УДК 004.89, 531

А. О. Яценко, Д. Г. Досин

ПОРІВНЯННЯ ЕФЕКТИВНОСТІ АЛГОРИТМІВ ПЛАНУВАННЯ, РЕАЛІЗОВАНИХ ДЛЯ МАРКІВСЬКОЇ МОДЕЛІ КЛІЄНТА ПОШУКОВОЇ СИСТЕМИ

Two basic algorithms for finding an optimal strategy for MDP plan had been investigated and compared. Such plan of an intelligent agent represents the information needs of a user of an information retrieval service. Algorithms and software implemented and studied on the test sample tasks.

Keywords: *Markov decision processes, optimal policy, utility functions, information search.*

Досліджено два базові алгоритми пошуку оптимальної стратегії інтелектуального агента для MDP плану, яким представлено інформаційні потреби клієнта служби інформаційного пошуку. Алгоритми програмно реалізовано і досліджено на тестових модельних задачах.

Ключові слова: *марківські процеси прийняття рішень, оптимальна стратегія, функція корисності, інформаційний пошук.*

Останніми десятиліттями надзвичайно гостро постає проблема пошуку потрібної інформації в практично необмеженому масиві, доступному через мережу Інтернет, автоматизації процесів добування знань з відкритих джерел без участі людини. Існуючі пошукові системи побудовані на засадах інтерактивного пошуку, за якого людина-клієнт пошукової служби задає деяку множину ключових слів як модель своєї інформаційної потреби і в процесі отримання результату пошуку ітеративно уточнює цю множину, додаючи або вилучаючи з неї ключові слова. Фактично, це – ручний метод пошуку, який не може бути застосований у більшості випадків, зокрема, для неперервного моніторингу повідомлень, фільтрації поштового спаму тощо. Автоматизації перешкоджає вкрай недосконала модель інформаційних потреб клієнта служби інформаційного пошуку. Єдиною ефективною альтернативою такій моделі, на наш погляд, є подана формальною мовою база знань інтелектуального агента, що міститиме інформацію про те, які задачі стоять перед клієнтом і які методи їх розв'язання він використовує [1]. Отже, фактично база знань являє собою план функціонування інтелектуального агента, що складається з множини станів (умов задачі та етапів її рішення) і множини дій (методів рішення окремих підзадач), що переводять агента зі стану в стан з певною імовірністю. Раціональний інтелектуальний агент будує оптимальну стратегію реалізації такого плану, максимізуючи загальний виграш. Теорія автоматичного планування пропонує математичний апарат марківських процесів прийняття рішень. Така формалізація може бути достатньо ефективною для чисельної оцінки релевантності текстового документа в сенсі потрібності клієнту для вирішення його задачі. При цьому залишається вибрати та реалізувати швидкий та ефективний алгоритм розрахунку оптимальної стратегії для заданого плану інтелектуального агента та відповідної такій стратегії сумарної корисності. Різниця корисності стратегії для планів до внесення інформації з досліджуваного документа та після врахування такої інформації і становитиме чисельну оцінку корисності цієї інформації та документа в цілому [2].

Постановка задачі. Ставимо задачу вибрати або розробити та дослідити алгоритми розрахунку оптимальної стратегії прийняття рішень для інтелектуального агента, поданого його планом діяльності, а також очікуваної корисності такої

стратегії з врахуванням того, що цей агент є клієнтом служби інформаційного пошуку і такий розрахунок виконується багаторазово з метою оцінення цінності знайденого текстового документа. Дослідження необхідно виконувати шляхом порівняння стратегій, отриманих у результаті роботи алгоритмів для ідентичних вхідних даних та структур середовища. Вибір оптимальної стратегії здійснюється за критерієм максимального загального виграшу (очікуваної корисності) від діяльності агента. Відповідно середовище моделюється шляхом накладання потрібної моделі на деякий універсальний граф середовища.

MDP для повністю спостережуваного середовища. Марківські процеси прийняття рішень MDP призначені для створення плану роботи агента у стохастичному середовищі з марківською моделлю переходів та множиною цілей. Їх поділяють на два типи: MDP Марківські процеси прийняття рішень у повністю спостережуваному середовищі та, POMDP Марківські процеси прийняття рішень в частково спостережуваному середовищі частковим випадком яких є MDP. Їх основна ідея полягає в поданні задачі планування як задачі оптимізації [3–5].

Марківські процеси прийняття рішень мають такі властивості:

1. Середовище моделюється як стохастична система, тобто це система з недетермінованими переходами між станами, з розподілом функції імовірності на кожен перехід. Отже, дії (переходи) моделюються на основі розподілу функції імовірності.

2. Цілі визначаються значеннями функції корисності.

3. Плани подані у вигляді стратегій, які визначають ті дії, які потрібно виконувати в тому чи іншому стані.

4. Проблему планування розглядають як задачу оптимізації, в якій алгоритми планування будують план, що максимізує функцію корисності.

5. Частково спостережуване середовище моделюється на основі імовірного стану. Імовірний стан – це розподіл імовірності перебування в деякому стані на усі можливі стани. Проблема планування при частковому спостережуваному середовищі розв'язується як задача планування при повністю спостережуваному середовищі в просторі імовірнісного стану і створенні стратегій, які зіставляють імовірні стани з діями.

Середовище описують за допомогою трьох компонентів [4, 5]:

S множина станів.

A множина дій.

$P_s(s' \mid a)$ імовірності переходу P з стану s в s' при виконанні дії a.

Результатом роботи планувальника є стратегія. Стратегія визначає ті дії, які потрібно виконувати в тому чи іншому стані. Стратегію π подають як функцію відображення множини станів на множину дій:

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

Якщо агент має повний опис стратегії, то він завжди знає, що робити далі незалежно від результату попередньої дії. При неодноразовому здійснені стратегії, починаючи з початкового стану, стохастичний характер середовища призводить до формування кожного разу іншої історії перебування агента в середовищі. Тому доцільно вважати за визначення ефективності стратегії очікувану корисність від можливих історій перебування в середовищі, які створюють за допомогою цієї стратегії. Оптимальна стратегія π^* – стратегія, яка забезпечує максимальну очікувану корисність [3, 4].

Корисність стану s визначають як різницю прибутку від перебування в стані s $R(s)$ та затрат на виконання дії a $C(s,a)$:

$$V(s,a) = R(s) - C(s,a). \tag{2}$$

Тоді можна вважати, що корисність стратегії $V(\pi)$ – це сума корисності станів історії побудованої згідно із стратегією π, де для кожного стану *s* виконується дія відповідно до стратегії $\pi(s)$.

$$V(\pi) = \sum_{i \geq 0} V(s_i, \pi(s_i)). \quad (3)$$

Підставивши сюди формулу (3) отримаємо

$$V(\pi) = \sum_{i \geq 0} (R(s_i) - C(s_i, \pi(s_i))). \quad (4)$$

Одна з проблем такого визначення у тому, що така функція корисності при нескінченній множині станів постійно наростатиме, що не дасть можливості порівняти різні історії стратегії. Поширений спосіб уникнення такої ситуації – це введення коефіцієнта знецінення γ, що описує перевагу поточних винагород над майбутніми і дає змогу навіть у нескінченній множині станів визначити підмножину, на основі якої можна чітко обчислити функцію корисності. Тоді формула (4) набуде такого вигляду:

$$V(\pi) = \sum_{i \geq 0} \gamma^i (R(s) - C(s_i, \pi(s_i))). \quad (5)$$

Отже, цінність будь-якої стратегії – це очікувана сума отриманих знецінених винагород. Звідси випливає, що оптимальна стратегія π^* – це стратегія з максимальною цінністю:

$$\pi^* = \arg\max_{\pi} V\left[\sum_{i \geq 0} \gamma^i (R(s) - C(s_i, \pi(s_i))) \mid \pi\right]. \quad (6)$$

Алгоритми планування. ***Корисність станів.*** Виконання стратегії вибудовує певну історію перебування в середовищі, а та, відповідно, є послідовністю станів, які відвідує агент, отже, цінність стану залежить не лише від власної корисності, а й від корисності станів, у які з нього можна потрапити [3–5]. У такому випадку можна вважати, що цінність деякого стану *s* дорівнює сумі безпосередньої винагороди за перебування в цьому стані та очікуваної знеціненої корисності наступного стану s', за умови, що агент вибирає оптимальну дію *a*.

Отже, цінність $V(s)$ стану *s* набуде такого вигляду:

$$V(s) = R(s) + \gamma \max_a \sum_{s'} (P_s(s' \mid a) V(s') - C(s, a)). \quad (7)$$

Рівняння (7) називають рівнянням Белмана. Отже, на основі рівнянь Белмана корисність деякого стану визначають з послідовності наступних станів.

Алгоритм ітерації за значеннями. В основі алгоритму ітерації за значеннями (див. алгоритм 1) лежить розв'язок рівнянь Белмана – по одному для кожного стану: якщо в середовищі є *n* станів, то кількість рівнянь Белмана також *n*. На першому кроці алгоритму потрібно випадковим чином обрати цінності $R_0(s)$ для кожного стану $s \in S$ середовища. Після цього відбувається багаторазове уточнення значення корисності. Для усіх станів *s* на кожному кроці *к* розраховують значення очікуваної корисності $V(s)_k$, базуючись на значеннях $V(s)_{k-1}$, які були розраховані на попередньому кроці. Алгоритм знаходить такі дії *a*, при яких $V(s)$ набуває максимальних значень, і записує їх у стратегію. Після проведення достатньої кількості ітерацій можна стверджувати, що похибка точності визначення корисності станів та відповідно побудованої стратегії набуває деякого значення *j*, яке відповідає поставленим вимогам [4–6].

$$\max_{s \in S}(V(s)_k - V(s)_{k-1}) < j\,. \qquad (8)$$

Алгоритм 1. Ітерація за значеннями.

Value_Iteration (S,A,γ)
  for each s ∈ S
    $R_0(s)$=random()
  i=1
  for $\max(V(s)_i - V(s)_{i-1}) < j$
    for each s ∈ S
      for each a ∈ A
        $V(s,a)=R(s)+\gamma \max_a \sum_{s'} (P_s(s'|a)V(s')-C(s,a))$
      $R_i(s)=\max_{a \in A} V(s,a)$
      $\pi(s)=\text{argmax}_{a \in A} V(s,a)$
  i=i+1
return(π)
end

Алгоритм ітерації за стратегією. Основна ідея алгоритму полягає в тому, щоб поступово ітеративно покращувати початково довільно обрану стратегію π (див. алгоритм 2). Алгоритм можна поділити на два основні етапи: 1-ий – етап визначення вартості стратегії, на цьому етапі визначають цінність поточної стратегії шляхом розв'язання рівняння Белмана; 2-ий – етап покращення стратегії, на якому стратегія покращується до нової стратегії з вищою корисністю шляхом порівняння корисності стану *s* при виконанні дій згідно із стратегією $V(s,\pi(s))$ з корисністю станів при виконанні альтернативних дій *a* у цьому стані $V(s,a)$. При вищій ефективності нової дії *a* вона записується в стратегію для цього стану. Алгоритм зупиняється, коли немає альтернативних дій, які можуть покращити стратегію [4, 5].

Алгоритм 2. Ітерація за стратегією.

Policy_Iteration (S,A,γ)
  $\pi = 0$
  setect any $\pi' \neq 0$
  for $\pi \neq \pi'$
    $\pi = \pi'$
    for each s ∈ S
      $V(s,\pi(s))=R(s)+\gamma \sum_{s'} (P_s(s'|\pi(s))V(s')-C(s,\pi(s)))$
  for each s ∈ S
    $V(s,a)=R(s)+\gamma \sum_{s'} (P_s(s'|a)V(s')-C(s,a))$
    if ∃a ∈ A exist $V(s,\pi(s))<V(s,a)$
      then $\pi' \leftarrow$ a
      else $\pi'(s) \leftarrow \pi(s)$
return(π)
end

Метою наших досліджень було порівняти два основні алгоритми MDP планування, алгоритм ітерації за значеннями та алгоритм ітерації за стратегією, для створення стратегії на так званих універсальних графах, шляхом накладання моделі середовища. Поняттям універсальний граф ми визначаємо граф, у якому з кожного стану можна потрапити у будь-який інший стан, наклавши деяку модель середовища, ми отримуємо обмежене середовище, у якому початковий стан – це вершина графа, у яку немає жодного вхідного вузла, а кінцевий стан – вершина без вихідних вузлів.

Дії поєднують усі стани, але можливості переходу агента в той чи інший стан, виконавши конкретну дію, обмежуються імовірністю переходу $P_s(s' \mid a)$, для деяких станів вона дорівнюватиме нулю, тому можливостей таких переходів можна і не розглядати.

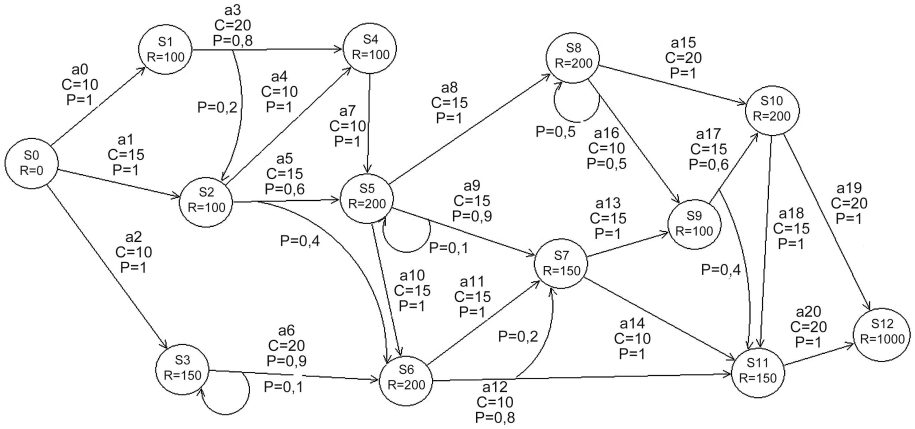

Рис. 1. Граф моделі середовища.

Дослідження, проведені на базі графа середовища (рис. 1). У цього графа є 13 вершин: вершина S_0 – початковий стан системи, у цьому стані найнижчий прибуток від перебування 0, вершина S12 – кінцевий стан з максимальним прибутком від потрапляння 1000. У проміжних станах прибуток від потрапляння знаходиться в межах від 0 до 200.

Досліди повторювали 1500 разів, після кожного досліду ми змінювали прибутки від перебування в станах $R(s)$, дії залишалися незмінними.

Для побудови стратегії за допомогою алгоритму ітерації за значеннями визначено коефіцієнт знецінення 0,6, а похибка точності 0,05 для алгоритму ітерації за стратегією коефіцієнт знецінення також 0,6.

Для алгоритму ітерації за значенням у 92% випадків було зафіксовано однакову стратегію. Зупинка алгоритму відбувалася після 8-ї ітерації, саме тоді значення максимальної різниці між корисностями станів з двох сусідніх ітерацій стало менше, ніж 0,05, зміна значень корисностей станів (рис. 2), зміна значень максимальної різниці корисностей (рис. 3). В результаті побудовано деяку стратегію (див. табл. 1) з найвищою корисністю.

Таблиця 1. Стратегія дій з найвищою корисністю

Стан	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Дія	a0	a3	a5	a6	a7	a8	a11	a13	a16	a17	a18	a20

Для побудови стратегії за допомогою алгоритму ітерації за стратегією на першому кроці алгоритму було обрано випадковим чином початкову стратегію (див. табл. 2), далі її удосконалено до оптимальної. Алгоритм у 78% випадків

виявив однакову стратегію, яка така, як в алгоритмі ітерації за значеннями (див. табл. 1), а у 72% випадків зупинявся після 4-ї ітерації, тоді не відбулося жодної зміни дій в стратегії, зміна значень корисностей станів (рис. 4).

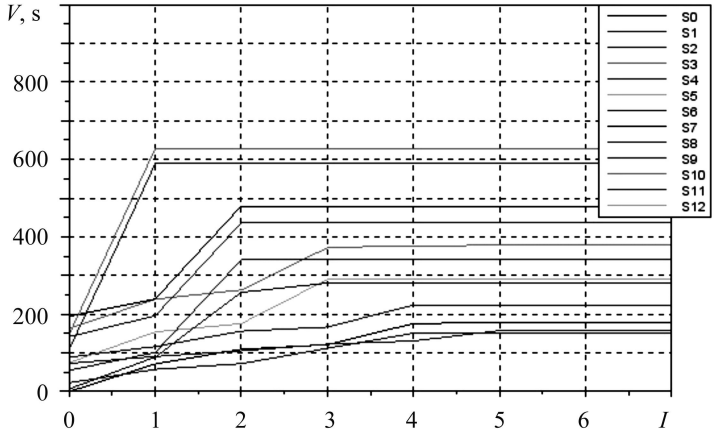

Рис. 2. Графіки зміни корисності станів ітерації за значеннями.

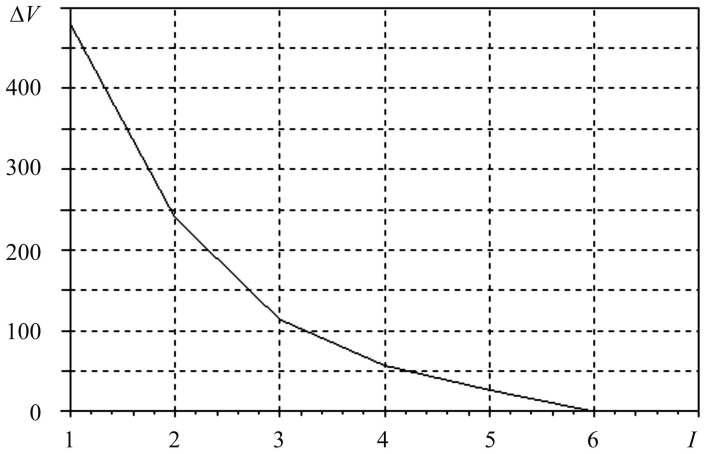

Рис. 3. Графіки зміни максимальної різниці корисностей.

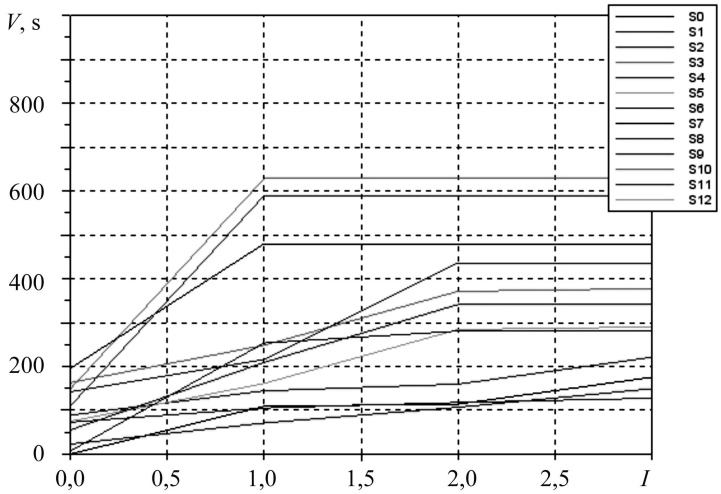

Рис. 4. Графіки зміни корисності станів ітерації за стратегією.

Таблиця 2. Початкова стратегія

Стан	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Дія	a0	a3	a4	a6	a7	a10	a12	a13	a15	a17	a19	a20

ВИСНОВКИ

У результаті проведених досліджень алгоритмів розрахунку оптимальної стратегії інтелектуального агента як MDP-моделі інформаційних потреб користувача системи інформаційного пошуку, а саме порівняння ефективності методу ітерацій за значеннями та методу ітерацій за стратегією, було виявлено:

– стратегії ідентичні у 78% дослідів;

– середня кількість ітерацій алгоритму, що реалізує метод ітерацій за значеннями – 8;

– середня кількість ітерацій алгоритму що реалізує метод ітерацій за стратегією – 4.

Усього було проведено 1500 дослідів, у кожному визначено різні початкові умови та параметри середовища.

Можна зробити висновок, що алгоритм що реалізує метод ітерацій, за значеннями забезпечує вищу точність при більших затратах обчислювальних ресурсів, тому його доцільно застосовувати для плануваннях у системах, де точність має вищий пріоритет і є можливість забезпечити потужний обчислювальний ресурс, щоб компенсувати часові затрати.

Алгоритм, що реалізує метод ітерацій за стратегією, є оптимальним для моделювання інформаційних потреб користувача через більшу швидкодію у зв'язку з необхідністю багаторазово повторювати розрахунок оптимальної стратегії для MDP-плану користувача для того, щоб оцінити релевантність кожного документа до інформаційних потреб користувача служби інформаційного пошуку. Нижча точність не є критичним чинником у досягненні прийнятних результатів.

Отже, на основі проведеного аналізу алгоритм, що реалізує метод ітерації за стратегією, можна рекомендувати як універсальний для пошуку інформації в мережі та проводити роботи з його модернізації.

1. *Методи* і засоби побудови онтології інтелектуального агента в галузі матеріалознавства / В. В. Литвин, Д. Г. Досин, Р. Р. Даревич, Н. В. Шкутяк, А. С. Мельник // Відбір і обробка інформації. – 2011. – Вип. 34 (110). – С. 129–134.
2. *Досин Д. Г., Ковалевич В. М.* Архітектура інтелектуальної системи інформаційного пошуку в мережі Інтернет // Искусственный интеллект. – 2012. – Вип. 3. – С. 241–252.
3. *Рассел С., Норвиг П.* Искусственный интеллект. – М.; СПб.; К.: Вильямс, 2006. – 1408 с.
4. *Ghallab M., Nau D., Traverso P.* Automated Planning Theory & Practice. – San Francisco: Morgan Knaufman, 2004. – 635 p.
5. *Martin L. Puterman* Markov decision processes discrete stochastic dynamic programming // John Wiley and Sons, Inc. – New Jersey: Hoboken, 2005. – 649 p.
6. *Spaan M. T. J. and Vlassis N.* Perseus: Randomized point-based value iteration for POMDPs // JAIR. – 2005. – **24**. – P. 195–220.

Фізико-механічний інститут ім. Г. В. Карпенка НАН України, Львів

Одержано 13.11.2012