INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTERAD 2022-06-16

EXAMENSARBETE "First return, then explore" Adapted and Evaluated for Dynamic Environments
STUDENTER Nicolas Petrisi, Fredrik Sjöström
HANDLEDARE Hampus Åström (LTH), Volker Krueger (LTH)
EXAMINATOR Elin A. Topp (LTH)

Go-Explore anpassad för dynamiska miljöer

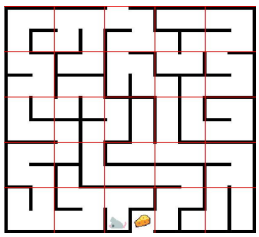
POPULÄRVETENSKAPLIG SAMMANFATTNING **Nicolas Petrisi, Fredrik Sjöström**

Genom förbättringar på den toppmoderna algoritmen Go-Explore fungerar den nu för dynamiska miljöer med mer än sju gånger bättre resultat. Dynamiska celler och "On The Fly"- (OTF) vägar ger Go-Explore förmågan att kunna navigera genom en labyrint med slumpade startpositioner, vilket den inte kunnat göra förr.

Go-Explore är en toppmodern algoritm inom förstärkningsinlärning gjord för att effektivt utforska miljöer. Den visar enastående resultat när den spelar Atari-spel såsom Montezuma's Revenge och Pitfall och i en robotsimulator klarar den att placera objekt i hyllor även när några av hyllorna har en hasp.

När Go-Explore-agenten navigerar genom miljön sparar den undan platserna den hittar i olika "celler". Alla tillstånd som är lika varandra, såsom att de är nära varandra på skärmen i en 2D-miljö, sägs tillhöra samma cell. Detta gör att en cell representerar flera olika tillstånd. När agenten utforskar och går genom dessa sparar den undan vägen den gått för att komma ihåg hur den har kommit fram till de olika cellerna.

Men problemet med Go-Explore är att den är byggd för att alltid börja och sluta på samma position, vilket inte alltid är verklighetstroget. När man flyttar på antingen start- eller slut-positionen fungerar inte längre algoritmen då vägarna som algoritmen sparar undan alltid antar att den börjar på samma position. Vidare så kan man inte heller anta att bara för att två platser på skärmen är nära varandra så behöver platserna i sig inte vara lika, vilket antas i orginalversionen. Att vara på ena eller andra sidan av en vägg i en labyrint kan



ha enorm betydelse för om man är nära att lösa labyrinten eller inte, som man kan se i figuren där det röda rutnätet delar upp miljön i sina celler.

Två stora ändringar i algoritmen är gjorda för att anpassa den till dynamiska miljöer. Närliggande celler slås ihop under körning vilket gör att de kan användas i områden där det är svårt att definiera bra celler av större storlek innan agenten undersökt området. Och istället för att komma ihåg exakt vilka celler agenten ska gå igenom för att komma till sitt mål så skapas OTF-vägar genom att kolla på vilka celler som är grannar för att bygga vägar som kan gå till målet oavsett var man startar.

Med ändringarna lyckas den anpassade Go-Explore lösa labyrinter av olika storlekar med slumpade startpositioner mer än sju gånger oftare än Go-Explore utan ändringarna.