Actor Based Simulation for Closed Loop Control of Supply Chain using Reinforcement Learning

Extended Abstract

Souvik Barat, Prashant Kumar, Monika Gajrani, Vinay Kulkarni
TCS Research, Pune 411013, India
souvik.barat@tcs.com

Hardik Meisheri, Vinita Baniwal, Harshad Khadilkar
TCS Research, Thane 400607, India
harshad.khadilkar@tcs.com

ABSTRACT

Reinforcement Learning (RL) has achieved a degree of success in control applications such as online gameplay and robotics, but has rarely been used to manage operations of business-critical systems such as supply chains. A key aspect of using RL in the real world is to train the agent before deployment, so as to minimise experimentation in live operation. While this is feasible for online gameplay (where the rules of the game are known) and robotics (where the dynamics are predictable), it is much more difficult for complex systems due to associated complexities, such as uncertainty, adaptability and emergent behaviour. In this paper, we describe a framework for effective integration of a reinforcement learning controller with an actor-based simulation of the complex networked system, in order to enable deployment of the RL agent in the real system with minimal further tuning.

KEYWORDS

Reinforcement learning; Simulation of complex systems; Model based simulation

ACM Reference Format:
Souvik Barat, Prashant Kumar, Monika Gajrani, Vinay Kulkarni and Hardik Meisheri, Vinita Baniwal, Harshad Khadilkar. 2019. Actor Based Simulation for Closed Loop Control of Supply Chain using Reinforcement Learning. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Business-critical systems need to continually make decisions to stay competitive and economically viable in a dynamic environment. Reinforcement Learning (RL) [9, 11] is a class of machine learning algorithms that can be used for controlling such complex systems in an adaptive and flexible manner. The goal of the system controller (also called *RL agent*) is to learn to take the best possible control actions in each possible state of the system, in order to maximise long-term system objectives. A crucial aspect of RL is the computation of next state and associated rewards for the chosen action(s), in a closed loop to enable learning. The setup is illustrated in Figure 1. This paper argues that the use of analytical expressions for modelling the environment is infeasible for complex systems, and advocates an agent/actor based modelling abstraction [1, 8] as an effective modelling aid to understand the dynamics of such complex systems. We present a framework that uses RL for exploring policies and deciding control actions, and actor-based simulation for performing accurate long-term rollouts of the policies, in order to optimise the operation of complex systems. We use the domain of supply chain replenishment as a representative example.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

2 PROBLEM FORMULATION

We illustrate the generic reinforcement learning problem in the context of supply chain replenishment, which presents well-known difficulties for effective control [7, 10]. The scenario is that of a grocery retailer with a network of stores and warehouses served by a fleet of trucks for transporting products. The goal of replenishment is to regulate the availability of the entire product range in each store at all times, subject to the spatio-temporal constraints imposed by available stocks, labour capacity, truck capacity, transportation times, and available shelf space for each product in each store. A schematic of the flow of products is shown in Figure 2.

From operational perspective, each store stocks $i = \{1, \ldots, k\}$ unique varieties of products, each with a maximum shelf capacity $c_{i,j}$ where $j \leq n$ is the index of the store. Further, let us denote by $x_{i,j}(t)$ the inventory of product i in store j at time t. The replenishment quantities (*actions*) for delivery moment d are denoted by $a_{i,j}(t_d)$, and are to be computed at time $(t_d - \Delta)$ where Δ is the lead time. The observation $O(t_d - \Delta)$ consists of the inventory of each product in each store at the time, the demand forecast for each product between the next two delivery moments, and metadata such as unit volume and weight, and shelf life. The inventory $x_{i,j}(t)$ depletes between two delivery moments $(d-1)$ and d, and undergoes a step increase by amount $a_{i,j}(t_d)$ at time t_d.

The reward $r(t_{d-1})$ is a function of the actions $a_{i,j}(t_{d-1})$ and the inventory $x_{i,j}(t)$ in $t \in [t_{d-1}, t_d)$. Two quantities are of particular interest: (i) the number of products that remain available throughout the time interval $[t_{d-1}, t_d)$, and (ii) the wastage of any products

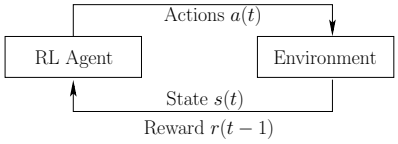

Figure 1: Interaction of RL agent with an environment.

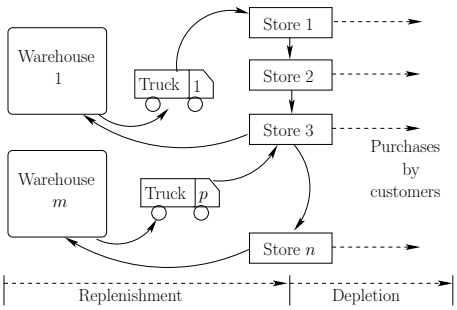

Figure 2: Schematic of supply chain replenishment use case.

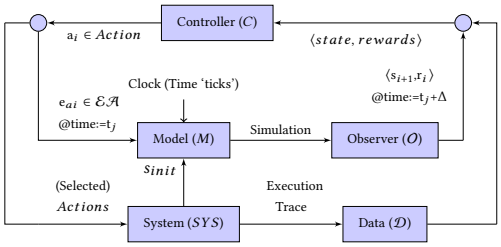

Figure 3: Proposed approach.

that exceed their shelf lives. Mathematically, we define this as,

$$r(t_{d-1}) = 1 - \frac{\text{count}(x_{i,j} < \rho)}{k\,n} - \frac{\sum_{i=1}^{k}\sum_{j=1}^{n} w_{i,j}(t_{d-1})}{\sum_{i=1}^{k}\sum_{j=1}^{n} X_{i,j}}, \quad (1)$$

where $\text{count}(x_{i,j} < \rho)$ is the number of products that run out of inventory (drop below fraction ρ) at some time $t \in [t_{d-1}, t_d)$, $w_{i,j}(t_{d-1})$ is the number of units of product i in store j that had to be discarded in the time interval because they exceeded their shelf lives, and $X_{i,j}$ is the shelf capacity for product i in store j.

3 METHODOLOGY

A reinforcement learning problem is described by a Markov Decision Process (MDP) [11] represented by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, P, \gamma)$. Here, $\mathcal{S}$ is the set of states of the system, $\mathcal{A}$ is the set of control actions, $\mathcal{R}$ is the set of possible rewards, P is the (possibly stochastic) transition function from $\{\mathcal{S}, \mathcal{A}\} \rightarrow \mathcal{S}$, and γ is a discount factor for future rewards. In several cases, the agent is unable to observe the state space entirely, resulting in a partially-observable MDP or POMDP [11]. Observations $\mathcal{O}$ are derived from $\mathcal{S}$ to represent what the agent can sense. The RL agent should compute a policy $\mathcal{O} \rightarrow \mathcal{A}$ that maximises the discounted long-term reward. We use a form of RL known as A2C [6] to compute the actions. The Critic evaluates the *goodness* of the current system state, while the Actor chooses an action that maximises the improvement in value in the next state.

We propose an actor based simulation framework [4] for training the RL agent in a synthetic environment as shown in Figure 3. The proposed framework contains two control loops: (i) a model centric loop for mapping $\mathcal{A} \rightarrow \mathcal{O}$ based on the actions of the RL agent and their effect on the system, and (ii) a real time control loop. We consider an extended form of *actor* model [3] to closely mimic the complex systems; and adopt simulation as an aid [2] to compute micro-behaviours and observe emerging macro behaviours, overall system *state*, *observations*, and *rewards* over time.

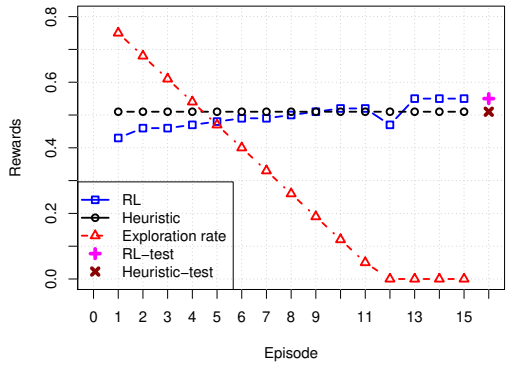

Figure 4: Evolution of rewards during training.

4 EXPERIMENTS AND VALIDATION

We use a data set spanning one year derived from a public source [5] for experimentation. A total of 220 products were chosen from the data set, and their meta-data (not originally available) was input manually. The time between successive delivery moments was set to 6 hours (leading to 4 deliveries per day). The lead time Δ was 3 hours. Forecasts were computed using a uniformly weighted 10-step trailing average for each product. The store capacity, truck volume and weight capacity, and labour counts were computed based on the order volumes seen in the data. We deliberately set the truck volume constraint such that the average order numbers would severely test the transportation capacity of the system. The initial normalised inventory level for each product is set to 0.5 at the start of each training 'episode', and the level below which penalty is imposed is set to $\rho = 0.25$. Of the order data set, the first 225 days (900 delivery moments) were used for training, while the remaining 124 days (496 delivery moments) were retained for testing.

Figure 4 shows the training of the reinforcement learning algorithm in conjunction with the actor-based simulation, over 15 episodes each spanning the 900 delivery moments from the training data set. The average reward, computed over all 220 products and all DM, is seen to increase as training proceeds. The reward is compared with a simplified version of an industry-standard replenishment heuristic, which aims to maintain the inventory levels of all products at a constant level. We see that the reward at the end of the training exercise exceeds the heuristic performance, and this advantage is retained on the test data set as well (plotted using separate markers at the ends of the curves).

5 SUMMARY

An efficient learning framework with realistic model is argued necessary to control complex business systems. A control framework that uses reinforcement learning and an actor-based simulation is presented to support our argument. Initial evaluations show that training and policy evaluation of RL agent using proposed approach is feasible (in terms of computational time and expense) and effective as compared to traditional aggregated analytical models.

REFERENCES

[1] Gul Agha. 1986. *Actors: A Model of Concurrent Computation in Distributed Systems.* MIT Press, Cambridge, MA, USA.

[2] Souvik Barat, Vinay Kulkarni, Tony Clark, and Balbir Barn. 2017. A method for effective use of enterprise modelling techniques in complex dynamic decision making. In *IFIP Working Conference on The Practice of Enterprise Modeling*. Springer, 319–330.

[3] Souvik Barat, Vinay Kulkarni, Tony Clark, and Balbir Barn. 2018. A Model Based Approach for Complex Dynamic Decision-Making. In *Communications in Computer and Information Science*, Vol. 880. Springer, 94–118.

[4] Tony Clark, Vinay Kulkarni, Souvik Barat, and Barn Barn. 2017. ESL: an actor-based platform for developing emergent behaviour organisation simulations. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*.

[5] Kaggle. Retrieved 08-2018. Instacart Market Basket Analysis Data. https://www.kaggle.com/c/instacart-market-basket-analysis/data. (Retrieved 08-2018).

[6] Vijay R Konda and John N Tsitsiklis. 2000. Actor-critic algorithms. In *Advances in neural information processing systems*. 1008–1014.

[7] Hau L Lee, Venkata Padmanabhan, and Seungjin Whang. 1997. Information distortion in a supply chain. *Management science* 43, 4 (1997), 546–558.

[8] Charles M Macal and Michael J North. 2010. Tutorial on agent-based modelling and simulation. *Journal of simulation* 4, 3 (2010), 151–162.

[9] Stuart J Russell and Peter Norvig. 2016. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

[10] Ehap H Sabri and Benita M Beamon. 2000. A multi-objective approach to simultaneous strategic and operational planning in supply chain design. *Omega* 28, 5 (2000), 581–598.

[11] R Sutton and A Barto. 2012. *Reinforcement learning: Introduction*. MIT Press.