Uso de Q-Learning en el Problema de Entrega de Concreto

Lic. Arnoldo Del Toro Peña

[1]Facultad de Ingeniería Mecánica y Elécctrica, Universidad Autónoma de Nuevo León

* Correo electrónico: arnoldo.toropn@uanl.edu.mx

Palabras claves— Q-Learning, CDP, Optimización, Concreto.

Desde un punto general los proveedores de concreto tienen múltiples problemas a los que se enfrentan, por ejemplo la adquisición de las materias primas, la entrega del concreto, administración de los conductores etc. En este artículo se presenta el Problema de Entrega de Concreto (Concrete Delivary Problem) que se centra en la parte logística y de distribución de la operación, dicho de otra manera: la planificación y el enrutamiento del concreto. El objetivo es encontrar rutas que cumplan con múltiples visitas (solo si es necesario) a diferentes depósitos de producción de concreto utilizando una flota de vehículos (heterogéneos) y cumpliendo con las entregas a los distintos sitios de de construcción, a todo esto se adhiere una planificación y restricciones de enrutamiento. Los parámetros se presentan en la tabla 1:

Parámetros	**Descripción**
P	Conjunto de sitios de producción.
C	Conjunto de Clientes.
0, $n+1$	Inicio y final en depósitos de los camiones.
V	$V = P \cup C \cup 0 \cup n+1$
K	Conjunto de camiones.
q_i	Concreto solicitado por el cliente $i \in C$.
q_k	Capacidad del camión $k \in K$.
p_k	Tiempo requerido para vaciar el camion $k \in K$.
a_i, b_i	Ventana de tiempo en la cual se tiene que realizar la entrega al cliente $i \in C$.
t_{ij}	Tiempo para viajar de i a j, $i, j \in V$.
γ	Tiempo máximo de espera entre dos entregas consecutivas.

Tabla 1: Parámetros

En la actualidad el Problema de Entrega de Concreto (Concrete Delivary Problem) se ha presentado en literaturas de maneras muy diferentes y de cantidad considerable, sin embargo sus amplias definiciones, variantes y su dificultad para obtener datos de manera pública tiene como consecuencia un obstáculo al momento de realizar sus comparaciones, por lo tanto utilizamos instancias de acceso público, y se toman dos enfoques: uno metaheurístico y otro exacto, para verificar los resultados obtenidos en el algoritmo Q-Learning implementado.

El algoritmo Q-Learning sigue una actualización por medio de la ecuación:

$$new\, Q \leftarrow Q(s, a, \theta_M) + \alpha(r + \gamma \underbrace{max}_{a'} Q(s', a', \theta_T) + Q(s, a, \theta_M))$$

se inicia el búfer de reproducción R en la cual definimos un estado inicial s_0 (vacío), se inicializan los pesos del modelo θ_M y los pesos del *Target* $\theta_T = \theta_M$, obtenemos el conjunto A como las acciones no enmascaradas para el estado s_t, si el conjunto es diferente del vacío entonces en base a una probabilidad ϵ seleccionaremos una acción $a_t \in A$ de lo contrario $a_t = \text{argmax}\, Q(s_t, a, \theta_M)$. Aplicamos la acción seleccionada, conseguimos la recompensa r_t y el estado s_{t+1}. Salvamos (s_t, a_t, r_t, s_{t+1}) en R. Realizamos un mini lote de muestra β desde R en este mini lote aplicaremos la ecuación 1, se realiza el descenso de gradiente por medio de $(\text{new}\, Q - Q(s, a, \theta_M))^2$ para actualizar θ_M. Cuando se termina el mini lote asignamos $s_{t+1} \leftarrow s_0$ (reiniciamos desde un estado vacío). Por último cada K iteraciones $\theta_T = \theta_M$.

Un estado s contiene información parcial de la solución, una acción corresponde a la visita al cliente c en el tiempo t, cuando la acción es aplicada, el agente escoge un vehículo para ejecutar la acción y el próximo estado es creado. $Q(s, a, \theta_M)$ es el valor Q obtenido usando la red neuronal *Model*. $Q(s, a, \theta_T)$ es el valor Q obtenido usando la red neuronal *Target*. La red neuronal *Target* es actualizada con los pesos de la red neuronal *Model* cada K iteraciones. La red neuronal *Model* es actualizada cada iteración usando el mini lote. La acción de enmascarar garantiza la factibilidad de la solución explorada.

Pasamos a la parte del constructor de la mejor solución, $S \leftarrow$ solución vacía y un estado s vacío, obtenemos el conjunto de acciones no enmascaradas A para el estado s, mientras el conjunto $A \neq \emptyset$ seleccionamos la acción a mediante

$$\underbrace{argmax}_{a' \in A} Q(s_t, a', \theta_M)$$

aplicamos la acción a y obtenemos la recompensa r, el siguiente estado s', y asignamos el vehículo v, salvamos $S \leftarrow S \cup \{(a, v)\}$, $s \leftarrow s'$ y actualizamos A como el conjunto de acciones no enmascaradas para el estado s, y por último recuperamos S.