Journal of
Information Systems Engineering
and Business Intelligence

Vol.7, No.2, October 2021
Available online at: http://e-journal.unair.ac.id/index.php/JISEBI

# Reinforcement Learning Approach for Efficient Inventory Policy in Multi-Echelon Supply Chain Under Various Assumptions and Constraints

**Ika Nurkasanah***

*Department of Information Systems, Institut Teknologi Sepuluh Nopember, Indonesia*
*Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya*
ika.nurkasanah@its.ac.id

***Abstract***

**Background:** Inventory policy highly influences Supply Chain Management (SCM) process. Evidence suggests that almost half of SCM costs are set off by stock-related expenses.
**Objective:** This paper aims to minimise total inventory cost in SCM by applying a multi-agent-based machine learning called Reinforcement Learning (RL).
**Methods:** The ability of RL in finding a hidden pattern of inventory policy is run under various constraints which have not been addressed together or simultaneously in previous research. These include capacitated manufacturer and warehouse, limitation of order to suppliers, stochastic demand, lead time uncertainty and multi-sourcing supply. RL was run through Q-Learning with four experiments and 1,000 iterations to examine its result consistency. Then, RL was contrasted to the previous mathematical method to check its efficiency in reducing inventory costs.
**Results:** After 1,000 trial-error simulations, the most striking finding is that RL can perform more efficiently than the mathematical approach by placing optimum order quantities at the right time. In addition, this result was achieved under complex constraints and assumptions which have not been simultaneously simulated in previous studies.
**Conclusion:** Results confirm that the RL approach will be invaluable when implemented to comparable supply network environments expressed in this project. Since RL still leads to higher shortages in this research, combining RL with other machine learning algorithms is suggested to have more robust end-to-end SCM analysis.

**Keywords**: Inventory Policy, Multi-Echelon, Reinforcement Learning, Supply Chain Management, Q-Learning

***Article history:*** Received 30 July 2021, first decision: 18 August 2021, accepted 24 September 2021, available online 28 October 2021

## I. INTRODUCTION

As a vital factor for a business to deliver competitive advantages, Supply Chain Management (SCM) directs the flow of information, products and cash through the entire business process. SCM is exceptionally affected by the functional decision of replenishment or inventory policy made by stakeholders[1][2][3][4]. Various studies revealed that almost half of SCM costs are set off by stock-related expenses [5]. Therefore, an inventory policy has drawn in much interest among researchers in recent decades [6].

Inventory consists of raw materials, components, work-in-process and finished goods that can be managed by considering several assumptions and constraints [7]. Inventory management, the core of SCM, has regularly encountered challenges when deciding on three essential issues: the recurrence of stock status reviews, the time renewal orders are to be placed, and the amounts of reordered items [8]. These choices are impacted by several factors that lead to intricacy; for example, stochastic requirement and lead time, limited storage capacity, and unstable machineability. As a result, inventory management costs arise when overstocking occurs. On the other hand, keeping too little stock will actually increase the possibility of shortages, thereby increasing backorder costs [9]. Inefficient replenishment policy will affect the inventory-related cost; such as 1) holding cost for managing space to store items; 2) ordering cost for some units to the upstream; 3) backorder cost to compute from any bookkeeping or delay costs; and 4) additional warehouse rental cost if internal storage is overcapacity.

In such conditions, past studies [6][10] noticed that operational mathematical methods and models remain popular due to their execution simplicity. Examples of these methods include four inventory control strategies: periodic review reorder quantity model (T, Q), periodic review reorder point-up to level (T, S), continuous order-up to level (s, S), and continuous reorder quantity (s, Q). Also, there is a condition when demand and lead time are uncertain, so reordering becomes more confounded since managers must determine the frequency of inventory status reviews (continuously or periodically) and the ideal quantity of reordered materials [10]. A few numerical

* Corresponding author

ISSN 2443-2555 (online) 2598-6333 (print) 
doi: http://dx.doi.org/10.20473/jisebi.7.2.138-148

controls were utilised to characterise the inventory policy under vulnerability, such as continuous-reorder point-order up to level (s,S), continuous-reorder quantity (s,Q), periodic order up to level (R, S), and the mix of (s,S) + (R,S) called (R,s,S)[8]. Nonetheless, considering the fluctuated conditions from dubious components, these methodologies might become mistaken when characterising an ideal stock approach. In this way, machine learning as an arising business analytics method has offered better capabilities to investigate complicated patterns and new knowledge within an immense amount of electronic information [5]. More precisely descriptive, predictive, and prescriptive analyses can be carried out by machine learning algorithms [9], such as experimental research. Then, it yields lower inventory costs generated by Reinforcement Learning (RL) when contrasted to the one-on-one inventory model[11].

However, RL's implementation in previous studies has not fully reflected the real supply chain in the industry where replenishment policy should deal with complex constraints. For example, RL approach is used for determining ordering policy, but with a relatively simple model that considers only one company in every stage [12]. RL has also optimised replenishment quantity from multi-stages supply chain, but with a single actor and only under deterministic demand and lead time [11][13][14]. In addition, these research overlooked several essential constraints, such as capability and capacitated machinery in manufacturers, inventory level to be maintained, order limitation to suppliers and product storage capacity in the warehouse. Meanwhile, only some of these constraints were particularly considered in other approaches like game theory [15].

Therefore, this paper's main contribution is examining the RL approach's efficiency in solving more complex inventory problems to reflect the nature of the supply chain process. The complexity is emphasised on aggregated constraints that were missing in the previous RL studies, such as supply chain with a multi-echelon model (multi-stages and more than one actor for each stage) and the involvement of crucial constraints mentioned in the previous paragraph. Finally, the main goal of this study is to optimise replenishment policy while reducing total inventory management costs in the multi-echelon supply chain under complexity (various constraints and stochastic conditions).

## II. Literature Review

### *A. Inventory Management in Multi-Echelon Model*

Stock replenishment is one of inventory management's activities to fulfill the demand. It is regulated by several decision-makers or agents in a multi-echelon model. Studies have modeled inventory management decision making in two stages: supply and manufacturing [16][17], while others have adopted MIT beer game in decentralising the supply chain in three main processes: supply, production and distribution [18][12][19]. Moreover, some researchers have considered retailers in the four echelons stages [15] [6].

In order to have efficient inventory management, decision-makers should consider various assumptions and constraints. The main assumptions are demand or requirements from downstream and inventory lead time, which are derived either from known (fixed amount during a period) or random (follow specific distribution pattern) [20] factors. Another assumption is a *control system* that determines how often an inventory status should be reviewed, how many units are to be ordered, and when they are expected to arrive. The number of orders that are less than the machine's minimum capability will cause the machine's inability to produce those orders; consequently, the upstream chain acknowledges a minimum order quantity or gathers the requested amount to the minimum batch size. On the other hand, suppliers and upstream plants have limited capacity to produce the goods needed by their downstreams, so there will be a maximum order quantity scheme. The warehouse makes for an additional stock constraint [7] because when inventory exceeds available capacity, additional costs may be charged for storage rental.

### *B. Reinforcement Learning (RL) Practice for Inventory Policy*

RL provides an opportunity for agents as decision-makers to learn rewards and penalties from the results of interactions with their environment through trial-error simulations [21][22]. Hence, RL takes care of more problematic issues since it learns through historical policies over a dynamic state-action pair and accumulates rewards until it accomplishes the near-optimal value [23].

RL is well known for its robustness in solving complex inventory management problems for multi-echelon SCM models, for example those related to demand and lead time uncertainty in continuous time inventory controls [12][24]. By embracing MIT Beer Game's network model, RL prompts lower stock expenses than a one-to-one technique and GA-based calculation [25]. Recent studies have also proven that RL yields minimum inventory costs on perishable products where demand has high variance and products have short lifetimes [26]. Unfortunately, the research was only carried out at the retailer's level (single echelon) with a single item. Another recent study has accommodated hundreds of product items and proven the effectiveness of RL in minimising inventory costs throughout a dynamic supply chain environment. Still, the focus is only on the last stage in the supply chain, namely the movement of products from local warehouse to the store [27]. It can be inferred that those studies has only captured a single supply chain's echelon. Meanwhile, previous research mentioned in the introduction have not considered more complex supply chain conditions where various assumptions and uncertain

environments can happen together and affect the total inventory costs. Meanwhile, such things are commonly used partially in various other non-RL research related to inventory management. The assumptions and constraints mentioned are as follows: multi-echelon (multi-agents and multi-stages) [11][13][14][15][28], capacitated manufacturing and warehouse [15][29][28], allowable backorder [12][29], reorder limitation to suppliers, and stochastic demand & lead time [15][29][28].

RL's five key elements interact with others (Fig. 1): agent, environment, state, action and reward[11]. During each interaction, the agent (decision-maker) observes the state $s$ or current inventory position in time $t$ ($s_t$), where $s \in S$. Then, agent selects actions $a \in A$ to order some quantity to the upstream, so the environment responds to them and presents penalty $p$ (inventory cost as a consequence of agent's action). Afterward, the response from environment stimulates a transition of previous state $s$ to the new state $s_{t+1}$. The decision made by the agent through action $a$ then affects the immediate payoff $r$ and the discounted payoff in the next period based on a Markov Decision Process (MDP)[30]. The next action-value calculation will be carried out prior to optimal value when the penalty is obtained through MDP. At that point, the value is defined by running Bellman Equation inside Q-Learning function[14].

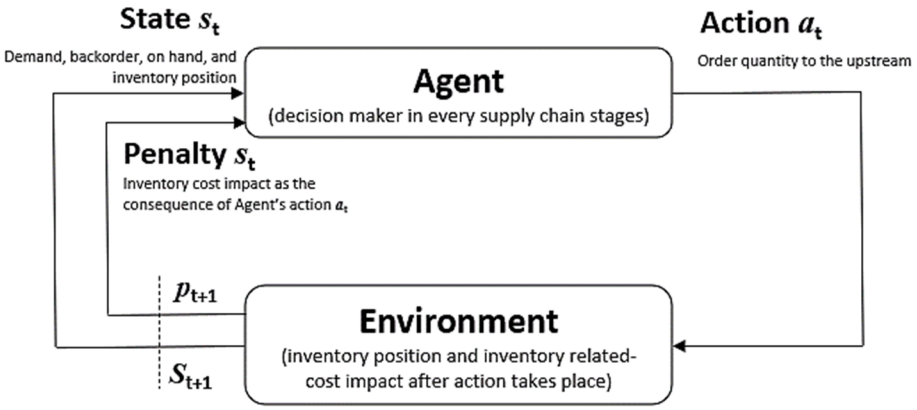
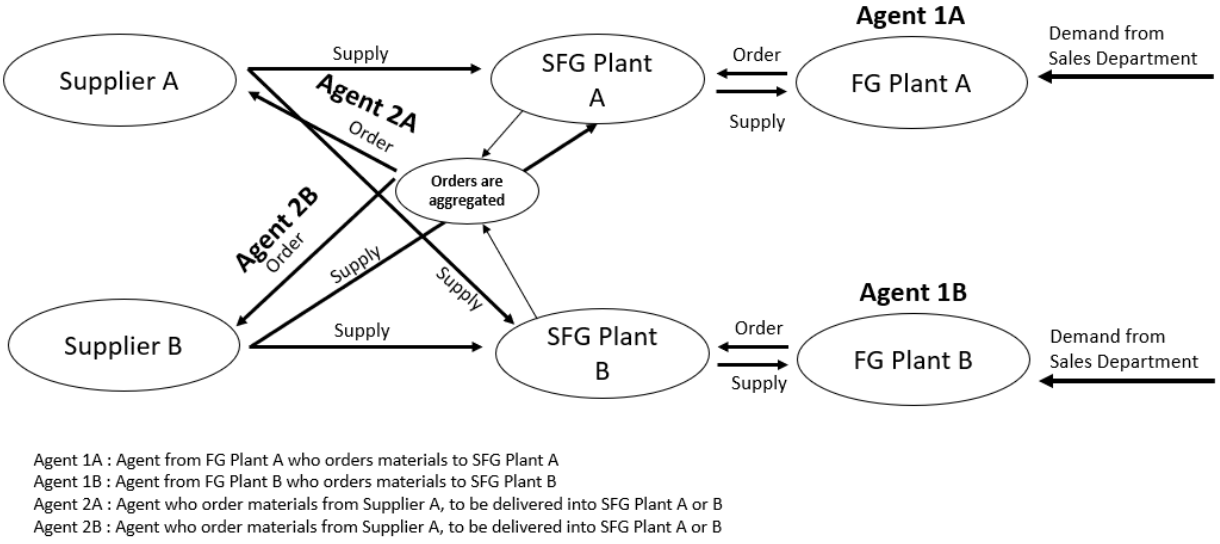



Fig. 1 Key Element Interaction of RL

### *A. Defining Supply Chain Environment*

The first step in this study is determining the scope of supply chain environment simulation. Inspired by previous literature, which is then combined with an actual make-to-stock company model, this research focuses on a multi-echelon supply chain consisting of multi Semi-Finished Good plants (SFG Plants), multi Finished-Good plants (FG Plants), and multi suppliers for a single item (Fig. 2).




Agent 1A : Agent from FG Plant A who orders materials to SFG Plant A
Agent 1B : Agent from FG Plant B who orders materials to SFG Plant B
Agent 2A : Agent who order materials from Supplier A, to be delivered into SFG Plant A or B
Agent 2B : Agent who order materials from Supplier A, to be delivered into SFG Plant A or B

Fig. 2 Multi-echelon supply chain's environment

Agents will occupy each FG and SFG plant as decision-makers who determine actions by ordering materials to the upstream supply chain: manufacturers or suppliers. Demand from sales department will be communicated to Agent 1A at FG Plant A and Agent 1B at FG Plant B. The demand, then, becomes the input for Agent 1A and 1B

to calculate SFG materials order quantity to SFG Plant A and B. After Agents 1A and 1B send their order quantity of materials, Agents 2A and 2B will catch it as demand for them. Agents 2A and 2B need to buy materials from suppliers before producing finished goods to meet this demand. The amount of material needs must be aggregated before being allocated to each supplier, where the purchase will be maximised first to supplier A. If the purchase to Supplier A has exceeded the maximum order quantity limit, then a number of orders will be allocated to Supplier B. Finally, by applying RL, this study has an objective to minimise overall inventory costs caused by order, backorder, holding and rented warehouse (additional storage) costs. This machine learning will consider some variables and constraints, such as demand, lead time and machine capacity.

As previous experiments have not considered various constraints simultaneously, we were unable to find data references. Therefore, to achieve the study's goal, we added arbitrary initial data, such as demand, lead time, initial inventory, and cost assumptions. Other limitations considered in the RL simulations are: 1) FG and SFG plants committed to one stock-keeping unit (SKU); 2) storage is devoted to an FG/SFG; 3) item-units must be integer number, e.g. data input (demand, beginning inventory, lead time, etc.) and output (optimum value of inventory); 4) order can be placed only at the beginning of the month; 5) backorder; and 6) additional external warehouse rental.

### *B. Q-learning Adjustment*

RL is utilised in this undertaking with some acclimation to align with the environment of supply chain network, such as demand, lead time, safety stock, reorder point limitation, capacitated manufacturing (minimum and maximum machine capacity), order batch size, capacitated warehouse (maximum storage) and allowable backorder. R version 3.6.1 is combined with Rstudio to code the RL algorithm, then further costs analysis and evaluation generated from those algorithms are extracted into Microsoft Excel spreadsheet.

Another adjustment is required for Bellman Equation where the original formula in the literature used reward and maximisation objectives. As this study focuses on reducing the total inventory cost, the goal is changed to minimise the inventory's penalty cost with Q-Function using Bellman Equation as in (1).

$$Q(s, a) \leftarrow E(p(s, a)) + \gamma \min Q(\delta(s, a), a')) \quad a \in A \qquad (1)$$

The outputs of Q-learning in the form of state-action pairs are stored in a Q-Table. Every time action $a$ (as an element of action series $A$) reaches the exit point of state $s$, the penalty $p$ is calculated and the Q-Table is updated with a new Q-function value denoted by Q'. Referring to (2) there is a learning rate with a value between 0 and 1 which illustrates how fast the agent impulse Q-value was in the Q-Table. Finally, at the end of the iterations, RL calculates the immediate rewards and the future aggregated payoff. Along these lines, this Q-function value makes up the worth that will be enhanced.

$$Q'(s, a) \leftarrow ((1 - \alpha) Q(s, a)) + (\alpha (p' + \gamma \min Q(s', a'))) \quad a \in A \qquad (2)$$

Finally, through 1,000 times of iterations, the Q-learning will minimise annual inventory cost (holding, backorder, with objective function as in (3), where time $t$ represent monthly order from 1st to 12th month, $k$ 1 and 2 represent number of echelons, and $n$ accounts number of iterations, $f$ is fixed cost, $o$ is order quantity, $oc$ is order cost per unit, $h$ is hold materials, $hc$ is holding_cost per unit, $w$ is materials quantity hold in external warehouse, $wc$ is additional rented warehouse cost per unit, $b$ is backorder, and $bc$ is backorder cost.

$$Minimize \sum_{t=1}^{12} \sum_{k=1}^{2} \sum_{n=1}^{1000} (f + (o * oc) + (h * hc) + (w * wc) (b \times bc)) \qquad (3)$$

### *C. RL Algorithm Development*

The proposed RL algorithm for all agents will follow Markov Decision Process (MDP) with the flow as in Fig. 3. The common MDP is used with different attributes aligned into the case: number of iteration $n$ equals to 1,000, order is only done at the beginning of the month, agent $K$ are 1A, 1B, 2A, and 2B represents multi-echelon decision-maker and constraints (manufacturer capacity, warehouse capacity, safety stock and backorder).

Following MDP, the pseudocodes for RL algorithm were developed with a similar nature for all agents. However, it has different parameter values of demand, lead time, initial inventory, safety stock, maximum order, minimum batch size, machine capacity and some cost assumptions. For example, finished good demand for Agent 1A and 1B that come directly from the market has a different unit of measurement with semi-finished good order for Agent 2A and 2B. Therefore, numbers of demands are automatically different among the agents. The pseudocodes of proposed RL algorithm can be seen in Algorithm 1. The codes are run firstly for Agent 1A and 1B to find the optimum result (order quantity and order time). Then, the result will become a demand for Agent 2A and 2B to decide action (order materials from suppliers).

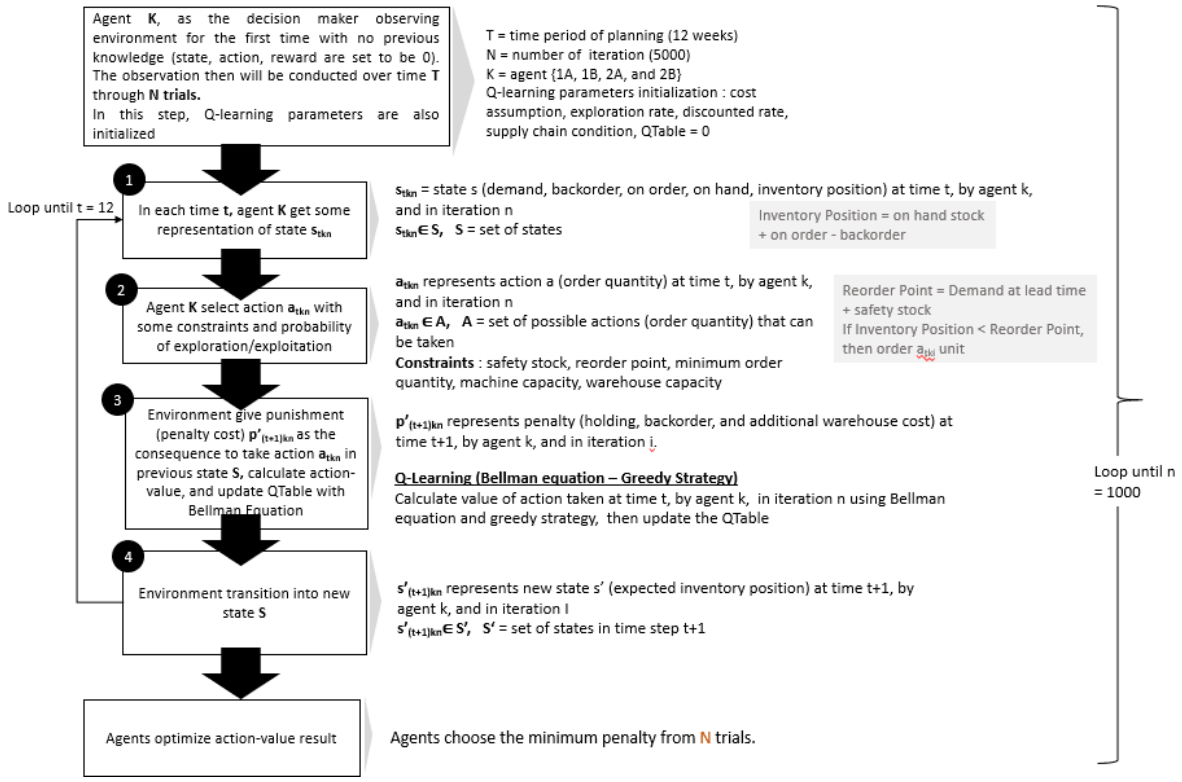



Fig. 3 Proposed Markov Decision Process Flow

```
    Codes
1   #1 Set cost assumption
2   #2 Set learning assumption
3   #3 Initial supply chain condition
4       #4 Create Q-Table
        Create Q-Table and set value = 0
5       #5 Start the Iteration
        For iteration in 1 to N {
6           #6 Put initial inventory value into Q-Table
            time = 1 # beginning of periodic review
7           For time in 1 to T {
8               Update Q-Table1 with values of:
                Iteration number, time step (week number), demand & lead time at time step
9           }
10          time = 1
11          #7 State Change
            For time 1 to T {
12              Update Q-Table1 with values of:
                initial inventory, inventory position
13              demand, beginning inventory, on order, backorder, inventory position
14                  Set exploration trade-off, then take an exploitation or exploration action
15                  Calculate penalty cost based on the action taken
16                  Calculate action value based on Q-Function formula
17                  Update Q-Table with penalty cost and action value
18                  Set new state
21          }
22          Exploration Decay
23      }
24  #8 Display the optimum value given state-action-value pairs
```

Algorithm 1. Proposed RL Algorithm

*#1 Set cost assumption*

All the numbers used in cost assumptions are arbitrary, and this approach is also used in previous research mentioned in the literature. Assumptions in code #1 were made for two modes: 1) same costs for all variables; and 2) different costs for the individual variable (order, backorder, holding, rented warehouse and shortage cost). Fixed cost is assumed to be $20.

When running a model where cost is the same for all variables, an arbitrary value of $2 is applied for both Agent 1A and 1B. Meanwhile, when different cost modes are run, it will adopt the common supply chain assumptions where agents must avoid backorder since it incurs more costs. Therefore, the backorder is set to be more than double the other costs (£5). Besides, an extra warehouse is also charged with an arbitrary cost £3, lower than backorder yet higher than the internal warehouse (holding cost) because it is natural that additional rental for an external warehouse is more costly than the inward holding cost.

Furthermore, the materials order costs from Agents 2A and 2B to Supplier A and Supplier B are distinguished to represent the nature of the supply chain where companies usually source their materials from multi-suppliers. The cost of Agent 2A to purchase materials to Supplier A is set to be lower ($2) than to Supplier B ($3), assuming the company will mainly order from Supplier A for cost-effective reasons. Then, fixed cost is only charged when order quantity >0 or since agents do not need to pay when no purchase order is required.

TABLE 1
LEARNING CONDITION

| Variables | Assumption Value | Source / Remarks |
|---|---|---|
| exploration rate | 1 | Following the concept of RL with Markov Decision Process[30][31] where the agent has no knowledge in the initial condition, the probability of exploration is 100% or 1. |
| max. exploration rate | 1 | The maximum exploration rate is 100%, so it is set to be 1 |
| min. exploration rate | 0.01 | It should be near 0. Unfortunately, they did not mention the exact number for the minimum exploration rate, so it is assumed to be 0.01 [11] |
| exploration decay rate | 0.01 | The least decay rate is equal to the min. exploration rate, which is 0.01 |
| discounted rate | 1 | Refers to [11] |
| learning rate | 0.9 | Refers to [11] |
| T | 12 | Own assumption following a number of months in a year |
| N | 1000 | Arbitrary number, but then it will be evaluated whether this iteration number is enough for agent to learn (convergence testing) |

TABLE 2
INITIAL SUPPLY CHAIN CONDITION

| Variables | Assumptions | | | |
|---|---|---|---|---|
| | Agent 1A | Agent 1B | Agent 2A | Agent 2B |
| **Demand** | Assumed stochastic demand with normal distribution (mean=159, standard deviation = 57) *this is random numbers generated by R | Assumed stochastic demand with normal distribution (mean=90, standard deviation = 42) *this is random numbers generated by R | Taken from optimum order quantity resulted by Agent 1A and 1B. The demand is multiplied by 3 as it is assumed that 1 unit of SFG requires 3 units of raw material. | |
| **Lead time** (for producing items in the manufacturing or ordering materials to supplier) | Assumed stochastic lead time with uniform distribution (minimum 1 day and maximum 2 month) | Assumed stochastic lead time with uniform distribution (minimum 1 day and maximum 3 months) | Assumed stochastic lead time with uniform distribution (minimum 1 day and maximum 2 days) | Assumed stochastic lead time with uniform distribution (minimum 1 day and maximum 3 days, longer than 2B because 2A is prioritized suppliers) |
| **Safety Stock** (number of stock to be kept in the warehouse) | Calculated based on safety stock formula in previous literature[8] | | | |
| **Reorder Point** The stock level that triggers to take an action (order an amount of unit to upstream) | Calculated based on reorder point formula in previous literature[8] | | | |
| **Maximum machine capacity** (Maximum quantity that machine can produce per month) | Assumed stochastic demand with normal distribution (mean=300, standard deviation = 50) | Assumed stochastic demand with normal distribution (mean=350, standard deviation = 50) | No production | No production |
| **Batch Size** (Minimum & Maximum order) | Min. = 150 units (arbitrary)<br>Max = max. machine capacity | | Min. = 400 units (arbitrary)<br>Max = 800 units (arbitrary) | Min. = 100 units (arbitrary)<br>Max = 800 units (arbitrary)<br>Assumed lower than supplier A since supplier B is backup plan |
| **Maximum warehouse capacity** (maximum internal storage) | 270 units for each FG plant | | 1000 units for each SFG Plants | |

*#2 Set Learning Condition*

Similar to the cost assumptions, RL agent's ability to study the environment also needs to be initialised through three parameters: 1) exploration rate (maximum, minimum and decay rate); 2) learning rate; and 3) discounted rate. *T* as the variable of periods per week and *N* as the number of iterations were also determined. In this case, the numbers 12 and 1,000 are designated as initial values of *T* and *N*. The complete initial conditions specified in TABLE and it is applied for all experiments are.

*#3 Set Initial Supply Chain Condition*

The constraint variables, namely demand, lead time, safety stock, reorder point, maximum machine capacity, batch size and maximum warehouse capacity are then initiated in

TABLE . All of those supply chains' constraints are assumed with arbitrary numbers (similar to previous research that uses the same approach).

*#4 Create Q-Table*

As in the first stage of exploration, the agent does not have any learning experience about the environment, thus, the Q-value inside Q-Table is assumed to be 0.

*#5 Start the Iteration*

The RL iteration is begun at1 and halted when it accomplishes the maximum iteration of 1,000. The agent will learn ideal state-action pairs during the iterations to search for the least penalty.

*#6 Put initial inventory value into Q-Table*

Several parameters, which are weekly demand, respective lead time to satisfy demand, iteration number and week number are embedded into Q-Table. Afterwards, the Q-Table will be refreshed with newest values (state-action pairs, action-value and penalty).

*#7 State Change*

The state referred to in the case of RL in this project represents the inventory position, which is calculated from the quantity on hand stock plus on order minus backorder. The value in the state triggers the agent to perform an action through exploration or exploitation. As explained earlier, the exploration rate is initially set at 1, but as the agent's ability to learn about its environment increases, epsilon becomes greedy so that the rate will decrease [12]. There will be a threshold that stores the agent's actions, and if the threshold is < exploration rate, the agent will choose to do exploration; otherwise, exploitation is selected.

The exploration rate will decay as agents improve its learning experience, so there should be codes to express this condition once the environment has moved to the new state where exploration rate = minimum exploration rate + (maximum exploration rate – minimum exploration rate) * exponent (exploration decay rate * iteration) [31]

Subsequently, formula (3) or the objective function is compiled. Then, at that point, through the learning cycles and augmented time step *t*, the new action-value is determined and embedded into Q-Table. Refers to Bellman Equation or (2), the new action-value *Q' (s,a)* made up a weighted amount of previous action-value *Q(s,a)* and the learned action-value *Q(s',a')*. As such, the new Q-value characterises the summation of the discounted penalty amassed toward the next period. Then, at that point, provided time *t* and iteration *n*, Q-Table is refreshed with the penalty, and action-value is determined. As requirement and supply lead time for each week has been characterised, the beginning stock and order arrival are refreshed and set as new. The exploration rate will rot as the agents develop their learning experience further, so they ought to be coded to reflect that this condition has moved to the new state where investigation rate = least investigation rate + (greatest investigation rate – least investigation rate) * example (investigation rot rate * cycle)

*#8 Show the optimum result (state-action-value pair)*

Once iterations are finished, the system will display iterations that have the least Q-Value and penalty. A set of solutions, in this case, consists of inventory position where the order must be created, order quantity for each month and total penalty cost.

### *D. RL Evaluation*

Alluded to past experiments on RL assessment, convergence assessment is conducted to review whether the iterations are sufficient for agents to learn state-action matches and look for the minimum accumulated inventory expenses [14]. Microsoft Excel's chart is then used to extract the solutions and visualise the convergence trend. In two experimental scenarios, the trend depicts all agents found a minimum inventory cost and Q-Value under 250 iterations. Hence, it revealed that 1,000 iterations set in this algorithm is sufficient to reach the convergence level.

The two RL experimental scenarios is then performed to inspect the consistency of RL algorithm in defining optimum inventory policy. The scenario considered the sensitivity of the most crucial inventory management assumptions: demand, supply lead time and cost assumptions (Table 3). Demand and lead time are assumed stochastic with a specific probability distribution for both scenarios. Meanwhile, costs are computed with two sensitivity scenarios: 1) flat cost for all variables; and 2) different cost setting—especially so for backorder and additional external warehouse since those two assumptions lead to adverse risk for supply chain and incur higher inventory cost assumptions. All the numbers of assumptions follow similar literature mentioned in section 2.3, in which arbitrary numbers are used.

After getting the RL experiments' results, the effectiveness of RL is evaluated by comparing its results with ones of the most used mathematical approaches of replenishment policy under uncertainty mentioned in the introduction section, which is (s,Q) [8].

TABLE 3
EXPERIMENTS DESIGN

| Scenarios | Demand & Lead Time Assumptions | Cost Assumptions |
|---|---|---|
| 1 | Stochastic with certain probabilistic | same cost for all variables |
| 2 | Stochastic with certain probabilistic | different costs for backorder and additional external warehouse |

## I. RESULTS

As explained in the method section, an evaluation of RL is carried out by contrasting its outcome with a mathematical method called (s,Q). TABLE illustrates that for scenario 1, where demand and lead time are probabilistic, and cost assumptions are flat for all variables, RL generated the total inventory cost, which is not significantly better than (s, Q). It happens since the difference is just by 7%. However, the backorder cost of RL is higher than (s,Q). In scenario 2, the result is consistent with the first observations. RL affirms to be a superior technique to decide efficient inventory policy in limiting stock expenses with 21% (£16,666) distinction than (s,Q). Similar to scenario 1, RL also suffers from backorder costs for scenario 2 (TABLE ).

TABLE 4
OVERALL COST BREAKDOWN – ALL AGENTS

| | Scenario 1 | | | Scenario 2 | | |
|---|---|---|---|---|---|---|
| **Cost Type** | **RL** | **(s,Q)** | **Difference** | **RL** | **(s,Q)** | **Difference** |
| Fixed | 620 | 620 | - | 580 | 680 | 17% |
| Order | 17,330 | 17,604 | 16% | 13,663 | 18,120 | 33% |
| Holding (internal _ external) | 45,883 | 51,644 | 13% | 45,883 | 58,412 | 30% |
| **Sub Total** | **63,833** | **69,868** | **9%** | **58,826** | **77,212** | **31%** |
| Backorder | 1,660 | 270 | 84% | 2,740 | 1,020 | 63% |
| **Sub Total** | **1,660** | **270** | **84%** | **2,740** | **1,020** | **63%** |
| **Total** | **65,493** | **70,138** | **7%** | **61,566** | **78,232** | **21%** |

For both scenarios, RL is unable to perform better for the overall cost of Agent 1B (TABLE ). A closer inspection to Agent 1B in scenario 1 shows that the (s,Q) strategy actually results in a lower accumulated cost than RL, £6,122 and £7,382, respectively.

(s,Q) method apparently manages the inventory more smoothly because the stock's holding cost is lower by £1,446 while backorder cost is slightly different by £2 (TABLE ). Moreover, better execution of RL's structure cost cannot make up at the less expensive cost coming about by (s,Q) strategy, as shown in TABLE .

TABLE 5
ACCUMULATED COST PER AGENT

| | Scenario 1 | | | Scenario 2 | | |
|---|---|---|---|---|---|---|
| Agents | **(s,Q)** | **(RL)** | **Difference** | **(s,Q)** | **(RL)** | **Difference** |
| 1A | 10,720 | 9,101 | 15% | 11,476 | 10,712 | 7% |
| 1B | 6,122 | 7,382 | 21% | 7,263 | 7,746 | 7% |
| 2A + 2B | 53,296 | 49,010 | 8% | 59,493 | 43,108 | 28% |

TABLE 6
COST BREAKDOWN OF AGENT 1B – SCENARIO 1

| Cost Type | (s,Q) | RL | Difference |
|---|---|---|---|
| Fixed | 240 | 240 | 0 |
| Order | 2,436 | 2,248 | 8% |
| Holding | 3,414 | 4,860 | 29% |
| Backorder | 32 | 34 | 6% |
| **Total** | **6,122** | **7,382** | **17%** |

Further inspection to Agent 1B on scenario 2 depicts the more expensive inventory cost in the RL method is mainly impacted by the cost to hold materials (Fig. ). From the first to the fifth period, the higher expense is always shown by RL with an accumulation of $3.891. Meanwhile, in the same period, the (s,Q) strategy only required an inventory cost of $1,594, meaning that only half of the RL's spending. This condition occurs in light of the fact that RL's Agent 1B ordered 342 units toward the start of the period and kept it for a significant stretch, prompting a large stock. Simultaneously, the demand in that period is relatively low (Fig. 5).

Conversely, (s,Q) strategy orders the items starting from the third and fourth months to appear in the fifth month when the requirement is high (Fig. ). The extended length of keeping the stocks also happens when RL's agent sets the accompanying request in the sixth and seventh months, while (s,Q) recharges the supply in the ninth to eleventh months (nearly year's end).


Fig. 4 Holding Cost Agent 1B - Scenario 2


Fig. 5 Order Quantity vs Demand of Agent 1B - Scenario 2

Referring back to Table 4, especially in scenario 1, RL provokes more backorders and even arrives at multiple times more than (s,Q) method, while (s,Q) permits the agent to spend just £270, RL costs more than five times (£1,660). Nevertheless, since all agents other than 1B provide better execution of order and holding cost, RL's RL's overall expense is cheaper by $4,645. Moreover, the subsequent situation shows that RL's Agent 1B triggers to more backorders, representing £2,740, which is beyond two-fold contrasted with (s,Q) that spends just £1,020. Such a condition is principally set off by RL's Agent 1A decision that produces delayed purchase or backorder units because of enormous deficiencies (Fig. )

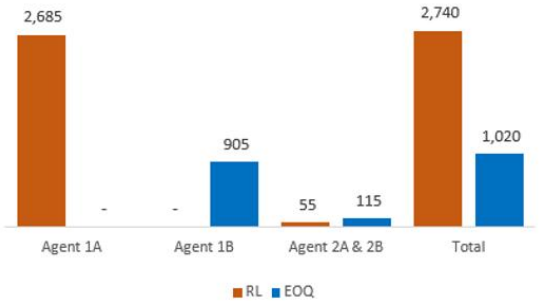



Fig. 6 Backorder Cost - Scenario 2

## II. Discussion

Considering overall agents (1A, 1B, 2A, and 2B), the aggregated expenses created by the RL approach are consistently lower than different strategies in each circumstance. Interestingly, a nearer assessment into individual agents shows that RL triggers a higher expense for Agent 1B since the agent cannot efficiently manage the replenishment when the demand is low or high. At the same time (s,Q) strategy allows a smoother way to order materials to anticipate high demand. Another drawback of RL is that backorders are generated higher than (s,Q) strategy. Notwithstanding, better execution in different expenses (fixed, holding, and order costs) repays the backorder; in this manner, the overall budget were lower than (s,Q). For instance, a less expensive spending from fixed, holding, and order cost by £6,035 repays lower execution of scenario 1 by £1,350 because of backorders. In scenario 2, the gap of backorder difference by £1,720 can not overweigh cheaper cost from fixed, order and holding cost by £18,386.

The results of this study complement the results of previous studies in several points. First, RL consistently produces lower total inventory costs, similar to the research conducted by [32][11][12] Kara and Dogan, Chaharsogi and Gianno. Those studies compare various RL methods, namely with the Genetic Algorithm, 1-1 strategy, and centralised periodic order policy. By comparing RL with mathematical models, this research is able to enrich the effectiveness of RL in producing efficient inventory policies.

Second, this research simulates various assumptions and constraints simultaneously. Meanwhile, previous studies only applied some restrictions partially, such as [32][27], which only simulated a single stage (echelon). Other research, such as [12][11][13][14], considered several stages, but only 1 actor in each stage. Therefore, this research contributes by assuming multi-echelon and multi-actors in every supply chain's stage.

Another assumption that was not adequately accommodated in previous studies is the uncertainty of demand and lead time [12][11][13][14]. In addition, the typical constraints of capacitated machinery in manufacturers, inventory level to be maintained, order limitation to suppliers, and product storage in the warehouse have generally been used in the following studies [12][11][13][14] [17][16]. However, they did not apply together in the same simulation. Now, those constraints are finally added simultaneously in this paper to prove the robustness of RL in handling complex supply chain problems.

## III. Conclusions

Overall, this study reveals that RL can efficiently minimise total inventory cost when the additional constraints to reflect the complexity of supply chain. Nonetheless, decision-makers should focus on whether backorder costs are essentially more costly than different expenses (e.g., dramatically increasing the holding, fixed and order costs). Since RL can result in higher delayed purchase costs, it influences the total inventory expense. At last, consolidating or contrasting RL and other machine learning calculations to deal with more complicated supply network circumstances will also be beneficial.

**Funding**: This research received no specific grant from any funding agency.

**Conflicts of Interest**: The authors declare no conflict of interest.

## References

[1] A. J. Clark and H. Scarf, "Optimal Policies for a Multi-Echelon Inventory Problem," *Manage. Sci.*, vol. 50, no. 12, 2004.
[2] H. Lee, V. Padmanabhan, and S. Whang, "Information distortion in a supply chain: the bullwhip effect," *Manage. Sci.*, vol. 43, pp. 546–558, 1997.
[3] H. Lee and S. Whang, "Decentralized multi-echelon supply chains: incentives and information," *Manage. Sci.*, vol. 45, pp. 633–640, 1999.
[4] T. Moyaux, B. Chaib-draa, and S. D'Amours, "Multi-agent coordination based on tokens," *Proc. Second Int. Jt. Conf. Auton. Agents Multiagent Syst. - AAMAS '03*, 2003.
[5] R. . Lancioni, "New Developments in Supply Chain Management for the Millennium," *Ind. Mark. Manag.*, vol. 29, pp. 1–6, 2000.
[6] C. Jiang and Z. Sheng, "Case-based reinforcement learning for dynamic inventory control in a multi-agent supply-chain system," *Expert Syst. Appl.*, vol. 36, no. 3 PART 2, pp. 6520–6526, 2009, doi: 10.1016/j.eswa.2008.07.036.
[7] D. Waters, *Inventory control and management*. Sussex: Wiley, 2003.
[8] D. F. Pyke, D. J. Thomas, and E. A. Silver, *Inventory and production management in supply chains*, 4th ed. Taylor & Francis Group, 2017.
[9] X. Guo, C. Liu, W. Xu, H. Yuan, and M. Wang, "A prediction-based inventory optimization using data mining models," in *Proceedings - 2014 7th International Joint Conference on Computational Sciences and Optimization, CSO 2014*, Oct. 2014, pp. 611–615, doi: 10.1109/CSO.2014.118.
[10] S. Nahmias and T. L. Olsen, *Production and Operations Analysis*, 7th ed. Long Grove III: Waveland Pr, 2015.
[11] S. K. Chaharsooghi, J. Heydari, and S. H. Zegordi, "A reinforcement learning model for supply chain ordering management: An application to the beer game," *Decis. Support Syst.*, vol. 45, no. 4, pp. 949–959, 2008, doi: 10.1016/j.dss.2008.03.007.

[12] I. Giannoccaro and P. Pontrandolfo, "Inventory management in supply chains: A reinforcement learning approach," *Int. J. Prod. Econ.*, vol. 78, no. 2, pp. 153–161, 2002, doi: 10.1016/S0925-5273(00)00156-0.
[13] A. Mortazavi, A. Arshadi Khamseh, and P. Azimi, "Designing of an intelligent self-adaptive model for supply chain ordering management system," *Eng. Appl. Artif. Intell.*, vol. 37, pp. 207–220, 2015, doi: 10.1016/j.engappai.2014.09.004.
[14] T. Van Tongeren, U. Kaymak, D. Naso, and E. Van Asperen, "Q-learning in a competitive supply chain," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, pp. 1211–1216, 2007, doi: 10.1109/ICSMC.2007.4414132.
[15] Q. Duan and T. Warren Liao, "Optimization of replenishment policies for decentralized and centralized capacitated supply chains under various demands," *Int. J. Prod. Econ.*, vol. 142, no. 1, pp. 194–204, 2013, doi: 10.1016/j.ijpe.2012.11.004.
[16] Z. Jemaï and F. Karaesmen, "Decentralized inventory control in a two-stage capacitated supply chain," *IIE Trans. (Institute Ind. Eng.*, vol. 39, no. 5, pp. 501–512, 2007, doi: 10.1080/07408170601180536.
[17] P. Toktas-Palut and F. Ülengin, "Coordination in a two-stage capacitated supply chain with multiple suppliers," *Eur. J. Oper. Res.*, 2011.
[18] F. Strozzi, J. Bosch, and J. M. Zaldívar, "Beer game order policy optimization under changing customer demand," *Decis. Support Syst.*, pp. 2153–2163, 2007.
[19] M. Li and Z. Wang, "An integrated robust replenishment/production/ distribution policy under inventory inaccuracy," *Int. J. Prod. Res.*, 2018.
[20] A. Gupta, C. D. Maranas, and C. M. McDonald, "Mid-term supply chain planning under demand uncertainty: customer demand satisfaction and inventory management," *Comput. Chem. Eng.*, vol. 24, 2000.
[21] A. J. Smith, "Applications of the self-organising map to reinforcement learning," *Neural Networks*, pp. 1107–1124, 2002.
[22] J. J. Rao, K. K. Ravulapati, and T. K. Das, "A simulation-based approach to study stochastic inventory-planning games," *Int. J. Syst. Sci.*, vol. 34, no. 12–13, pp. 717–730, 2003, doi: 10.1080/00207720310001640755.
[23] Marcus A. Maloof, "Incremental rule learning with partial instance memory for changing concepts," *Proc. Int. Jt. Conf. Neural Networks*, pp. 2764–2769, 2003.
[24] A. Mahadevan, S., Marchalleck, N., Das, K.T., Gosavi, "Self-improving factory simulation using continuous-time average-reward reinforcement learning," *Proc. 14th Int. Conf. Mach. Learn.*, pp. 202–210, 1997.
[25] S. O. Kimbrough, D. J. Wu, and F. Zhong, "Computers play the beer game: Can artificial agents manage supply chains?," *Decis. Support Syst.*, vol. 33, no. 3, pp. 323–333, 2002, doi: 10.1016/S0167-9236(02)00019-2.
[26] A. Kara and I. Dogan, "Reinforcement learning approaches for specifying ordering policies of perishable inventory systems," *Expert Syst. Appl.*, vol. 91, pp. 150–158, 2018, doi: 10.1016/j.eswa.2017.08.046.
[27] H. Meisheri, V. Baniwal, N. N. Sultana, H. Khadilkar, and B. Ravindran, "Using Reinforcement Learning for a Large Variable-Dimensional Inventory Management Problem," 2020.
[28] P. Tokta-Palut and F. Ülengin, "Coordination in a two-stage capacitated supply chain with multiple suppliers," *Eur. J. Oper. Res.*, vol. 212, no. 1, pp. 43–53, 2011, doi: 10.1016/j.ejor.2011.01.018.
[29] Z. Jemai and F. Karaesmen, "Coordination in a two-stage capacitated supply chain with multiple suppliers," *IIE Trans. (Institute Ind. Eng.*, vol. 5, no. 39, pp. 510–512, 2007.
[30] Steven D Whitehead and L.-J. Lin, "Reinforcement learning of non-Markov decision processes," *Artif. Intell.*, vol. 73, pp. 271–306, 1995.
[31] Deeplizard.com, "Train Q-learning agent with python - reinforcement learning code project," 2018. https://deeplizard.com/learn/video/HGeI30uATws (accessed Aug. 03, 2019).
[32] A. Kara and I. Dogan, "Reinforcement Learning Approaches for Specifying Ordering Policies of Perishable Inventory Systems," *Expert Syst. Appl.*, 2017, doi: 10.1016/j.eswa.2017.08.046.