

# Programación Estadística: Búsqueda Montecarlo

**Adrián Sosa**

*Universidad Veracruzana*

## *Búsqueda Montecarlo*

Monte Carlo es un metodo numérico muy versatil y facil de implementar, se puede aplicar a problemas de  $N$ -dimensiones, en contraste con búsqueda en malla, el método consiste en la elaboración de  $N$  puntos aleatorios usando una distribución de probabilidad sobre el dominio del problema, La complejidad del esfuerzo computacional es  $\Phi(N)$ .

El árbol de búsqueda de Monte Carlo, es un algoritmo de búsqueda heurístico para algunos tipos de procesos de toma de decisiones sobre todo los que trabajan con juegos. Así como el método de operación el enfoque de búsqueda de Monte Carlo se encuentra en el análisis de los movimientos mas prometedores, ampliando el árbol de búsqueda, basado en un muestreo aleatorio.

## *Ventajas*

- Se puede aplicar a juegos con un factor de ramificación tan grande que resultan inabordables con las técnicas tradicionales.
- Se puede configurar para que se detenga después de una cantidad de tiempo prefijada, haciendo que con tiempos mayores se obtengan jugadores más fuertes.
- No requiere necesariamente de un conocimiento específico del juego, por lo que puede ser utilizado como motor de juego general; y es adaptable a juegos que incorporan aleatoriedad en las reglas.

### Procedimiento

*Selección:* empezar desde la raíz  $R$  y seleccionar nodos hijos sucesivos hasta alcanzar un nodo hoja  $L$ . La selección describe una manera de elegir los nodos hijos, que permitan que el árbol se expanda hacia movimientos mas prometedores, que es la esencia del árbol de búsqueda Monte Carlo.

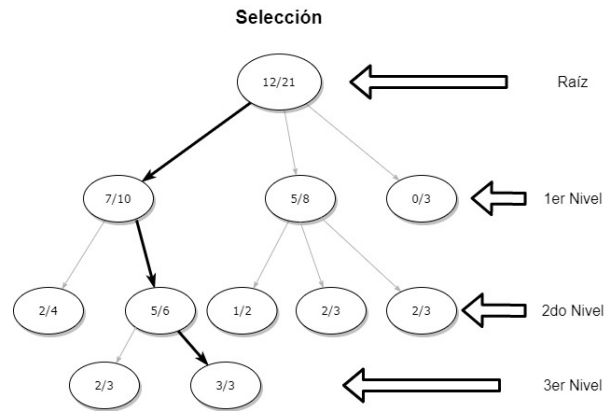


Figure 1: Selección

*Expansión:* empezar desde la raíz  $R$  y seleccionar nodos hijos sucesivos hasta alcanzar un nodo hoja  $L$ . La selección describe una manera de elegir los nodos hijos, que permitan que el árbol se expanda hacia movimientos mas prometedores, que es la esencia del árbol de búsqueda Monte Carlo.

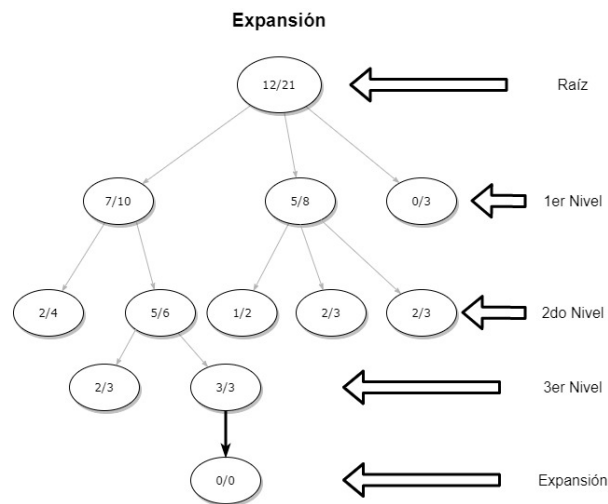


Figure 2: Expansión

*Simulación:* realizar una simulación aleatoria desde el nodo C.

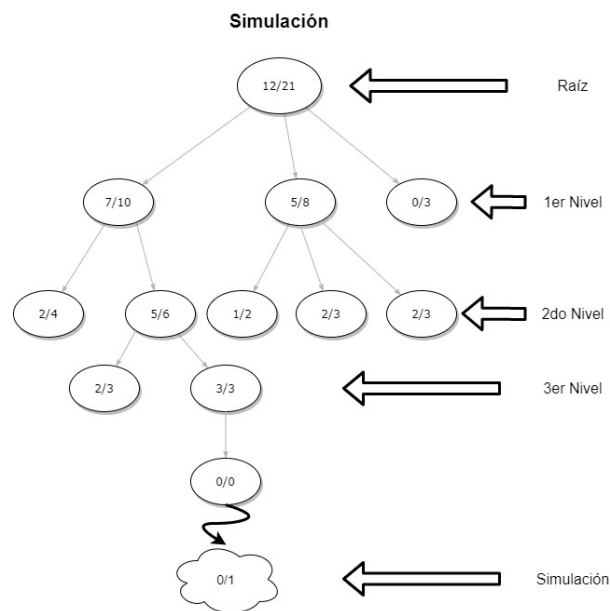


Figure 3: Simulación

*Retro Propagación:* utilizar el resultado de la simulación para actualizar la información en los nodos en el camino de C a R.

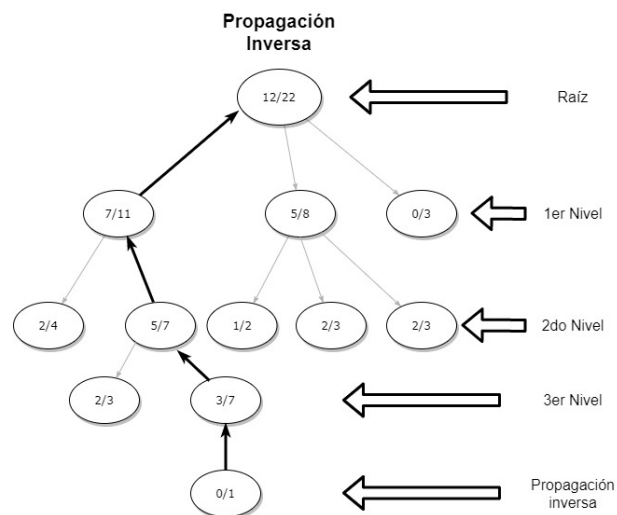


Figure 4: Retro Propagación

## Codificación

La codificación de Búsqueda Monte Carlo es relativamente sencilla, al hacer uso de el método de búsqueda ciega:

```
mcsearch
```

```
## function (N, lower, upper, Fx, type = "min", ...)  
## {  
##   D <- length(lower)  
##   s <- matrix(nrow = N, ncol = D)  
##   for (i in 1:N) s[i, ] = runif(D, lower, upper)  
##   fsearch(sol = s, Fx = Fx, type = type, ...)  
## }
```

Su implementación de igual manera es sencilla:

```
N=10000 # se define el número de muestras  
#para la función de esfera:  
D <- c(2,30)  
label="esfera"  
for(i in 1:length(D))  
{  
  S=mcsearch(N,rep(-5.2,D[i]),rep(5.2,D[i]),sphere,"min")  
  cat(label,"D:",D[i],"s:",S$sol,"f:",S$eval,"\n")  
}
```

```
## esfera D: 2 s: -0.02428153 f: 0.003577936  
## esfera D: 30 s: -0.6641743 f: 108.1639
```

```
label="Rastrigin"  
for(i in 1:length(D))  
{ S=mcsearch(N,rep(-5.2,D[i]),rep(5.2,D[i]),rastrigin,"min")  
  cat(label,"D:",D[i],"s:",S$sol,"f:",S$eval,"\n")  
}
```

```
## Rastrigin D: 2 s: -0.003405824 f: 0.04997579  
## Rastrigin D: 30 s: -2.882636 f: 356.9194
```