DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Aprendizaje por Refuerzos

2012 **Aprendizaje por Refuerzos**

Integrante	LU	Correo electrónico
Mariano De Sousa	??	marian_sabianaa@hotmail.com
Mariano Bianchi	92/08	marianobianchi08@gmail.com
Pablo Brusco	527/08	pablo.brusco@gmail.com

1. Problema

Para este trabajo, se utilizo una versión simplificada del Tower Blocks, el cual se basa en un edificio de bloques al que hay que agregarle nuevos pisos utilizando una grúa. La dificultad radica en que el jugador no puede manejar la grúa, solo puede ejecutar la acción de soltar un nuevo piso sobre el edificio ya construido. De acuerdo a la presición con que se deposite el piso, la torre gana o pierde estabilidad.

Ya que el juego tiene una gran variedad de niveles y variaciones, se opto por un modelo simple en el cual tenemos que agregar pisos (bloques) a un edificio intentando llegar a una cierta cantidad y sin haber hecho que se derrumbe. Para lo cual existe una grúa que se mueve de manera **constante** sobre el edificio y nos permite ejecutar la acción de tirar o no tirar.

2. Ambiente

Para la implementación del ambiente La grúa es un objeto que tiene un movimiento lineal entre dos posiciones (en este caso -49 y +49), a velocidad constante (1) se mueve en una u otra dirección, de donde se tirará el nuevo piso a agregar.

Por otro lado, el edificio, posee un movimiento pendular que va variando en velocidad de acuerdo a que tan desbalanceado está, debido a los tiros previos del jugador.

El ambiente, mantiene un estado visible para el exterior que contiene, la posición y velocidad de la torre, y además, la posición y dirección de la grúa.

Dados los parametros que mantiene el ambiente, la cantidad de estados está dada por:

$$\#estados = \#posiciones\_grua * \#direcciones\_grua * \#posiciones\_torre * \#velocidad\_torre$$

Que dada la configuración que se utilizo, serian:

$$\#estados = 99 * 2 * 99 * 11 = 215622$$

Aunque no todos ellos son alcanzables dependiendo de otro factor. Este otro factor es el ángulo máximo que se admite que tenga la torre antes de colapsar y que fuimos variando para encontrar buena dinamica, dejandolo en 30 grados.

3. Agentes

Para implementar los jugadores, utilizamos 2 tipos de agentes. Un agente que aprende utilizando la tecnica de Q-Learning y otro que aprende usando el algoritmo Sarsa-λ (ambos utilizando los algoritmos vistos en clase). Estos agentes, recibian estimulos según sus posibles acciones:

- Tirar
- Pasar

En donde el ambiente podia responder con distintos refuerzos que contemplaban los siguientes casos:

- Refuerzo por pasar (negativo chico)
- Refuerzo por fallar y no golpear a la torre (negativo medio)
- Refuerzo por pegarle a la torre y lograr que se caiga (negativo grande)
- Refuerzo por tiro exitoso (golpeó la torre y se agregó un nuevo piso)

Veamos algunas comparaciones en función de ciertos parametros que fuimos variando para estudiar

3.1. Experimentos y comparaciones

Para todos los experimentos normalizamos el ambiente para que sea un edificio de 20 pisos al que hay que llegar como objetivo del juego.

La primera de las preguntas que nos hicimos fue, que funcionara mejor, Q-Learning o Sarsa-λ para parametros similares.

Para ello corrimos el siguiente experimento:

Agente	ϵ	γ	α	λ
Q-Learning	0.001	0.8	0.7	-
Sarsa-λ	0.001	0.8	0.7	0.001
Sarsa-λ	0.001	0.8	0.7	0.3
Sarsa-λ	0.001	0.8	0.7	0.7

Obteniendo el siguiente resultado: