SIMULATED ANNEALING PARA LA BÚSQUEDA DE POLÍTICAS ÓPTIMAS EN PROCESOS DE DECISIÓN DE MARKOV

Roberto Emilio Salas Ruiz[*]

Resumen

El presente articulo muestra como implementar el algoritmo del simulated annealing, para resolver un ejemplo práctico de procesos de decisión de Markov (MDP). Se presenta una conceptualización básica de los MDP y del algoritmo del simulated annealing, así como descripción del problema y la forma de cómo se realizó el mismo y los resultados obtenidos.

Palabras claves

Procesos de decisión de Markov, *Simulated Annealing*, Cadena de Markov, política.

Abstract

This paper shows who to implement the simulated annealing algorithm in order to resolve a practical example of Markov decision processes (MDP). It presents basic concepts of MDP and the simulated annealing algorithm; it also shows a description of the problem, the way it was formulated and the gotten results.

[*] Ingeniero de Sistemas de la Universidad del Norte, Magíster en Ingeniería de Sistemas de la Universidad Nacional de Colombia. Profesor Universidad Distrital "Francisco José de Caldas" – Facultad Tecnológica. Correo electrónico resalas72@yahoo.com

Keywords

Markov Decision Processes, Simulated annealing, Markov Chain, Policy.

Introducción

Este articulo se centra en un modelo para la toma de decisiones secuénciales bajo incertidumbre, los llamados procesos de decisión de Markov (PDM). Este modelo consta de un conjunto de estados, un conjunto de decisiones disponibles (acciones) para cada estado, un costo o recompensa de acuerdo a la acción que se tome en cada estado. Dependiendo del estado en que se encuentre el sistema y de la acción que se tome en ese estado se transitará con cierta probabilidad a otro estado, esta probabilidad es independiente de los estados y acciones pasadas en el sistema. La idea del tomador de decisiones es encontrar una secuencia de acciones (política) a seguir en cada estado de tal manera que el sistema, después de un número determinado de iteraciones o en ciertos casos a largo plazo, produzca una recompensa o costo óptimas.

En las últimas décadas ha habido un notable resurgimiento en la investigación teórica y aplicada de estos modelos. Estos modelos surgieron como una ramificación de la investigación de operaciones en la década de los 50, y en años recientes han ganado reconocimiento en diversos campos como la ecología, economía e ingeniería [4]. La tarea fundamental en el estudio de estos modelos es el diseño de algoritmos eficientes para encontrar políticas óptimas.

El objetivo principal de este artículo es el uso del algoritmo del *simulated annealing* como una alternativa para encontrar buenas políticas (“cercanas a las óptimas”) en este tipo de modelo. Para lo anterior se aplico el referido algoritmo en un ejemplo clásico de este tipo de problemas, tal y como el problema del reemplazo del automóvil.

1. Generalidades

1.1. Procesos de decisión de Markov

Un proceso de decisión de Markov, es similar a una cadena de Markov, con la diferencia que la matriz de transición depende de la acción tomada por un agente en cada paso del tiempo.

El objetivo es hallar una función llamada política, la cual especifica que acción tomar en cada estado de modo que optimice alguna función de costo o de recompensa.

Un proceso de decisión de Markov, contiene:

- Un conjunto de posibles estados *S.*
- Un conjunto de posibles acciones *A.*
- Una función real de costo o recompensa *R(s,a)*
- Una descripción *T* de los efectos de cada acción sobre cada estado.

Se asume la propiedad de Markov: "Los efectos de una acción tomada en un estado, dependen solo del estado actual y no de la historia anterior".

Las acciones que se toman pueden ser de dos clases:

- Acciones Deterministicas: $T: SxA \rightarrow S$. Para cada estado y acción, especificamos un nuevo estado.

- Acciones probabilísticas: $T: SxA \rightarrow Prob(S)$. Para cada estado y acción, especificamos una distribución de probabilidad sobre los próximos estados. La distribución se representa por $P(S'|s,a)$.

1.2. Políticas

Una política es una regla que específica que acción tomar en cada punto en el tiempo.

En general, las decisiones especificadas por una política pueden:

- Depender del estado actual del proceso que describe al sistema.

- Ser aleatoria, es decir, dependen, de algún evento externo aleatorio.

- O depender de estados pasados y/o decisiones.

Una política estacionaria es definida por una función de acción que asigna una función a cada estado, independiente de previos estados, previas acciones y tiempo.

Bajo una política estacionaria, un proceso de decisión de Markov es una cadena de Markov.

1.3. Obtención de la política óptima en procesos de decisión de Markov

Para hallar la política óptima de un proceso de decisión de Markov, existen varios métodos para obtenerla, entre los cuales podemos mencionar:

- **Solución por enumeración exhaustiva**. Por este método, se le halla el costo o recompensa a todas las políticas posibles en el sistema y se escoge la solución óptima. Las acciones son deterministicas y cada política esta determinada por un conjunto de acciones o decisiones. Está política determina una cadena de Markov, a la cual se le hallan las probabilidades de estado estacionario y se multiplica por el costo o la recompensa de estar en ese estado y se halla el costo

promedio o a la larga de esa cadena de Markov, lo cual nos indica el costo de esa política. Este mismo procedimiento se realiza para todas las políticas y se halla cual es la más optima entre todas. El costo esperado de cada política es:

$$E(C(D)) = \sum_{i=0}^{n} C_{ia}\pi_i \quad (1)$$

Donde:

D es la política.

n es el número de estados menos uno.

C_{ia} es el costo de tomar la decisión *a* en el estado *i*.

π_i: es la probabilidad de estado estable en *i* bajo la política *D*.

La solución por enumeración exhaustiva, se hace bajo políticas estacionarias y por lo tanto enumerables. Este método es factible utilizarlo cuando el numero de políticas es reducido y en el cual se pueden hallar todas las posibles políticas y así determinar la mejor.

- **Solución por programación lineal**. Por este método se plantea una función lineal objetivo sujeta a unas restricciones lineales, la cual se puede resolver por el método simplex. La variable de decisión es Y_{ia} y representa la probabilidad incondicional de estado estable de que el sistema se encuentre en el estado *i* se tome la acción *a*. Esta variable toma valores entre cero y uno. Bajo un problema pequeño el simplex lo puede manejar bien y resolver sin problemas, el inconveniente es cuando el número de variables es bastante grande ya que el algoritmo simple tiene una complejidad exponencial.

- **Solución por el algoritmo de mejoramiento de políticas**. Fue propuesto por Ronald Howard [3]. Por este método se parte de una política arbitraria y para determinar la política óptima se sigue un proceso iterativo, en la cual en cada iteración se halla una política mejor. El proceso iterativo termina cuando dos políticas sucesivas son idénticas. Y siempre converge a la misma

política, independiente del estado en que se comience. Eventualmente este algoritmo hallará una política óptima. La complejidad de este algoritmo es n x m, donde n es el número de estados y m es el número de posibles acciones en cada estado.

De hecho los anteriores métodos permiten hallar políticas óptimas pero los mismos funcionan bien cuando el problema es relativamente pequeño ya que para un proceso de decisión de Markov de n estados y m posibles acciones, existen a lo más m^n políticas estacionarias posibles que se pueden tomar.

1.4. *Simulated Annealing*

Simulated annealing es una técnica de optimización numérica basa en principios de termodinámica, motivada por una analogía en el templado de sólidos.

La idea del *Simulated annealing* viene de un artículo publicado por Metrópolis en 1953. El algoritmo en este artículo simuló el enfriamiento de un material en una tina caliente. Este es un proceso que se conoce como "templado".

Si se calienta un sólido pasado su punto de fundición y entonces se enfría, las propiedades estructurales del sólido dependen de la tasa de enfriamiento. Si el líquido se enfría lentamente, grandes cristales se formaran. No obstante, si el líquido se enfría rápidamente (apagado), los cristales contendrán imperfecciones.

El algoritmo de Metrópolis simuló el material como un sistema de partículas.

El algoritmo del *Simulated annealing*, simula el proceso de enfriamiento, bajando gradualmente la temperatura del sistema hasta que este converja a un estado estable de congelamiento.

1.5. Presentación del algoritmo del *Simulated Annealing*

A continuación se describen los pasos del algoritmo del *Simulated Annealing* de manera general:

Paso 1: Inicializar la temperatura (*T*) y generar una solución aleatoria.

Paso 2: Calcular el costo de la solución generada. Esta se tiene como solución actual del sistema.

Paso 3: Generar una solución vecina a la que se tiene como solución actual.

Paso 4: Calcular el nuevo costo

Paso 5: Si Δ(costo actual – nuevo costo) es menor o igual a cero, tomar la nueva solución como la solución actual al sistema.

Paso 5.1 Si Δ(costo actual – nuevo costo) es mayor que cero, se calcula la expresión , si esta expresión es mayor que un número aleatorio que se genera entre cero y uno, se toma la nueva solución, como la solución actual al sistema, sino se rechaza.

Paso 6: Repetir los pasos 3, 4 y 5 un numero determinado de iteraciones.

Paso 7: Decrecer la temperatura.

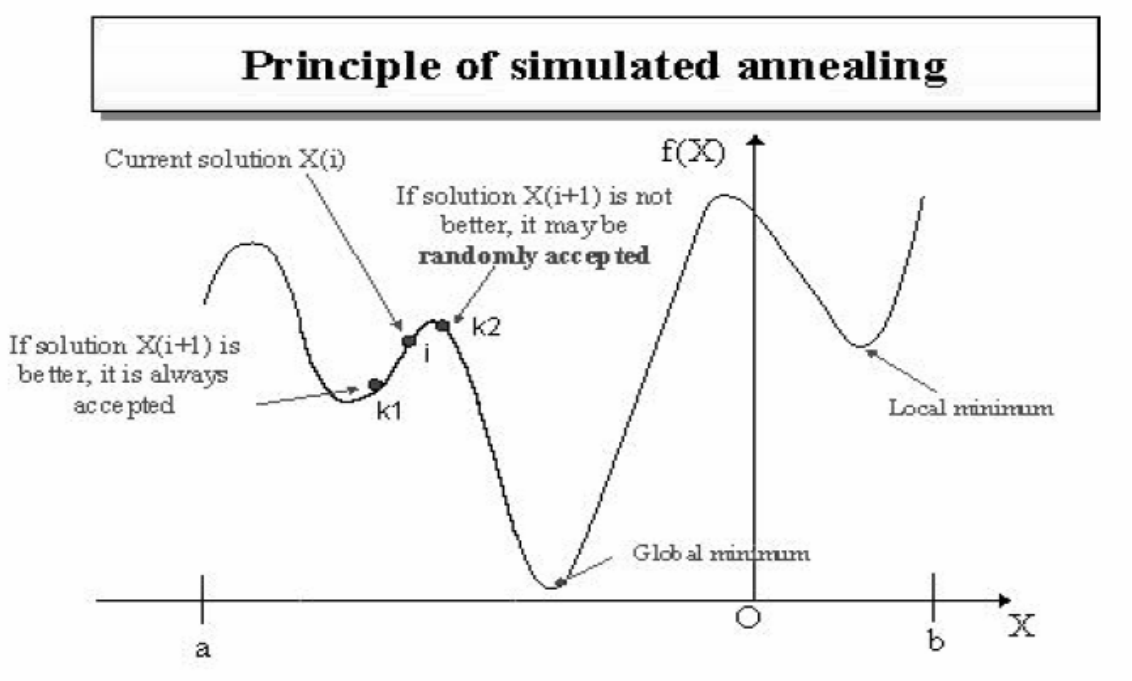
Paso 8: Repetir los pasos 3, 4, 5, 6 y 7 un numero determinado de iteraciones.

Fin del algoritmo. (La que se tiene como solución actual, es la solución al sistema).

Tal y como se puede observar, la idea es muy simple, es generar soluciones un numero determinado de iteraciones, y estas soluciones se aceptan o rechaza según algún criterio. Lo que el algoritmo busca, es no quedarse atrapado en mínimos locales y tratar de localizar un mínimo global.

La figura 1 muestra el principio del *simulated annealing*, donde se aprecia que $X(i)$ es la solución actual, y si la solución $X(i+1)$ es mejor (la ubicada en el punto *k1*) se acepta, sino, se puede aceptar con cierta probabilidad (punto *k2*).

Figura 1. Selección de nuevos estados en el *simulated annealing*



fuente: http://www.ep.liu.se/exjobb/isy/2002/3339/exjobb.pdf. pag. 10.

Lo importante en este algoritmo, es el esquema de reducción de la temperatura, llamado la programación del enfriamiento (*cooling schedule*). Sobre el particular, se han propuesto varios es quemas, se puede utilizar el templado de Cauchy que es $T=To/k$, en el tiempo k. O el templado de Boltzmann $T=To$ / ln k. Igualmente el paquete de software *Adaptve Simulated Annealing* (ASA)[6] utiliza como templado $T= To \exp(-ck^{1/D})$, donde c es una constante y D hace referencia al espacio D-dimensional.

Análogo a los procesos físicos, la temperatura es levemente reducida, causando que la probabilidad de aceptar un peor movimiento, decrezca con el tiempo.

2. Problema del reemplazo del automóvil

Ahora se presente un problema particular, al cual se le va a aplicar el algoritmo del *simulated annealing* y el cual se describe a continuación[1]:

Consideremos el problema de reemplazo de un automóvil sobre un intervalo de tiempo de diez años. Estamos de acuerdo de revisar nuestra actual situación cada tres meses y tomar una decisión de mantener nuestro actual carro o negociarlo por uno nuevo en ese momento. El estado del sistema, *i,* es descrito por la edad del carro en periodos de tres meses; *i* puede ir desde 0 hasta 40. A fin de mantener, el numero de estados finitos, un carro de edad 40 permanece como un carro de edad 40 por siempre (Es considerado que ya está gastado). Las alternativas disponibles en cada estado son dos, mantener el carro actual o negociarlo por uno nuevo. En el estado 0 como el carro es nuevo, no es factible tomar la decisión de negociarlo. Así las cosas, tenemos 40 estados en los cuales se pueden tomar dos decisiones posibles, de modo que hay 2^{40} posibles políticas, lo que es más de un billón de políticas.

Los datos suministrados son los siguientes:

El costo de un carro nuevo es de $2000.

T_i, es el valor de negociar un carro de edad *i*.

E_i, es el costo de operar un carro de edad *i* hasta que alcance la edad *i+1*.

P_i, es la probabilidad de que un carro de edad *i* sobreviva a la edad *i+1*, sin incurrir en un costo prohibitivo de reparación.

La probabilidad definida encima es necesaria para limitar el número de estados a los que puede transitar. Un carro de cualquier edad que tiene una avería irreparable es inmediatamente enviado al

[1] Adaptado de Bellman R. E. y Dreyfus S. E. *Applied Dynamic Programming*, pag. 312.

estado 40. Naturalmente, p_{40}=0, siendo este un estado absorbente si se toma la decisión de mantener el carro actual.

La tabla 1 resume los datos anteriormente expuestos.

Tabla 1. Datos del problema del reemplazo del automovil

Estado	Negocio (T_i)	Operación (E_i)	Probabilidad de Sobrevivir (P_i)	Costo de reemplazar por uno nuevo (C_i)
0	_	50	1.000	_
1	1460	53	0.999	590
2	1340	56	0.998	710
3	1230	59	0.997	820
4	1050	62	0.996	1000
5	980	65	0.994	1070
6	910	68	0.991	1140
7	840	71	0.988	1210
8	710	75	0.985	1340
9	650	78	0.983	1400
10	600	81	0.980	1450
11	550	84	0.975	1500
12	480	87	0.970	1570
13	430	90	0.965	1620
14	390	93	0.960	1660
15	360	96	0.955	1690
16	330	100	0.950	1720
17	310	103	0.945	1740
18	290	106	0.940	1760
19	270	109	0.935	1780
20	255	112	0.930	1795
21	240	115	0.925	1810
22	225	118	0.919	1825
23	210	121	0.910	1840
24	200	125	0.900	1850
25	190	129	0.890	1860
26	180	133	0.880	1870
27	170	137	0.865	1880
28	160	141	0.850	1890
29	150	145	0.820	1900
30	145	150	0.790	1905
31	140	155	0.760	1910
32	135	160	0.730	1915
33	130	167	0.660	1920
34	120	175	0.590	1930
35	115	182	0.510	1935
36	110	190	0.430	1940
37	105	205	0.300	1945
38	95	220	0.200	1955
39	87	235	0.100	1963
40	80	250	0	1970

Este problema resuelto por el algoritmo de mejoramiento de políticas, encontró que la política óptima es: Del estado 0 al 30 mantener el automóvil actual y del 31 al 40, la mejor decisión es reemplazarlo por uno nuevo. Está política tiene un costo de 172.559.

3. Implementación del *Simulated Annealing* en el problema del reemplazo del automóvil

El algoritmo del *simulated annealing*, el cual se utiliza para generar las políticas, de manera general y para el problema en particular se implementó así:

Pasos:

1. Inicializar la Temperatura

2. Generar una política aleatoriamente

3. Proceso iterativo

a. Para la política generada, entonces es calculado el costo asociado a la cadena de Markov producida por esa política, esto se calcula hallando las probabilidades de estado estable de esa política multiplicado por el costo de estar en ese estado.

b. (Después de la primera iteración) Si la nueva solución es mejor que la previa, esta es aceptada; sino, es aceptada con cierta probabilidad; si esta probabilidad es alta, la nueva solución es seleccionada.

c. Generar una nueva política, vecina a la que se tiene como solución actual al sistema.

d. Se repiten los pasos a hasta el c, un número determinado de iteraciones con la misma temperatura.

e. Se disminuye la temperatura.

f. Pasos a hasta el e son repetidos para un determinado numero de iteraciones.

Básicamente en el anterior algoritmo, no necesariamente, va a converger a la política óptima, pero si va a generar una buena solución. Claro esto también depende de la temperatura inicial que se tome y de cómo esta se vaya disminuyendo.

La temperatura se inicializó en *To*=500 y el programa de reducción de temperaturas que se utilizó es el templado de Cauchy *T=To/j*, donde *j* es la *j*-esima iteración.

La política vecina se generó, tomando un estado aleatoriamente y modificándole la decisión que se toma en el mismo, por ejemplo si al estado que le vamos a cambiar la decisión tiene actualmente la decisión de mantener el carro, se le modifica por la decisión de cambiar el carro por uno nuevo y viceversa. Es de anotar que esto se hace para un solo estado nada más, no para varios.

De acuerdo a la política que se genere, para esa política existe una cadena de Markov asociada, entonces se genera la política y simultáneamente se arma la matriz de transición (cadena de Markov) ligada a esa política. A esta cadena de Markov se les halla las probabilidades de estado estable y se calcula el costo aplicando la ecuación (1).

Las probabilidades de transición de acuerdo a la política que se tomen están dadas por las siguientes expresiones:

Si *a*=0 (Mantener el automóvil actual)

$$p_{ij}^{a} = \begin{cases} p_i, & para \quad j = i+1 \\ 1-p_i, & para \quad j = 40 \\ 0, & para \quad otra \quad j \end{cases}$$

Si *a*=1 (Reemplazar el automóvil por uno nuevo)

$$p_{ij}^{a} = \begin{cases} 1, & para \quad j = 1 \\ 0, & para \quad otra \quad j \end{cases}$$

El costo en el estado *i*, de acuerdo a la acción que se tome está dado por:

$$Costo_i = \begin{cases} E_i, & para \quad a = 0 \\ C_i, & para \quad a = 1 \end{cases}$$

Donde E_i es el valor que se muestra en la segunda columna de la tabla 1 y C_i es el valor que se muestra en la quinta columna de la tabla 1.

4. Resultados obtenidos

El algoritmo, se implementó en matlab 5.3, y se comenzó con una temperatura inicial de 500, para una misma temperatura se realizaron 400 iteraciones, y se realizaron 2000 iteraciones con temperaturas diferentes, es decir, el algoritmo tuvo en total 800000 iteraciones. Para la disminución de la temperatura se utilizó el templado de Cauchy, y como son 2000 temperaturas diferentes, se empezó en T=500 terminando en T=0.249.

La mejor política que se obtuvo se muestra en la tabla 2. El costo total de esa política es de 172.559.

Tabla 2. Mejor política obtenida

Estado(E)	Acción(a)
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0

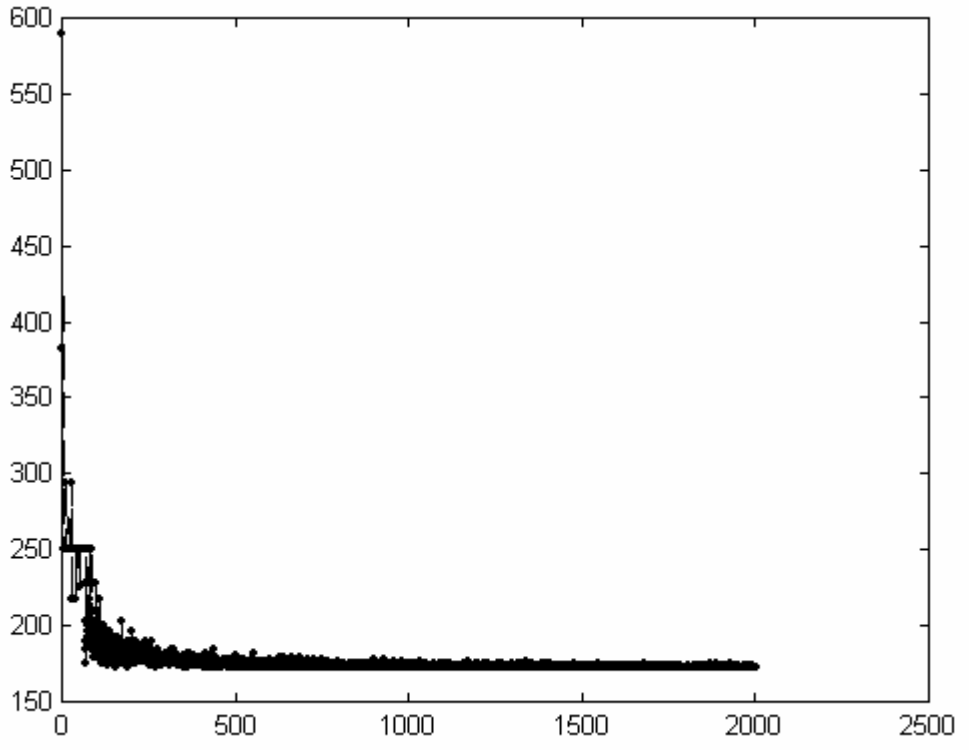
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0
31	1
32	0
33	1
34	1
35	1
36	1
37	0
38	1
39	0
40	1

Por los resultados obtenidos, se observa que la mejor decisión que se puede tomar en los estado del 0 al 30 es la de mantener el carro, después del estado 30, la mejor decisión es cambiarlo por uno nuevo; y en el estado 40 la mejor decisión obviamente es cambiarlo por uno nuevo. Estos resultados parten de nuestra idea intuitiva de que si un carro esta en una edad donde existe baja probabilidad de que tenga un daño irremediable, lo mejor es mantenerlo; por el contrario si el carro es “viejo” la mejor decisión es negociarlo y comprar uno nuevo. La solución hallada tiene el mismo costo que la que obtiene el algoritmo de mejoramiento de políticas, por lo que se puede decir que la política obtenida es óptima, aunque no es la misma que se obtuvo con el citado algoritmo, dando como conclusión que este es un problema de solución múltiple.

La figura 2 muestra la evolución del valor de los costos para 2000 temperaturas diferentes aplicando *simulated annealing*. El eje de las *x* representan el numero de iteración y el eje de las *y* el valor del costo de la política que se tiene como solución actual al sistema en esa iteración se observa que es

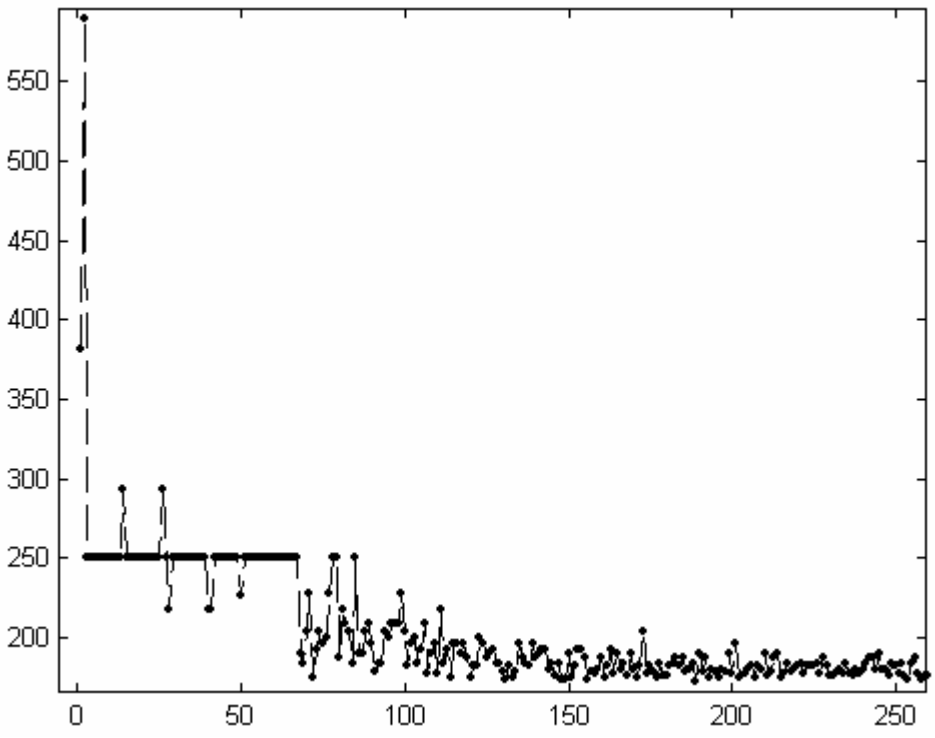
un comportamiento descendente y en las últimas iteraciones, las variaciones son mínimas y se puede decir que ya está en un punto de congelamiento y si se siguiera iterando más no se iban a obtener mejores resultados.

Figura 2. Evolución de los costos en el problema del reemplazo del automóvil en 2000 iteraciones aplicando *simulated annealing*



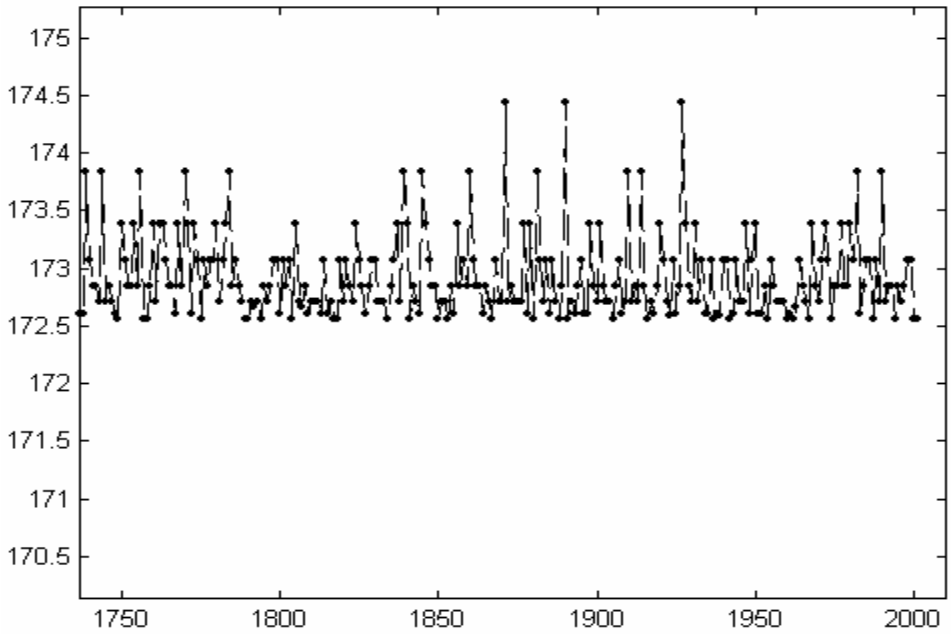
fuente: autor

Figura 3. Primeras 250 iteraciones



fuente: autor

Figura 4. Ultimas 250 iteraciones



fuente: autor

La figura 3 y la figura 4 son acercamientos de la figura 2, donde la primer mencionada figura muestra las primeras 250 iteraciones y los costos asociados a las políticas obtenidas en esas

iteraciones y se observa una gran variación en los valores de los costos (varían aproximadamente entre 174 y 590), lo cual es lógico ya que en estas primeras iteraciones los valores de las temperaturas son altos. La segunda mencionada figura muestra las últimas 250 iteraciones, y se observa que el valor de los costos tienen variación mínima (varían aproximadamente entre 172.559 y 174.5) y se puede decir que se esta en un punto de congelamiento.

Conclusiones

1. A través del presente trabajo, se ha demostrado que el *simulated annealing*, es una técnica válida para resolver procesos de decisión de Markov con políticas estacionarias. Para el caso presentado se probó que se puede obtener la política óptima y diferente a la que obtiene la rutina de mejoramiento de políticas.

2. Si bien, el algoritmo del *simulated annealing* se probó en un problema relativamente pequeño, parece factible utilizar el mismo en problemas mucho más grandes, donde son inaplicables técnicas como la rutina de mejoramiento de políticas por lo costosa que es, o en problemas donde las probabilidades de transición no están disponibles y por ende no se puede obtener la cadena de Markov.

3. Próximos trabajos en este campo son probar si por medio del *simulated annealing* se pueden resolver procesos de decisión de Markov que tengan políticas aleatorizadas y ver si este algoritmo halla buenas solución, otro enfoque en el que se trabajará es el de resolver estos procesos de decisión markovianos con algoritmos genéticos, lo cual se esperan que obtengan mejores resultados que el *simulated annealing* ya que estos trabajan con varias soluciones al mismo tiempo y se pueden realizar más operaciones con ellos.

Referencias bibliográficas

[1] Bellman R. E. y Dreyfus S. E. *Applied Dynamic Programming*, 1a ed. Princeton, New Jersey: Princeton University Press, 1962.

[2] Hillier F. S. y Lieberman G. J. *Introducción a la investigación de operaciones*, 6a ed. Bogota, Colombia: McGraw Hill, 1999.

[3] Howard R. Dynamic Programming and Markov Process. [Cambridge, MA]: MIT press, 1960.

[4] Ingber L. A. (1993) Simulated annealing: Practice versus theory.

[5] Ingber L. A. (1996) Adaptive simulated annealing (ASA): Lessons learned.

[6] Puterman M. L., Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc. 1994.

[7] Moins S. (2002) Implementation of a simulated annealing algorithm for Matlab.

[8] Taha H. A., *Investigación de operaciones*, 5a ed. Ciudad de Mexico, Mexico: Alfaomega, 1992.

Infografía

http://www.ingber.com/asa93 sapvt.pdf.

http://www.ingber.com/asa96 lessons.pdf.

http://www.ep.liu.se/exjobb/isy/2002/3339/exjobb.pdf.