automatiki i telemechaniki. – T. XX1V, N . – P. 3–27. 9. Wilson I.D. Foundations of hierarhical control / Wilson I.D. // "International Journal of Control" – 1979. N 6. – P. 899–933. 10. Lee K.Y. Coordinated control of distributed-parameter systems / Lee K.Y. // "Distrib. Parameter Contr. Syst.", Oxford. – 1982. – P. 213–238. 11. Michalska H. Joint coordination method for the steady-state control of large-scale systems / Michalska H., Ellis J.E., Roberts P.D. // Int. J. Syst. Sci. – 1985. – N 5. – P. 605–618. 12. Бернштейн Н.А. Архив биологических наук: Проблема взаимоотношений координации и локализации. Т. 38, вып. 1. – 1935. 13. Васильев Ю.В. Теория управления / Ю.В. Васильев, В.Н. Парахина, Л.И. Ушвицкий (ред.) – М.: Финансы и статистика, 2008. – 608 с. 14. Гапоненко А.Л. Теория управления / А.Л. Гапоненко, А.П. Панкрухина. – М.: РАГС, 2003. – 558 с. 15. Фатхутдинов Р.А. Стратегический менеджмент / Фатхутдинов Р.А. – 2-е изд., доп. – М.: ЗАО "Бизнес-школа "Интел-Синтез", 1998. – 416 с.

УДК 004.832.2; 004.852; 004.942

П.О. Кравець, О.М. Проданюк
Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

МАРКІВСЬКІ МЕТОДИ НАВЧАННЯ У СИСТЕМАХ ПРИЙНЯТТЯ РІШЕНЬ

Досліджується проблема оптимального прийняття рішень за допомогою марківських методів навчання. Сформульовано задачу прийняття рішень, описано методи детермінованого та стохастичного навчання. Розроблено алгоритмічне та програмне забезпечення для моделювання прийняття рішень в умовах невизначеності. Наведено та проаналізовано результати комп’ютерного моделювання процесу прийняття рішень у клітинному просторі.

The problem of optimum decision-making by the Markovian learning methods is investigated. The definition of a decision making task is executed, the methods of the deterministic and stochastic learning are described. The algorithmic and software tools for the modelling of decision making in uncertainty conditions are developed. The results of computer simulation of decision-making process in cellular space are resulted and analysed.

Вступ

Сучасні організаційні, економічні, технічні, інформаційні та інші системи є динамічними, ієрархічними, розподіленими, з децентралізованим або комбінованим керуванням, масштабованими, відкритими до структурно-функціональної модифікації та взаємодії. Такі системи, як правило, функціонують в умовах апріорної невизначеності, обумовленої структурно-параметричною неточністю їх математичної моделі [1].

Бажані (оптимальні) режими функціонування системи забезпечуються за допомогою одного або декількох введених у контур зворотного зв’язку агентів [2]. Агент – це автономна активна інтелектуальна система прийняття рішень, здатна опрацьовувати реакції середовища для цілеспрямованого впливу на його стани за допомогою набору керуючих дій.

Проблема оптимального прийняття рішень є однією з найважливіших в теорії штучного інтелекту [3]. В умовах невизначеності вона не вирішується за допомогою одного чи декількох етапів оптимізації, характерних для детермінованих задач математичного програмування [4]. Прийняття рішень в динамічних системах з елементами невизначеності здійснюється на основі ітераційної взаємодії середовища та агента на досить великому проміжку часу, достатньому для розвідування агентом станів середовища, збирання даних, необхідних для побудови оптимальних стратегій поведінки [5, 6]. Оптимальне прийняття рішень забезпечується навчанням агента [7, 8], яке полягає у формуванні адаптивних стратегій, здатних відфільтровувати невизначеності системи. Серед базових методів ітераційного навчання виділимо навчання, підкріплене поточними реакціями середовища [9, 10]. Загалом такий метод не вимагає наявності моделі середовища, що важливо для великих (складних) систем прийняття рішень в умовах невизначеності.

Незважаючи на значний історичний період розвитку, дослідження процесів агентного (машинного) навчання не втрачають своєї актуальності у зв'язку із необхідністю зменшення ризиків прийняття рішень у традиційних та нових галузях людської діяльності [11].

Сучасні дослідження методів навчання агентів переважно ґрунтуються на властивостях марківських випадкових процесів [12,13]. Такі методи є відносно простими для математичного дослідження та зручними для практичної реалізації, оскільки не враховують передісторії динаміки системи прийняття рішень.

Залежності від кількості агентів та організації середовища розрізняють такі підходи до дослідження процесів прийняття рішень: 1) один агент, один стан – навчання одного автомата [14, 15]; 2) декілька агентів, один стан – ігри автоматів [16]; 3) один агент, декілька станів – марківські процеси прийняття рішень [17]; 4) декілька агентів, декілька станів – марківські ігри агентів [18].

Застосування марківських моделей є унікальним для кожного формулювання задачі оптимального прийняття рішень, оскільки залежить від природи середовища та динаміки станів досліджуваної системи. Тому з науково-практичного погляду важливим є формування загальних системотворчих засад прийняття рішень в умовах невизначеності та побудова програмних інструментальних засобів дослідження ефективності методів навчання для вироблення оптимальних рішень у реальному масштабі часу.

Метою роботи є систематизація та формалізація марківських методів навчання агентів, розроблення алгоритмів та програмних засобів моделювання одноагентних систем прийняття рішень для ефективного розв'язування практичних задач.

Марківські процеси прийняття рішень

Марківський процес – це кортеж (S, A, p, r), де S – набір усіх станів системи, A – набір можливих дій агента, $p: S \times A \to \Delta(S)$ – функція зміни станів системи, $r: S \times A \to R$ – функція винагороди. Тут $\Delta(S)$ – набір усіх розподілів імовірностей на множині S. Імовірність зміни станів p залежить тільки від поточного стану середовища і поточних дій агентів:

$$p\left(s_{t+1} = s' \mid (s_t, a_t), t = 0,1,2,...,t\right) = p\left(s_{t+1} = s' \mid s_t, a_t\right).$$

У кожен момент часу t середовище перебуває у стані $s \in S$ і агент вибирає та реалізує дію $a \in A(s)$. В окремих випадках для спрощення можна прийняти, що $A(s) = A, \forall s \in S$. На основі вибору $a \in A$ середовище змінює свій стан згідно з розподілом імовірностей $p(s,a)$, і агент отримує випадковий виграш r. Агент взаємодіє з недетермінованим середовищем, модель якого в загальному випадку йому не відома. Агенту доступні для спостереження лише поточні стани середовища та власні стани і стратегії поведінки.

Функція розподілу станів середовища p набуває значення на відрізку $[0,1]$:

$$\forall s \in S,\ \forall a \in A \quad \sum_{s' \in S} p(s,a,s') = 1,$$

де s – поточний стан системи, s' – стан у наступний момент часу.

Центральною концепцією марківського прийняття рішень є функція p вибору стратегій, яка описує реальну поведінку агента, тобто спосіб перетворення станів у дії:

$$p : S \to A. \tag{1}$$

Функція p визначає імовірності вибору стратегій $a \in A$ агентом у кожному стані середовища:

$$\forall s \in S \qquad \sum_{a \in A} p(s,a) = 1.$$

Розподіл p набуває значення на відрізку $[0,1]$. Якщо $p(s,a) \in \{0,1\}$, то агент здійснює детермінований вибір варіантів рішень.

Метою агента є максимізація функції сумарних виграшів R за рахунок формування ефективної стратегії p :

$$E_p\left[R\right] \to \max_p. \tag{2}$$

Цільова функція очікуваного виграшу (2) залежить від станів середовища, дій агента, функції вибору стратегій та ін., які обумовлюють стохастичну природу системи прийняття рішень. Оскільки поточний виграш є випадковою величиною, то функція очікуваного виграшу формулюється у кумулятивному вигляді, наприклад, сумарної, середньої або сумарної дисконтованої винагороди.

Функція сумарної винагороди нагромаджує поточні виграші за обмежену кількість кроків взаємодії агента з середовищем:

$$R_t = \sum_{i=0}^{h} r_{t+i} \ , \tag{3}$$

де r_t – значення поточних виграшів у момент часу t, h – часовий горизонт.

На відміну від (3), функція середньої винагороди виконує усереднення виграшів у часі:

$$R_t = \frac{1}{h}\sum_{i=0}^{h} r_{t+i} \ . \tag{4}$$

Функції очікуваної винагороди у вигляді (3) та (4) використовується переважно для детермінованих та обмежених за кількістю станів систем.

Критерії оптимальності стохастичних систем, як правило, формулюються на безмежному відрізку часу, що дає змогу отримати стійкі значення характеристик системи прийняття рішень за рахунок згладжування їх випадкових складових. Наприклад, функція (4) може бути визначена так:

$$R_t = \lim_{h \to \infty} \frac{1}{h}\sum_{i=0}^{h} r_{t+i} \ .$$

Функція дисконтованої винагороди визначається на безмежному відрізку часу, причому поточні виграші зважуються додатними коефіцієнтами $g \in (0,1]$, які визначають співвідношення між поточними та прогнозованими (майбутніми) виграшами:

$$R_t = \sum_{i=0}^{\infty} g^{\,i} r_{t+i} \ . \tag{5}$$

Дисконтування поточних виграшів здійснюється за законом геометричної прогресії і при $g < 1$ забезпечує швидку стабілізацію функції очікуваного виграшу. Крім того, значення коефіцієнта g визначає характер прийняття рішень агентами. Близьке до 0 значення коефіцієнта надає перевагу виграшам на короткому відрізку часу (за принципом „краще синиця у руці, ніж журавель у небі”). Навпаки, близьке до 1 значення цього коефіцієнта надає перевагу перспективним виграшам на довгому відрізку часу.

З перерахованих цільових функцій найчастіше у самонавчальних системах прийняття рішень використовується функція (5) з дисконтуванням виграшів. Переважно її використання обумовлено результативністю та легкістю математичних перетворень.

Ефективність (вартість) станів системи визначається значенням функції

$$V_p(s) = E_p\left[R \mid s_0 = s\right], \tag{6}$$

де E_p – очікувана винагорода агента за реалізацію стратегії p, починаючи зі стану середовища s.

Обчислення (6) може бути виконано у рекурсивній формі, відомій у літературі як рівняння Беллмана. Враховуючи (5), після нескладних перетворень отримаємо:

$$V_p(s \mid s_t = s) = E(r_t) + g\sum_{k=0}^{\infty} g^k E(r_{t+k+1}) = E(r_t) + gV_p(s_{t+1}) = r(s, p(s)) + g\sum_{s' \in S} p(s' \mid s, p(s))V_p(s'). \tag{7}$$

де s' – можливі майбутні стани системи.

Метою агента є знаходження функції вибору стратегій p^*, яка максимізує функцію (6) для всіх станів середовища:

$$\forall p\ \forall s \in S \qquad V^{p^*}(s) \geq V^p(s).$$

Оскільки вибирають варіанти дій випадково, то для порівняння ефективності дій, коли система перебуває у стані $s \in S$, поточні виграші корисно отримати з (6). Для цього використовується спеціально побудована Q-функція середніх виграшів, яка визначає ціну дії – сумарний виграш агента, який у стані s вибрав дію a:

$$Q_p(s, a) = E_p\left[R \middle| s_0 = s, a_0 = a\right], \tag{8}$$

Тут $Q_p(s, a)$ є табличною функцією значень варіантів дій a у станах s.

Аналогічно до (7) отримаємо:

$$Q_p(s, a) = r(s, a) + g\sum_{s' \in S} p(s' \mid s, a)V_p(s'). \tag{9}$$

Дотримання принципу оптимальності Беллмана забезпечує оптимальний виграш агента з досягнутого поточного стану $s \in S$ в усі майбутні моменти часу. Застосування цього принципу для усіх станів забезпечує досягнення глобального оптимального розв'язку.

Для оптимальної функції вибору стратегій p^* для кожного стану $s \in S$ отримаємо:

$$V_{p^*}(s) = \max_{a \in A}\left[r(s, a) + g\sum_{s' \in S} p(s' \mid s, a)V_{p^*}(s')\right]. \tag{10}$$

З (10) можна отримати оптимальну функцію вибору стратегій

$$p^*(s) = \arg\max_{a \in A} Q_{p^*}(s, a). \tag{11}$$

Оптимізація (11) може бути виконана методами динамічного програмування [17].

Методи навчання агента

Рівняння Беллмана (9) покладено в основу марківських навчальних систем, які використовуються при невідомій функції виграшів або функції зміни станів системи.

Відомі два базові методи для обчислення оптимальних стратегій поведінки агентів – ітерація за стратегіями (метод послідовного наближення у просторі стратегій) та ітерація за критеріями (метод послідовних наближень у просторі функцій) [9, 10].

Ітерації за стратегіями ґрунтуються на оцінюванні поточного значення стратегії, яке потім вдосконалюється, з використанням алгоритму „жадібної" оптимізації [19]. Суть цього алгоритму полягає у прийнятті на кожному етапі локально-оптимальних рішень, вважаючи, що результуюче рішення буде глобально-оптимальним. Ітерації за критеріями ґрунтуються на послідовних

наближеннях значення функції, причому немає необхідності у повторному обчисленні її точного значення.

Алгоритм методу ітерації за стратегіями зводиться до такої послідовності кроків:

1) задати $t = 0$; ініціалізувати функції виграшів V_t, наприклад, $V_t(s) = 0$; ініціалізувати функцію вибору стратегій $p(s) \in A(s) \ \forall s \in S$;

2) для всіх $s \in S$ виконати ітерації

$$V_{t+1}(s) = r(s, p(s)) + g \sum_{s' \in S} p(s' \mid s, p(s)) V_t(s')$$

за моментами часу $t := t + 1$ поки не виконається умова точності $|V_{t+1}(s) - V_t(s)| \le e$;

3) запам'ятати поточне значення функції $p_{old}(s) = p(s)$, $\forall s \in S$;

4) для всіх $s \in S$ знайти уточнене значення

$$p(s) = \arg\max_{a \in A} \left[r(s, a) + g \sum_{s' \in S} p(s' \mid s, a) V_{t+1}(s') \right];$$

5) якщо $p_{old} <> p$, то перейти на крок 2.

Алгоритм ітерації по критеріях складається з таких кроків:

1) для всіх $s \in S$ виконати кроки 2 – 6;

2) задати $\Delta = 0$, $t = 0$; ініціалізувати функції виграшів $V_t(s)$ довільними значеннями, наприклад, $V_t(s) = 0$;

3) присвоїти $v = V_t(s)$ і для всіх $a \in A$ виконати

$$Q(s, a) = r(s, a) + g \sum_{s' \in S} p(s' \mid s, a) V_t(s');$$

4) визначити $V_{t+1}(s) = \max_{a \in A} Q(s, a)$;

5) обчислити $\Delta = \max(\Delta, | v - V_{t+1}(s) |)$;

6) якщо $\Delta \ge e$ (поки функція вибору стратегій є не добре визначеною), то задати наступний момент часу $t := t + 1$ і перейти на крок 3.

7) вивести значення функції вибору стратегій p для кожного $s \in S$:

$$p(s) = \arg\max_{a \in A} Q(s, a).$$

Обидва методи оптимізації оперують з характеристичною функцією системи $V(s)$. В умовах невизначеності ця функція може бути обчислена методом її поновлення на основі нагромадження даних у кожній ітерації взаємодії агента з середовищем:

$$V_{t+1}(s_t) = (1 - a_t) V_t(s_t) + a_t \left[r_{t+1} + g V_t(s_{t+1}) \right], \tag{12}$$

де $a_t \in [0, 1]$ – параметр, який визначає швидкість навчання.

Метод (12) відомий у літературі під назвою *Temporal Difference Learning* або $TD(0)$ [10]. Головною ідеєю цього методу є наближення функції $V(s_t)$ у напрямку бажаного значення $r_{t+1} + g V_t(s_{t+1})$. Інакше, метод враховує реакцію середовища на один крок вперед.

Метод $TD(l)$ будується на основі врахування декількох попередніх кроків:

$$\forall s \in S \quad V_{t+1}(s_t) = V_t(s_t) + a_t \left[r_{t+1} + g V_t(s_{t+1}) - V_t(s_t) \right] e(s_t), \tag{13}$$

де $e(s_t)$ – коефіцієнт відповідності (важливості) стану.

Значення коефіцієнта $e(s_t)$ пропорційне кількості відвідувань стану s в минулому:

$$e(\$_t) = \sum_{k=1}^{t} (lg)^{t-k} c[\$_t = s_k],$$

де $l \in [0,1]$; $c[\$_t = s_k] \in \{0,1\}$ – індикаторна функція події.

Коефіцієнт $e(\$_t)$ можна обчислити у реальному часі:

$$e(\$_t) = lg\, e(\$_{t-1}) + c(\$_t = s_t).$$

Інший спосіб числової ідентифікації середовища прийняття рішень полягає у застосуванні методу Q-*навчання* [9, 10]:

$$Q_{t+1}(s_t, a_t) = (1 - a_t) Q_t(s_t, a_t) + a_t \left[r_{t+1} + g \max_{b \in A} Q_t(s_{t+1}, b) \right]. \quad (14)$$

На відміну від функції $V(s)$, яка визначена у просторі станів середовища, функція $Q(s,a)$ визначена у просторі станів та дій агентів. Це робить зручним її використання у системах керування, коли критеріальна функція визначається для кожної пари стан–дія.

Метод (14) виконує додаткову операцію максимізації Q-функції у стані s_{t+1}. Оскільки Q-функція (8) залежить від стратегій p, то її оптимізація без значних труднощів можлива для стаціонарних стратегій. На відміну від (14), наступний метод SARSA-навчання (назва походить від імплікації $(s,a,r) \to (s',a')$) оперує з поточними значеннями стратегій і може використовуватися для нестаціонарних стратегій:

$$Q_{t+1}(s_t, a_t) = (1 - a_t) Q_t(s_t, a_t) + a_t \left[r_{t+1} + g\, Q_t(s_{t+1}, a_{t+1}) \right]. \quad (15)$$

Використання методу (15) накладає ряд додаткових обмежень на шукані стратегії. Так, цей метод забезпечує оптимальність Q-функції, якщо стратегії p в асимптотиці часу забезпечують оптимальний вибір дій. Однак, метод (15) має переваги при його застосуванні у задачах прогнозування та оптимального керування.

На початковому відрізку часу агент є ненавченим, і його стратегії повинні забезпечувати можливість максимально глибокого розвідування пошукового простору, який визначається станами середовища та можливими діями агента. Це забезпечує можливість отримання під час навчання глобального оптимального значення Q-функції. Навчений агент реалізує „жадібний" алгоритм поведінки, вибираючи дії, які відповідають максимальним значенням Q-функції, що зменшує час розв'язування оптимізаційної задачі.

Аналогічно до (13) можна побудувати $Q(l)$ -метод:

$$\forall \$ \in S, \forall \alpha \in A \quad Q_{t+1}(\$_t, \alpha_t) = Q_t(\$_t, \alpha_t) + a_t \left[r_{t+1} + g\, Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t) \right] e(\$_t, \alpha_t), \quad (16)$$

$$e(\$_t, \alpha_t) = lg\, e(\$_{t-1}, \alpha_{t-1}) + c(\$_t = s_t \wedge \alpha_t = a_t).$$

Методи (12) – (16) виконують обчислення оптимальних середніх виграшів на основі поточних стимулів і фактично є методами теорії стохастичної апроксимації. Умови їх збіжності визначаються положеннями цієї теорії [20]. Так, при $t \to \infty$ функція $Q_{t+1}(s_t, a_t)$ збігається до оптимального значення

$$Q^*(s,a) = V(s) = \sum_a p(a \mid s) \max_{a \in A} Q(s,a)$$

при дотриманні базових умов стохастичної апроксимації для спадних невід'ємних послідовностей величин $a_t = t^{-l}$ $(l > 0)$:

$$\sum_{t=0}^{\infty} a_t = \infty, \qquad \sum_{t=0}^{\infty} a_t^2 < \infty.$$

Методи (12) – (16) є базовими для побудови інших методів навчання у системах прийняття рішень.

Обчислення стратегій агента

Для детермінованих середовищ стратегії агента визначаються за максимальним значенням Q-функції:

$$p(a \mid s) = \arg\max_a Q(s,a). \tag{17}$$

В умовах невизначеності розвідування простору стан–дія системи прийняття рішень може бути виконано на основі випадкового розподілу, пропорційно до значень функцій $Q(s,a)$.

Одним із варіантів імовірнісного обчислення стратегій є використання "e-жадібного" алгоритму випадкового вибору:

$$\forall a' <> a \;\; p(a' \mid s) = e << 1;\; p(a \mid s) = 1 - e,\; a = arg\max_a Q(s,a). \tag{18}$$

Імовірність вибору агентом дії a, якщо середовище перебуває у стані s, інакше може бути визначена так:

$$p(a \mid s) = \frac{Q(s,a) + e}{\sum_a Q(s,a) + \mid A_s \mid e}, \tag{19}$$

де $Q(s,a) \geq 0$; $\mid A_s \mid$ – потужність множини (кількість) варіантів дій агента у стані s; $0 < e << 1$ – зміщення, необхідне для невиродженості відношення (19) при $Q(s,a) = 0 \;\; \forall a \in A_s$.

Інший відомий вид розподілу – це розподіл Больцмана:

$$p(a \mid s) = \frac{e^{Q(s,a)/T}}{\sum_a e^{Q(s,a)/T}}, \tag{20}$$

де T – температурний параметр системи. Для великих значень T реалізується близький до рівномірного випадковий розподіл. Для малих значень T реалізується розподіл, близький до „жадібного" вибору дій агентів.

У реальних задачах великий розмір пошукового простору призводить до значного зростання часу навчання. Якщо цей час є критичним, то, за раціональними міркуваннями, виконують редукцію пошукового простору, обмежуючись пошуком у локальних областях. Загалом це призводить до отримання субоптимальних розв'язків задачі.

Прийняття рішень у задачах клітинного світу

Задачі клітинного світу – це різноманітні задачі штучного інтелекту, які можуть бути зведені до оптимальної поведінки у дискретизованому середовищі, наприклад, розвідування простору, переслідування, позиційні ігри, пошук виходів із лабіринтів, моделювання гри у футбол, розподіл ресурсів та інші.

Для прикладу розглянемо задачу пошуку автономним агентом джерела живлення за мінімальну кількість кроків (знаходження найкоротшого шляху, який максимізує функцію $V(s) \;\; \forall s \in S$) [11].

Агент, починаючи зі стану $s_{init} = \{w\}$, з імовірністю $P(w \mid t = 0) = 1$ потрапляє у клітинний простір, зображений на рис. 1 у вигляді графу переходів. У моменти часу $t = 1, 2, \ldots$, перебуваючи у стані $s = \{i * 4 + j\}$, $i = \overline{0,2}$, $j = \overline{0,3}$, агент вибирає один із варіантів дій $a \in A = \{r, l, d, u, n\}$, прямуючи до поглинаючого стану s_{end} з оцінкою $V(s_{end}) = 0.0$. У сусідніх до поглинаючого $s_{end}^{neighbor}$ станах агент отримує поточний виграш $r(s_{end}^{neighbor}, a, s_{end}) = 100$. В усіх інших станах за кожну вибрану дію агент отримує поточний виграш $r(s,a) = 0$.

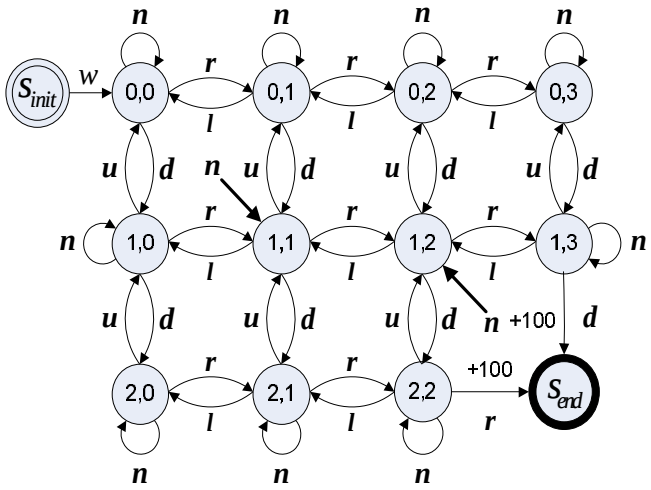

Рис. 1. Простір станів клітинного середовища

Ця задача може бути розв'язана у детермінованому або стохастичному формулюванні.

У детермінованій задачі моделі середовища та агента не залежать від випадкових факторів. Виграші агента є відомими апріорі для кожної пари стан–дія, а стратегії агента вибираються за правилом (17).

Розв'язування детермінованої задачі виконаємо методом динамічного програмування, виконавши оцінювання Q-функції методом (14) з параметрами $a = 1$ та $g = 0.9$ для початкових значень $Q(s,a) = 0$, $\forall s \in S, \forall a \in A_s$. Оцінювання Q-функції виконано методом ітерації за критеріями з точністю $e = 10^{-6}$.

Результати розв'язування детермінованої задачі зображено на рис. 2. У вершинах графу подано оптимальне значення функції вартості станів $V_{p^*}(s)$, а біля ребер графу – значення функції ефективності дій $Q_{p^*}(s,a)$ агента.

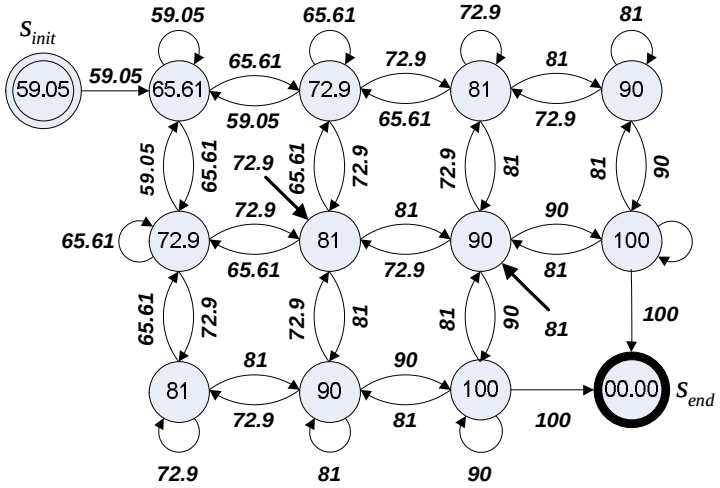

Рис. 2. Оптимальні значення функцій вартості станів $V_{p^*}(s)$

та ефективності дій $Q_{p^*}(s,a)$ агента

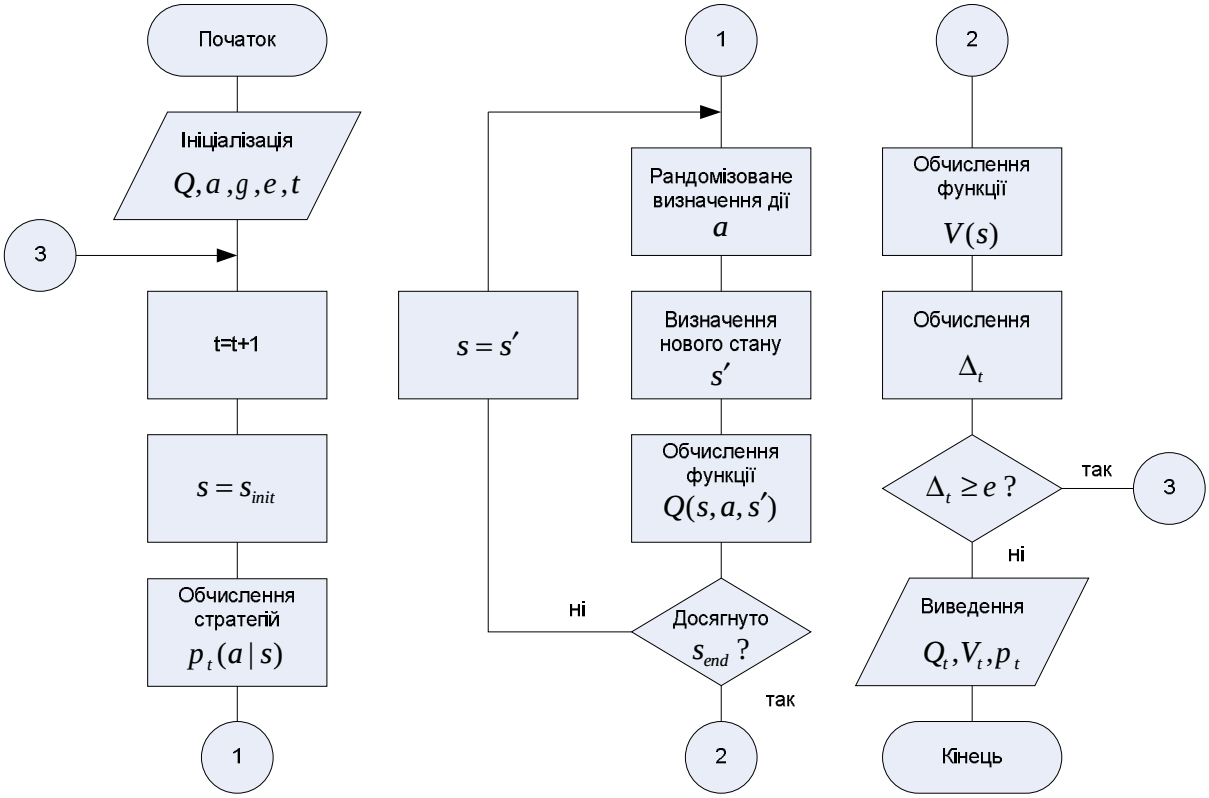

Рис. 3. Графічна схема алгоритму рандомізованого Q-навчання

Для знаходження оптимальної стратегії p, яка забезпечує перехід від стану s_{init} до стану s_{end} найкоротшим шляхом, необхідно для кожного стану $s \in S$ визначити дію $a \in A_s$, яка максимізує значення критерію $Q(s,a)$. Для детермінованої задачі визначення оптимальної стратегії здійснюється згідно із (17). Як видно із рис. 2, сформульована задача має декілька оптимальних розв'язків.

Сформульована задача може бути розв'язана також за допомогою механізму випадкового вибору стратегій, який може бути рівноймовірним або ефективнішим, адаптивним чи побудованим на основі раціональних евристик, подібних до (18) – (20). Застосування випадкових стратегій особливо доцільне у випадку стохастичної природи середовища прийняття рішень. Як правило, стохастичний вибір стратегій вимагає більше кроків для прийняття оптимальних рішень порівняно з детермінованим вибором.

Дослідження ефективності стохастичної поведінки агента виконаємо на основі функції розподілу стратегій (19). Збіжність методу зі стохастичним вибором стратегій оцінимо за допомогою різницевого варіанта критеріальної функції

$$\Delta_t = |S|^{-1} \sum_{s \in S} |V_{t+1}(s) - V_t(s)|, \qquad (21)$$

де $|S|$ – потужність множини станів системи.

Алгоритм Q-навчання з випадковим вибором стратегій зображено на рис. 3.

Результати розв'язування оптимізаційної задачі зі стохастичним вибором варіантів дій агента подано на рис. 4 у вигляді графіків усередненої у часі різницевої функції:

$$\overline{\Delta}_t = t^{-1} \sum_{t=0}^{t} \Delta_t \ .$$

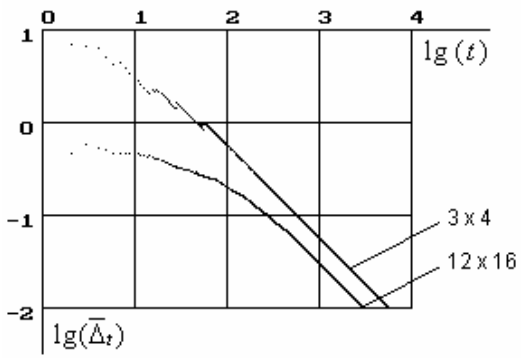

Рис. 4. Збіжність стохастичного методу Q-навчання

Графіки рис. 4 зображено у логарифмічному масштабі для двох розмірностей клітинного простору: 3×4 та 12×16 станів. Зменшення значення різницевої функції у часі свідчить про збіжність методу випадкового вибору стратегій. Швидкість збіжності методу можна оцінити пропорційно куту нахилу лінійної апроксимації функції Δ_t з віссю часу. Як видно з рис. 4, із збільшенням кількості станів системи швидкість збіжності досліджуваного методу зменшується.

На рис. 5. подано відповідні рис. 4 графіки функції ефективності дій агента:

$$\overline{Q}_t = t^{-1}\sum_{t=0}^{t} Q_t \ .$$

Поточне значення критеріальної функції обчислено усередненням за всіма станами середовища та варіантами дій агента:

$$Q_t = |S|^{-1}\sum_{s\in S}|A_s|^{-1}\sum_{a\in A_s} Q_t(s,a).$$

Під час ітераційного розв'язування задачі стабілізується середнє значення Q-функції, що підтверджує збіжність методу Q-навчання.

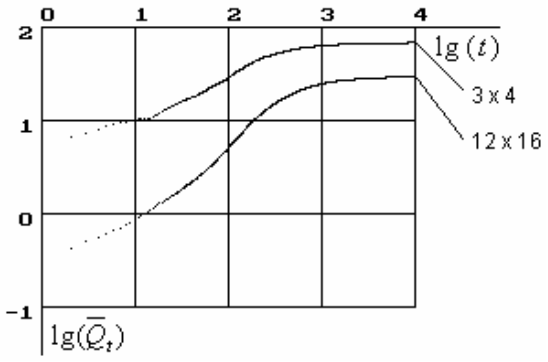

Рис. 5. Динаміка Q-функції у часі

Залежно від розміру клітинного простору для знаходження оптимальних розв'язків задачі необхідно від декількох сотень до тисяч кроків. Відповідні результати, отримані із зростанням розміру клітинного простору, подано у вигляді графіків на рис. 6. Вісь абсцис визначає кратність k базового простору розміром 3×4 стани за кожним координатним напрямком, а вісь ординат $\bar{t}$ – середню кількість кроків, необхідних для досягнення розв'язку з точністю $e=10^{-6}$.

Середню кількість кроків, необхідних для розв'язування пошукової задачі, обчислено за $m = 100$ реалізаціями випадкових послідовностей:

$$\bar{t} = m^{-1} \sum_{i=1}^{m} t_{out}(i),$$

де $t_{out} = \{t \mid \Delta_t < e\}$ – момент завершення розв'язування задачі з точністю e.

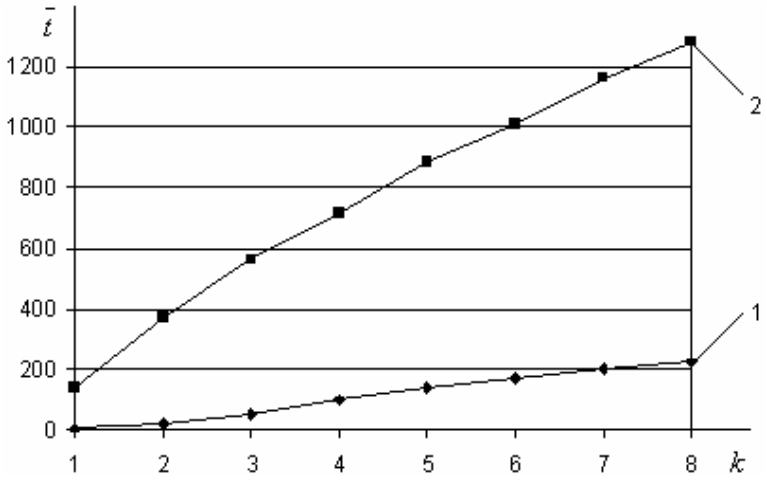

Рис. 6. Залежність середньої кількості кроків стохастичного Q-навчання від розміру клітинного простору $k^2 * (3 \times 4)$: 1 – часткове розвідування станів; 2 – повне розвідування станів

Враховуючи (21), для цієї задачі можливе виконання умови точності $\Delta_t = 0$ при неповному розвідуванні пошукового простору завдяки багатоваріантності оптимальних розв'язків та випадковому характеру (19) вибору стратегій агентом. У цьому випадку отримуємо один (або декілька) з множини оптимальних розв'язків задачі. Графік 1 на рис. 6 показує середню кількість кроків, необхідних для оптимізації Q-функції при частковому, а графік 2 – при повному розвідуванні станів клітинного простору. Виконання умови повного розвідування простору призводить до зростання у десятки разів часу розв'язування задачі.

Висновки

Розв'язано задачу марківського навчання агента для прийняття оптимальних рішень в детермінованому середовищі на прикладі пошуку найкоротшого шляху у клітинному просторі. Виконано математичне формулювання задачі прийняття рішень з детермінованими та стохастичними стратегіями на основі рекурентного оцінювання критеріальних функцій корисності станів та ефективності варіантів дій агента. Оцінювання критеріальних функцій відбувається у реальному масштабі часу на основі підкріпленого Q-навчання і не вимагає наявності моделі середовища, що важливо для практичних застосувань прийняття рішень в умовах невизначеності. Досягнутий методом навчання оптимальний розв'язок задачі є стійким до механізму випадкового вибору стратегій, про що свідчать зменшення різницевої функції та стабілізація значення Q-функції з часом. Отримані результати та розроблене алгоритмічно-програмне забезпечення можуть бути адаптовані для розв'язування задач оптимального прийняття рішень у системах з іншим інформаційним базисом.

Окремої уваги вимагають розподілені системи, в яких прийняття рішень здійснюється множиною агентів, що значно ускладнює задачу за рахунок зростання кількості станів середовища та рівня невизначеності за рахунок дій інших агентів. Для дослідження процесів прийняття рішень в

мультиагентних системах ефективними є ігрові моделі, які дають змогу дослідити та оптимізувати процеси самоорганізації, координації, взаємодії, комунікації, конкуренції та кооперації дій агентів. Розв'язування задач прийняття рішень в ігровому формулюванні є предметом перспективних досліджень у галузі розподіленого штучного інтелекту.

1. *Бурков В.Н. Теория активных систем: состояние и перспективы [Текст] / В.Н. Бурков, Д.А. Новиков. – М.: СИНТЕГ, 1999. – 128 с. 2. Wooldridge, M. An Introduction to Multiagent Systems [Текст] / M. Wooldridge. – John Wiley & Sons (Chichester, England), 2002. – 366 pp. 3. Катренко А.В. Теорія прийняття рішень / А.В. Катренко, В.В. Пасічник, В.П. Пасько. – Київ: BHV, 2009. – 450 с. 4. Трухаев Р.И. Модели принятия решений в условиях неопределенности / Р.И. Трухаев. – М.: Наука, 1981. – 257 с. 5. Растригин Л.А. Адаптация случайного поиска [Текст] / Л.А. Растригин, К.К. Рипа, Г.С. Тарасенко. – Рига: Зинатне, 1978. – 244 с. 6. Назин, А.В. Адаптивный выбор вариантов: Рекуррентные алгоритмы [Текст] / А.В. Назин, А.С. Позняк. – М.: Наука, 1986. – 288 с. 7. Weiss G. Adaptation and Learning in Multiagent Systems [Текст] / Gerhard Weiss, Sandip Sen, editors. – Berlin: Springer Verlag, 1996. – 585 pp. 8. Stone, P. Layered Learning in Multiagent Systems [Текст] / P. Stone. – MIT Press, 2000. – 300 pp. 9. Watkins, C.J.C.H. Q-Learning [Текст] / C.J.C.H. Watkins, P. Dayan // Machine Learning, No. 8. – Kluwer Academic Publishers, Boston. – 1992. – PP. 279–292. 10. Sutton, R. S. Reinforcement Learning: An Introduction [Текст] / Richard S. Sutton, Andrew G. Barto. – MIT Press, 1998. – 322 pp. 11. Mitchell, T.M. Machine Learning [Текст] / T.M. Mitchell. – New York: McGraw-Hill, 1997. – 414 pp. 12. Майн, Х. Марковские процессы принятия решений [Текст] / Х. Майн, С. Осаки. – М.: Наука, 1977. – 176 pp. 13. Filar, J. Competitive Markov Decision Processes [Текст] / Jerzy Filar, Koos Vrieze. – Springer-Verlag, 1997. – 393 pp. 14. Цетлин М.Л. Исследования по теории автоматов и моделированию биологических систем [Текст] / М.Л. Цетлин. – М.: Наука, 1969. – 316 с. 15. Поспелов Д.А. Вероятностные автоматы [Текст] / Д.А. Поспелов. – М: Энергия, 1970. – 88 с. 16. Варшавский В.И. Коллективное поведение автоматов [Текст] / В.И. Варшавский. – М.: Наука, 1973. – 408 с. 17. Puterman, M. L. Markov Decision Processes: Discrete Stochastic Dynamic Programming [Текст] / M. L. Puterman. – John Wiley & Sons, New York, 2005. – 649 pp. 18. Fudenberg, D. The Theory of Learning in Games [Текст] / D. Fudenberg, D. K. Levine. – Cambridge, MA: MIT Press, 1998. – 276 pp. 19. Кормен Томас Х. Алгоритмы: построение и анализ. – 2-е изд. [Текст] / Томас Х. Кормен и др. – М.: Вильямс, 2006. – 1296 с. 20. Вазан М. Стохастическая аппроксимация [Текст] / М. Вазан. – М.: Мир, 1972. – 295 с.*