando la característica humana del apetito de riesgo.

Key words: Prueba, DMS, DSS, Sistema Multi-agente, Decisión Cognitiva, Alimentación de Proceso, Riesgo Asociado.

Introducción

La presente investigación tiene como objetivo fundamental realizar un primer acercamiento al uso de los sistemas multi-agente no sólo como un sistema de soporte a las decisiones como tradicionalmente se ha realizado, sino como un sistema de toma de decisiones capaz de recibir información sobre las variables involucradas en un proceso de producción, determinar el resultado con las decisiones tomadas, y analizar el mismo.

La metodología empleada en el trabajo consiste en un sistema multi-agente entrenado por medio de aprendizaje por reforzamiento o RL, específicamente mediante el método DDQN. Esta metodología de trabajo consiste en alterar el apetito de riesgo del sistema creado mediante los puntajes asignados a las decisiones por tomar, lo cual permitirá desarrollar un algoritmo autónomo para un sistema de producción DBR.

La cantidad de investigaciones sobre este tópico son reducidas, lo cual demuestra que es un tema poco explorado para su uso en las organizaciones. Sin embargo, podría resultar en múltiples beneficios si se es aplicado en las empresas, sobretodo en un contexto industrial como el de hoy en día con el creciente auge de la industria 4.0 y la automatización de procesos. Esta razón impulsa el sujeto de investigación del presente artículo, para así demostrar los beneficios de seguir explorando aplicaciones como esta.

Estado del Arte

La inteligencia artificial puede ser definida como una herramienta de toma de decisiones. Este tipo de inteligencia es demostrada por máquinas o softwares que comprenden su entorno y con base en esa data, toman decisiones a través de

una red neurológica artificial. La principal característica y a su vez, el principal reto de este tipo de inteligencias es simular el razonamiento humano al tomar decisiones con base en lo percibido. En el ser humano la cognición se relaciona con el pensamiento y la comprensión; es la expresión de cómo funciona la mente humana. Este es el resultado de una serie de procesos: percepción, atención, pensamiento, memoria y lenguaje; es un proceso humano de carácter psicológico que tiene lugar a nivel individual.[1■]

Esta simulación se logra al desarrollar sistemas complejos que realicen dichas tareas recolectando y analizando esta data utilizando algoritmos predefinidos, generando información útil, y luego utilizando esta información en la toma de decisiones.[2■][3■].

Esta toma de decisiones puede ser generada por agentes artificiales de igual manera, si las decisiones son tomadas por un conjunto de agentes se le denomina sistema multi-agente. Los sistemas multiagente son organizaciones distribuidas donde los componentes o agentes, colaboran para lograr una serie de metas. Los agentes son entidades autónomas que pueden ser caracterizadas por sus capacidades las cuales pueden incluir autonomía, proactividad, reactividad, organización, entre otras.

Estas características hacen de los agentes apropiados para ser aplicados en escenarios de inteligencia empresarial y gestión del riesgo. Los agentes pueden presentar diferentes roles y establecer un modelo organizacional donde se emulan los comportamientos humanos y los procesos de gestión.

Los agentes pueden ser entrenados para especializarse en funciones específicas y trabajar en conjunto para incrementar la productividad y eficiencia del sistema. Estos agentes pueden funcionar como un interfaz entre los usuarios humanos y los sistemas, o bien pueden trabajar independientemente en la

toma de decisiones, este es el camino a seguir para el presente artículo. [4■]

La etapa de entrenamiento de los agentes será crucial para una toma de decisiones exitosa y fundamentada. Existen diferentes formas de crear este aprendizaje, la utilizada en el presente artículo corresponde al aprendizaje por refuerzo. El aprendizaje por refuerzo, o RL por sus siglas en inglés (Reinforcement Learning), tiene su origen en los primeros días de la cibernética y trabajo en estadística, psicología, neurociencia, y la informática. En este tipo de aprendizaje se definen las decisiones a tomar al maximizar la premiación numérica. De esta manera, a la persona que aprende no se le indica qué acciones tomar, sino que debe intentar descubrir la combinación de variables, es decir, la decisión que indique la mayor cantidad de premiación.

Existen diferentes maneras de realizar este tipo de aprendizaje, en algunos casos, las acciones tomadas no afectan sólo la recompensa sino las situaciones subsecuentes. El proceso detrás de este tipo de aprendizaje se visualiza en la siguiente imagen [5■]

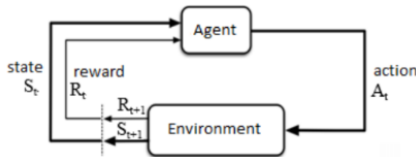


Fig. 1. Diagrama aprendizaje por refuerzo.

Deep Q-Learning (DQN), es uno de los métodos paramétricos más populares, es una adaptación de Q-Learning que utiliza una red neuronal profunda Deep Neural Network, DNN) como modelo de aproximación a las funciones estado-acción. DDQN tiende a ser más escéptico con las acciones elegidas, es decir, a la hora de calcular el valor de Q objetivo para realizar la acción en cuestión. De forma general, DDQN ayuda a reducir la sobreestimación de los valores q, lo que permite minimizar el tiempo de entrenamiento y tener un aprendizaje con mayor porcentaje de estabilidad. [6■] [7■]

La inteligencia artificial y los sistemas multi-agente normalmente se utiliza como DSS, o sistema de soporte a las decisiones, por sus siglas en inglés. Un DSS se define como un sistema interactivo basado en computador que da soporte a los tomadores de decisiones en vez de reemplazarlos, utiliza datos y modelos para resolver problemas con diferentes grados de estructura y se enfoca más en la efectividad que en la eficiencia, el cual es un aspecto clave para el desarrollo de la propuesta del presente artículo. Mientras que por el contrario, un DMS, o sistema de toma de decisiones, por sus siglas en inglés, corresponde a un sistema capaz de resolver estos problemas pero adicionalmente, posee la capacidad de tomar decisiones con base en los resultados. [8■]

Existen diferentes aspectos que los agentes deben tomar en cuenta en la toma de decisiones, en primer lugar, el tipo de alimentación de la línea de producción. Existen dos tipos de alimentación para una línea de producción: tipo push y tipo pull. Los sistemas push, empujar en inglés, implican que el material es movido a través del sistema de producción pasando de un proceso a otro, sin importar lo que esté sucediendo adelante de él en el sistema de producción, de ahí su nombre, el material es empujado en el sistema. Es importante mencionar, que este tipo de sistema se caracteriza porque la orden es

procesada lo más antes posible y al máximo de capacidad, además de que se controla el throughput y se observa el WIP. [9■]

Por otro lado, los sistemas de tipo pull, jalar en inglés, manejan su producción según la demanda, no empujan la producción hacia el cliente, como sucede con los sistemas push. En estos sistemas la orden se procesa con la capacidad mínima requerida, se controla el WIP y se observa el throughput. Estos sistemas generan productos de alta calidad ya que se intenta utilizar las cantidades necesarias de todas las variables con el fin de evitar el desperdicio y los altos costos. [10■]

El WIP se refiere al producto semi terminado, o por sus siglas en inglés, Work In Progress. El procesar materia prima en producto terminado implica que este producto vaya a través de varios procesos, donde el producto de salida se denomina WIP. El WIP está presente en todo proceso de manufactura que produce un producto terminado, es necesario mantener un seguimiento del mismo, ya que permite tener una idea de la eficiencia del proceso. [11■]

Otro aspecto importante está relacionado con la secuenciación de operaciones y el manejo de pedidos urgentes. Existen diferentes reglas de despacho, los cuales se rigen por diferentes aspectos, y el comportamiento de ciertas variables. Primero está el modelo PEPS (Primero en entrar, primero en salir), donde se asume que los primeros costos que entran al inventario son los primeros que salen al costo de mercadería vendida, es decir, lo primero que se compra es lo primero en venderse; por otro lado, se tiene un método contrario denominado UEPS, este funciona de manera diferente, donde los últimos costos que entran al inventario, son los primeros que salen al costo de mercadería vendida. Sin embargo, este último método presenta desventajas como sobrevalorar el inventario restante, pero sobre todo, las Normas Internacionales de Información Financiera ya no autorizan este método para que sea usado en las empresas.

La siguiente regla de despacho corresponde a la mínima fecha de entrega, MINDD por sus siglas en inglés, donde aquel pedido con la fecha de entrega más temprana se procesa primero, en este caso se podría inferir que aquellos pedidos urgentes tienen prioridad. Seguidamente, fecha mínima planeada de inicio, MINSI por sus siglas en inglés, para esta regla la empresa debería contar con un sistema de planeación de actividades, para finalmente elegir aquella que tenga la fecha más temprana. La siguiente regla corresponde a tiempo mínimo de procesamiento, o MINPRT también por sus siglas en inglés, donde aquella que tenga la menor duración en la máquina, va en primer lugar, MINSOP o mínimo tiempo de holgura se refiere a aquella operación que tenga la menor diferencia entre tiempo de procesamiento y el tiempo disponible a la fecha de entrega, LSU o alistamiento mínimo como su nombre lo indica, se rige por el producto cuyo alistamiento tenga el menor tiempo, y finalmente, RANDOM simplemente se refiere a elegir al azar el pedido a procesar. [9■] [12■]

Un aspecto importante de la producción son las tandas de transferencia, estas se refieren a la cantidad de unidades que representan el bloque de traslado de material entre operaciones en el proceso de producción, y la definición del número correcto de tanda de transferencia depende de las variables involucradas en el proceso, por esta razón es buena idea determinar estos criterios, los cuales pueden ser: tiempo de producción, tiempo de documentación para trazabilidad, entre otros. Es sumamente importante elegir una tanda de transferencia adecuada ya que esta tendrá un efecto sobre los niveles de inventario, el tiempo de ciclo, throughput, calidad, entre otros. Para definir la tanda

de transferencia se puede intentar variar en pocas cantidades la tanda de transferencia utilizada actualmente y vigilar el comportamiento de las variables involucradas, para determinar el impacto en la producción de acuerdo con el objetivo de la empresa. [13■]

Propuesta

Los sistemas de planeación de producción emplean las inteligencias artificiales como sistemas de soporte a las decisiones (DSS), al procesar y resumir cantidades grandes de información, o generando escenarios bases donde los humanos son los encargados de tomar las decisiones con base en la respuesta de la inteligencia artificial. Otros estudios mencionan la capacidad de la inteligencia artificial, en particular los sistemas multi-agente, al tomar decisiones cognitivas y/o asumir riesgos, sin embargo, estas aplicaciones se realizan en entornos controlados o condiciones simplificadas a la realidad.

La propuesta para la investigación se basa en la ejecución de una prueba que coloque al sistema multi-agente en un escenario cercano a la realidad industrial, donde resolverá un problema de producción tomando cierta decisión después de haber sido entrenada, y analizará el resultado después de haber tomado la decisión.

Esta prueba estará relacionada con la aplicación de un MAS sobre un problema de alimentación de línea bajo el modelo DBR, donde el sistema, apoyado en los algoritmos tradicionales donde es utilizada como sistema de soporte a las decisiones, deberá tomar decisiones cognitivas con un riesgo asociado. Este será entrenado bajo un modelo de aprendizaje por refuerzo de Q-Learning donde las recompensas por los resultados de sus decisiones se variarán en dos niveles por cada recompensa. Esto con el fin de alterar el apetito de riesgo del sistema en sus decisiones, esto ya que en la realidad de una empresa, las decisiones pueden tomarse desde un enfoque más arriesgado o conservador.

El problema a utilizar para el sistema multi-agente contemplará un sistema DBR que cuenta con un número de cinco máquinas, donde solamente una de ellas tendrá scrap y reproceso, y el cuello de botella se encontrará en la máquina 3.

Las variables de entrada para el sistema serán el tamaño de lote, reprocesos, velocidad de máquinas con un cuello de botella estático, y órdenes. Las variables de decisión para la inteligencia artificial creada serán: completar lotes, ingresar órdenes con alta prioridad, tomar en cuenta el buffer, y el drum.

Metodología

La inteligencia artificial se sustenta en algoritmos inteligentes o en algoritmos de aprendizaje que, entre muchos otros fines, se utilizan para identificar tendencias económicas, predecir nuestros comportamientos digitales, etc.

Un algoritmo se puede definir como un conjunto preciso de instrucciones o reglas, o como una serie metódica de pasos que puede utilizarse para hacer cálculos, resolver problemas y tomar decisiones. El algoritmo es la fórmula que se emplea para hacer un cálculo.

Para la base de datos utilizada con el fin de entrenar al sistema multi-agente se crearon distintos códigos. Para el primer código creado, se definieron los valores de las variables con funciones aleatorias, el tamaño de lote con números entre 300 y 600, los porcentajes de reproceso se definieron de la misma manera con números aleatorios menores a 0.25, el porcentaje

de scrap menor a 0.10. Con el fin de definir las velocidades de procesamiento de las máquinas se generaron números aleatorios entre 60 y 200, definiendo que el número menor correspondería a la velocidad en el cuello de botella. Adicionalmente, se definen 5 tamaños de órdenes con números aleatorios entre 1200 a 4500 donde el orden de los mismos se encuentra definido según el método de Gupta. No siempre existirán órdenes de emergencia por lo que la cantidad de órdenes de emergencia dependerá de otro número aleatorio entre 0 y 2, donde el tamaño de estas órdenes igual estará definido de esta manera, con valores entre 600 y 1100. Se cuenta con DSS como Gupta antes mencionado y Markov para los reprocesos presentes en máquina 2.

Posteriormente, para el siguiente código se realizó mediante OOP, programación orientada a objetos, por sus siglas en inglés. Se define una primera clase en referencia a la máquina 1, la necesidad de las máquinas está definida por la cantidad de WIP por las máquinas, la velocidad está definida por el código anterior, y en este segundo código se define una velocidad acumulada donde la velocidad acumulada de la máquina 1 será la velocidad de la máquina 1, la velocidad acumulada de la máquina 2 será la velocidad de la máquina 1 más la velocidad de la máquina 2, y de la misma manera para las siguientes. Las máquinas 4 y 5 funcionan de la misma manera que la máquina 1 en el código.

La máquina 1 es la encargada de gestionar las órdenes entrantes, y permite la entrada de las mismas dependiendo del estado del drum y del buffer.

La máquina 2 corresponde a aquella que cuenta con scrap y reproceso, cuyos porcentajes se definen en el primer código, se genera también en este segundo código la cantidad de unidades procesadas tomando en cuenta el scrap. El reproceso generado en la máquina 2 se dirige hacia la máquina 1.

Para la máquina 3, se define un drum y un buffer donde cada vez que se llega al buffer se detienen las dos máquinas y cada vez que se llega al drum se reactiva el proceso, con el fin de que la máquina que representa el cuello de botella pueda funcionar sin ningún problema. El funcionamiento del buffer se define mediante un parámetro booleano que activa la señal cuando sea necesario. Se devuelve la señal a la máquina 1 para la toma de decisiones.

Se define además una función que permite calcular el WIP en distintos momentos del proceso para permitir trazabilidad y vigilancia del mismo. Se obtiene además la cantidad de WIP y la velocidad para cada máquina, la velocidad acumulada, la cantidad de unidades de scrap, y la cantidad de unidades con reproceso. Cabe destacar que, la cantidad de unidades con scrap en caso de corresponder a un número real, el valor se redondeará a la unidad superior más cercana.

En cuanto al ecosistema, las decisiones tomadas por el sistema alteran el ecosistema o el problema. Seguidamente, en cuanto a los agentes utilizados, se utilizarán 5 agentes, cada uno con funciones distintas: completar lotes, ingresar órdenes urgentes 1, ingresar órdenes urgentes 2, establecer buffer, y establecer drum, respectivamente. Existen entradas que definen la información que cada agente puede ver, estas variables corresponden a: tamaño de lote de transferencia, total de tamaño de órdenes, las velocidades de las máquinas, la cantidad de reproceso, la cantidad de scrap, y las órdenes de emergencia 1 y 2.

En cuanto a la inicialización de la simulación, la máquina 1 procesa las órdenes y envía la señal a la máquina 2 de cuando se han procesado, seguidamente la máquina 2 comienza a procesar las órdenes junto con el scrap y el reproceso además de indicar a la máquina 1 sobre este reproceso.

Un aspecto clave del sistema multi-agente creada corresponde a la sección de recompensas ya que en ese momento se establece el apetito de riesgo y la personalidad de la misma. Se realiza además, una comparación entre los resultados y la predicción del sistema y se generan las recompensas, mediante estas recompensas y la variación de los números, se modifica la personalidad de la inteligencia.

El entrenamiento de la misma se realizó mediante DDQN, donde se establece que en base a una política, en este caso, una política colaborativa, en la cual el objetivo de los agentes es obtener la mejor puntuación para llegar a una solución óptima al problema presentado.

Se utiliza además el método PER o Prioritized Experience Replay, En este método se pretende mejorar la etapa de entrenamiento ya que se da prioridad a las situaciones donde existe una brecha más grande entre la predicción de la inteligencia artificial y el resultado como tal, de tal manera que se almacenan o se recuerdan más las situaciones en las que se obtuvieron mejores recompensas. [14■]

Finalmente, se realiza una corrida con el fin de controlar el funcionamiento de la inteligencia y se realiza la evaluación de los resultados obtenidos.

Análisis

En lo que respecta al análisis de los datos recopilados del entrenamiento de la inteligencia, se recopilaron los puntajes obtenidos dadas las circunstancias de cada simulación de línea de producción y las decocciones tomadas por el sistema multi-agente en respuesta a éstas. Estos resultados se grafican para el número de iteraciones en la gráfica 2. Lo que se espera observar típicamente en una gráfica de este estilo es un comportamiento logarítmico del resultado, indicando que el sistema logró encontrar un modelo que se adapte al tipo de problema que se presenta. A continuación, se presenta la gráfica y su comportamiento con el puntaje de aprendizaje a lo largo del tiempo.

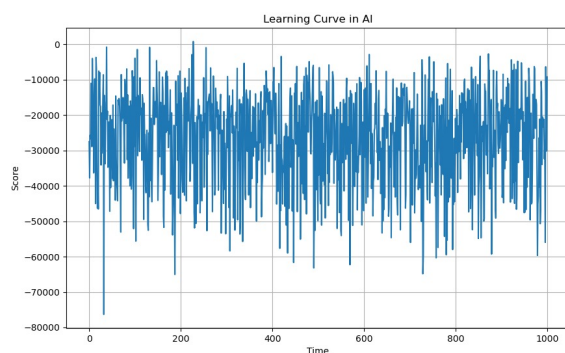


Fig. 2. Gráfica de Aprendizaje del Sistema Multi-agente.

A cómo se explicó previamente del código obtiene la gráfica de aprendizaje de la 2 en la cual se denotan los ejes "x" y "y" con los nombres "score" y "tiempo" respectivamente, en el tiempo se puede observar la cantidad de iteraciones realizadas por la AI a través de su curva de aprendizaje, esta debe de ir demostrando un comportamiento creciente a lo largo del tiempo conforme el objetivo final del sistema multiagente se

basa en aprender poco a poco de los errores que la misma va cometiendo, esta inicia por valores cercanos a los -27 000 puntos denotando que la misma inicia fallando y debe de ir buscando aquellas configuraciones pertinentes para avanzar cada vez más en el total de "score"; se debe de aclarar que además y el otro parámetro de análisis es este último mencionado en donde se obtiene un puntaje acumulado por cada decisión correcta o incorrecta y proviene del nombre en inglés "puntaje" y se asigna según los valores que se consideraron ideales en el desarrollo del sistema multiagente, en esta se obtienen valores que multiplicados por 2000 se acumulan (como ejemplo de funcionamiento) para decisiones de completar lotes de producción o inclusive puntajes negativos como scrap en el sistema, ordenes no completadas u otros indicadores que puedan o no llamar la atención a través del proceso. Es importante recalcar que el sistema multiagente recibe datos aleatorios siempre, por lo que su curva de aprendizaje no va a responder a una distribución de tipo normal si no que esta va a responder a parámetros que cada vez más se van a ir alejando de los valores más negativos de los que parte en un inicio. En cuanto a los resultados propios del proceso, este finaliza con valores de puntaje cercanos a -7000 puntos por lo que se considera alentador el uso de un sistema multiagente en este contexto, una perspectiva real y más acercada a la cotidianidad de lo que se vive en una empresa denota que el modelo puede llegar a utilizarse en contextos actuales y cambiantes en donde los sistemas multiagente denotan el futuro de la civilización para temas de planeamientos de producción y sus áreas relacionadas. La creciente gráfica de aprendizaje demuestra no solo un funcionamiento ideal del código sino también además una capacidad de aprendizaje que puede utilizarse en procesos de producción rápidamente cambiantes y que adoptan modelos de mercado que responden a constantes pedidos de emergencia, o lotes de producción que pueden o no ser completados. Se demuestra además que las mejoras por más de que son levemente notorias en el sistema multiagente debe de existir una mayor capacidad de almacenamiento, así como mejores algoritmos de análisis del sistema multiagente para que el mismo pueda llegar a usarse de manera comercial o en un ambiente de producción.

Conclusiones

El uso de este tipo de sistemas no sólo como un soporte a las decisiones sino como un sistema de toma de decisiones es un aspecto todavía por ser aplicado en las industrias de la actualidad y que traería consigo múltiples beneficios para las mismas. La investigación pretende dar un primer paso a la innovadora utilización de la misma en los procesos industriales, para así lograr aún más automatización y mejorar la capacidad de respuesta de los sistemas actuales.

El rápido movimiento de la industria 4.0 ha generado la necesidad de adaptarse a los cambios tecnológicos que trae consigo y a comenzar la búsqueda de nuevas técnicas que den como resultado procesos más eficientes, y decisiones más exactas en tiempos más cortos.

La aplicación de estas nuevas tecnologías puede permitir la reducción de tiempos, y la reducción de mano de obra dedicada a esta tarea específica, lo cual abre la posibilidad de asignar recursos al sector de investigación.

En las situaciones reales de producción se presentan una gran cantidad de variables que de alguna u otra manera modifican los procesos y sus resultados, por esta razón se mostró en la

investigación realizada el potencial presentado por un sistema como este de actuar dentro de un entorno donde las variables son asignadas aleatoriamente.

Existen muchos factores a tomar en cuenta y es todo un reto presentar una aplicación de los algoritmos fuera del entorno en el que han sido utilizados comúnmente sin embargo, el resultado es exitoso y el futuro es prometedor. Se ha logrado modelar un sistema multi-agente que se asemeja a un contexto real dentro de una empresa, al adaptarse a la simulación del comportamiento de una línea de producción, se demostró el correcto funcionamiento del modelo multi-agente por medio de los recursos planteados, sin embargo, aún falta información e investigación en sistemas de aprendizaje multi-agente para poder llegar a soluciones óptimas con mejores algoritmos, por lo que se abren las puertas para más investigación en sistemas de aprendizaje para lograr obtener soluciones o decir, un sistema multi-agente que pueda ser de mayor utilidad a nivel industrial o comercial.

Con la creación de este tipo de algoritmos se pretende mostrar un campo abierto de investigación y poco explorado por el momento pero que sin duda, será de gran valor en el futuro al enfrentarnos a una nueva era de la tecnología, donde es vital que las compañías logren adaptar sus sistemas para demostrar tener buena competitividad y relevancia en el mercado. al mismo tiempo que logran una mejor comunicación con más personas en el mundo.

References

- [1■]a. Barón, “Los sesgos cognitivos: De la psicología cognitiva a la perspectiva cognitiva de la organización y su relación con los procesos de toma de decisiones gerenciales” 2018.
- [2■]Slaveykova, “Basic guidelines in artificial intelligence (ai) applications in erp systems in the last 5 years” s.f.
- [3■]Hrishev and Shakev, “Artificial intelligence in erp systems” s.f.
- [4■]Bajo, Borrajo, D Paz, Corchado, and Pellicer, “A multi-agent system for web-based risk management in small and medium business,” *ELSEVIER* 2012.
- [5■]Fonseca-Reyna, Martínez-Jimenez, Verdecia, and Rodríguez, “Optimization of heavily constrained hybrid-flexible flowshop problems using a multi-agent reinforcement learning approach,” *Investigacion Operacional* vol. 40, no. 1, pp. 100–111, 2019.
- [6■]M M and M R “A gently but rigorous introduction to reinforcement learning,” *Centro de Investigación en Computación del IPN* 2023.
- [7■]V del Moral, “Algoritmo deep q learning para el aprendizaje por refuerzo de una estrategia de conducción en 2d,” *Universidad Publica de Navarra* p. 51, 2021.
- [8■]Ruíz, Hernández, and Giraldo, “Aplicación de los sistemas de soporte a la decisión (dss) en el comercio electrónico,” *Revista Ingeniería e Investigación* vol. 29, no. 2, pp. 94–99, 2009.
- [9■]Castrillo, “Clase 9. secuenciación ve,” *II-0703 Ingeniería de Operaciones. Mediación Virtual* 2023.
- [10■]Vargas-Sánchez, Jiménez-García, Toro-Galvis, and Rodríguez-García, “Comparación por simulación de sistemas de manufactura tipo push y pull,” *Ciencia e Ingeniería Neogranadina* vol. 29, no. 1, pp. 81–94, 2019. DOI: 10.18359/rcin.3075.
- [11■]H Wai Tan Rijal Jamaludin, “Work-in-progress inventory control case study in lean management,” *International Journal of Engineering Technology* vol. 7, no. 3.4, pp. 181–187, 2018.
- [12■]Trujillo, “Sistema para el control de inventarios en la empresa “inversiones novillo de oro s.a.s”,” *Ingeniería, Matemáticas, y Ciencias de la Información* vol. 7, no. 14, pp. 105–116, 2020. [Online]. Available: <http://dx.doi.org/10.21017/rimci.2020.v7.n14.a87>.
- [13■]A Godínez Rojas, “Desarrollo de una propuesta de programación de la producción para una línea de alta mezcla y bajo volumen en cer 4 de la empresa healthex,” *SIBDI* 2013.
- [14■]Mohammad, “Deep q learning (dqn) for multi-agent reinforcement learning,” *Github* 2018.

Anexos

Github privado bajo el nombre :
InteligenciaArtificialTomadoradeDecisionesUCR

Sobre los Autores



José Pablo Castro Martínez. Nacido en Costa Rica. Estudiante de licenciatura en Ingeniería Industrial en la Universidad de Costa Rica. Integrante del Grupo de Ingeniería Aeroespacial de la misma institución. Sus áreas de interés laboral se dirigen a la investigación y desarrollo aeroespacial, producción y logística sostenible y energías renovables



David Jesús Jiménez Marín Nacido en Alajuela, Costa Rica el 29 de junio del 2001. Estudiante de licenciatura en la carrera de Ingeniería Industrial en la Universidad de Costa Rica. Asistente de la escuela de Ingeniería Industrial y del curso de Logística 2 en la Universidad de Costa Rica. Sus áreas de interés laboral se dirigen a la práctica en herramientas logísticas así a como producción sostenible.



Rachel Cisneros Quesada. Nacida en San José, Costa Rica el 11 de febrero del 1999. Estudiante de licenciatura Ingeniería Industrial en la Universidad de Costa Rica. Especialista en Control de Documentación en Abbott Medical. Sus áreas de interés laboral se dirigen a sistemas de calidad.



Natalia Jireth Molina Quesada. Natalia Jireth Molina Quesada. Nacida en San José, Costa Rica el 28 de setiembre del 2001. Graduada del Colegio Madre del Divino Pastor, actualmente cursando la licenciatura de Ingeniería Industrial en la Universidad de Costa Rica. Su área de interés laboral se dirige hacia sistemas automatizados, aseguramiento de la calidad, y logística.