Atari-fying the Vehicle Routing Problem with Stochastic Service Requests

Nicholas D. Kullman
Universitè de Tours, LIFAT EA 6300, CNRS, ROOT ERL CNRS 7002, Tours, France, nicholas.kullman@etu.univ-tours.fr
Jorge E. Mendoza, Martin Cousineau
HEC Montréal and CIRRELT, Montréal, Canada, {jorge.mendoza, martin.cousineau}@hec.ca
Justin C. Goodson
Richard A. Chaifetz School of Business, Saint Louis University, Saint Louis, MO, goodson@slu.edu

1. Motivation

Deep reinforcement learning (RL) has seen much recent success in tasks that involve sequential decision making under uncertainty. Perhaps the most well-known application domain of deep RL is gaming, having been used to train agents capable of superhuman performance in, e.g., Go (Silver et al. 2018), StarCraft II (Vinyals et al. 2019), and, of interest here, Atari (Mnih et al. 2015).

For Atari, Mnih et al. established a single solution method — a single agent architecture — capable of outperforming humans on a majority of Atari games. The use of a single architecture to accomplish this feat is notable as these games differ in their appearance, goals, rewards, actions, etc. Given the diversity of the games on which this architecture was successful, we pose the question of whether it would perform comparably well on any game with a similar (Atari-like) format. If this is true, it presents the opportunity of using this agent architecture to solve research problems, permitted that they can be *Atari-fied*, formulated as an Atari-like game.[1] We explore this opportunity here for the well-studied vehicle routing problem with stochastic service requests (VRPSSR) (e.g., Gendreau et al. (1999), Bent and Van Hentenryck (2004), and Ulmer et al. (2018)). Our primary contributions are 1) the introduction of the concept of Atari-fying research problems, 2) demonstrating the Atari-fication of the VRPSSR, and 3) (by way of (2)), offering the first application of deep RL to this problem. Broadly, we hope to show that there is an opportunity to extend the groundbreaking achievements from the deep reinforcement learning community to research problems in other domains through reformulations of classical problem models.

[1] We do not provide a precise definition of what makes a game *Atari-like*, but broadly mean that it is a two-dimensional third-player game with a total pixel count on the order of 10^5 or less.

2. Problem

Deep RL recap. As defined in Sutton, Barto et al. (1998), RL refers to the process through which an agent, sequentially interacting with some environment, learns what to do (i.e., a *policy* mapping states to actions) so as to maximize a numerical reward signal from the environment. Often the agent seeks to learn the value of choosing a particular action a from some state s, known as the state-action pair's *Q-value* ($Q(s,a)$), equal to the immediate reward plus the expected sum of future rewards earned by taking action a from state s. With knowledge of Q-values for all state-action pairs, the agent's policy is then to choose the action with the largest Q-value. In practical problems, however, the number of unique state-action pairs is too large to learn and store a value for each, so a functional approximation of these Q-values is learned. When deep artificial neural networks (ANNs) are used for this approximation, the method is called *deep reinforcement learning.*

Problem statement. In the VRPSSR, the agent is responsible for dynamically routing a vehicle to serve customer service requests that arise randomly throughout some service region over the course of a work day of duration T. The locations of potential customer requests are known in advance, but it is not known which customers will request service or when they will do so. The problem begins with the vehicle at the depot at time 0, and it must return to the depot before the end of the day at time T. The agent moves the vehicle among the customers $\mathcal{N} = \{1, \ldots, N\}$ and the depot (defined to be node 0) via edges in some underlying graph (which varies by formulation, as discussed below). We assume known fixed travel times along the edges. At each step, beginning at time $t = 0$, the agent either instructs the vehicle to wait at its current location, or it selects an adjacent node in the graph to which to move. The next step occurs either when the vehicle arrives to a new node (agent chose to move) or after some predefined waiting time $\bar{t}$ (agent chose to wait). If the vehicle visits a node representing a customer that is currently requesting service, then the customer is marked as served, and the vehicle earns a reward. The agent's objective is to find a policy that dynamically routes the vehicle so as to maximize the expected sum of rewards.

Atari-fied graph formulation. In the traditional VRPSSR formulation, the agent moves the vehicle among the customers and depot via edges in the complete graph G with vertices $\mathcal{N} \cup \{0\}$. Conversely, in the Atari-fied formulation the agent moves the vehicle along edges in the graph G', a Manhattan-style grid representation of G, where some nodes (intersections) represent customer locations (see Figure 1). We note the opportunities for additional dynamism in the Atari-fied formulation, as movements between customers can be effectively pre-empted at each intersection.

Atari-fied state representation. The primary difference between the traditional and Atari-fied formulations is the representation of the state that the agent receives to perform decision-making. In the traditional formulation, the state is a tuple consisting of the vehicle's current

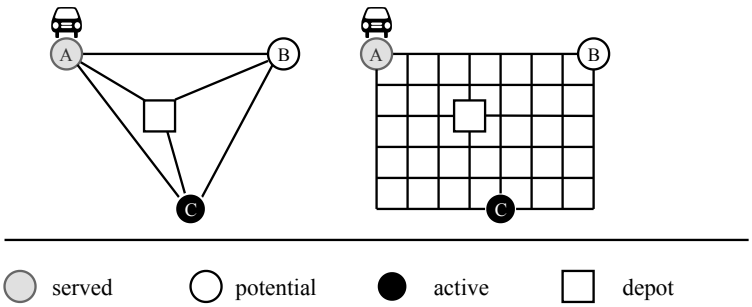


Figure 1 **Analogous graphs for the traditional (left) and Atari-fied (right) formulations of the VRPSSR.**

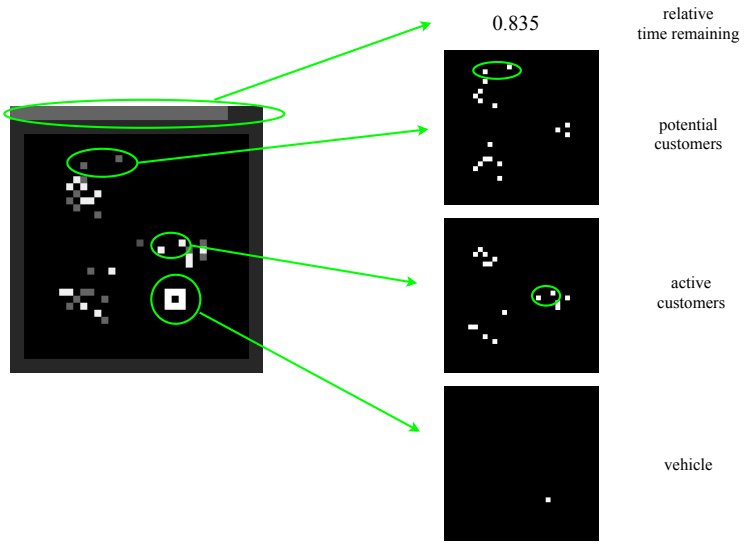


Figure 2 **Two different visual representations of the state in the Atari-fied formulation. At left is a frame from the VRPSSR game; at right is its corresponding feature layer representation.**

location, the time t, the customers currently requesting service, and the customers that have not yet requested service. With Atari-fication, the agent receives the same information but represented visually. The visual rendering of the game that we use in our Atari-fication of the VRPSSR is shown at left in Figure 2. It includes the playable area — a simplified depiction of G' with each pixel representing a node, bordered by its adjacent nodes from G' — surrounded by a thin border and topped with a bar representing the relative amount of time remaining ($\frac{T-t}{T}$). In practice we use a *feature layer* representation of the state (at right in Figure 2), which preserves core visual and spatial information while separating game elements from one another.

3. Results

Computational experiments are summarized in Figure 3. The agent successfully learns and improves its performance in the VRPSSR environment, eventually achieving a mean score of 125.02 over the last 5000 training episodes. While we suspect these results do not yet compete with more traditional methods, they show promise in the process of Atari-fication, at least for the VRPSSR.



Figure 3 **Agent's performance over 25000 training episodes.**

4. Conclusion

We present a new general approach to modeling research problems as Atari-like videogames to make them amenable to recent celebrated solution methods from the deep reinforcement learning community. The approach is flexible, applicable to a wide range of problems. We demonstrate its application on a well known VRP. Our preliminary results on this problem, though not transformative, show signs of success and suggest that Atari-fication may be a useful modeling approach for researchers studying problems involving sequential decision making under uncertainty.

References

Bent RW, Van Hentenryck P, 2004 *Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research* 52(6):977–987.

Gendreau M, Guertin F, Potvin JY, Taillard E, 1999 *Parallel tabu search for real-time vehicle routing and dispatching. Transportation science* 33(4):381–390.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al., 2015 *Human-level control through deep reinforcement learning. Nature* 518(7540):529.

Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, et al., 2018 *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science* 362(6419):1140–1144.

Sutton RS, Barto AG, et al., 1998 *Introduction to reinforcement learning*, volume 2 (MIT press Cambridge).

Ulmer MW, Goodson JC, Mattfeld DC, Hennig M, 2018 *Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. Transportation Science* 53(1):185–202.

Vinyals O, Babuschkin I, Czarnecki WM, Mathieu M, Dudzik A, Chung J, Choi DH, Powell R, Ewalds T, Georgiev P, et al., 2019 *Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature* 1–5.