Benchmarks for Online Stochastic Operations Research Problems

Anonymous Author(s)
Affiliation
Address
email

Abstract

Benchmarks to compare reinforcement learning algorithms with state of the art for solving online stochastic operations research problems

1 Introduction

Operations Research is an awesome field. Many online stochastic problems exist in practice but solutions make simplistic assumption. Reinforcement learning is a good fit to these problems and there have been some initial results. However, no standard set of problems or algorithms to compare against exist. We prepare these benchmarks to compare different algorithms and push the community towards developing better ones.

2 Related Work

3 Bin Packing

In the classic bin packing problem, we fit given items of varying sizes into as few fixed size bins as possible. In the online stochastic version of this problem, each item arrives one at a time and its size varies as per an unknown distribution. We formulate the problem as a Markov Decision Process and compare reinforcement learning algorithms against a well known baseline called Sum of Squares that asymptotically converges to a good solution regardless of the item size distribution.

3.1 Related Work

Online stochastic bin packing [**?**]

3.2 Problem Formulation

Items can be of different types $j \in \{1, ..., J\}$. The size of type j is s_j and the probability that an item is of type j is p_j. Without loss of generality, we assume item types are enumerated in the increasing order of their size: $s_1 < s_2 < ... < s_J$. Items arrive one at a time and are packed into bins of size B, where $s_J < B < \infty$. We assume the item sizes s_j and bin size B are integers. We assume the number of bins are unlimited and denote the sum of item sizes in a bin as *level* h. After n items have been packed, we denote the number of bins at level h as $N_h(n)$, where $h = 1, ..., B$.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

Our objective is to reduce the number of non-empty bins. We can achieve this by minimizing the total waste (i.e. empty space) in the partially filled bins. Hence our objective is to minimize waste at any point in time:

$$W(n) \triangleq \sum_{h=1}^{B-1} N_h(n)(B-h) \tag{1}$$

3.3 Baseline Algorithm

3.4 Reinforcement Learning Algorithm

3.5 Results

4 Newsvendor

5 Vehicle Routing

5.1 Literature Review

5.2 Problem Formulation

We consider a version of VRP that is of a food delivery driver (e.g. using Amazon Restaurants). Orders arrive at the driver's phone app over time in a dynamic manner over time. Each order has a reward (e.g. delivery fee and tip) associated with it and it is assigned to a specific restaurant in the city. "City" here means the whole Euclidean space in which the VRP problem lives. Orders arrive according to a Poisson process and the rate depends on the region of the city that the order is created in. Also, the reward of an order comes from a Gamma distribution, again, with parameters specific to the region. The driver has to accept an order and pick the items up from the restaurant the order is mapped to. The order needs to be delivered within a time window since its creation, which is imposed as a hard constraint. If the driver does not accept a specific order, it remains open for a while and then disappears according to a probability distribution, meaning that either it has expired or accepted by some other driver. There is a capacity on the number of orders a driver can carry in the vehicle, however there is no limit on the number of orders that are accepted (but not delivered) by the driver at the same time. Finally, there is a cost associated with travel The driver's goal is to maximize the average reward over an infinite horizon.

This problem is known as stochastic and dynamic capacitated vehicle routing problem with pick up and delivery, time windows and service guarantee. (SDPDPTW with service guarantee).

5.3 Baseline Solution

The problem for a given set of orders can be expressed and solved using the following Mixed Integer Programming formulation.

Sets

V : Current vehicle location, $V = \{0\}$
P : Pick up nodes (copies of the restaurant nodes, associated with the orders that are not in transit)
D : Delivery nodes representing the orders that are not in transit
A : Nodes representing the orders that are accepted by the driver; $A \subset D$
T : Delivery nodes representing the orders that are in transit
R : Nodes representing the restaurants, used for final return)
N : Set of all nodes in the graph, $N = V \cup P \cup D \cup T \cup R$
E : Set of all edges, $E = \{(i,j), \forall i,j \in N\}$

Decision variables

$$
\begin{aligned}
x_{ij} &: \text{Binary variable, 1 if the vehicle uses the arc from node } i \text{ to } j\text{, 0 otherwise; } i, j \in N \\
y_i &: \text{Binary variable, 1 if the order } i \text{ is accepted, 0 otherwise; } i \in P \\
Q_i &: \text{Auxiliary variable to track the capacity usage as of node } i\text{; } i \in N \\
B_i &: \text{Auxiliary variable to track the time as of node } i\text{; } i \in N
\end{aligned}
$$

Parameters

$$
\begin{aligned}
n &: \text{Number of orders available to pick up, } n = |P| \\
C_{ij} &: \text{Symmetric Manhattan distance (in miles) matrix between node } i \text{ and } j\text{, } (i, j) \in E \\
q_i &: \text{Supply (demand) at node } i\text{, } q_0 = |T|; q_i = 1, \forall i \in P; q_i = -1, \forall i \in D \cup T; q_i = 0 \in R \\
m &: \text{Travel cost per mile} \\
r_i &: \text{Revenue for order associated with pick up node } i\text{, } i \in P \\
U &: \text{Vehicle capacity} \\
M &: \text{A very big number} \\
t &: \text{Time to travel one mile} \\
d &: \text{A constant positive service time per stop}
\end{aligned}
$$

Model

$$
\begin{aligned}
\underset{x, y, Q, B}{\text{maximize}} \quad & \sum_i r_i y_i - m \sum_{(i,j)\in E} C_{ij} x_{ij} && && \text{(2a)} \\
\text{subject to} \quad & \sum_{j\in N} x_{ij} = y_i && \forall i \in P, && \text{(2b)} \\
& \sum_{j\in N} x_{ij} - \sum_{j\in N} x_{i+n,j} = 0 && \forall i \in P, && \text{(2c)} \\
& y_i = 1 && \forall i \in A, && \text{(2d)} \\
& \sum_{j\in N} x_{ij} = 1 && \forall i \in T, && \text{(2e)} \\
& \sum_{j\in N} x_{0j} = 1, && && \text{(2f)} \\
& \sum_{j\in N\setminus R} x_{ji} = 1 && \forall i \in R, && \text{(2g)} \\
& \sum_{j\in N\setminus R} x_{ji} - \sum_{j\in N} x_{ij} = 0 && \forall i \in P \cup D \cup T, && \text{(2h)} \\
& Q_i + q_j - M(1 - x_{ij}) \leqslant Q_j && \forall i, j \in N, && \text{(2i)} \\
& \max(0, q_i) \leqslant Q_i && \forall i \in N, && \text{(2j)} \\
& \min(U, U + q_i) \geqslant Q_i && \forall i \in N, && \text{(2k)} \\
& B_i + d + C_{ij}t - M(1 - x_{ij}) \leqslant B_j && \forall i, j \in N, && \text{(2l)} \\
& B_i + C_{i,j+n}t \leqslant B_{i+n} && \forall i \in P, && \text{(2m)} \\
& x_{ij} \in \{0, 1\} && \forall i, j \in N, && \text{(2n)} \\
& y_i \in \{0, 1\} && \forall i \in P && \text{(2o)}
\end{aligned}
$$

6 Conclusion

Amazing! Awesome results! Accept!

Acknowledgments

Do not include acknowledgments in the anonymized submission, only in the final paper.