A SIMULATION ENVIRONMENT FOR REDUCING FOOD WASTE VIA REINFORCEMENT LEARNING

Sebastian Pilarski
Aman Sidhu

Electrical and Computer Engineering
McGill University
3480 University Street
Montreal, QC, CANADA
{sebastian.pilarski, aman.sidhu}@mail.mcgill.ca

Dániel Varró

Computer and Information Science (IDA)
Linköping University
SE-581 83
Linköping, SWEDEN
daniel.varro@liu.se

ABSTRACT

Food retailers discard astonishing amounts of spoiled produce as waste. Reinforcement learning could greatly improve retail store product ordering and pricing decisions to simultaneously reduce food waste and improve business profits. We present a discrete-event-based food retail simulation framework which simulates wholesaler, store, and customer interactions. This simulator is essential for driving development and testing of future reinforcement learning methods to help economically reduce food waste for food retail stores. Simulation provides an efficient learning feedback system across a massive number of possible scenarios, which cannot be replicated from live observation or pure historical data alone. We demonstrate our simulator on an example built from historical consumption and price data. A simple realistic baseline resulted in more than US$1M of food waste (2010-2015). A soft actor critic (SAC) RL agent increased profit by 42% and reduced food waste by almost US$500k over three years (2012-2015) after learning from simulations (2010-2012).

Keywords: discrete event simulation, reinforcement learning, food retail, food waste.

1 INTRODUCTION

Today, over one-third of all food ($1T worth) is thrown away, with experts calculating that most retail grocers waste 40% of fresh products such as produce when measured as the difference between inventory delivered and inventory sold. Such waste is neither environmentally conscious nor sustainable. It represents an alarming missed potential access to nutrition, and also negatively affects business profits. Food waste contributes to tight profit margins for food retailers with the industry average sitting at 2.2%. Clearly, there exists a large opportunity for reducing food waste given the economic incentive for businesses in doing so.

Artificial intelligence (AI) presents exciting new possibilities and solutions to modern problems. As a motivating example, reinforcement learning (RL) has shown the capability to greatly outperform expert human decision-making in complex strategic games such as chess and Go. Our long-term vision aims to exploit RL techniques to optimize food retail operations to both reduce food waste and improve business profitability.

In food retail, consumer habits are dynamic with many variables affecting purchasing behavior. Discounts, weather, the season, and product quality are all factors which may be a difference-maker in whether a customer decides to purchase a product or not. For a retailer, it is a difficult challenge to optimize order quantities and pricing of products for simultaneous food waste reduction and profitability, especially for perishable products. Too few products in inventory or extreme discounts can result in products running out-of-stock, which presents a loss in potential revenue for retailers. Thus, from a business perspective, there is a bias towards overstocking perishable products in case there is a variable surge in consumer demand. Retailers maintain significant markups compared to wholesale prices; the revenue from one product sold to consumers covers the loss from more than one product of waste.

An RL-driven decision-making system which decreases food waste, while avoiding understocked inventory, would greatly improve sustainability and retail profitability. Such a system, however, would need to have demonstrated reliability and adaptability to changing environments. It must be capable of operating and reacting effectively to both expected events (e.g., holiday season) and unexpected events (e.g., pandemic, drought, new health trend). Both types of events can affect wholesale prices of products, as well as consumer spending behaviours. Consequently, any intelligent RL in food retail needs to be rigorously evaluated on a wide gamut of possible events prior to deployment to a real environment. *Simulation plays a crucial role for training, evaluating, and testing such RL-based decision-making systems.*

Problem Statement Adapting an RL approach for store food waste reduction and improved profitability requires a learning and testing environment where a myriad of events can be learned from and relevant data can be accessed. Unfortunately, despite a significant amount of conceptual results, existing simulators do not provide interfaces to food waste, product deliveries, and purchase tracking as well as pricing/ordering control, which is a prerequisite for an RL approach in food retail.

Objectives This paper represents the first stage of an NSERC I2I innovation project which aims to tailor advanced RL techniques for intelligent decision making in food retail. The main objective of this paper is to (1) develop a simulation framework that can be used for the development, validation, and testing of RL methods and to (2) provide an initial demonstration in a food retail context. In a next phase, the simulation framework will serve as the foundation for customizing decision making to individual stores in food retail.

Contributions This paper presents technical contributions in the following areas:

1. *A food retail simulator* capable of simulating a food retail environment with event, seasonal, and day-to-day demand variations, revenue, food spoilage (waste), product deliveries, and product pricing.
2. *Conceptual methodology for applying RL* to the retail food waste problem via simulation.
3. *Initial experimental evaluation* for strawberries and potatoes derived from historical data.

Significance The customizable food retail simulator presented in this paper enables a sandbox environment to develop and test RL solutions for retail pricing and inventory management of perishable items. This simulator enables a learning feedback loop where an RL algorithm can experience, experiment, and learn from a vast number of unique realities and events which otherwise would be limited to live and historical observation. Furthermore, it contributes a conceptual methodology to apply RL techniques which will lead to improved retail practices which significantly reduce food spoilage and waste.

2 BACKGROUND

Decisions in Food Retail Stores Successful store management requires effective decision-making to maintain profitability despite low-profit margins. Food retail stores have many tools and tricks at their disposal, including arranging and organizing products within a store in a manner to drive customer traffic to certain products. However, the most fundamental decisions a store makes are (a) what price to set for products, and (b) how many products to order from wholesale to stock inventories. These decisions are

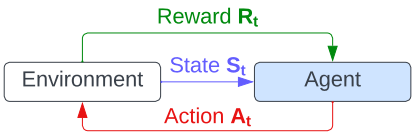

Figure 1: RL formulation.

coupled and directly affect business profitability. Changes in the price of a product can affect customer demand requiring a different order quantity. Excess product in inventory due to a large order could require discounting product prices to avoid sunk costs due to food waste.

Discrete Event Simulation (DES) $DES = (\Sigma, Ev)$ models the behavior of a system with a set of states Σ (value assignments to state variables) and a set of discrete (instantaneous) events Ev. When an event occurs e_{t_i} at time t_i, the state trajectory $\Sigma(t_i)$ of the system is changed (by updating state variables), and simulation time is set to t_i. The state of the system remains stable until when a next event $e'_{t_{i+1}}$ occurs at time t_{i+1}. We intentionally keep this DES definition at a high-level of abstraction, but note that RL could also be integrated with other simulation formalisms (e.g., DEVS (Zeigler et al. 2000)) for smart retail optimization.

Reinforcement Learning RL is a machine learning technique which seeks to learn what actions to perform in a given situation via feedback from rewards. RL problems can be represented by Markov Decision Processes $MDP = (S, A, R, T)$; $\pi : S \rightarrow A$ (shown in Figure 1). An RL agent interacts with its environment (Sutton and Barto 2020). At discrete moments of time, $t, t+1, \ldots$, the agent observes an environment *state* S_t. The agent determines what *action* A_t to perform as the next decision. This action changes the state of the system S_t to S_{t+1} according to a probabilistic state transition function T. At any time t, the agent may receive a *reward* R_t, which is used as feedback to learn better action strategies. Formally, an action strategy is called a *policy* $\pi(a|s)$ which determines an action a for a given state s. The agent learns along *episodes* consisting of a sequence of states $S_0, S_1, \ldots, S_n$ ending on a terminal state S_n (e.g., playing a game). Most RL techniques require significant numbers of episodes to learn effective policies. A non-exhaustive list of RL algorithm types includes multi-armed bandit, actor-critic, and Q-learning. Each has its benefits and disadvantages. Finding effective RL solutions for challenging real-world problems may require experimentation with many algorithm types.

In an RL setting, pricing and ordering decisions can be mapped to *agent actions*. The *state of the environment* can encapsulate information such as the season and number of products in inventory. With a carefully crafted (custom) *reward function* and sufficient learning episodes (e.g., via simulation), an RL agent could learn a *pricing policy* and *ordering policy* used in given environment states.

3 RELATED WORK

Related work spans the fields of simulation and machine learning, and includes literature on sales forecasting (Lv et al. 2008), price optimization (Esmaeili, Naghavi, and Ghahghaei 2018), customer behaviour (Jandera and Skovranek 2022), food waste (Teller et al. 2018), and inventory restock policy (Kim et al. 2005). Our work differentiates itself from others by simulating a food retail environment which enables RL-driven store management. Moreover, most existing simulators are either (a) not available off-the-shelf or (b) cannot be tailored to the needs of *food waste reduction of individual retail stores*. Furthermore, we experiment with new RL techniques not previously used in the food retail domain.

Simulation (Baydar 2003) creates an agent-based simulation environment for a grocery store and uses optimization to determine discount amounts for particular customers to balance pricing, sales-volume, and customer satisfaction. While this is the most similar work discovered in our literature search, it does not factor in wholesalers, control over delivery order sizes, nor model food waste. (Chung and Li 2014), how-

ever, develop a single store simulator for food waste relying upon the Poisson distribution. This simulation is limited to need-based purchases. Customers always select the cheapest product available that meets their remaining days of freshness requirement. (Welling et al. 2021) provide a discrete event simulator by following the 12 step simulation development process to study lead-time based pricing for semiconductor supply chains. (Lau, Xie, and Zhao 2008) create a simulator studying the effects of inventory policy on supply chain performance with four retailers and one supplier. This simulator generates random demand and production capacities and models retailer and supplier decisions. (Christensen et al. 2021) propose forecasting methods utilizing product shelf life and validate these methods through simple simulation.

Reinforcement Learning (Afridi et al. 2020) utilize simulation to develop and test a deep Q learning based approach to replenishment policy in vendor management inventory for semiconductors which leads to significant improvements over a baseline. Similarly, (Oroojlooyjadid et al. 2017) use simulation and a Q learning based approach for the inventory optimization problem to achieve near optimal order quantities in the beer game. (Gijsbrechts et al. 2019) adapt A3C deep RL for inventory management problems and match state-of-the-art heuristic and dynamic programming solutions in simulation. Finally, (David and Syriani 2022) propose the use of RL techniques to create DEVS models.

4 ARCHITECTURE AND METHODOLOGY OF FOOD RETAIL SIMULATOR

Our simulation framework is designed to enable RL-based control of individual stores. A user provides a list of stores, wholesalers, and (groups of) customers characterized by their respective behavioral functions as *input* to the simulation *environment.* Ideally, real food retail data is used to create some underlying demand functions which govern customer purchasing behaviour. The discrete event simulation simulates a sequence of events resulting from interactions between entities on a day-by-day basis. Each event is recorded.

Simulation Architecture Figure 2 presents the simulation architecture in details. A store's behaviour is easily mapped onto an RL problem formulation consisting of an environment and an agent where the environment provides states and rewards and the agent performs actions for a given state. A store (RL agent) receives information from the environment in the form of events (e.g., product purchased or food spoilage). Such events (marked in blue), can be used to designate/update a state. Store orders and pricing decisions (in red) can be represented as RL actions (e.g., set product price). The environment controls when the store can perform an action. Thus, at the time of a request, a state is constructed from prior event information from the environment for which the action is performed. The action is then transmitted to the environment in the form of a pricing or order event. From the simulator's perspective these constitute observable events. Finally, information encapsulated within events can likewise be used to form a (green) reward (e.g., some combination of profit and food waste). For each simulated entity, experienced events are recorded and saved.

Note that our simulator does not impose a concrete RL formulation for the retail food waste problem. Thus our solution is not restricted to a particular definition of a reward or state representation. Our abstraction is to provide and pass events from which states and rewards can then be formed. This provides great flexibility to experiment with any class of RL technique and try different ways of defining rewards/states to find an effective solution to the real food retail problem. The retail food waste problem has many valid RL problem formulations. An initial solution using soft actor critic (Raffin et al. 2021) is investigated in Section 6.

Food Retail Environment To simulate a food retail environment, we follow a discrete-event simulation framework where time elapses in (discrete) days. The simulation environment consists of three independent entity types. (1) **Wholesalers** sell products to *stores*. They control the price at which they sell a particular quantity of products on a particular day. (2) **Stores** purchase products from *wholesalers* for their inventory which *customers* can then buy. Stores set the prices at which to sell products to customers and when to order additional products from wholesalers. (3) **Customers** purchase products from *stores*. Each customer may represent an individual or group of customers with similar purchasing behaviour.

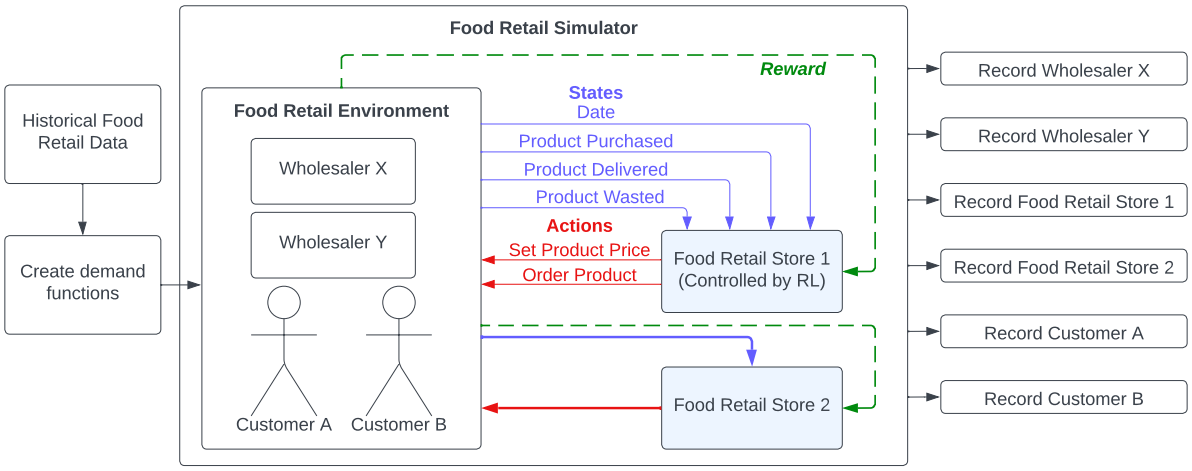

Figure 2: Simulator architecture with state information colored in blue, actions in red, and rewards in green.

Each entity independently makes *decisions* (pricing, purchasing, ordering) by some underlying control function. Interactions between these entities and their actions produce events that affect the *state of the environment* (e.g., products in-store inventories, wasted products, deliveries from wholesalers).

Entities may have certain *limits* regarding when or what kind of decisions (actions) they can make. For example, stores may only be allowed to order from a particular wholesaler on Mondays or Thursdays. Another limit may set maximum price increases for designated products.

All interactions between entities such as wholesalers, stores, and customers constitute *events* which are consumed and handled by appropriate entities potentially triggering state transitions in them. For example, a purchase event designates an interaction between a store and a customer where the customer purchases certain food products. Generated events are validated on the fly via type and instance checking.

5 FOOD RETAIL SIMULATION

Simulation enables training, validation, and testing of RL agents through many expected and unexpected events in a food retail environment. An effective simulator must be computationally efficient, customizable, and provide access to pertinent data and statistics. Our simulator captures the food retail environment, correctly generates states, and acts accordingly to actions undertaken by store controlling RL agents.

Configuration The configuration of the simulator details the simulation start and end dates, number of episodes to perform, and number of threads to use. Furthermore, the configuration provides links to wholesaler, store, and customer entity definitions, the number of instances, their relations, and underlying behaviour functions. Product types (e.g., strawberries) are also defined. The base configuration file (in TOML format) provides a means for documenting the simulations for reruns or future extensions.

Behaviour Functions Customer purchase behaviour is a key input for food retail simulation. Customer behaviour is governed by a demand function which measures the desired demand for a product (i.e., how much to purchase for a given price/product). Our definition of a demand function is flexible, with many possible inputs available such as the price of the product, the season/date, time to expiration, etc. Ideally, such demand functions are derived from historical data. E.g., our example in Section 6 derives daily demand data from monthly and yearly statistics via interpolation. The simulator is also designed in such a way to enable RL agent control of wholesalers and customers in the future. This could serve as an interesting

Algorithm 1: Simulate Day

```
Input: date, wholesalers, stores, deliveries, customers
InOut : eventLog
foreach store in stores do
    foreach delivery in deliveries[store] do
        de = new ProductDeliveredEvent(date, store, delivery.wholesaler, delivery.product)
        store.update(de) ;                    // add delivered products to inventory
        eventLog.append(de)
    foreach ep in store.expiringProducts do
        we = new ProductWastedEvent(date, store, ep)
        store.update(we) ;                    // expired products are removed
        eventLog.append(we)
    orderActions = store.getOrderActions()
    foreach oAct in orderActions do
        oe = new ProductOrderedEvent(date, store, oAct.wholesaler, oAct.prodType, oAct.orderPrice)
        wholesaler.update(oe) ;               // fulfill product order
        store.update(oe) ;                    // product order confirmation
        eventLog.append(oe)
    priceActions = store.getPriceActions()
    foreach pAct in priceActions do
        pe = new PriceSetEvent(date, store, pAct.product, pAct.price)
        store.update(pe) ;                    // change price of product in store
        eventLog.append(pe)
while customers not done shopping do
    customer = select from customers
    purchase = customer.shop()
    se = new ProductSoldEvent(purchase.store, purchase.product, purchase.price)
    purchase.store.update(se) ;               // remove product from inventory
    eventLog.append(se)
eod = new EODEvent(date)
foreach entity in (wholesalers, stores, customers, deliveries) do
    entity.update(eod) ;                      // next day starts
eventLog.append(eod)
return eventLog
```

example, with customers more effectively searching for better deals on food and wholesalers also optimizing their operations. Such future studies could also help reduce food waste on the wholesaler side.

Discrete Event Simulation Environment for Food Retail Simulation (see Algorithm 1) occurs on a day-by-day basis with each day following a consistent pattern of operations which produce and consume events. There are six primary (discrete) event types which signal an environment state change:
(1) PriceSetEvent is generated when a store agent changes the price of a product.
(2) ProductOrderedEvent is created when a store agent places an order to a wholesaler.
(3) ProductSoldEvent occurs from a customer purchase. The product is removed from inventory.
(4) ProductDeliveredEvent alerts that a delivery from a wholeseller has arrived. Product added.
(5) ProductWastedEvent is triggered when a product has reached its expiration date. Product removed.
(6) EODEvent marks the end of the day. The date is moved forward by one day.

Algorithm 1 shows how a day is simulated. Each day follows a fixed high-level order of events.

- The day begins with deliveries (lines 2-5): a ProductDeliveredEvent is created for an arriving product. The delivered product is added to the store's inventory by using event data.
- Next, expiring food is removed from a store's inventory (lines 6-9). A ProductWastedEvent is created which provides a link to the store and the expiring product. The store is then updated via this event, wherein it removes the product from its inventory.
- The store then has the option to order more product (lines 10-15). A store's agent provides a list of order actions it would like to perform. These order actions are then transformed into ProductOrderEvents which update a wholesaler from which the product is purchased and the store. These updates begin the order and delivery process and confirm to the store that the order was accepted.
- Similarly, the store then makes pricing decisions (lines 16-20). An agent generates a list of price actions. From this list, PriceSetEvents are created to record how the store changes the product price.
- At this time, the store becomes open for customers to shop (lines 21-26). A customer is selected until all customer groups have completed their shopping. This customer determines their next purchase, which then is used to form a ProductSoldEvent. This event is then used to update the store to remove the purchased product from inventory. Such implementation allows randomization of customer orders (e.g., different customers purchasing last product leads to different behaviour).
- Once the customers have completed their shopping, an EODEvent is created to update all entities that the day is over. A new day can then be simulated.

Recorder Driving environment changes through events has the benefit of ascertaining simulation correctness and consistency as well as providing an event log for tracking statistics and behaviour across simulations. Statistical histories for each entity can be tracked (e.g., per customer, store, wholesaler, product, etc.). A recorder maintains base statistics across simulations.

Multiprocessing As each simulation run is independent, the simulator features multiprocessing capabilities to leverage many CPU cores to parallelize simulations. This allows for significant simulation speedup.

6 DEMONSTRATION: STRAWBERRIES AND POTATOES

In this section, we show a sample food retail setup demonstrating the use of our simulation platform in the context of a store that sells strawberries and potatoes. These two product types are selected due to available statistics and product differences which are used to model customer and wholesaler behaviour functions. Strawberries and potatoes exhibit different demand over time (including seasonality). Furthermore, strawberries spoil much faster than potatoes. In our simulation, we assume strawberries expire in $\mathcal{N}(8,1)$ days and potatoes in $\mathcal{N}(75,5)$ days.

6.1 Integrating Realistic Consumption Data

We rely on the Statista database for statistical records regarding quantities of strawberries and potatoes sold/consumed to create realistic simulations of customer demand throughout the year. While we merge data from different countries, all are in the Northern hemisphere and should exhibit similar seasonal patterns. As there always exists variation in demand for a particular location, any variations between these data sources should not overwhelmingly affect realism.

Strawberries The strawberry demand model is derived from statistics of fresh strawberry consumption per capita in the U.S. between 2000-2020 (US Department of Agriculture 2022) and the monthly fruit consumption in Spain in 2020 (Spain Ministry of Agriculture, Food and Environment 2022b). We believe that the shape of the fruit consumption demand over time curve is similar to that of strawberry consumption.

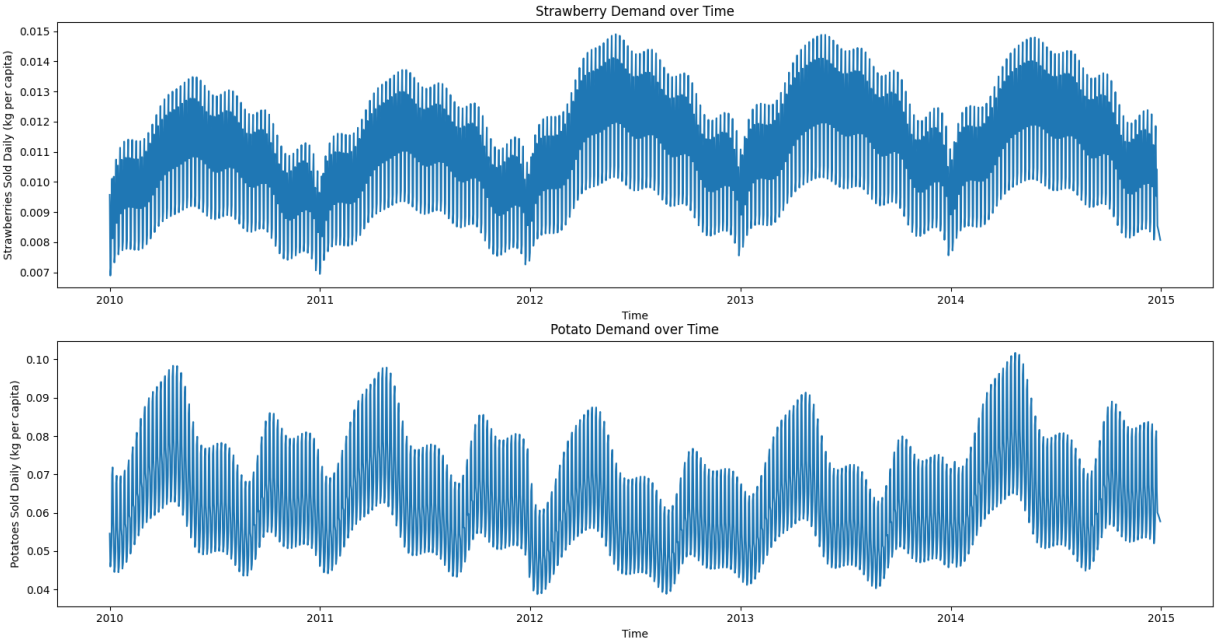

Figure 3: Daily demand functions per customer derived via interpolation from historical statistics.

Potatoes Similarly, the basis of the potato demand function is from yearly potato sales (USDA Foreign Agricultural Service 2022) in Canada 2009-2019, as well as monthly sales by month in Spain in 2020. (Spain Ministry of Agriculture, Food and Environment 2022a).

Interpolation A proper simulation model for the described food retail problem requires simulating daily purchases and therefore requires daily demand. As available statistics are limited in granularity to monthly and yearly reference points, we develop a method utilizing interpolation and optimization. This method determines a series of basis points corresponding to the first day of each month via linear interpolation for the given set of years for the simulation study. The first day of each month then becomes a variable for an optimization function, with the basis points serving as the starting baseline. An SLSQP optimization algorithm is run, wherein the objective is to minimize the difference between the monthly sum from interpolated daily demands and the desired monthly total (based on statistics) (SciPy 2023). As available real monthly statistics are limited to one year, we assume that the general yearly shape of seasonality is similar, but add light noise and interpolate to higher or lower yearly values from the yearly statistics. This provides smooth demand curves with realistic qualities and seasonality. To further enhance realism, we extract usual store traffic for each day of the week from (Google Maps 2022) — assuming that store traffic increases linearly with purchases. Such traffic is incorporated as a weighted window from the total sum of purchases over a week. The resulting demand plot can be seen in Figure 4.

6.2 Simulation Campaign

As a feasibility demonstration and initial evaluation of the simulator, we run a simulation campaign.

Setup We set up the simulation for 1,000 runs with the following entities and parameters. (1) *One wholesaler* whose pricing reflects the average historical weekly farm price for a product (Western Growers 2022). (2) *One customer group* with population of 10,000 whose purchase demand reflects Section 6.1. We model normal demand variation with Gaussian noise and omit any catastrophic unexpected events (e.g., pandemic).

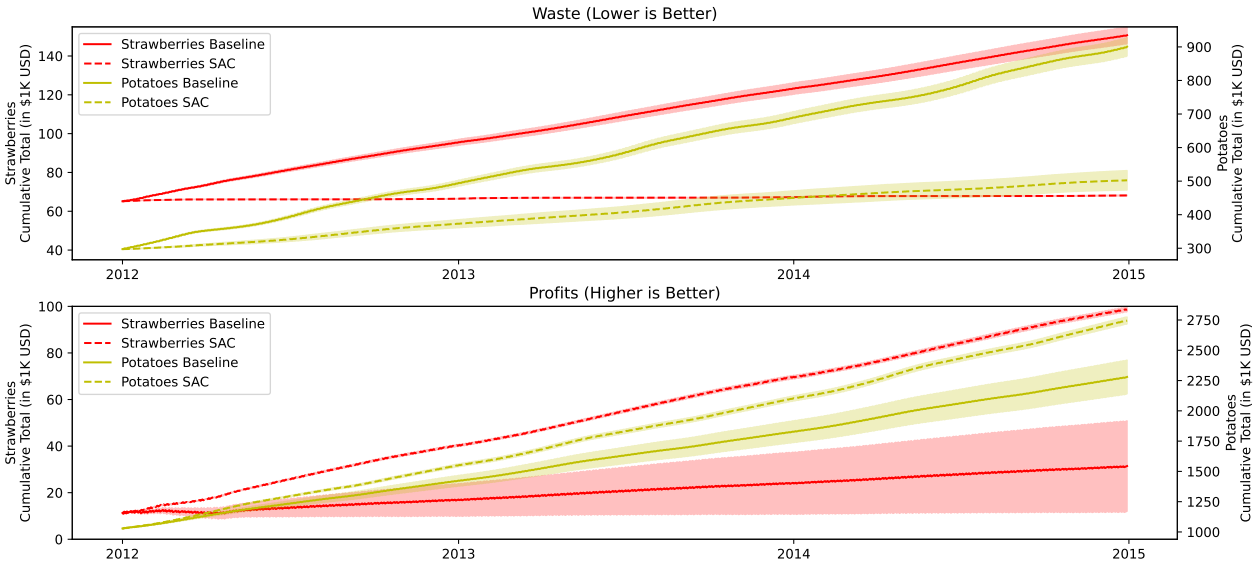

Figure 4: Cumulative waste (top) and cumulative profit (bottom) for strawberries (red) and potatoes (yellow) in USD across 1,000 simulations for the baseline (solid) and SAC (dashed) agents: mean and std. deviation.

(3) *One store* which can only order products on Mondays and Thursdays and set prices on Mondays. (4) The *agent* must set the price of a product to the historical average retail price in Atlanta, USA (Western Growers 2022) for that week, as the present customer demand functions do not incorporate price. Agents are allowed to learn on simulations of days between Jan 1, 2010 and Jan 1, 2012. Then evaluations will take place from Jan 1, 2012 to Jan 1, 2015.

Baseline Agent This agent orders the sum of product purchases from the past week. While this agent is not RL-based, it utilizes the interface available to an RL agent and serves as a simple realistic baseline for future work. It provides an estimation of real sunk cost of waste as it results in typical near 40% food waste.

RL Agent As an example RL agent for this demonstration, we apply Stable-Baseline3's off-the-shelf Soft Actor Critic (SAC) algorithm (Raffin et al. 2021). We set the observed state to be a tuple consisting of the day of the year (*doy*), day of the week (*dow*), the present number of items in an inventory (*inv*), and the quantity customers purchased in the last seven days (*purch*). Each Monday and Thursday the agent observes the tuple $(doy, dow, inv, purch)$ and provides the amount of product to order as an action. The agent then receives a reward which we formulate as food waste subtracted from revenue since the last order. Food waste is measured as sunk cost dollars paid to the wholesaler for the products which were wasted. This agent is trained through repeated simulations of 2010-2012 and evaluated on never experienced 2012-2015.

Results Our simulation results in Figure 4 measure the cumulative sunk cost waste and profits in USD for strawberries and potatoes. The line shading represents the standard deviation of the simulation outcomes following the agent's strategy. For illustration purposes, we assume that the store followed the baseline strategy until Jan 1, 2012. This provides a consistent starting point for both agents, but prevents extraneous plot overlap from both strawberries and potatoes starting at zero profit and waste.

The baseline agent's ordering strategy resulted in approximately 36% of potatoes wasted ($902k USD) and 40% of strawberries wasted ($149k USD) which aligns with general food waste statistics in retail. This adds up to over $1 Million USD of strawberry and potato food waste over 5 years (2010-2015).

The SAC RL agent demonstrates that RL could be an effective strategy to reduce food waste and improve business profitability. The SAC RL agent only wasted 3.5% of strawberries and 14.9% of potatoes. This

amounts to $82k USD and $400k USD less waste than the baseline strategy for strawberries and potatoes, respectively. It improved profits by $67k USD for strawberries and $466k USD for potatoes. Furthermore, despite increasing profitability by $500k+ USD, the standard deviation of expected results was much lower. The SAC RL agent reduced waste by nearly 70% and improved profits by 42% over the baseline.

When utilizing multiprocessing, 1,000 baseline simulations can be completed within 1 minute of computation time on a Mac M1 chip with 8 threads using a Python implementation without just-in-time compilation.

7 DISCUSSION

Demonstration Our demonstration highlights that the simulation framework is capable of properly simulating the food retail environment, handling events, and is ready for RL agents. Food waste, product deliveries, and purchases are properly tracked and interfaces provide proper access for product price and order actions. The customer group is able to select between available products.

Likewise, our RL agent showcases how simulation-trained RL can effectively reduce food waste and improve business profitability. Our RL agent could still undoubtedly be improved by better algorithms, rewards, and state configurations. For example, too few strawberries were sometimes ordered, thereby not maximizing possible revenue. For a real business scenario additional factors may need to be incorporated into reward structures. E.g., to maintain customer satisfaction there should always be products available even though it may result in additional food waste. Nevertheless, it still demonstrates the applicability of simulation-trained RL for food retail.

Currently, customer demand in the demonstration is related purely to historical consumption (correlated with historical pricing). As the prices are set to historical values, this does not affect the example. However, for an example where an RL agent controls product prices, customer behaviour must factor in price in its purchase decisions. Fortunately, as our framework allows for easy customization and configuration, it will be easy to replace the existing demand function with a new one.

Ideally, store demand could be approximated via data analytics from real store data. However, as such data is generally confidential, we have shown the usefulness of our simulator environment and ability to create examples even when store-specific data is not available. Our example extended with realistic customer reactions to price differences could serve as an example to compare RL techniques on a public problem.

Finally, the demonstration shows the business impact food waste can have on food retailers. Around 40% waste is realistic (we do not have concrete product-specific values). With the volume of waste, improvements in pricing and ordering decision-making could have a large business and environmental impact.

Simulator Design We designed our simulator by following a discrete event simulation (DES) framework rather than a discrete EVent system specification (DEVS) framework as our primary objective of RL agent control of store pricing and decision-making can be modeled without the need for continuous time. With DES, we were able to design and test the initial version of the simulation environment more rapidly. As our primary objective is to exploit RL for optimizing pricing and ordering control in stores, the more simple DES framework is sufficient to handle high-level (daily) information. In future work, we aim to extend our simulation environment to a DEVS framework to allow for more fine-grained, continuous-time customer behaviour (e.g., shopping or delivery at particular hours).

RL agents improve with experience. For this reason, faster simulation times result in faster learning: more experience per second of computation. While our simulator exhibits sufficient computation speed for initial experiments, it could still be increased. To further improve on just-in-time compilation with python, one can rewrite parts of the simulator in C++ and connect them to *python* interfaces via *python* wrappers. Further-

more, the simulator can be deployed as a software service with a public interface, which enables a smooth transition towards integrated solutions to be used by individual food retailers.

8 CONCLUSIONS

This paper presents a simulation framework that enables the design and validation of robust reinforcement learning agents to be used in a food retail context to reduce food waste and improve business profitability. The goal of this simulation environment is to use it as a platform to train, validate, and test reinforcement learning agents for reducing food waste and improving retail store profitability. Our discrete event simulation environment simulates the interaction between wholesalers, stores, and customers on a day-by-day basis. Reinforcement learning agents can be used to control entities. During simulation, purchases, deliveries, and food waste are all recorded and available for data analysis. We provide a simulation demonstration consisting of strawberries and potatoes derived from yearly and monthly statistics and interpolation. In this demonstration we show how RL can significantly improve upon baseline strategies. Our SAC algorithm reduced food waste sunk cost by almost $500k USD over 3 years and improved profits by even more. This is a 70% reduction in food waste and 42% improvement in profit for strawberries and potatoes.

In future work, we will extend our simulation to follow a DEVS framework for increased operations realism, expand our example to include demand changes due to pricing, and develop various RL agents (e.g., using multi-armed bandits handling delays (Pilarski, Pilarski, and Varró 2021)) to control pricing and order quantities.

ACKNOWLEDGEMENTS

This research was partially supported by Natural Sciences and Engineering Research Council of Canada Idea-to-Innovation grant (NSERC I2IPJ 576543-22), a TechAccelR (#243794) grant and the Invention To Impact (I-to-I) program of McGill Engine. The third author was also supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

Afridi, M. T., S. Nieto-Isaza, H. Ehm, T. Ponsignon, and A. Hamed. 2020. “A deep reinforcement learning approach for optimal replenishment policy in a vendor managed inventory setting for semiconductors”. In *2020 Winter Simulation Conference (WSC)*, pp. 1753–1764. Institute of Electrical and Electronics Engineers, Inc.

Baydar 2003. “Agent-based modeling and simulation of store performance for personalized pricing”. In *Proceedings of the 2003 Winter Simulation Conference, 2003.*, Volume 2, pp. 1759–1764 vol.2.

Christensen, F. M. M., C. Solheim-Bojer, I. Dukovska-Popovska, and K. Steger-Jensen. 2021. “Developing new forecasting accuracy measure considering Product’s shelf life: Effect on availability and waste”. *Journal of Cleaner Production* vol. 288, pp. 125594.

Chung, J., and D. Li. 2014. “A simulation of the impacts of dynamic price management for perishable foods on retailer performance in the presence of need-driven purchasing consumers”. *Journal of the Operational Research Society* vol. 65 (8), pp. 1177–1188.

David, I., and E. Syriani. 2022. “DEVS Model Construction As A Reinforcement Learning Problem”. In *2022 Annual Modeling and Simulation Conference (ANNSIM)*, pp. 30–41. Institute of Electrical and Electronics Engineers, Inc.

Esmaeili, M., M. Naghavi, and A. Ghahghaei. 2018. "Optimal (R, Q) policy and pricing for two-echelon supply chain with lead time and retailer's service-level incomplete information". *Journal of Industrial Engineering International* vol. 14 (1), pp. 43–53.

Gijsbrechts, J., R. Boute, D. Zhang, and J. Van Mieghem. 2019, 07. "Can deep reinforcement learning improve inventory management? Performance on dual sourcing, lost sales and multi-echelon problems". *SSRN Electronic Journal*.

Google Maps 2022. "Real Canadian Superstore". http://bit.ly/3jhYJa7. Accessed Dec. 28, 2022.

Jandera, A., and T. Skovranek. 2022. "Customer Behaviour Hidden Markov Model". *Mathematics* vol. 10 (8), pp. 1230.

Kim, C. O., J. Jun, J. Baek, R. Smith, and Y.-D. Kim. 2005. "Adaptive inventory control models for supply chain management". *Journal of Advanced Manufacturing Technology* vol. 26 (9), pp. 1184–1192.

Lau, R. S. M., J. Xie, and X. Zhao. 2008. "Effects of inventory policy on supply chain performance: A simulation study of critical decision parameters". *Computers and Industrial Engineering* vol. 55 (3), pp. 620–633.

Lv, H. R., X. X. Bai, W. J. Yin, and J. Dong. 2008. "Simulation based sales forecasting on retail small stores". In *2008 Winter Simulation Conference*, pp. 1711–1716. Institute of Electrical and Electronics Engineers, Inc.

Oroojlooyjadid, A., M. Nazari, L. V. Snyder, and M. Takác. 2017. "A Deep Q-Network for the Beer Game with Partial Information". *Computing Research Repository* vol. abs/1708.05924.

Pilarski, S., S. Pilarski, and D. Varró. 2021. "Delayed reward bernoulli bandits: Optimal policy and predictive meta-algorithm PARDI". *IEEE Transactions on Artificial Intelligence* vol. 3 (2), pp. 152–163.

Raffin, A., A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. 2021. "Stable-Baselines3: Reliable Reinforcement Learning Implementations". *Journal of Machine Learning Research* vol. 22 (268), pp. 1–8.

SciPy 2023. "Scipy.optimize.minimize". *SciPy v1.10.0 Manual*.

Spain Ministry of Agriculture, Food and Environment 2022a. "Spain: monthly consumption of potatoes 2020". http://bit.ly/3TiBWZG. Accessed: 2022-12-20.

Spain Ministry of Agriculture, Food and Environment 2022b. "Spain: monthly fresh fruit consumption 2020". http://bit.ly/3Lo3B9r. Accessed: 2022-12-20.

Sutton, R. S., and A. G. Barto. 2020. *Reinforcement learning: An introduction*. The MIT Press.

Teller, C., C. Holweg, G. Reiner, and H. Kotzab. 2018. "Retail store operations and food waste". *Journal of Cleaner Production* vol. 185, pp. 981–997.

US Department of Agriculture 2022. "U.S. fresh strawberries consumption per capita from 2000 to 2020". http://bit.ly/3YHTcZ7. Accessed: 2022-12-20.

USDA Foreign Agricultural Service 2022. "Consumption of fresh potatoes in Canada 2009-2019". http://bit.ly/3FpQK2F. Accessed: 2022-12-20.

Welling, T. L., L. Q. Noel, and A. Ismail. 2021. "Identifying Potentials and Impacts of Lead-Time Based Pricing in Semiconductor Supply Chains with Discrete-Event Simulation". In *2021 Winter Simulation Conference*, pp. 1–12.

Western Growers 2022. "Produce Price Index". Accessed Dec. 10, 2022.

Zeigler, B. P., T. G. Kim, and H. Praehofer. 2000. *Theory of modeling and simulation*. Academic press.

AUTHOR BIOGRAPHIES

SEBASTIAN PILARSKI is a Ph.D student in the Department of Electrical and Computer Engineering at McGill University. His research interests include software engineering, machine learning, reinforcement learning, and their applications to systems engineering. He has been investigating applications of artificial intelligence in gas turbine design and control with Siemens Energy. His email address is sebastian.pilarski@mail.mcgill.ca.

AMAN SIDHU is an undergraduate Honour's Thesis student at the Department of Electrical and Computer Engineering at McGill University. His research interest includes software engineering, reinforcement learning, quantum computing, and their applications. His email address is aman.sidhu@mail.mcgill.ca.

DÁNIEL VARRÓ is a WASP professor at Linköping University and an adjunct professor at McGill University. He is a co-author of over 190 scientific papers with seven Distinguished Paper Awards, and three Most Influential Paper Awards. He serves on the editorial board of Software and Systems Modeling journal, and served as a program co-chair of MODELS 2021, SLE 2016, ICMT 2014, FASE 2013 conferences. He is a co-founder of the VIATRA open source model query and transformation framework, and IncQuery Labs, a technology-intensive company. His email address is daniel.varro@liu.se.