Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра математики

Дипломна робота

освітній ступінь – бакалавр

на тему: «**Напівмарковські процеси прийняття рішень з рандомізованим дисконтом**»

Виконав: студент 4-го року навчання,

Освітньої програми «Прикладна математика», 113

Галдецький Андрій Володимирович

Керівник В.М. Михалевич,_____

доктор фіз.-мат. наук, доцент

Рецензент _____

(прізвище та ініціали)

Дипломна захищена

з оцінкою _____

Секретар ЕК _____

«____» _____ 20____ р.

Київ – 20___

ЗМІСТ

Стор.

ВСТУП 3

Постановка задачі

1.1. Напівмарковські процеси прийняття рішень. 5

1.2. Дисконт 6

1.3. Напівмарковські процеси прийняття рішень з рандомізованим дисконтом 6

Керування напівмарковським процесом(опис)

2.1. Керування на скінченному горизонті. 10

2.2. Вимірні умови вибору 13

2.3. Керування на нескінченному горизонті 15

Керування напівмарковським процесом(приклади) 18

Висновки. 24

Література 25

Додаток А. Реалізація на мові програмування Python 26

Вступ

Напівмарковські процеси, які розглядаються у цій роботі, фактично є подальшим розвитком та певним узагальненням ланцюгів Маркова та марковських процесів. Чим відрізняється напівмарковський процес від марковського процесу? Насамперед тим, що марковський процес виходить з того, що час переходу з одного стану в інший завжди однаковий, а напівмарковський процес допускає, що перехід з одного стану в інший може займати різний відрізок часу.

Сучасний світ та сучасну економіку можна розглядати як безліч систем масового обслуговування мільйонів чи, навіть, мільярдів людей та безперервний пошук більш ефективних моделей обслуговування, прогнозування, резервування (бронювання).

Для сучасних складних моделей масового обслуговування більш доцільно використовувати напівмарковський процес прийняття рішень. На практиці це доцільно робити для розрахунку оптимального часу обслуговування клієнтів у черзі чи, наприклад, для прогнозування процесу зносу (виходу з ладу) деталей у складних пристроях і оптимального розподілу ресурсів та коштів на їх заміну або підтримку. Можна вважати, що оптимальна довгострокова стратегія напівмарковської моделі, не відрізняється від стратегії однорідних ланцюгів Маркова за критеріями середніх витрат, якщо очікуваний час перебування в кожному стані умов однаковий для обох моделей. Однак соціально-економічні чинники відрізняються у кожній моделі, у тому числі залежно від параметрів розподілу часу перебування в кожному стані. Через це використання напівмарковської моделі більш підходить до реальних потреб соціуму та економіки.

У теорії випадкових процесів напівмарковські процеси прийняття рішень є одним з напрямків, що наразі активно розвивається. Це пов'язано, по-перше, з тим, що напівмарковські процеси є природним і важливим узагальненням ланцюгів і процесів Маркова, і, по-друге, з тим, що напівмарковські процеси дозволяють ефективно моделювати реальні системи масового обслуговування чи, наприклад, стохастичні автомати.

Робота складається з трьох розділів і додатку. У перших двох наведена ознайомча інформація стосовно напівмарковських процесів, поняття дисконту і керування напівмарковським процесами з рандомізованим дисконтом. У третьому розділі розглядаються більш конкретні приклади напівмарковських процесів з рандомізованим дисконтом. У додатку подано реалізацію процесу пошуку оптимального шляху на мові програмування Python.

Метою даної роботи є ознайомлення з принципами та способами практичного використання напівмарковських процесів з рандомізованим дисконтом.

Постанова задачі

У 50-ті роки минулого століття незалежно один від одного Поль Леві, Джон Сміт та Лайош Такач у своїх роботах запропонували розглядати стохастичні системи, еволюціонуючі як системи Маркова – напівмарковськими. Поняття напівмарковських процесів (НМП) є природним узагальненням ланцюгів Маркова та марковських процесів. Отже, якщо визначити стохастичний процес для ланцюгів Маркова, то це буде напівмарковським процесом. Основна відмінність марковського процессу від напівмарківського полягає в тому, що марковський визначається як набір станів і часів, а напівмарковський – є еволюціонуючим процесом, і будь-яка реалізація процесу має певний стан для будь-якого даного часу. Нижче я буду розглядати напівмарковські процеси прийняття рішень з рандомізованим дисконтом та їх застосування.

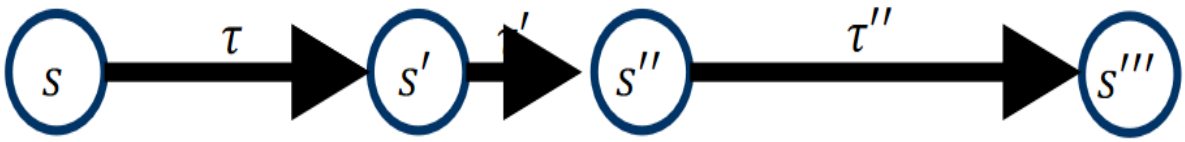
Напівмарковські процеси прийняття рішень

Постанову напівмарковського процесу можна лаконічно викласти, схожим чином до марковського процесу : Напівмарковським процесом прийняття рішень є сукупність об'єктів :

$$\mathbf{P} = \langle S, \mathrm{A}, P_A, R_A \rangle$$

1. Набір станів – S, який називається – простором станів.
2. Набір поведінок – A, який називається – простором керуючих станів.
3. Функція відображення - $P_A(s, s') = \Pr(s', \tau | s, a)$
4. Функція винагороди - $R_A(s, s') = E[r|s, a]$, де E – це очікувана винагорода.
5. Дисконт фактор – 0 < y < 1(дисконтований – y < 1, не дисконтований – y = 1).
6. Горизонт (кількість кроків) h. Скінченна – $h \in N$, або нескінченна $h = \infty$

Простір станів можна розділити на 2 типи: скінченні або нескінченні. Деякі нескінченні процеси можна звести до скінченного виду напівмарковського процесу.



Так виглядає перехід системи з одного стану в іншому у напівмарковському процесі.

Дисконт

Дисконт функцію можна розкласти на два поняття – константна або не константна. Якщо дисконт-функція - константна для різних галузей, товарів і проектів, то ця константна називається дисконт-фактором, або просто дисконтом.

Використаємо дисконт функцію для підрахування знецінення валюти з часом. Візьмемо рівень інфляції 12% у рік. Іншими словами 1 одиниця валюти через рік буде коштувати 1.12. Відповідно значення дисконту на наступний рік буде – 0.89.

Позначимо дисконт буквою β, $0 < \beta < 1$. Тоді, якщо r – це відсоткова ставка, то дисконт-функція визначається як – $\beta = {}^{1}/_{(1+r)}$. Відзначимо, що при такому підході вважають, що банківські відсотки плати за депозит однакові у всіх банках. Більш правильно було б вважати r, а тому і β, нечислових величинами, а саме, інтервалами $[r_1, r_2]$ і $[\beta_1, \beta_2]$. Де можна визначити зв'язок між інтервалами наступними формулами – $\beta_1 = {}^{1}/_{(1+r_2)}$ і $\beta_2 = {}^{1}/_{(1+r_1)}$

Напівмарковські процеси прийняття рішень з рандомізованим дисконтом

Далі будуть розглядатися напівмарковські процеси прийняття рішень, з рандомізованим дисконт-фактором $\beta = (1+r)^{-1}$. Тобто, процеси де $\beta = (1+r)^{-1}$ є рандомізованою величиною.

Напівмарковським процесом прийняття рішень з рандомізованим дисконтом (або керованим процесом) називатимемо сукупність об'єктів [1]:

$$\mathbf{P} = \langle (\mathrm{X}, \Xi), (\mathrm{A}, \mathrm{A}), \mathrm{F}, \Delta, \mathrm{Q}, \Phi, \mathrm{c} \rangle$$

у якому:

1. Будемо позначати простір станів системи як – $X = X' \times R^+$, а Ξ — σ-алгебра борелівських підмножин простору X;
2. Простір керуючих станів позначимо літерою — A, а множина A — σ-алгебра борелівських підмножин простору A;
3. F — відображення, яке ставить відповідність кожному $(x, \alpha) \in X$ деяку непорожню замкнену підмножину $A_{(x,\alpha)}$ множини A, $A_{(x,\alpha)}$ — множина допустимих керуючих впливів системи в деякому стані (x, α);
4. $\Delta = \{(x, \alpha, a \mid (x, \alpha) \in X, a \in A_{(x,\alpha)})\}$ — множина можливих наборів станів і керуючих впливів, вимірна підмножина $X \times A$;
5. $Q(\cdot \mid x, \alpha, a)$ — напівмарківське ядро, яке задане на X, породжене Δ, називається законом переходу;
6. $\Phi(\cdot \mid x', \alpha', a, x'', \alpha'')$ — функція розподілу додатної випадкової величини τ, яка є часом перебування системи в стані (x', α') за умови, що прийнято рішення $a \in A_{(x,\alpha)}$ і наступним станом системи є $(x'', \alpha'') \in X$;
7. $c(x, \alpha, a)$ — вимірна дійснозначна функція на Δ, що позначає очікувану вартість перебування у стані $(x, \alpha) \in X$ за умови прийняття рішення a.

Визначення σ-алгебра: підмножина Σ з множини Ω називається σ-алгебра, якщо вона задовольняє наступним умовам:

- $\Omega \in \Sigma$
- $\mathrm{A} \in \Sigma \Rightarrow \overline{\mathrm{A}} \in \Sigma$
- $\mathrm{A}_i \in \Sigma, i = 1 \ldots \infty \Rightarrow \bigcup_{i=1}^{\infty} \mathrm{A}_i \in \Sigma$

Означення 1. Набір K містить графік вимірюваної функції від X до A. Це припущення гарантує, що множини у Визначенні 1 не є порожніми.

Для кожного моменту часу $n \in \boldsymbol{Z}^+$ визначимо простір H_n допустимих історій до n-го моменту часу таким чином [1]:

$$H_0 = X \text{ (множина початкових станів);}$$

$$H_n = \left(\prod_{i=0}^{n-1}(\Delta \times \boldsymbol{R}^+)\right) \times X, n \in \boldsymbol{N}.$$

Узагальнений елемент H_n, $n \in \boldsymbol{Z}^+$, який називається допустимою історією, є вектором форми [1]:

$$h_n = (x_0, \alpha_0, a_0, t_0, \dots, x_{n-1}, \alpha_{n-1}, a_{n-1}, t_{n-1}, x_n, \alpha_n),$$

де $(x_i, \alpha_i, a_i) \in \Delta$, $i = \overline{0,n}$, t_i — реалізація випадкової величини τ, t_i — час перебування системи в стані (x_i, α_i).

При цьому для кожного $n \in \boldsymbol{N}$, множина H_n є підмножиною [1]:

$$H_n \subseteq \widehat{H}_n = \left(\prod_{i=0}^{n-1}(X \times A \times \boldsymbol{R}^+)\right) \times X, H_0 \subseteq \widehat{H}_0 = X.$$

Слід зауважити, що — $\overline{H_0} = H_0 = X_0$

Стратегія - це послідовність дій, що здійснюються керуванням, тобто:

Означення 2. Стратегію допустимих дій π — називатимемо послідовність $\{\pi_n, n \in \boldsymbol{Z}^+\}$, де $\pi_n(\cdot \mid h_n)$ — ймовірнісна міра на (A, A), зосереджена на $A_{(x_n,\alpha_n)}$, тобто [1]:

$$\pi_n\left(A_{(x_n,\alpha_n)} \mid h_n\right) = 1 \text{ для всіх } h_n \in H_n, n \in \boldsymbol{Z}^+$$

і вимірно залежить від n -історії керованого процесу $\boldsymbol{P}$.

Визначимо критерій керування для подальшої роботи.

Означення 3. Критерієм керування буде функціонал наступного вигляду [1]:

$$\Re(\pi, x, \alpha) := E^{\pi}_{(x,\alpha)}[\sum_{n=0}^{\infty} e^{-s_n} c(x_n, \alpha_n, a_n)], \pi \in S_R, (x, \alpha) \in X_0,$$

де $E^{\pi}_{(x,\alpha)}$ — математичне сподівання, що відповідає процесу $\boldsymbol{P}$, керованому стратегією π, з початковим станом $(x, \alpha) \in X$;

$$s_0 = 0,\ s_n = \sum_{i=0}^{n-1} \alpha_i \tau_i,\ n \in \boldsymbol{N},$$

Стратегію $\pi^* \in S_R$ називатимемо оптимальною, якщо [1]:

$$\Re(\pi^*, x, \alpha) = \inf_{\pi \in S_R} \Re(\pi, x, \alpha) =: \Re^*(x, \alpha) \text{ для всіх} (x, \alpha) \in X_0,$$

А функцію $\Re^*$ — функцією вартості.

Керування напівмарковським процесом

Керування на скінченному горизонті

У цьому розділі ми розглянемо модель керування напівмарковським процесом на скінченному горизонті

$$\boldsymbol{P} = \langle (X, \Xi), (A, A), F, \Delta, \mathrm{Q}, \Phi, c \rangle$$

Тобто, випадок коли процес відбувається протягом N-кроків.

Поточна вартість витрат на стадії m задається значенням [1]:

$$e^{-s_m} c(x_m, \alpha_m, a_m) = 0 \ \ \forall m > N.$$

Де, $s_0 = 0$ та $s_m = a_0 + a_1 + \cdots + a_{m-1}$ для $\mathrm{m} = 1, \ldots, \mathrm{N} - 1$. На останньому кроці рішення не приймається, і ми вважаємо, що на етапі N існує кінцева вартість $c = c(x_N, \alpha_N)$. Тобто проблема, яка нас цікавить, полягає в мінімізації критерію на скінченному горизонті [1]:

$$\Re(\pi, x, \alpha) := E^{\pi}_{(x,\alpha)}[\textstyle\sum_{n=0}^{N-1} e^{-s_n} c(x_n, \alpha_n, a_n) + e^{-s_N} c(x_N, \alpha_N)], \pi \in S_R, (x, \alpha) \in X_0, \quad (1)$$

Таким чином, визначимо $\Re^*$ функцію вартості як [1]:

$$\Re^*(x, \alpha) = \inf_{\pi \in S_R} \Re(\pi, x, \alpha) \ \forall (x, \alpha) \in X_0. \quad (2)$$

Проблема полягає у знаходженні стратегії $\pi^* \in S_R$ для якої функція вартості буде:

$$\Re(\pi^*, x, \alpha) = \Re^*(x, \alpha). \quad (3)$$

Головним результатом у розділі є наступна теорема про динамічне програмування (DP), яка дає функцію ціни та детерміновану оптимальну стратегію.

Теорема 1. Для $n = \overline{0; N}$ визначимо функцію V_n, задану на X, починаючи з останнього кроку $n = N$, на якому ми не приймаємо рішення, отже:

$$V_N(x, \alpha) := c(x, \alpha); \quad (4)$$

а для $n = N - 1, N - 2, \ldots, 0$ функція визначається як [1]:

$$V_n(x,\alpha) := \min_{a \in A_{(x,\alpha)}} \left\{ e^{-\alpha\tau(x,\alpha,a)} \left[c(x,\alpha,a) + \int_X V_{n+1}(y,\beta) Q(d(y,\beta)|x,\alpha,a) \right] \right\}. \quad (5)$$

Припустимо, що функції на кожному кроці $n = \overline{0; N-1}$ вимірні і для кожного з них існує деякий селектор $f_n \in F$, такий як $f_n(x,\alpha) \in A_{(x,\alpha)}$, який досягає мінімуму у (5) для всіх $(x,\alpha) \in X$. Тобто для всіх $(x,\alpha) \in X$ і $n = \overline{0; N-1}$

$$V_n(x,\alpha) = e^{-\alpha\tau(x,\alpha,a)} \left[c(x,\alpha,f_n) + \int_X V_{n+1}(y,\beta) Q(d(y,\beta)|x,\alpha,f_n) \right]. \quad (6)$$

Тоді стратегія $\pi^* = \{f_0, \dots, f_{N-1}\}$ — оптимальна і $\Re^* = V_0$, тобто

$$\Re^*(x,\alpha) = V_0(x,\alpha) = \Re(\pi^*, x, \alpha) \text{ для всіх } (x,\alpha) \in X. \quad (7)$$

Доведення. [1] Нехай $\pi = \{\pi_n\}$ - довільна стратегія, і $C_n(\pi,x,\alpha)$ - очікувана загальна витрата за період з кроку n до кінцевого часу N, враховуючи зі стану $(x_n,\alpha_n) = (x,\alpha)$ на кроці n:

$$C_n(\pi,x,\alpha) := E^\pi \left[\sum_{k=n}^{N-1} e^{s_n - s_k} c(x_n,\alpha_n,a_n) + e^{s_n - s_N} c(x_N,\alpha_N) \middle| x_n = x, \alpha_n = \alpha \right] \quad (8)$$

для $n = 0, \dots, N-1$

$$C_N(\pi,x,\alpha) := E^\pi [c(x_N,\alpha_N)|x_N = x, \alpha_N = \alpha] = c(x,\alpha).$$

$C_n(\pi,x,\alpha)$ називається „cost-to-go” або очікувана вартість за період з n і далі при використанні стратегії π та $(x_n,\alpha_n) = (x,\alpha)$. Зокрема, зауважимо, що з (1) та (8)

$$\Re(\pi,x,\alpha) = C_0(\pi,x,\alpha). \quad (9)$$

Щоб довести цю теорему, ми покажемо це для всіх $(x,\alpha) \in X$ і $n = \overline{0; N}$ виконується:

$$C_n(\pi,x,\alpha) \geq V_n(x,\alpha) \quad (10)$$

рівність якщо $\pi = \pi^*$:

$$C_n(\pi^*,x,\alpha) = V_n(x,\alpha). \quad (11)$$

Зокрема, для $n = 0$

$$\Re(\pi,x,\alpha) \geq V_0(x,\alpha) \text{ і } \Re(\pi^*,x,\alpha) = V_0(x,\alpha) \; \forall (x,\alpha) \in X,$$

що дає бажаний висновок для доведення (7), бо $\Re(\pi,\cdot,\cdot) \geq V_0(\cdot,\cdot)$ для довільних стратегії π, з цього випливає, що $\Re^*(\cdot,\cdot) \geq V_0(\cdot,\cdot)$.

Доказ (10) та (11) - зворотною індукцією. Зауважимо, що (10) та (11) тривіально виконуються при $n = N$, оскільки з (9) та (3) слідує $C_N(\pi, x, \alpha) = V_N(x, \alpha) = c(x, \alpha)$.

Далі за гіпотезою індукції припустимо, що для деяких $n = \overline{N-1;0}$

$$C_{n+1}(\pi, x, \alpha) \geq V_{n+1}(x, \alpha)\ \forall (x, \alpha) \in X. \qquad (12)$$

Тоді :

$$C_n(\pi, x, \alpha) = E^{\pi}\left[\sum_{k=n}^{N-1} e^{s_n - s_k} c(x_k, \alpha_k, a_k) + e^{s_n - s_N} c(x_N, \alpha_N)\middle| x_n = x, \alpha_n = \alpha\right] =$$

$$= \int_A e^{-\alpha\tau(x,\alpha,a)}\left[c(x, \alpha, a) + \int_X C_{n+1}(\pi, y, \beta) Q(d(y, \beta)|x, \alpha, a)\right]\pi_n(da|x, \alpha).$$

Отже,

$$C_n(\pi, x, \alpha) \geq \min_{a \in A_{(x,\alpha)}}\left\{e^{-\alpha\tau(x,\alpha,a)}\left[c(x, \alpha, a) + \int_X V_{n+1}(y, \beta) Q(d(y, \beta)|x, \alpha, a)\right]\right\} = V_n(x, \alpha),$$

Це доводить (10). З іншого боку, якщо в (12) виконується рівність $\pi = \pi^*$ таким чином, що $\pi_n(\cdot\,|h_n)$ є мірою Дірака, зосередженою на $f(x_n, \alpha_n)$, то рівність виконується протягом усіх попередніх обрахунків, що і є доказом (11).

Міра Дірака – міра, яка привласнює розмір набору виключно на підставі того, чи містить він фіксований елемент x, чи ні. Це один з способів формалізації дельта-функції Дірака.

Формально міру Дірака можна висловити як міра δ_x на множині X, є визначеною для даної множини $x \in X$ і будь-якого набору $A \subseteq X$ через:

$$\delta_x(A) = I_A(x) = \begin{cases} 0, & x \notin A \\ 1, & x \in A \end{cases}$$

Де I – функція індикатора з A, яка приймає значення 1, якщо він містить точку a, і 0 в іншому випадку.

Вимірні умови відбору

У попередній теоремі ми припустили існування вимірного селектора, який задовольняє співвідношення (1) - (6). У цьому розділі визначимо умови для напівмарковських процесів, які забезпечують існування селектора. Для цього потрібно сформулювати наступне припущення:

Припущення 1. Для даної вимірюваної функції $u: X \to \boldsymbol{R}$ функція u^* від X до $\boldsymbol{R}$ визначена як [1]:

$$u^*(x,\alpha) := \inf_{a \in A_{(x,\alpha)}} \left\{ e^{-\alpha\tau(x,\alpha,a)} \left[c(x,\alpha,a) + \int_X u(y,\beta) Q(d(y,\beta)|x,\alpha,a) \right] \right\}, \quad (13)$$

і вона є вимірною, $f \in F$ – деякий селектор, такий, що $e^{-\alpha\tau(x,\alpha,f)}\left[c(x,\alpha,f) + \int_X u(y,\beta) Q(d(y,\beta)|x,\alpha,f)\right]$ досягає свого мінімуму при $f(x,\alpha) \in A_{(x,\alpha)}$ для всіх $(x,\alpha) \in X$, тобто можна визначити як:

$$u^*(x,\alpha) := e^{-\alpha\tau(x,\alpha,f)}\left[c(x,\alpha,f) + \int_X u(y,\beta) Q(d(y,\beta)|x,\alpha,f)\right] \forall (x,\alpha) \in X.$$

Перед подальшими умовами запишемо деякі визначення, які будуть використовуватися в умовах нижче.

Визначимо метричний простір Y, а v - функція відображення з Y в $\boldsymbol{R} \cup \{\infty\}$ така, що $v(y) < \infty$ (у) виконується принаймні для однієї точки $y \in Y$. Зазначено, що функція v є напівнеперервною знизу точки при $y \in Y$, якщо $\liminf_{y_n \to y} v(y_n) \geq v(y))$ для будь-якої послідовності $\{y_n\}$ в Y, які збігається до у. Функція v називається напівнеперервною знизу, якщо вона є напівнеперервною знизу в кожній точці з Y. Функцію $v: \Delta \to \boldsymbol{R}$ називають некомпактною (*інфімум-компактною*) на Δ, якщо для кожного $(x,\alpha) \in X$ та $r \in \boldsymbol{R}$ множина $\left\{a \in A_{(x,\alpha)} \middle| v(x,\alpha,a) \leq r\right\}$ компактний. Багатозначна функція ψ з X до A називається напівнеперервною зверху, якщо $\psi^{-1}[F]$ замкнена в X для кожного замкненої множини $F \subset A$. Нехай $\boldsymbol{B}(X)$ - родина вимірюваних обмежених функцій на X, а $\boldsymbol{C}(X) \subset \boldsymbol{B}(X)$ - підродина неперервних функцій.

Тепер ми розглянемо три умови.

Умова 1.

1) Множина станів $A_{(x,\alpha)}$ компактна для всіх $(x,\alpha) \in X$.

2) Функція вартості $c(x,\cdot)$, є напівнеперервною знизу на $A_{(x,\alpha)}$ для кожного $(x,\alpha) \in X$.

3) Функція відображення [1]:

$$v'(x,\alpha,a) := \int_X v(y,\beta)Q(d(y,\beta)|x,\alpha,a) \qquad (14)$$

на Δ задовольняє одну з двох наступних умов:

I. Функція $v'(x,\alpha,\cdot)$ — є напівнеперервною знизу на $A_{(x,\alpha)}$ для всіх $(x,\alpha) \in X$ і будь-якої $v \in \boldsymbol{C}(X)$;

II. Функція $v'(x,\alpha,\cdot)$ — є напівнеперервною знизу на $A_{(x,\alpha)}$ для всіх $(x,\alpha) \in X$ і будь-якої $v \in \boldsymbol{B}(X)$.

Умова 2.

1) $A_{(x,\alpha)}$ є компактом для всіх $(x,\alpha) \in X$, а багатозначна функція $(x,\alpha) \mapsto A_{(x,\alpha)}$ —напівнеперервна зверху;

2) Одноетапна вартість c напівнеперервна і обмежена нижче.

3) Закон переходу (перехідне ядро) Q є або:

I. Q — слабо неперервне, тобто для довільної функції $v \in \boldsymbol{C}(X)$ функція v' у (14), неперервна і обмежена на Δ;

II. Q — строго неперервне, тобто v' неперервна і обмежена на Δ для кожного $v \in \boldsymbol{B}(X)$.

Умова 3.

1) Функція вартості c - напівнеперервна, обмежена знизу та інфімум–компактна на Δ;

2) Також як і в попередньому припущенні(пункт 3) перехідне ядро Q або:

2.1) слабо неперервне.

2.2) строго неперервне.

Далі ми покажемо, як ці три умови стосувались **Припущення 1**.

Теорема 2. **Перші дві** умови передбачають **припущення 1** для будь-якої невід'ємної вимірюваної функції $u: X \to \boldsymbol{R}$. Більше того, якщо виконуються умови 1(3.1) і 2(3.1) достатньо взяти u невід'ємне і напівнеперервне знизу. А у випадку виконання умови 8 (1, 2, 3.1) функція u^* в (13) — напівнеперервна знизу.

Умова 3 передбачає **припущення 1**, якщо в пункті 3(2.1) u — невід'ємне та напівнеперервне знизу, або, за 3(2.2), якщо u є невід'ємною вимірюваною функцією. Якщо, крім того, багатозначна функція $(x,\alpha) \mapsto A^*_{(x,\alpha)}$ де $A^*_{(x,\alpha)}$ дорівнює [1]:

$$\left\{a \in A_{(x,\alpha)} \middle| u^*(x,\alpha) = e^{-\alpha\tau(x,\alpha,a)}\left[c(x,\alpha,a) + \int_X u(y,\beta)Q(d(y,\beta)|x,\alpha,a)\right]\right\},$$

напівнеперервна знизу, тоді u^* також напівнеперервна знизу.

Керування на нескінченному горизонті

У цьому розділі ми розглянемо керування напівмарковською моделлю з рандомізовиним дисконтом на нескінченному горизонті, тобто коли процес відбувається протягом зліченної кількості кроків. Враховуючи стаціонарну модель управління як напівмарковську модель, критерій ефективності, який слід мінімізувати, має наступний вигляд [1]:

$$\Re_n(\pi,x,\alpha) := E^{\pi}_{(x,\alpha)}[\sum_{k=0}^{n-1} e^{-s_k}c(x_k,\alpha_k,a_k)],\ \pi \in S_R,\ (x,\alpha) \in X_0. \qquad (15)$$

де s_k як і у керуванні на скінченному горизонті. Стратегія π^* задовольняє

$$\Re(\pi^*,x,\alpha) = \Re^*(x,\alpha).$$

називається оптимальною.

Протягом усього наступного, ми припускаємо, що функція вартості c - невід'ємна (хоча насправді, щоб практично всі результати були істинними, достатньо припустити, що c обмежена знизу).

(15) можна подати для визначення критерію якості керування за допомогою теореми монотонної збіжності як :

$$\Re(\pi,x,\alpha) = \lim_{n\to\infty} \Re_n(\pi,x,\alpha) \qquad (16)$$

Вимірна функція $v: X \to \boldsymbol{R}$ називається розв'язком для рівняння оптимальності, якщо вона задовольняє рівняння [1]:

$$v(x,\alpha) = \min_{a\in A_{(x,\alpha)}} \left\{e^{-\alpha\tau(x,\alpha,a)}\left[c(x,\alpha,a) + \int_X v(y,\beta)Q(d(y,\beta)|x,\alpha,a)\right]\right\}. \quad (17)$$

Для подальшої роботи необхідно сформувати декілька припущень

Припущення 2.

1) Вартість c напівнеперервна знизу, невід'ємна та інфінум-компактна на ІК.
2) Ядро переходу Q строго неперервне.

Припущення 3. Існує стратегія π така, що $\Re(\pi, x, \alpha) < \infty$ виконується для кожного $(x, \alpha) \in X$.

Щоб аргументувати існування селектора на нескінченній множині можна використати Теор. 1 і сформувати наступне означення.

Теорема 3. Нехай виконуються припущення 2 і 3. Тоді:

1) Функція вартості $\Re^*$ є мінімально можливим рішенням рівняння оптимальності [1]:

$$\Re^*(x,\alpha) = \min_{a \in A_{(x,\alpha)}} \left\{ e^{-\alpha\tau(x,\alpha,a)} \left[c(x,\alpha,a) + \int_X \Re^*(y,\beta) Q(d(y,\beta)|x,\alpha,a) \right] \right\} \quad (18)$$

для всіх $(x, \alpha) \in X$ і якщо u - інше рішення рівняння оптимальності, то має виконуватиметься наступна нерівність : $u(\cdot,\cdot) \geq \Re^*(\cdot,\cdot)$;

2) Існує селектор $f_* \in F$ такий, що $f_*(x, \alpha) \in A_{(x,\alpha)}$ досягає мінімуму в (18)[1]

$$\Re^*(x,\alpha) = e^{-\alpha\tau(x,\alpha,f_*)} \left[c(x,\alpha,f_*) + \int_X \Re^*(y,\beta) Q(d(y,\beta)|x,\alpha,f_*) \right] \ \forall (x,\alpha) \in X \quad (19)$$

а нерандомізована стаціонарна стратегія f_*^∞ є оптимальною. І навпаки, якщо f_*^∞ - стаціонарна нерандомізована оптимальна стратегія, то вона задовольняє (19).

3) Якщо π^* є такою стратегією, що $\Re(\pi^*,\cdot,\cdot)$ є рішенням оптимального рівняння і задовольняє [1]

$$\lim_{n\to\infty} E^{\pi}_{(x,\alpha)} [e^{-s_n} \Re(\pi^*, x_n, \alpha_n)] = 0 \text{ при довільних } \pi \in S_R \text{ і } (x,\alpha) \in X, \quad (20)$$

тоді $\Re(\pi^*,\cdot,\cdot) = \Re^*(\cdot,\cdot)$ і π^* є оптимальним з дисконтуванням. Тобто, якщо виконується (20), то π^* є оптимальним тоді і лише тоді, коли $\Re(\pi^*,\cdot,\cdot)$ задовольняє рішення оптимального рівняння (19).

4) Якщо існує оптимальна стратегія, то існує така стратегія, яка буде оптимальною нерандомізованою стаціонарною.

Лема 1. Нехай u і u_n, де $n = 1, 2 \ldots$ буде напівнеперервною та обмеженою знизу і інфінум-компактоною на Δ, Якщо $u_n \uparrow u$, то [1]:

$$\lim_{n\to\infty} \min_{a\in A_{(x,\alpha)}} u_n(x,\alpha,a) = \min_{a\in A_{(x,\alpha)}} u(x,\alpha,a)\ \forall(x,\alpha) \in X. \qquad (20)$$

Доказ можна опустити через те, що він аналогічний до леми 4.2.4 у [7].

У цьому випадку нам також потрібно існування вимірюваних селекторів, які задовольняють рівняння Динамічного програмування. Для цього ми використовуємо **теорему 1** та наступне визначення.

Означення 4. Позначимо набір невід'ємних вимірних функцій на X як $M(X)^+$, а для кожної функції $u \in M(X)^+$ введемо такий оператор T , що для нього Tu — функція, визначена на X співвідношенням [1]:

$$Tu(x,\alpha) := \min_{a\in A_{(x,\alpha)}} \left\{e^{-\alpha\tau(x,\alpha,a)}\left[c(x,\alpha,a) + \int_X u(y,\beta)Q(d(y,\beta)|x,\alpha,a)\right]\right\}. \quad (21)$$

Лема 2. З 2 припущення T відображає $M(X)^+$ в самого себе, для кожного u в $M(X)^+$ - Tu також є $M(X)^+$, більш того, існує селектор $f \in F$ такий, що для нього [1]:

$$Tu := e^{-\alpha\tau(x,\alpha,f)}\left[c(x,\alpha,f) + \int_X u(y,\beta)Q(d(y,\beta)|x,\alpha,f)\right]\ \forall(x,\alpha) \in X.$$

Лема 3. Припустимо, що виконуються припущення 2 і 3, тоді [1]:

1) Якщо $u \in M(X)^+$ є таким, що $u \geq Tu$, тоді за попередньою лемою (2) можна обрати $f \in F$ такий, що $u \geq \Re^*$.
2) Якщо $u: X \to \boldsymbol{R}$ - це вимірювана функція, що Tu є чітко визначеною на і, крім того, u≤Tu та

$$\lim_{n\to\infty} E^{\pi}_{(x,\alpha)}[e^{-s_n}u(x_n,\alpha_n)] = 0\ \ \forall\pi \in S_R \text{ і } \forall(x,\alpha) \in X. \qquad (22)$$

то $u \leq \Re^*$.

Доведення аналогічне до доведення леми (16) у [1]

Керування напівмарковським процесом (приклад)

Приклад 1. Розглянемо більш конкретний приклад використання напівмарковського процесу прийняття рішень. У якості процесу ми розглянемо умовний капітал банку(x_τ), який потрібно розподілити найкращим чином між інвестиціями – стратегіями (a_τ) і витратами на ці стратегії ($x_\tau - a_\tau$) за N кроків. Для прикладу будемо вважати, що банк не бере кредит для своїх інвестицій, а має лише власний капітал $A_{(x,y)} = [0, x]$. Отже, потрібно максимізувати функцію вартості. Зв'язок між вибором інвестицій і накопиченим капіталом можна подати формулою :

$$x_{\tau+1} = a_\tau \xi_\tau$$

$$\alpha_{\tau+1} = r\alpha_\tau + \eta_\tau$$

де $\{\xi_\tau\}$ та $\{\eta_\tau\}$ - це незалежні послідовності незалежних та однаково-розподілених величин і вони не залежать від початкового стану (x_0, α_0). Припустимо, що $E(\xi_0) = e^m$ з m > 0 і 0 < r <1; передбачається, що віддача за крок і — $d\,(x_i, a_i)$, а разом з витратами — $d\,(x_i, a_i) := b(x_i - a_i)$ з b > 0, і ми хочемо максимізувати очікувану загальну знижену вартість.

Припустимо, що параметри r, m, α та сподівання E задовольняють :

$$E(e^{-\frac{\eta_0}{1-r}}) \geq 1 \text{ і } \alpha_0 + \frac{M}{1-r} \leq m \quad (23)$$

де, $|\eta_0| \leq M$.

При зміні на max в алгоритмі динамічного програмування і $r_{N+1}(x) = 0$ маємо

$$V_N(x, a) = \max_{a\in(0,x)} \{e^{-a\tau_N}\big[\, b(x-a) + E\big(V_{N+1}(y)\big)\big]\}$$

І функція $f_N(x, \alpha) = 0$.

$$V_{N-1}(x, a) = \max_{a\in(0,x)} \{e^{-a\tau_{N-1}}\big[\, b(x-a) + E\big(V_N(y, \beta)\big)\big]\} =$$

$$= b\left(\max_{a\in(0,x)} \{e^{-a\tau_{N-1}}(x + a(-1 + e^m)\}\right)$$

$$= b\left(\max_{a\in(0,x)} \{(xe^{-a\tau_{N-1}} - ae^{-a\tau_{N-1}} + ae^{m-a\tau_{N-1}}\}\right)$$

$$= b(xe^{m-a\tau_{N-1}})$$

Дана нерівність випливає з (23). Оскільки,

$$|a| \le \left| r^{N-2}\alpha_0 + \sum_{i=0}^{N-2} r^i \eta_i \right| \le \alpha_0 + \frac{M}{1-r} \le m$$

А в загальному випадку, для кроку - $V_{N-t}(x,a)$ – маємо функцію вартості :

$$V_{N-n}(x,a) = \max_{a\in(0,x)} \left\{ e^{-a\tau_{N-1}} \left[b(x-a) + E\left(V_{N-t+1}(y,\beta)\right)\right]\right\}$$

для всіх кроків $n = \overline{2;N}$.

Отже, з нерівності (23) випливає, що

$$r^{n-1}\alpha \le m + \ln\left(E\left(e^{-(1+\cdots+r^{n-2})\eta_0}\right)\right) \le \cdots \le d\alpha \le m + \ln\left(E(e^{-\eta_0})\right)$$

З цього випливає, що функція вартості на кроці $N-n$:

$$V_{N-n}(x,a) = b\left(e^{mn-\tau(1+\cdots+r^{n-1})\alpha}\right)E(e^{-\eta_0})E\left(e^{-(1+r)\eta_0}\right)\ldots E\left(e^{-\tau(1+\cdots+r^{n-2})}\right)x$$

і селектор $f_{N-n}(x,a) = x$. Продовжуючи, отримаємо, що оптимальним значенням

$$V_0(x,a) = b\left(e^{mN-\tau(1+\cdots+r^{N-1})\alpha}\right)E(e^{-\eta_0})E\left(e^{-(1+r)\eta_0}\right)\ldots E\left(e^{-\tau(1+\cdots+r^{N-2})}\right)x$$

Отже, стратегія, що приносить максимальний дохід – вкладати усі кошти кожний крок.

Приклад 2. Визначимо стани системи x_1 і x_2. Визначимо простір керуючих станів як 2 стани : a_1 і a_2. Ймовірність переходу для кожного стану а задається матрицями переходу :

$$p_1 = \begin{bmatrix} p_{1,1} = 0.7 & p_{1,2} = 0.3 \\ p_{2,1} = 0.4 & p_{2,2} = 0.6 \end{bmatrix}$$

$$p_2 = \begin{bmatrix} p_{1,1} = 0.9 & p_{1,2} = 0.1 \\ p_{2,1} = 0.2 & p_{2,2} = 0.8 \end{bmatrix}$$

Винагороди та витрати за кожну дію а на :

$$c_1 = \begin{vmatrix} 6 & -5 \\ 7 & 12 \end{vmatrix} \text{ і } c_2 = \begin{vmatrix} 10 & 17 \\ -14 & 13 \end{vmatrix}$$

Припустимо, що математичне сподівання дисконту $E(\beta) = e^m$ з -0.2 < m < -0.1, отже математичне сподівання дисконту буде $E(\beta) = e^{-0.15}$ і 0 < r <1 і час перебування у відповідних станах має нормальний розподіл $0 \le t_1 \le 5$ і $2 \le t_2 \le 7$. Знайдемо оптимальну стратегію на 2 кроки.

Отже, метою агенту у даній задачі є знаходження максимально вигідної стратегії π*. Для визначення оптимальної стратегії введемо поняття оптимальності. У даній задачі метою агенту – є максимізація винагороди(для x_1 і x_2 початкового стану).

1) Знайдемо функції вартості для кожного кроку при стратегії $\pi = \{a_1, a_2\}$:

Для $n = 2$

$V_2(x, a_2) := \max_a \{e^{-\alpha\tau_n}[c(x, \alpha, a) + 0]\}.$

$V_2(1,1) = e^{-0.15*4.5}[10 + 0] = 5.092$

$V_2(1,2) = e^{-0.15*4.5}[17 + 0] = 8.656$

$V_2(2,1) = e^{-0.15*4.5}[-14 + 0] = -7.128$

$V_2(2,2) = e^{-0.15*4.5}[13 + 0] = 6.619$

Для $n = 1$

$V_1(x, a_1) := \max_a \{e^{-\alpha\tau_n}[c(x, \alpha, a) + E(V_2(x, a_2))]\}.$

$E(V_2(1, a_2)) = 5.092 * 0.9 + 8.656 * 0.1 = 5.4484$

$E(V_2(2, a_2)) = -7.128 * 0.2 + 6.619 * 0.8 = 3.8696$

$V_1(1,1) = e^{-0.15*2.5}[6 + E(V_2(1, a_2))] = 4.124 + 3.745 = 7.869$

$V_1(1,2) = e^{-0.15*2.5}[-5 + E(V_2(2, a_2))] = -3.436 + 2.660 = -0.776$

$V_1(2,1) = e^{-0.15*2.5}[7 + E(V_2(1, a_2))] = 4.811 + 3.745 = 8.556$

$V_1(2,2) = e^{-0.15*2.5}[12 + E(V_2(2, a_2))] = 8.247 + 2.660 = 10.907$

$V_0 = 0$, для 0-го кроку і час проведений у цьому кроці – 0.

$V_0 = E(V_1(1, a_1)) = 7.869 * 0.7 - 0.776 * 0.3 = 5.276$

$V_0 = E(V_1(2, a_1)) = 8.556 * 0.4 + 10.907 * 0.6 = 9.967$

2) Теж саме для стратегії $\pi = \{a_2, a_1\}$

Для $n = 2$

$V_2(x, a_2) := \max_a \{e^{-\alpha\tau_n}[c(x, \alpha, a) + 0]\}.$

$V_2(1,1) = e^{-0.15*2.5}[6 + 0] = 4.124$

$V_2(1,2) = e^{-0.15*2.5}[-5 + 0] = -3.436$

$V_2(2,1) = e^{-0.15*2.5}[7 + 0] = 4.811$

$V_2(2,2) = e^{-0.15*2.5}[12 + 0] = 8.247$

Для $n = 1$

$V_1(x, a_2) := \max_a \{e^{-\alpha\tau_n}[c(x, \alpha, a) + E(V_2(x, a_1))]\}.$

$E(V_2(1, a_1)) = 4.124 * 0.7 + -3.436 * 0.3 = 1.856$

$E(V_2(2, a_1)) = 4.811 * 0.4 + 8.247 * 0.6 = 6.8726$

$V_1(1,1) = e^{-0.15*4.5}[10 + E(V_2(1, a_1))] = 5.092 + 0.945 = 6.037$

$V_1(1,2) = e^{-0.15*4.5}[17 + E(V_2(2, a_1))] = 8.656 + 3.499 = 12.155$

$V_1(2,1) = e^{-0.15*4.5}[-14 + E(V_2(1, a_1))] = -7.128 + 0.945 = -6.183$

$V_1(2,2) = e^{-0.15*4.5}[13 + E(V_2(2, a_1))] = 6.619 + 3.499 = 10.118$

$V_0 = 0$, для 0-го кроку і час проведений у цьому кроці – 0.

$V_0 = E(V_1(1, a_2)) = 6.037 * 0.9 - 12.155 * 0.1 = 4.2178$

$V_0 = E(V_1(2, a_2)) = -6.183 \ * 0.2 + 10.118 * 0.8 = 6.8578$

3) $\pi = \{a_1, a_1\}$

Для $n = 2$

$V_2(x, a_2) := \max_a \{e^{-\alpha\tau_n}[c(x, \alpha, a) + 0]\}.$

$V_2(1,1) = e^{-0.15*2.5}[6 + 0] = 4.124$

$V_2(1,2) = e^{-0.15*2.5}[-5 + 0] = -3.436$

$V_2(2,1) = e^{-0.15*2.5}[7 + 0] = 4.811$

$V_2(2,2) = e^{-0.15*2.5}[12 + 0] = 8.247$

Для $n = 1$

$V_1(x, a_2) := \max_a \{e^{-\alpha\tau_n}[c(x, \alpha, a) + E(V_2(x, a_1))]\}.$

$E(V_2(1, a_1)) = 4.124 * 0.7 + -3.436 * 0.3 = 1.856$

$E(V_2(2, a_1)) = 4.811 * 0.4 + 8.247 * 0.6 = 6.8726$

$V_1(1,1) = e^{-0.15*2.5}[6 + E(V_2(1, a_1))] = 4.124 + 1.276 = 5.4$

$V_1(1,2) = e^{-0.15*2.5}[-5 + E(V_2(2, a_1))] = -3.436 + 4.723 = 1.287$

$V_1(2,1) = e^{-0.15*2.5}[7 + E(V_2(1, a_1))] = 4.811 + 1.276 = 6.087$

$V_1(2,2) = e^{-0.15*2.5}[12 + E(V_2(2, a_1))] = 8.247 + 4.723 = 12.97$

$V_0 = 0$, для 0-го кроку і час проведений у цьому кроці – 0.

$V_0 = E(V_1(1, a_1)) = 5.4 * 0.7 + 1.287 * 0.3 = 4.166$

$V_0 = E(V_1(2, a_1)) = 6.087 * 0.4 + 12.97 * 0.6 = 10.217$

4) $\pi = \{a_2, a_2\}$

Для $n = 2$

$V_2(x, a_2) := \max_a \{e^{-\alpha\tau_n}[c(x, \alpha, a) + 0]\}.$

$V_2(1,1) = e^{-0.15*4.5}[10 + 0] = 5.092$

$V_2(1,2) = e^{-0.15*4.5}[17 + 0] = 8.656$

$V_2(2,1) = e^{-0.15*4.5}[-14 + 0] = -7.128$

$V_2(2,2) = e^{-0.15*4.5}[13 + 0] = 6.619$

Для $n = 1$

$V_1(x, a_1) := \max_a \{e^{-\alpha\tau_n}[c(x, \alpha, a) + E(V_2(x, a_2))]\}.$

$E(V_2(1, a_2)) = 5.092 * 0.9 + 8.656 * 0.1 = 5.4484$

$$E\left(V_2(2,a_2)\right) = -7.128 * 0.2 + 6.619 * 0.8 = 3.8696$$

$$V_1(1,1) = e^{-0.15*4.5}\left[10 + E\left(V_2(1,a_1)\right)\right] = 5.092 + 2.774 = 7.866$$
$$V_1(1,2) = e^{-0.15*4.5}\left[17 + E\left(V_2(2,a_1)\right)\right] = 8.656 + 1.97 = 10.626$$
$$V_1(2,1) = e^{-0.15*4.5}\left[-14 + E\left(V_2(1,a_1)\right)\right] = -7.128 + 2.774 = -4.354$$
$$V_1(2,2) = e^{-0.15*4.5}\left[13 + E\left(V_2(2,a_1)\right)\right] = 6.619 + 1.97 = 8.589$$

$V_0 = 0$, для 0-го кроку і час проведений у цьому кроці – 0.

$$V_0 = E\left(V_1(1,a_2)\right) = 7.866 * 0.9 + 10.626 * 0.1 = 8.142$$
$$V_0 = E\left(V_1(2,a_2)\right) = -4.354 \ * 0.2 + 8.589 * 0.8 = 6.0004$$

Отже, виходить що $\pi = \{a_2, a_2\}$ оптимальна випадку, коли починаємо з x_1, а $\pi = \{a_1, a_1\}$ оптимальний для x_2

Висновки

Таким чином, у цій роботі ми розглянули напівмарковський процес прийняття рішень з рандомізованим дисконтом і його практичне застосування, який дозволяє ефективно моделювати системи масового обслуговування, чи, наприклад, у інвестуванні та розподілі капіталу або вкладенні коштів. Також напівмарковський процес можна використовувати для моделювання системи зносу та заміни деталей, пристроїв, приладів у складному обладнанні.

Сучасний світ можна розглядати як безліч систем масового обслуговування (великих, середніх та дрібних). Якщо у дрібних та деяких середніх системах масового обслуговування можна покладатися на інтуїцію інвестора чи спритність менеджменту, то у великих системах масового обслуговування вже ніяк не можна без детальних розрахунків та математичного аналізу. Моделювання цих бізнес-процесів доцільно проводити шляхом використання напівмарковських процесів - як на стадії складання стратегії, так і на подальших стадіях реалізації цієї стратегії і ефективного розподілу прибутку або витрат.

Тобто, можна прогнозувати, що у глобальній економіці по мірі подальшого розвитку світової торгівлі та стрімкого зростання систем масового обслуговування, автоматизації процесів та ускладнення техніки буде зростати потреба у математичному моделюванні процесів, у тому числі шляхом застосування напівмарковських процесів прийняття рішень.

Література

1. Наталія Ковальчук, Руслан Чорней – Напівмарковські процеси прийняття рішень з рандомізованим дисконтом, НАУКОВІ ЗАПИСКИ. Том 151. Комп'ютерні науки, УДК 519.21, 07.08.2013 р.
2. González-Hernández J. Markov control processes with randomized discounted cost / Juan González-Hernández, Raquiel R. López-Martínez, J. Rubén Pérez-Hernández // Math. Meth. Oper. Res. — 2007. — Vol. 65, No. 1.
3. Chornei R. K. Control of Spatially Structured Random Processes and Random Fields with Applications / Ruslan K. Chornei, Hans Daduna, Pavel S. Knopov. — New York: Springer Science + Business Media, Inc., 2006.
4. Wessels, J., & van Nunen, J. A. E. E. . Discounted semi-Markov decision processes : linear programmingand policy iteration. (Memorandum COSOR; Vol. 7401). Eindhoven: Technische Hogeschool Eindhoven. 1974.
5. Fernando Luque-Vasquez, J. Adolfo Minjarez-Sosa - Semi-Markov control processes with unknown holding times distribution under a discounted criterion, Math Meth Oper Res 2005.
6. Arash Khodadadi, Pegah Fakhari, and Jerome R. Busemeyer. Learning to maximize reward rate: a model based on semi-Markov decision processes, Published online 2014 May 23. doi: doi.org/10.3389/fnins.2014.00101
7. Hernández-Lerma O., Lasserre J. B. Discrete-time Markov control processes: basic optimality criteria / O. Hernández-Lerma, J. B. Lasserre. — Berlin Heidelberg New York: Springer, 1996.

Додаток 1

У додатки представлено реалізацію напівмарковського процесу прийняття рішень на мові програмування Python. Дані задаються через 2 таблиці csv.

Дані у «costs.csv» повинні мати наступний вигляд

(x x1), c

x – стан у якому перебуває система

x1 – стан у який переходить система

c – вартість перебування у стані x за умови стратегії, яка переходить в x1

Наприклад:

(0 1), -0.1

translationAndAction.csv

(стан, дія, результат-стан, ймовірність, час перебування у стані)

Наприклад:

(0 1),A,(0 0),0.1,4.0

де,

(0 1) – стан у якому перебуває система,

A – дія,

(0 0) – стан у який перейшла система

4.0 – час який провела система в стані (0 1) за умови стратегії A і переходу у (0 0)

```
import math
import random
import csv

Transitions = {}
TimeOfTransitions = {}
Cost = {}
```

```
def dataCSV():
    #CSV input
    with open('action_time.csv', 'r') as csvfile:
        reader = csv.reader(csvfile, delimiter=',')
        for row in reader:
            if row[0] in Transitions:
                if row[1] in Transitions[row[0]]:
                    Transitions[row[0]][row[1]].append((float(row[3]), row[2]))
                    TimeOfTransitions[row[0]] = float(row[4])
                else:
                    Transitions[row[0]][row[1]] = [(float(row[3]), row[2])]
            else:
                Transitions[row[0]] = {row[1]:[(float(row[3]),row[2])]}

    with open('costs_reward.csv', 'r') as csvfile:
        reader = csv.reader(csvfile, delimiter=',')
        for row in reader:
            Cost[row[0]] = float(row[1]) if row[1] != '0' else None

dataCSV()

class SMDP:

    #Semi-marcov init
    def __init__(self, transition, cost, time):
        #collect all nodes from the transition models
        self.states = transition.keys()
        #initialize transition
        self.transition = transition
        #initialize cost
```

```
        self.cost = cost
        #initialize time
        self.time = time

    def S(self, state):
        #cost
        return self.cost[state]

    def actions(self, state):
        #set of actions
        return self.transition[state].keys()

    def transl(self, state, action):
        #list of transition(prop and res)
        return self.transition[state][action]

    def time(self, state):
        #time
        return self.time[state[0]]



#Initialize the SMDP
smdp = SMDP(transition=Transitions, cost=Cost, time=TimeOfTransitions)



#time bounds
#time = random.uniform(1, 5)
#discount factor bounds
discount = random.uniform(-0.2, -0.1)
#epsilon error
```

```
ep = 0.0005
#randomize discount
def randomD():
    return random.uniform(-0.2, -0.1)



#float -> decimal
def truncate(f, n):
    s = '{}'.format(f)
    if 'e' in s or 'E' in s:
        return '{0:.{1}f}'.format(f, n)
    i, p, d = s.partition('.')
    return '.'.join([i, (d+'0'*n)[:n]])



#discount and time
#res1 = float(truncate(math.pow(math.e, (discount*time)), 4))
curTime = 0
def discountiong(discount, time):
    return float(truncate(math.pow(math.e, (discount*time)), 4))
def randomD_Ta(time):
    dNew = random.uniform(-0.2, -0.1)
    curTime = float(truncate(math.pow(math.e, (dNew*time)), 4))

#value function
def funcV1(V_prev, factor, curr, Cost, transition, actions):
    return factor*(Cost(curr) + max([ sum([p * V_prev[s1] for (p, s1) in transition(curr, a)]) for
a in actions(curr)]))
```

```
#value_at_index = dic.values()[index]
def V_iteration():
    #utility values
    states = smdp.states
    ac = smdp.actions
    S = smdp.S
    tr = smdp.transl
    time = list(smdp.time.values())
    timeS = list(smdp.time)

    #initialize utility
    #states from last to 0 (k=0 case)
    V1 = {s: 0 for s in states}
    while True:
        V = V1.copy()
        difference = 0
        for s in states:
            for i in range(0, len(timeS)):
                if(timeS[i] == s):
                    discount = randomD()
                    curDiscount = discountiong(discount, time[i])
            V1[s] = funcV1(V, curDiscount, s, S, tr, ac)
            difference = abs(V1[s] - V[s])

        #check if difference < epsilon, and break if true
        if difference < ep * (1 - curDiscount) / curDiscount:
            return V



def policy(V):
```

```
    #find optimal
    states = smdp.states
    ac = smdp.actions
    optPolicy = {}
    for s in states:
        optPolicy[s] = max(ac(s), key=lambda a: exceptedR(a, s, V))
    return optPolicy


def exceptedR(a, s, V):
    #SMDP expected V utility of doing a in state s
    tr = smdp.transl
    return sum([p * V[s1] for (p, s1) in tr(s, a)])


#call value iteration
V = V_iteration()

#results
print('State - Value')
for s in V:
    print(s)
    print('Predicted reward or cost: ', V[s])
#best policy for stages
optPolicy = policy(V)
print('Optimal policy for State - Action')
for s in optPolicy:
    print( s, ' - ', optPolicy[s])
```