ПРОГРЕСИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

PROGRESSIVE INFORMATION TECHNOLOGIES

ПРОГРЕССИВНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.[852+94]; 519.837.3

МОДЕЛЮВАННЯ ІГРОВОЇ ЗАДАЧІ ПРИЗНАЧЕННЯ ПЕРСОНАЛУ ДЛЯ ВИКОНАННЯ ІТ-ПРОЕКТІВ НА ОСНОВІ ОНТОЛОГІЙ

Кравець П. О. – канд. техн. наук, доцент, доцент кафедри «Інформаційні системи та мережі», Національний університет «Львівська політехніка», Львів, Україна.

Литвин В. В. – д-р техн. наук, професор, завідувач кафедри «Інформаційні системи та мережі», Національний університет «Львівська політехніка», Львів, Україна.

Висоцька В. А. – канд. техн. наук, доцент, доцент кафедри «Інформаційні системи та мережі», Національний університет «Львівська політехніка», Львів, Україна.

АНОТАЦІЯ

Актуальність. У цій статті описано розв'язування ігрової задачі призначення персоналу для роботи над проектами на основі онтологічного підходу. Суть задачі полягає у такому. Існує потреба у створенні команд для виконання декількох проектів. Кожен проект задається набором необхідних онтологічних знань. Для виконання проектів менеджери залучають кваліфікованих спеціалістів (агентів), здібності яких також задаються наборами онтологій. Склад команд повинен бути таким, щоб об'єднані онтології їх агентів покривали множини онтологій відповідних проектів. Кожен агент з певними імовірностями може прийняти послідовну участь у виконанні декількох проектів. Одночасна робота агента над різними проектами не допускається. Необхідно визначити порядок виконання проектів і відповідний йому порядок призначення персоналу.

Метою дослідження є розроблення математичної моделі стохастичної гри, рекурентних марковських методів для її розв'язування, алгоритмічного та програмного забезпечення, проведення комп'ютерного експерименту, аналіз результатів та виробленням рекомендацій щодо їх практичного застосування.

Метод. Для планування виконання проектів використано стохастичний ігровий алгоритм розфарбовування неорієнтованого випадкового графа. Для цього кількість вершин графа прийнята рівною кількості проектів. Ребрами з'єднано ті вершини графа проектів, для виконання яких залучено одного і того ж агента. З урахуванням відновлювальних відмов агентів зв'язки між вершинами графа динамічно змінюються. Необхідно досягнути правильного розфарбування випадкового графа. Тоді проекти з однаково зафарбованими вершинами графа можуть бути виконані паралельно, а проекти з різними кольорами вершин – послідовно.

Результати. У статті побудовано математичну модель стохастичної гри та самонавчальний марковський метод для її розв'язування. Кожна вершина графа контролюється гравцем. Чистими стратегіями гравця є елементи палітри кольорів. Після вибору кольору власної вершини кожен гравець обчислює поточний програш як відносну кількість однакових кольорів у локальній множині сусідніх гравців. Мета гравців полягає у мінімізації функцій середніх програшів. Марковський рекурентний метод забезпечує адаптивний вибір кольорів вершин випадкового графа на основі динамічних векторів змішаних стратегій, значення яких залежать від поточних програшів гравців. Результатом стохастичної гри є асимптотично правильно розфарбований випадковий граф, коли кожному ребру початкового детермінованого графа будуть відповідати у середньому різні кольори вершин.

Висновки. Проведено комп'ютерний експеримент, який підтвердив збіжність стохастичної гри для задачі розфарбовування випадкового графа. Це дало можливість визначити порядок призначення персоналу для виконання проектів.

КЛЮЧОВІ СЛОВА: проект, агент, онтологія, призначення персоналу, розфарбовування випадкового графа, стохастична гра, марковський рекурентний метод, адаптація, самонавчання.

АБРЕВІАТУРА

ІС – інформаційна система;

ІТ – інформаційна технологія.

DOI 10.15588/1607-3274-2022-1-14

НОМЕНКЛАТУРА

O – онтологія;

Ω – бібліотека онтологій;

Π – множина IT-проектів;

L – кількість IT-проектів або кількість вершин графа (кількість команд агентів, кількість гравців);

Π_i – набори онтологічних знань або компетенцій, необхідних для виконання i-го проекту;

Γ – скінченна множина команд агентів;

K – загальна кількість агентів, що можуть бути залучені для виконання усіх проектів;

A_k – агент з номером k або набір онтологій, що визначають здібності агента;

$q_{i,k}$ – імовірність участі i-го агента у виконанні k-го проекту;

V – скінченна множина вершин;

E – множина ребер;

D_i – множина номерів сусідніх вершин для i-ї вершини графа;

t – момент часу;

N – кількість чистих стратегій гравців або кількість фарб;

X_i – вектор чистих стратегій i-го гравця;

$x_{i,j}$ – колір j-ї фарби, вибраної i-м гравцем;

p_i – змішана стратегія i-го гравця;

$p_{i,j}$ – умовна імовірність вибору i-м гравцем j-го кольору;

γ – параметр кроку навчання;

α – коефіцієнт порядку кроку навчання;

ε – параметр ε-симплекса;

β – коефіцієнт порядку розширення ε-симплекса;

λ – ваговий коефіцієнт;

χ – індикаторна функція події;

ξ_i – значення поточного програшу;

Ξ_i – поточні значення функцій середніх програшів (або втрат);

$t_{\max}$ – максимальна кількість кроків методу.

ВСТУП

У сучасному інформаційному просторі з розвиненими комп'ютерними мережами та засобами телекомунікації задача формування віртуальних професійних команд є важливою для організації виконання проектів, особливо в умовах дистанційної роботи [1]. Якісний підбір та належна організація командної роботи кваліфікованого персоналу є запорукою оперативного та успішного виконання проекту [2].

Проблема залучення персоналу для виконання проектів є подібною до класичної задачі про призначення засобів (устаткування або людей) для виконання робіт. Однак, у формулювання цієї задачі використані суттєві обмеження – для виконання кожної роботи можна призначити тільки один засіб, усі засоби взаємозамінні та абсолютно надійні. У практиці формування команд виконавців проектів існує протилежний підхід. Для виконання проекту потрібно декілька різних спеціалістів, персонал може бути частково взаємозамінний і характеризується імовірністю відмов. Відмови можуть бути обумовлені недостатнім досвідом, нескоординованою роботою, зміною стану здоров'я виконавців, іншими непередбачуваними чинниками і впливають на якість і терміни виконання проекту.

Враховуючи вказані розбіжності, для розв'язування задачі про призначення персоналу для виконання проектів можна застосувати метод розфарбовування графа. Граф будується так, що вершини позначають проекти, а ребра зв'язують тільки ті вершини проектів, для виконання яких залучено однакових виконавців. Вершини на протилежних сторонах ребер повинні бути зафарбовані у різні кольори. Тоді проекти, вершини яких мають однаковий колір можна виконати паралельно, а ті, що мають різні кольори – послідовно у часі.

Відновлювальні відмови виконавців проектів формують динамічну структуру зв'язків між вершинами графа. Тому, замість детермінованого, необхідно розглядати випадковий граф проектів. Задача розфарбовування випадкового графа належить до класу NP-складних (nondeterministic polynomial time) [3, 4]. Для її розв'язування необхідно використати або розробити самонавчальні чи адаптивні методи, які можуть забезпечити наближене до оптимального розфарбовування за допустимий поліноміальний час.

Враховуючи, що граф є структурною моделлю розподіленої системи, і те, що задача розфарбовування випадкового графа має аспекти локальної конкуренції та глобальної узгодженості цілей, для її розв'язування використаємо адаптивний стохастичний ігровий метод. Адаптивні методи можуть прямо або опосередковано усереднювати контрольовані випадкові процеси, селективно формувати розв'язки, які забезпечують оптимізацію середніх значень випадкових величин або їх стохастичних моментів.

Метою дослідження є розв'язування задачі призначення персоналу для виконання IT-проектів на основі моделі стохастичної гри розфарбовування випадкового графа. Досягнення мети забезпечується розробленням математичної моделі стохастичної гри, рекурентних марковських методів для її розв'язування, алгоритмічного та програмного забезпечення, проведенням комп'ютерного експерименту, аналізом результатів та виробленням рекомендацій щодо їх практичного застосування.

Об'єктом дослідження є процес призначення кваліфікованого персоналу для роботи над проектами на основі онтологічного підходу в умовах невизначеності.

DOI 10.15588/1607-3274-2022-1-14

Предметом дослідження є стохастичний ігровий метод розфарбовування графів для розв'язування задачі безконфліктного розподілення персоналу між проектами.

1 ПОСТАНОВКА ЗАДАЧІ

Нехай необхідно організувати виконання L проектів $\Pi = \{\Pi_1, \Pi_2, ..., \Pi_L\}$, кожен з яких задається необхідними для його виконання компетенціями $\Pi_i = \{O_1, O_2, ..., O_r\}$ у вигляді набору онтологій. Кожна онтологія формально описує знання у певній проблемній галузі, які необхідні для виконання проекту [5].

Для онтологічної підтримки проектів необхідно залучити кваліфікованих спеціалістів у потрібних галузях знань. Використовуючи термінологію мультиагентних систем, інформаційну модель виконавців проектів робіт далі будемо називати агентами. Нехай ринок праці пропонує множину агентів $A = \{A_1, A_2, ..., A_K\}$, $K \geq L$, які можуть бути залучені для виконання проектів. Кожен агент A_k визначається набором онтологій $A_k = \{O_{k,1}, O_{k,2}, ..., O_{k,s}\}$ $k = 1..K$, які описують його здібності в одній або декількох галузях знань. Онтології агентів можуть мати непорожній перетин $A_i \cap A_j \neq \varnothing$, тобто агенти частково володіють однаковими здібностями. Припускається, що сукупні онтологічні знання агентів є достатніми для виконання усіх проектів.

Для виконання проектів необхідно сформувати множину команд $\Gamma = \{\Gamma_1, \Gamma_2, ..., \Gamma_L\}$, кожна з яких є організованою групою агентів $\Gamma_i = \{A_{i,1}, A_{i,2}, ..., A_{i,g}\}$, $i = 1..L$, де $\bigcup\limits_{i=1..L} \Gamma_i = A$.

Необхідною умовою успішного виконання проекту є його повна онтологічна підтримка командою агентів. Здібності команди агентів повинні покривати компетенції, необхідні для виконання проекту: $\bigcup\limits_{\forall A_k^i \in \Gamma_i} A_k^i \supseteq \Pi_i$, $i = 1..L$.

Ігровий підхід до покриття потрібних для виконання проектів онтологій наявними онтологіями агентів розглянуто у [6].

Підбір команд агентів виконують менеджери проектів незалежно один від одного. Тоді в умовах обмеженої кількості кваліфікованих спеціалістів деякі з агентів можуть бути залучені для виконання різних проектів, тобто $\Gamma_i \cap \Gamma_j \neq \varnothing$.

Припустимо, що виконання кожного проекту не може бути перервано і виконавці не можуть переходити з одного проекту на інший до моменту його повного завершення. У зв'язку з цим виникає задача визначення порядку виконання проектів у часі. Подібно до формулювання задачі розподілу устаткування для виконання певних робіт будемо вважати, що час виконання кожного проекту є однаковим. Тоді задачу планування послідовності виконання проектів можна звести до задачі розфарбовування неорієнтованого графа $G = (\Gamma, E)$. У такому графі вершини розмічені множинами команд агентів Γ_i, $i = 1..L$, залучених до виконання відповідного проекту, а ребра зв'язують ті вершини графа проектів, що містять однакових агентів:

$$E_i = \left\{ e_{i,j} \mid \chi\left(\Gamma_i \cap \Gamma_j \neq \varnothing\right) \right\},$$

де $\chi() \in \{0,1\}$ – індикаторна функція події. Значення $\chi() = 1$ вказує на наявність відповідного ребра $e_{i,j}$ у графі. Якщо потужність перетину більша від 1, тобто $|\Gamma_i \cap \Gamma_j| > 1$, то розглядається мультиграф.

Тепер необхідно розфарбувати вершини графа проектів так, щоб вершини з'єднані ребрами $e_{i,j}$, $i, j = 1..L$ мали різні кольори. Тоді за таким розфарбуванням можна визначити порядок виконання проектів. Проекти, вершини яких мають однакові кольори, можуть бути виконані одночасно, а з різними кольорами – послідовно.

Однак у момент формування команд існує певна невизначеність участі агентів у виконанні того або іншого проекту. Цю невизначеність задамо імовірностями $q_{i,k}$ домовленостей між менеджером i-го проекту та k-м агентом. Ці імовірності охоплюють ряд факторів ризику невиконання проекту, наприклад, непокриття проекту необхідними онтологіями у зв'язку з недостатніми здібностями агентів, неможливість виконання проекту за станом здоров'я виконавців, непередбачувані зовнішні чинники тощо. Значення $q_{i,k}$ можуть бути визначені зі сторони менеджерів проектів, як ступінь їх впевненості у тому, що відповідний агент прийме участь у виконанні проекту, або зі сторони агента, як його схильність до виконання того або іншого проекту, або комплексно – з обох договірних сторін. Спрощено можна вважати, що $q_{i,k} = q_k$ – це імовірність участі k-го агента у виконанні будь-якого проекту. Тоді $1 - q_k$ – це імовірність відмови k-го агента. Надалі будемо вважати, що відмови агентів є відновлювальними.

Враховуючи таку стохастичну природу агентів, замість детермінованого графа необхідно розглядати випадковий граф зв'язків між проектами. Відновлювальна відмова одного із агентів, завдяки якому у графі є ребро між вершинами-проектами, призводить до тимчасової втрати цього ребра. Отже, замість заданого детермінованого графа у певні проміжки часу будуть спостерігатися його випадкові реалізації у вигляді усіх можливих підграфів.

Для розфарбовування випадкового графа детерміновані методи не можуть бути застосовані,

DOI 10.15588/1607-3274-2022-1-14

оскільки на кожному кроці гри змінюються реалізації графа і значення поточних програшів є випадковими величинами. Для цього необхідно використати адаптивні стохастичні методи, які зможуть пристосовуватись до випадкової зміни структури графа. Для безконфліктного планування призначення виконавців проектів нами пропонується метод на основі багатокрокової стохастичної гри [7].

2 ОГЛЯД ЛІТЕРАТУРИ

Задача про призначення персоналу для виконання проектів робіт формулюється подібно до задач розподілу устаткування, ресурсів, завдань, програм, складання розкладів, класифікації та інших аналогічних задач [8–13].

Класичне формулювання узагальненої задачі про призначення полягає у наступному. Нехай є L видів робіт або завдань, для виконання яких можна використати $K \ge L$ засобів (машин, пристроїв, роботів, програмних агентів, людей). Відомі витрати $c_{i,j}$ використання j-го засобу ($j = 1..K$) для виконання i-ї ($i = 1..L$) роботи. Кожен засіб можна використати тільки для одного виду роботи. Вважається, що засоби є взаємно замінними.

Необхідно знайти такий план $u = \left[u_{i,j} \in \{0,1\} \mid i = 1..L, j = 1..K \right]$, де $u_{i,j} \in \{0,1\}$, виконання робіт (розподілити засоби між роботами), щоб сумарні витрати $f(u)$ були мінімальними. Змінна $u_{i,j} = 1$, якщо засіб j виконує роботу i, та $u_{i,j} = 0$ – в іншому випадку.

Математична модель сформульованої задачі має вигляд:

1) цільова функція, яка визначає загальні витрати від використання засобів для виконання робіт:

$$f(u) = \sum_{i=1}^{L} \sum_{j=1}^{K} c_{i,j} u_{i,j} \to \min_{x} ;$$

2) система обмежень:

$\sum_{j=1}^{K} u_{i,j} = 1$, $i = 1..L$ – для виконання i-ї роботи можна призначити тільки один j-й засіб;

$\sum_{i=1}^{L} u_{i,j} \le 1$, $j = 1..K$ – деякі засоби можуть бути незадіяними, якщо $K > L$.

Формулювання задачі може змінюватися залежно від інтерпретації параметрів задачі, як правило, розглядається вартість або час виконання проектів.

У частковому випадку, якщо $K = L$, то $\sum_{i=1}^{L} u_{i,j} = 1$, $j = 1..K$. Така задача називається основною або лінійною задачею про призначення. Вона є однією із фундаментальних задач комбінаторної оптимізації.

Для розв’язування задач про призначення можна використати методи лінійного програмування (угорський алгоритм, симплексний метод), цілочисельного програмування (метод гілок та меж, метод Гоморі) [14], мурашиної колонії [15], генетичного алгоритму [16], розфарбовування графів [17], штучних нейронних мереж [18], евристичні методи [19].

У виконаному вище формулюванні практичне використання задачі про призначення є обмеженим. На практиці для виконання однієї роботи як правило потрібно одночасно застосувати декілька різнотипних засобів, тобто $1 < \sum_{j=1}^{K} u_{i,j} \le K$. Якщо ті ж засоби будуть потрібні і для інших робіт, то $0 \le \sum_{i=1}^{L} u_{i,j} \le L$ для $K \ge L$. У такому разі матриця $u = \left[u_{i,j} \right]$ вважається заданою і потрібно перейти до формування розкладу використання однакових засобів для виконання різних робіт. Розв’язком такої задачі є визначення порядку виконання робіт у часі, щоб один і той же засіб виконання робіт можна було використати тільки послідовно. Цю задачу можна звести до задачі розфарбовування вершин графа [20, 21].

Побудуємо неорієнтований граф, вершини якого позначають робóти (завдання), а ребра – засоби для їх виконання. Ребра з’єднують тільки ті робóти, для виконання яких потрібен однаковий засіб. Для кожного засобу $j = 1..K$ побудуємо кліку графа, з’єднавши між собою вершини, для яких $u_{i,j} = 1$, $i = 1..L$.

У результаті отримаємо неорієнтований граф $G = (V, E)$. Якщо для двох або більше робіт потрібно декілька однакових засобів, то отримаємо мультиграф.

Розфарбовуванням називають відображення $g : V \to X$, де $X = (x_1, x_2, ..., x_N)$ ($N \le L$) – палітра кольорів. Розфарбовування є правильним, якщо $g(k) \ne g(l)$ для кожного наявного у графі ребра, тобто $\forall e_{k,l} \in E$, $k, l \in V$.

Із розфарбованого графу можна отримати план виконання робіт у часі. Роботи, вершини яких зафарбовано однаковим кольором, можуть бути виконані одночасно, а зафарбовані різними кольорами – послідовно.

Розфарбовування графа великого порядку $|V|$ обмеженою кількістю N фарб вважають NP-складною задачею, яка не може бути розв’язана повним перебором варіантів за прийнятний поліноміальний час.

Для пришвидшення пошуку можна використати метод бектрекінгу (пошук з поверненням), який виключає з розгляду значну кількість варіантів однією перевіркою, будуючи дерево рішень та здійснюючи

DOI 10.15588/1607-3274-2022-1-14

його обхід у глибину [22]. Хоча метод класифікується як метаевристика, але він гарантовано знаходить всі розв'язки скінченної дискретної задачі за обмежений час.

Нехай вершини графа позначено літерами латинського алфавіту, а кольори – послідовними цілими числами від 1 до n. Спочатку вершина $v(a)$ фарбується в колір 1. Якщо вершина $v(b)$ не є суміжною з $v(a)$, то вона фарбується у колір 1, інакше – у колір 2. Далі розглядається вершина $v(c)$, для якої здійснюється спроба фарбування у найменший колір 1. Якщо це не можливо, то вибирається наступний допустимий колір. Після досягнення вершини, яку не можна розфарбувати жодним із n кольорів, відбувається повернення до останньої розфарбованої вершини, її колір замінюється на наступний можливий колір із впорядкованого списку кольорів і здійснюється спроба її розфарбування. Якщо і це неможливо, то відбувається повернення ще до попередньої вершини. Процес продовжується аналогічно до тих пір, поки не буде досягнуте правильне розфарбування графа, або з'ясується, що граф не можна розфарбувати в n кольорів.

Математична модель задачі розфарбовування графа може бути сформульована як задача 0 – 1 цілочисельного програмування [23]:

1) цільова функція, яка вказує на те, що граф потрібно зафарбувати мінімальною кількістю кольорів:

$$z = \sum_{j=1}^{Q} \sum_{i=1}^{L} w_j c_{i,j} \to \min_{c} .$$

2) система обмежень:

$\sum_{j=1}^{Q} c_{i,j} = 1 \quad \forall i = 1..L$ – кожна вершина може бути зафарбована тільки одним кольором;

$M \cdot (1 - c_{i,j}) - \sum_{k=1}^{L} a_{i,k} c_{k,j} L \ge 0 \quad \forall i = 1..L$, $\forall j = 1..Q$ – кожна пара суміжних вершин не має однакового кольору.

Тут $\left[a_{i,j} \,|\, a_{i,j} \in \{0,1\}, i = 1..L, j = 1..L \right]$ – матриця суміжностей вершин графа; $\left[c_{i,j} \,|\, c_{i,j} \in \{0,1\}, i = 1..L, j = 1..Q \right]$ – матриця зафарбованих вершин графа, де $c_{i,j} = 1$, якщо вершина v_i має колір j; $\left(w_j \,|\, w_{j+1} > L \cdot w_j, w_1 = 1, j = 1..Q \right)$ – вектор додатних ваг кольорів; $M > L$.

Для великих порядків графа визначення оптимального розв'язку задачі методами цілочисельного програмування може не дати задовільного результату у зв'язку з великою розмірністю матриць. Тому рекомендується використовувати інші методи з поліноміальним часом розв'язування задачі, наприклад, основані на оптимізації перебору варіантів, «м'яких» обчисленнях або евристичних здогадках.

Огляд методів розфарбовування графів можна знайти у роботах [24, 25]. Для розв'язування різноманітних практичних задач використовуються такі методи розфарбовування графів:

1) динамічного програмування [26];
2) жадібний [27];
3) генетичний [28];
4) штучних нейронних мереж [29];
5) ройовий [30];
6) мурашиний [31];
7) багатоагентний [32];
8) ігровий [33, 34].

Задача розфарбовування значно ускладнюється для часових випадкових графів $G(t) = (V(t), E(t))$, ребра яких розмічені імовірностями їх належності графу [35, 36]. Тоді у кожен момент часу $t = 1, 2, ...$ граф проявляється однією із своїх можливих реалізацій.

Дослідження випадкових графів в основному пов'язанні в отриманні імовірнісних асимптотичних оцінок його параметрів, у тому числі хроматичного числа, а інформація про ефективні методи розфарбовування випадкового графа недостатньо висвітлена у наукових працях.

Детерміновані методи розфарбовування випадкового графа є непродуктивними. Необхідно використати або розробити багатокрокові уточнюючі методи з елементами самонавчання, побудовані на основі «м'яких» обчислень або різноманітних евристик. Робота таких методів повинна бути спрямована на покращення хроматичної картини для досягнення правильного розмалювання графа в асимптотиці часу. Для розфарбовування випадкового графа можуть бути використані модифікації методів 1 – 8. Особливу увагу слід приділити розробленню стохастичних варіантів реалізації цих методів.

Для розв'язування задачі розфарбовування випадкового графа, побудованого на основі онтологічної підтримки проектів, нами пропонується застосування стохастичного ігрового методу, який має властивості самонавчання та адаптації до невизначеностей.

3 МАТЕРІАЛИ І МЕТОДИ

З кожною вершиною (проектом) графа пов'яжемо гравця, чисті стратегії якого визначають палітру пронумерованих кольорів $X_i = \{x_{i,1}, x_{i,2}, ..., x_{i,N}\}$, де $x_{i,j}$ – номер кольору; N – кількість елементів палітри кольорів, яка обмежується значенням $N = L$, що є необхідним для розфарбовування повнозв'язного графа.

Вибір чистих стратегій $x_i(t) \in X_i$ здійснюється гравцями випадково і незалежно у моменти часу $t = 1, 2, ...$. У зв'язку з відмовами гравців, на кожному кроці гри визначається випадкова реалізація графа $G(t) = (\Gamma(t), E(t)) \subseteq G$, у якому змінюється лише

DOI 10.15588/1607-3274-2022-1-14

розмітка вершин $\Gamma(t)=\{\Gamma_i(t)\,|\,i=1..L\}$ множинами агентів

$$\Gamma_i(t)=\{A_{i,1},A_{i,2},...,A_{i,g}\}\setminus\{A_{i,k}(t)\,|\,k=1..l,l\le g\}$$

та відповідні зв'язки $E(t)$ між вершинами. Тут $\{A_{i,k}(t)\}$ – агенти, що відмовили у момент часу t. Кількість вершин L і відповідних до них проектів Π_i залишаються незмінними. Гравці не володіють інформацією про поточну реалізацію графа в цілому. Кожному гравцю відомий тільки їх локальний підграф – множина сусідніх вершин, що приєднані ребрами до контрольованої гравцем вершини графа.

Нехай $K_i^{loc}(t)=|E_i(t)|$ – кількість ребер i-ї вершини випадкового графа у момент часу t. Тоді i-й гравець приймає участь у грі, якщо йому відповідає неізольована вершина, тобто $K_i^{loc}(t)\ge 1$.

Після завершення вибору чистих стратегій усіма гравцями кожен з них обчислює значення поточного програшу, як середню кількість однакових кольорів сусідніх вершин графа:

$$\xi_i(t)=\left(K_i^{loc}(t)\right)^{-1}\sum_{j\in D_i(t)}\chi\left(x_i(t)=x_j(t)\right), \tag{1}$$

де $\chi()\in\{0,1\}$ – індикаторна функція події; $D_i(t)=\{\underset{j}{index}(e_{i,j}(t))\}$ – множина номерів сусідніх вершин для i-ї вершини випадкового графа. Сусідніми до вершини i є вершини випадкового графа, що мають з нею безпосередній зв'язок у момент часу t. Очевидно, що $|D_i(t)|=K_i^{loc}(t)$.

Гравці оцінюють свої дії у ході гри за допомогою поточних значень функцій середніх програшів (або втрат):

$$\Xi_i(t)=t^{-1}\sum_{\tau=1}^{n}\xi_i(\tau),\ i=1..L. \tag{2}$$

Хід стохастичної гри в цілому можна контролювати за допомогою системної функції середніх програшів:

Стратегія поведінки кожного гравця повинна бути спрямована на мінімізацію власних функцій середніх програшів (на мінімізацію співпадань кольорів сусідніх вершин графа):

$$\overline{\lim_{t\to\infty}}\,\Xi_i(t)\to\min_{\{x_i(t)\}},\ i=1..L. \tag{4}$$

Розв'язки задачі багатокритеріальної оптимізації (4) необхідно шукати у множинах точок колективної оптимальності, наприклад, Слейтера, Неша, Парето або інших [37]. Найчастіше у задачах без обміну поточною інформацією про стратегії, стани та програші гравців (або з мінімально потрібним обміном) використовується критерій рівноваги за Нешем, суть якого полягає у наступному. У точці рівноваги за Нешем кожному гравцю не вигідно змінювати свою стратегію, якщо усі інші гравці притримуються точки рівноваги:

$$\overline{\lim_{t\to\infty}}\left[\Xi_i(t,\{x^{D_i}(t)\})-\Xi_i(t,\{\tilde{x}^{D_i}(t)\})\right]\le 0,\ i=1..L. \tag{5}$$

У (5) використано такі позначення: D_i – локальна множина сусідніх гравців (або вершин графа) для i-го гравця (або вершини); $X^{D_i}=\underset{j\in D_i}{\otimes}X_j$ – простір комбінованих чистих стратегій гравців з локальної множини D_i; $X^{D_i}\subseteq X$; $X=\underset{i=1..L}{\otimes}X_i$ – простір комбінованих чистих стратегій усієї множини гравців; $\otimes$ – операція декартового добутку; $x^{D_i}\in X^{D_i}$; $\tilde{x}_i$ – відхилена від точки рівноваги стратегія i-го гравця; $\tilde{x}^{D_i}=x^{D_i}\setminus x_i+\tilde{x}_i\in X^{D_i}$ – комбінована стратегія сусідніх гравців з підмножини D_i після заміни стратегії i-го гравця; $x_i,\tilde{x}_i\in X_i$.

Отже, спостерігаючи випадкові поточні програші (1), гравці повинні навчитися вибирати фарби $x_i(t)\in X_i$ із палітри кольорів X^i так, щоб сформована послідовність варіантів $\{x_i(t)\}$ забезпечила виконання мети (4) в асимптотиці часу $t\to\infty$. На практиці кількість кроків стохастичної гри обмежується деяким максимальним значенням або досягненням правильного розфарбовуванням графа.

Для генерування послідовностей стратегій $\{x_i(t)\,|\,i=1..L\}$, $t=1,2,...$, які забезпечують виконання критеріїв (4), побудуємо імовірнісний механізм на основі змішаних стратегій гравців $\{p_i(t)\,|\,i=1..L\}$. Змішана стратегія $p_i(t)=\left(p_{i,1}(t),p_{i,2}(t),...,p_{i,N}(t)\right)$ складається з умовних імовірностей вибору чистих стратегій:

$$p_{i,j}(t)=P\left\{x_i(t)=x_{i,j}\,\middle|\,u_i(\tau),\xi_i(\tau),\tau=1,2,...,t-1\right\},$$
$$j=1..N,$$

де $\{x_i(\tau)\}$ – передісторія чистих стратегій, вибраних гравцем з номером i; $\{\xi_i(\tau)\}$ – передісторія отриманих за це програшів.

Для формування послідовностей з потрібними властивостями змішані стратегії на кожному кроці гри змінюються за рекурентним методом [38]:

$$p_i(t+1)=\pi^N_{\varepsilon(t+1)}\left\{p_i(t)-\gamma(t)R(p_i(t),x_i(t),\xi_i(t))\right\}, \tag{6}$$

DOI 10.15588/1607-3274-2022-1-14

де $\pi^N_{\varepsilon_{t+1}}$ – проектор на одиничний ε-симплекс $S^N_\varepsilon \subseteq S^N \subset R^N$ (тут верхній індекс не є показником степеня, а вказує на кількість вимірів простору дійсних чисел); $\gamma(t)$ – монотонно спадна послідовність невід'ємних величин, яка регулює величину кроку методу; R – крок методу; $\varepsilon(t)$ – монотонно спадна послідовність невід'ємних величин, яка регулює швидкість розширення ε-симплексу.

Координати точок одиничного симплексу є нормовані так, що їх сума дорівнює 1:

$$S^N = \left\{ p \left| \sum_{j=1}^{N} p_j = 1;\ p_j \ge 0 \ \ (j = 1..N) \right. \right\}.$$

Одиничний ε-симплекс є компактною підмножиною одиничного симплексу:

$$S^N_\varepsilon = \left\{ p_i \mid p_i \in S^N;\ p_{i,j} \ge \varepsilon \ \ (j = 1..N) \right\},$$
$$\varepsilon \in (0,1/N),\ p_i(t) \in S^N_\varepsilon .$$

Рекурентний метод (6) потрібно побудувати так, щоб при виборі стратегії $x_{i,j}(t)$ відповідна імовірність $p_{i,j}(t)$ зменшилася пропорційно величині поточного програшу $\xi_i(t)$. Решта елементів змішаної стратегії не змінюється, або зростає пропорційно $\xi_i(t)$. Тобто метод повинен підсилювати імовірності вибору більш вдалих стратегій, корисних для виконання критеріїв мінімізації середніх програшів гравців (4). Метод з такими властивостями називається адаптивним або самонавчальним.

Після обчислення нових значень векторів змішаних стратегій відбувається їх проектування на ε-симплекс S^N_ε. Оператор проектування $\pi^N_{\varepsilon_{t+1}}$ задовольняє умови: $\left\| p_i - \pi^N_\varepsilon\{q_i\} \right\| \le \left\| p_i - q_i \right\|$; $\pi^N_\varepsilon\{q_i\} \in S^N_\varepsilon$ $i = 1..L,\ \forall p_i \in S^N_\varepsilon, \forall q_i \in R^N$.

Проектування на розширюваний ε-симплекс забезпечує виконання умови $p_{i,j}(t) \ge \varepsilon(t),\ j = 1..N$, яка необхідна для повноти статистичної інформації щодо вибору чистих стратегій.

Побудову рекурентних методів виду (6) виконаємо методом стохастичної апроксимації [39, 40]. Припустимо, що математичні сподівання випадкових величин $M\{\xi_i(x,t)\} = v_i(x)$ відомі для всіх комбінованих стратегій $x \in X = \underset{i=1..L}{\otimes} X_i$.

Визначимо полілінійну функцію середніх програшів матричної гри:

$$V_i(p^{D_i}) = \sum_{x^{D_i} \in X^{D_i}} v_i(x^{D_i}) \prod_{j \in D_i; x_j \in x^{D_i}} p_j(x_j), \qquad (7)$$

де D_i – локальна множина сусідніх гравців; $X^{D_i} = \underset{j \in D_i}{\otimes} X_j$ – простір комбінованих чистих стратегій гравців з локальної множини D_i; $x^{D_i} \in X^{D_i}$ – одна із комбінованих локальних стратегій гравців; $p^{D_i} \in S^{D_i} = \prod_{j \in D_i} S^N_j$; $p_i \in S^N$.

Метою матричної гри є мінімізація функцій середніх програшів (7):

$$V_i(p^{D_i}) \to \min_{p_i},\ i = 1..L. \qquad (8)$$

Розв'язок матричної гри у точці рівноваги за Нешем у змішаних стратегіях визначається такою умовою:

$$V_i(p_*^{D_i}) - V_i(p_*^{D_i \setminus \{i\}}, p_i) \le 0,\ i = 1..L,$$

де $V_i(p_*^{D_i})$ – функція середніх програшів, визначена у точці Неша $p_*^{D_i} \in S^{D_i}$ на локальному симплексі $S^{D_i} = \prod_{j \in D_i} S^N$ комбінованих змішаних стратегій гравців з множини D_i сусідніх гравців; $V_i(p_*^{D_i}, p_i)$ – функція середніх програшів, визначена на симплексі S^{D_i} для будь-якого відхилення змішаної стратегії i-го гравця від точки Неша.

Згідно із теорією стохастичної апроксимації [38–40], для мінімізації системи функцій (8) визначимо вектор руху R рекурентного методу (6) так, щоб його математичне сподівання було градієнтом функції середніх програшів (7):

$$M\{R(p_i(t), x_i(t), \xi_i(t))\} = \nabla_{p_i} V_i(p^{D_i}).$$

Враховуючи, що

$$\nabla_{p_i} V_i = M\left\{ \frac{\xi_i(t)}{e^{\mathrm{T}}(x_i(t))\, p_i(t)} e(x_i(t)) \right| \left. p_i(t) = p_i \right\},$$

де $e(x_i(t))$ – одиничний вектор–індикатор вибору чистої стратегії $x_i(t) \in X_i$, $e^{\mathrm{T}}(x_i(t))$ – транспонований вектор, отримаємо градієнтний рекурентний метод розв'язування ігрової задачі:

DOI 10.15588/1607-3274-2022-1-14

$$p_i(t+1)=\pi^N_{\varepsilon(t+1)}\left\{p_i(t)-\gamma(t)\frac{\xi_i(t)e(x_i(t))}{e^{\mathrm{T}}(x_i(t))p_i(t)}\right\}. \qquad (9)$$

Інші рекурентні методи можна отримати з умови доповняльної нежорсткості [41, 42], яка виконується для точок рівноваги за Нешем у повністю змішаних стратегіях:

$$\nabla_{p_i}V_i=V_ie_N\ ,\quad i=1..L\ , \qquad (10)$$

де e_N – вектор, що складається з N одиниць. Умова доповняльної нежорсткості описує незалежність функцій середніх втрат гравців від їх власних змішаних стратегій у точці Неша. Як би не змінювалася змішана стратегія $p_i(t)$ гравця з номером i ($i=1..L$) на одиничному симплексі, коли усі інші гравці притримуються своїх стратегій у точці Неша, значення функції виграшів V_i залишається постійним.

Оскільки

$$\nabla_{p_i}V_i-V_ie_N=M\left\{\xi_i(t)\left[\frac{e(x_i(t))}{e^{\mathrm{T}}(x_i(t))p_i(t)}-e_N\right]\middle|\,p_i(t)=p_i\right\},$$

то методом стохастичної апроксимації отримаємо рекурентний метод доповняльної нежорсткості:

$$p_i(t+1)=\pi^N_{\varepsilon(t+1)}\left\{p_i(t)-\gamma(t)\xi_i(t)\left[\frac{e(x_i(t))}{e^{\mathrm{T}}(x_i(t))p_i(t)}-e_N\right]\right\}. \qquad (11)$$

Для врахування розв'язків на межі одиничного симплексу виконаємо зважування векторної умови (10) елементами вектора p_i:

$$diag(p_i)[V_ie_N-\nabla V_i]=0\,, \qquad (12)$$

де $diag(p_i)$ – квадратна діагональна матриця порядку N, складена з елементів вектора p_i.

Враховуючи, що

$$diag(p_i)(V_ie_N-\nabla_{p_i}V_i)=M\left\{\xi_i(t)\left[p_i(t)-e(x_i(t))\right]\middle|\,p_i(t)=p_i\right\},$$

методом стохастичної апроксимації отримаємо рекурентний метод зваженої доповняльної нежорсткості (12):

$$p_i(t+1)=\pi^N_{\varepsilon(t+1)}\left\{p_i(t)-\gamma(t)\xi_i(t)\left[e(x_i(t))-p_i(t)\right]\right\}. \qquad (13)$$

Завдяки такій динамічній реорганізації змішаних стратегій на основі опрацювання поточних програшів, методи (9), (11), (13) забезпечують адаптивний вибір чистих стратегій у часі.

DOI 10.15588/1607-3274-2022-1-14

Параметри $\gamma(t)$ та $\varepsilon(t)$ є монотонно спадними послідовностями невід'ємних величин і використовуються для керування збіжністю рекурентних методів. Ці параметри можна обчислити так:

$$\gamma(t)=\gamma(0)t^{-\alpha}\ ,\ \varepsilon(t)=\varepsilon(0)t^{-\beta}\,, \qquad (14)$$

де $\gamma_0,\alpha,\beta>0$, $\varepsilon(0)\in(0,N^{-1})$. Збіжність змішаних стратегій $p_i(t)$ $i=1..L$ до оптимальних значень з імовірністю 1 або у середньоквадратичному визначається співвідношеннями параметрів γ_t та ε_t, які повинні задовольняти фундаментальні умови стохастичної апроксимації [38–40].

Працездатність (у сенсі виконання критеріїв (4)) рекурентних алгоритмів забезпечується виконанням умови псевдоградієнтності вектора R відносно функції Ляпунова $\Delta(p)$ [38]:

$$\left\langle M\{R\{x_i(t),p_i(t),\xi_i(t)\,|\,p_i(t)=p_i\},\nabla_{p_i}(\Delta(p))\right\rangle\ge 0\,,$$

де $\langle\cdot,\cdot\rangle$ – скалярний добуток векторів в евклідовому просторі; $p_i\in S^N$; $p\in S=\prod_{i=1}^{L}S_i^N$.

Функція Ляпунова Δ повинна бути диференційованою по p_i, $i=1..L$, мати корені в точках асимптотичної оптимальності $\Delta(p^*)=0$, бути знакододатною $\Delta(p)>0$ на комбінованому одиничному симплексі $\forall p\in S$, $p\ne p^*$. Для оптимізації функції середніх виграшів на системі одиничних симплексів можна прийняти $\Delta(t)=\sum_{i=1}^{L}\left\|p_i(t)-p_i^*(t)\right\|$, де $p_i^*(t)$ – асимптотично-оптимальний розв'язок у змішаних стратегіях для i-го гравця.

Для розглянутих рекурентних методів функція Ляпунова $\Delta(t)$ може бути визначена як похибка умови доповняльної нежорсткості (квадрат норми різниці змішаних стратегій):

$$\Delta(t)=\sum_{i=1}^{L}\left\|p_i(t)-\tilde p_i(t)\right\|^2\,,$$

де $\tilde p_i(t)=diag(p_i(t))(\nabla_{p_i(t)}V_i(t))\big/V_i(t)$ – зважена змішана стратегія i-го гравця, для якої виконується умова доповняльної нежорсткості.

Середньоквадратичну швидкість збіжності рекурентних методів можна оцінити асимптотичним методом моментів Чжуна [38]:

$$\overline{\lim_{n\to\infty}}\,n^{\theta}M\{\Delta(t)\}=\vartheta\,, \qquad (15)$$

де θ – параметр порядку, ϑ – величина швидкості збіжності. Більшим значенням θ та меншим ϑ відповідає більша швидкість збіжності ігрового методу.

У знакододатних середовищах, для яких $V_i(p^{D_i}) > 0$ на системі одиничних симплексів, теоретичний порядок середньоквадратичної збіжності методів (9) та (11) становить $\theta = \min(1+\beta-\alpha, \alpha-\beta)$ при обмеженнях на параметри $\alpha \in (0,1]$; $0 < \beta < \alpha$. Теоретичний порядок середньоквадратичної швидкості збіжності методу (13) дорівнює $\theta = \min(1+\beta-\alpha, \alpha)$ при обмеженнях $\alpha \in (0,1]$; $\beta > 0$ [42].

Вибір чистих стратегій (кольорів вершин) $x_{i,k}(t)$, $i = 1..L$ здійснюється гравцями випадково на основі змішаних стратегій $p_i(t) = \left(p_{i,1}(t), p_{i,2}(t), ..., p_{i,N}(t)\right)$:

$$k = \arg\left(\min_{k=1..N} \sum_{j=1}^{k} p_{i,j}(t) > \omega\right) \in \{1..N\}, \qquad (16)$$

де $\omega \in [0, 1]$ – дійсне випадкове число з рівномірним розподілом.

Стохастична гра розпочинається з ненавчених змішаних стратегій зі значеннями елементів $p_{i,j}(0) = 1/N$, де $j = 1..N$. На протязі наступних моментів часу динаміка векторів змішаних стратегій визначається за за одним із марковських рекурентних методів. Рекурентні методи (9), (11), (13) забезпечують адаптацію стратегій гравців як до зміни реалізацій випадкового графа так і до обчислених на їх основі апріорі невідомих поточних втрат.

Отже, в моменти часу $t = 1, 2, ...$ кожен гравець на основі змішаної стратегії $p_i(t)$ вибирає чисту стратегію $x_i(t)$ (16) і до моменту часу $t+1$ отримує поточний програш $\xi_i(t)$ (1), після чого обчислює змішану стратегію $p_i(t+1)$ згідно одного із методів (9), (11), (13).

Кольори вершин випадкового графа визначаються як заокруглене до цілого значення математичне сподівання можливих кольорів, обчислене для останнього кроку стохастичної гри:

$$\overline{x}_i(t) = \mathrm{int}\left(\sum_{i=1}^{N} p_i(t) x_i(t)\right), \; i = 1..L. \qquad (17)$$

Алгоритм стохастичної гри.

Кроки 1–2 визначають ініціалізацію даних та виконують підготовчі дії, а кроки 3–11 реалізують стохастичну гру розфарбовування вершин випадкового графа.

Крок 1. Задати початкові значення параметрів:

$t = 0$ – початковий момент часу; L, K, $N = L$ – кількість чистих стратегій гравців (кількість кольорів палітри фарб);

$\Omega = \{O_1, O_2, ..., O_m\}$ – бібліотека онтологій;

$\Pi_i = \{O_{i,1}, O_{i,2}, ..., O_{i,r}\} \subseteq \Omega$, $i = 1..L$ – набори онтологічних знань або компетенцій, необхідних для виконання проектів;

$A_k = \{O_{k,1}, O_{k,2}, ..., O_{k,s}\} \subseteq \Omega$, $k = 1..K$ – набори онтологій, що визначають здібності агентів;

$q_{i,k}$, $i = 1..L$, $k = 1..K$ – імовірності участі агентів у виконанні проектів;

$U_i = \{u_{i,1}, u_{i,2}, ..., u_{i,N}\}$, $i = 1..L$ – вектори чистих стратегій гравців;

$p_i(0) = \left((1/N)_j \,\middle|\, j = 1..N\right)$, $i = 1..L$ – початкові значення змішаних стратегій гравців;

$\gamma > 0$ – параметр кроку навчання;

$\alpha \in (0,1]$ – коефіцієнт порядку кроку навчання;

ε – параметр ε-симплекса;

$\beta > 0$ – коефіцієнт порядку розширення ε-симплекса;

$\lambda \in [0,1]$ – ваговий коефіцієнт;

$t_{\max}$ – максимальна кількість кроків методу.

Крок 2. Виконати підготовчі дії:

2.1. Виконати покриття проектів онтологіями, залучивши до виконання проектів відповідних агентів.

2.2. Побудувати граф, вершини якого позначають проекти (команди агентів), а ребра з'єднують ті вершини (проекти), для виконання яких залучені однакові агенти. Сформувати початкову матрицю суміжностей вершин графа.

2.3. З кожною вершиною графа пов'язати гравців, які вибирають варіанти поточних кольорів вершин графа.

Крок 3. Визначити поточний склад команд агентів, залучених до виконання проектів з імовірністю $q_{i,k}$ та виконати нову розмітку вершин графа.

Крок 4. Визначити поточну матрицю суміжностей вершин графа.

Крок 5. Вибрати чисті стратегії (кольори вершин графа) $x_i(t) \in X_i$ гравців $i = 1..L$ згідно з (16).

Крок 6. Обчислити значення поточних програшів $\xi_i(t)$, $i = 1..L$ згідно з (1).

Крок 7. Обчислити параметри $\gamma(t)$ та $\varepsilon(t)$ згідно з (14).

Крок 8. Обчислити елементи векторів змішаних стратегій $p_i(t)$, $i = 1..L$ згідно з (13).

Крок 9. Обчислити поточні значення функцій середніх програшів $\Xi_i(t)$ (2) кожного гравця і на їх основі обчислити системну функцію середніх програшів $\Xi(t)$ (3) стохастичної гри розфарбовування графа.

Крок 10. Задати наступний момент часу $t := t+1$.

DOI 10.15588/1607-3274-2022-1-14

Крок 11. Якщо $t < t_{\max}$, то перейти на крок 3, інакше – на крок 12.

Крок 12. Обчислити усереднені значення кольорів $\overline{x}_i(t)$, $i = 1..L$ для вершин графа згідно з (17). Кінець гри.

4 ЕКСПЕРИМЕНТИ

Нехай задані:

1) бібліотека онтологій $\Omega = \{O_1, O_2, O_3, O_4, O_5\}$;

2) компетенції, необхідні для виконання проектів $\Pi_1 = \{O_1, O_3, O_4\}$, $\Pi_2 = \{O_2, O_3, O_5\}$, $\Pi_1 = \{O_1, O_4, O_5\}$, $\Pi_4 = \{O_1, O_3, O_5\}$;

3) здібності агентів $A_1 = \{O_1, O_4\}$, $A_2 = \{O_2, O_3\}$, $A_3 = \{O_1, O_5\}$, $A_4 = \{O_3, O_4\}$, які можуть бути залучені для виконання проектів.

Виходячи з цих даних, можливі такі покриття проектів агентами: $\Pi_1 = \{A_1, A_2\}$, $\Pi_2 = \{A_2, A_3\}$, $\Pi_1 = \{A_1, A_3\}$, $\Pi_4 = \{A_3, A_4\}$. Дійсно, для заданих проектів справедливі наступні відношення покриття онтологій:

$\Pi_1 : A_1 \cup A_2 = \{O_1, O_2, O_3, O_4\} \supseteq \{O_1, O_3, O_4\}$;

$\Pi_2 : A_2 \cup A_3 = \{O_1, O_2, O_3, O_5\} \supseteq \{O_2, O_3, O_5\}$;

$\Pi_3 : A_1 \cup A_3 = \{O_1, O_4, O_5\} \supseteq \{O_1, O_4, O_5\}$;

$\Pi_4 : A_3 \cup A_4 = \{O_1, O_3, O_4, O_5\} \supseteq \{O_1, O_3, O_5\}$.

На рис. 1 зображено граф, вершинами якого є проекти Π_i, а ребрами – зв'язки між тими проектами, для виконання яких планується залучити одних і тих же агентів. У вершинах графа наведено список агентів A_k, $k = 1..K_i$, залучених до виконання проектів. З кожною вершиною (проектом) пов'язується гравець, який виконує ходи стохастичної гри щодо вибору поточного кольору вершини, залежно від кольорів з'єднаних ребрами сусідніх вершин графа.

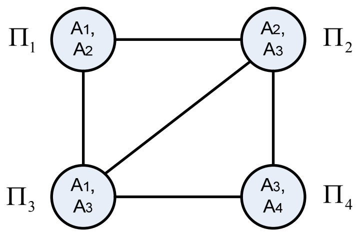

Рисунок 1 – Граф залежностей проектів від виконавців

Відмови агентів призводять до зміни складу команд виконавців проектів і до відповідної зміни зв'язків між вершинами графа. У результаті, замість детермінованого графа, отримаємо випадковий граф, декілька можливих реалізацій якого зображено на рис. 2.

На рис. 2a зображено реалізацію випадкового графа для випадку, коли відмовив агент A1 проекту П1. Така ж структура графа буде у випадку відмови агента A1 проекту П3, або відмови агента A1 для обох проектів П1 та П3. Результатом цього є втрата зв'язку між проектами П1 та П3.

Відмова агента A2 проекту П2 призведе до реалізації випадкового графа, зображеної на рис. 2b. Аналогічний результат отримаємо у випадку відмови агента A2 проекту П1, або відмови агента A2 проектів П1 та П2.

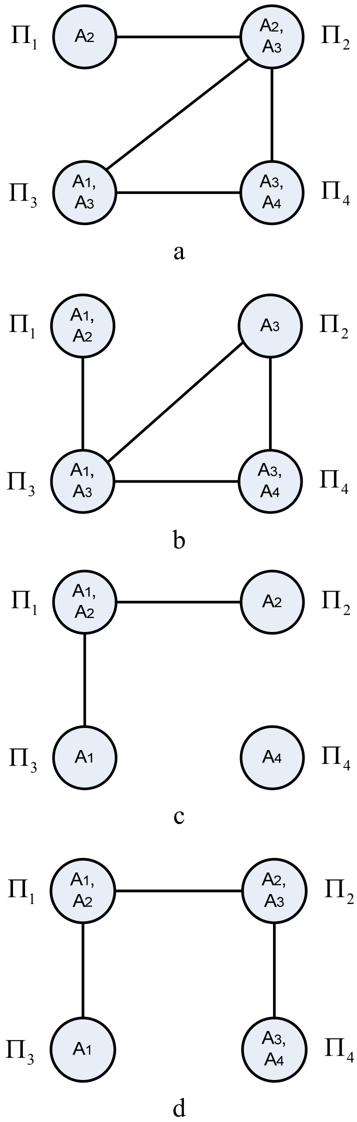

Рисунок 2 – Реалізації випадкового графа

Випадок відмови агента A3, залученого до виконання проектів П2, П3 та П4, зображено на рис. 2c. На цьому рисунку видно, що відмови агентів можуть призвести до порушення зв'язності графа проектів.

Реалізація графа, зображеного на рис. 2d, отримана у результаті відмови агента A3 проекту П3.

Для заданого випадкового графа необхідно визначити такий порядок виконання проектів, для якого можлива участь кожного агента є послідовною у часі.

DOI 10.15588/1607-3274-2022-1-14

5 РЕЗУЛЬТАТИ

Розв'язування цієї задачі виконаємо стохастичним ігровим методом (13) розфарбовування випадкового графа. Для цього задамо такі параметри стохастичної гри: $L = 4$; $N = L$; $K = 4$; $\gamma_0 = 1$; $\alpha = 0,01$; $\varepsilon_0 = 0,999 N^{-1}$; $\beta = 2$.

Вплив імовірностей $q_k = q$, $i = 1..K$ участі агентів у виконанні проектів на збіжність стохастичної гри зображено на рис. 3 у логарифмічному масштабі.

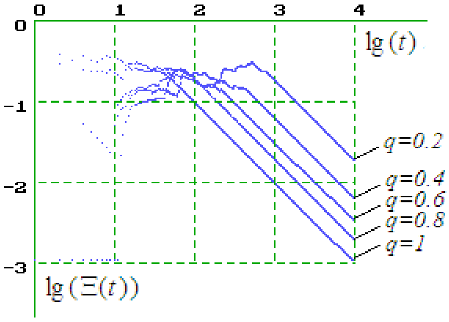

Рисунок 3 – Залежність системної функції середніх втрат від імовірностей участі агентів у виконанні проектів

Параметр θ порядку швидкості збіжності (15) ігрового методу визначається тангенсом кута, утвореного лінійною апроксимацією графіка системної функції середніх втрат та віссю часу. Як видно на рис. 3, середній порядок швидкості збіжності ігрового методу є наближеним до 1. Він практично не змінюється для різних імовірностей участі агентів у виконанні проектів. Із зменшенням значення цих імовірностей лише зростає час навчання стохастичної гри.

На рис. 4 зображено апроксимовану показникову залежність середньої кількості кроків $\overline{t}$, необхідних для розфарбовування випадкового графа, від імовірностей $q_k = q$, $i = 1..K$ участі агентів у грі.

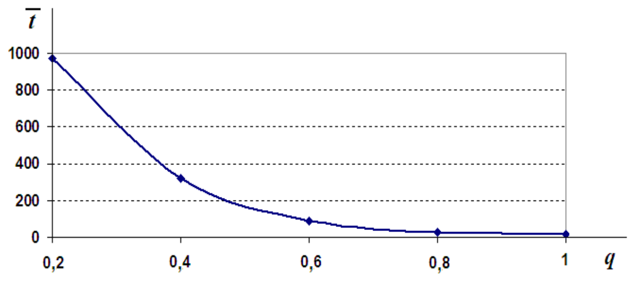

Рисунок 4 – Середня кількість кроків навчання стохастичної гри

Зростання імовірності $q \to 1$ участі агентів у виконанні проектів призводить до збільшення подібності реалізацій випадкового графа $G(t)$ до заданого детермінованого графа G:

$$\delta(t) = \mathfrak{M}(G(t), G) \to 0,$$

де $\delta(t) \in [0,1]$ – міра близькості графів у моменти часу $t = 1, 2, ...$ [43]. Результатом цього є зменшення кількості кроків навчання стохастичної гри. Для абсолютно надійних агентів ($q = 1$) потрібно 10–20 кроків стохастичної гри для формування правильного розфарбовування зображеного на рис. 1 графа.

Розв'язком стохастичної гри є зображений на рис. 5 розфарбований граф, отриманий для імовірностей $q_k = q = 0,8$, $i = 1..K$ участі агентів у виконанні проектів.

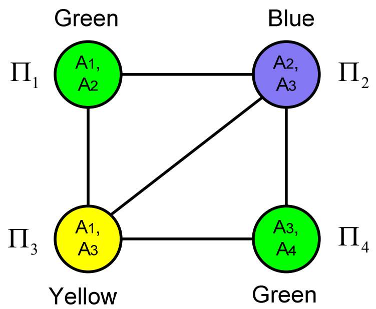

Рисунок 5 – Розфарбований граф проектів

Проекти, що відповідають вершинам з однаковим кольором, можуть виконуватись одночасно (паралельно). Дві послідовності виконання проектів із шести можливих зображено на рис. 6.

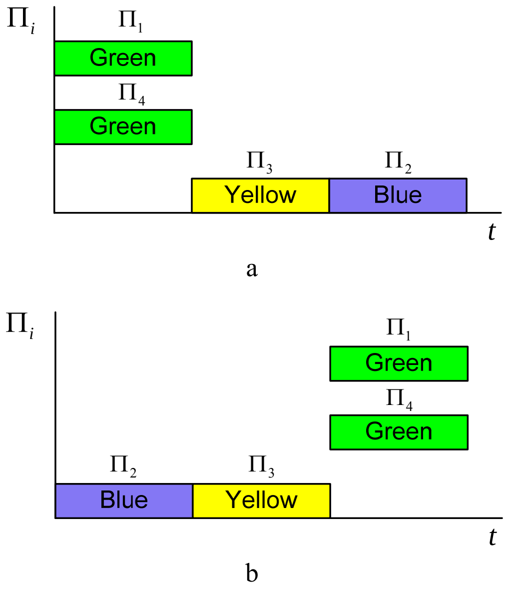

Рисунок 6 – Послідовності виконання проектів

Як видно на рис. 6a та рис. 6b, проекти П1 та П4 можуть виконуватися одночасно, а проекти П2 та П3 – послідовно з цими паралельними проектами. Застосування алгоритму розфарбовування графа визначає лише порядок виконання проектів без врахування їх, можливо, різної тривалості.

DOI 10.15588/1607-3274-2022-1-14

6 ОБГОВОРЕННЯ

В умовах апріорної невизначеності випадкового графа проектів важливою є максимізація імовірностей покриття проектів.

Нехай $Z_i = 2^{K_i}$ – повний простір комбінованих станів агентів, залучених до виконання i-го проекту ($i = 1..L$), де K_i – кількість таких агентів. Значення стану $s_{i,j} = 1$ сигналізує про участь $j \Leftrightarrow k$-го агента, а значення $s_{i,j} = 0$ – про його відмову від участі у виконанні проекту Π_i. Тут операція $\Leftrightarrow$ виконує взаємно однозначне відображення послідовних номерів $j = 1..K_i$ агентів A_j проекту Π_i та реальних номерів агентів A_k. Тоді імовірність покриття проекту $p_{\text{cov}}(\Pi_i)$ визначається як сума імовірностей тих комбінованих станів $(s_{i,1}s_{i,2}...s_{i,K_i})$, для яких індивідуальні стани агентів дорівнюють 1 і об'єднані онтології цих агентів покривають даний проект.

Наприклад, для зображеного на рис. 1 графа кожен проект Π_i, $i = 1..L$ має такий простір станів для двох залучених для його виконання агентів:

$$Z_i = \{(s_{i,1}s_{i,2}) \mid s_{i,j} \in \{0,1\}, j = 1,2\} = \{00, 01, 10, 11\} .$$

Очевидно, що імовірність реалізації усіх комбінованих станів дорівнює 1: $(1-q_{i,l})(1-q_{i,m}) + (1-q_{i,l})q_{i,m} + q_{i,l}(1-q_{i,m}) + q_{i,l}q_{i,m} = 1$. Тут перший індекс позначає номер проекту, а другий – номер агента.

Враховуючи склад множин онтологій, імовірність покриття проекту, наприклад Π_1, визначається так: $p_{\text{cov}}(\Pi_1) = q_{1,1}q_{1,2}$, де $\Pi_1 = \{A_1, A_2\}$. Якщо $q_{1,k} = q$, $k = 1,2$, то матимемо квадратичну залежність імовірності покриття $p_{\text{cov}}(\Pi_1) = q^2$ проекту від імовірності відмов агентів.

Імовірність покриття можна підвищити залученням у проект надлишкових агентів, онтології яких хоча б частково входять у множину онтологій проекту. Для прикладу введемо у проект Π_1 додаткового агента A_4, що призведе до появи у графі нового зв'язку між вершинами Π_1 та Π_4. У результаті цього простір станів агентів матиме вигляд:

$$Z_1 = \{(s_{1,1}s_{1,2}s_{1,3}) \mid s_{1,j} \in \{0,1\}, j = 1..3\} = \{000, 001, 010, 011, 100, 101, 110, 111\} .$$

З урахуванням нового набору онтологій імовірність покриття проекту $\Pi_1 = \{A_1, A_2, A_4\}$ тепер буде дорівнювати:

$$p'_{\text{cov}}(\Pi_1) = q_{1,1}(1-q_{1,2})q_{1,4} + q_{1,1}q_{1,2}(1-q_{1,4}) + q_{1,1}q_{1,2}q_{1,4} .$$

Легко перевірити, що $p'_{\text{cov}}(\Pi_1) \ge p_{\text{cov}}(\Pi_1)$. В окремому випадку для однакових значень імовірностей участі агентів у виконанні проекту маємо $p'_{\text{cov}}(\Pi_1) = 2q^2(1-q) + q^3 \ge q^2$.

Графіки імовірностей $p_{\text{cov}}(\Pi_1)$ та $p'_{\text{cov}}(\Pi_1)$ покриття проекту Π_1 для $q_{1,k} = q \in [0,1]$, $k = 1..K$ наведено на рис. 7.

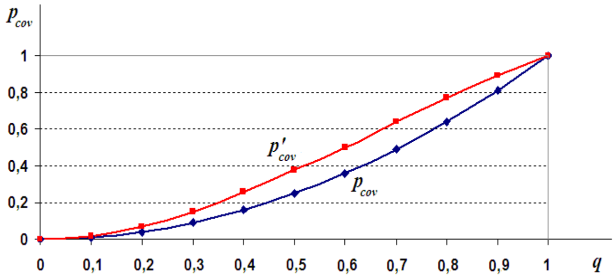

Рисунок 7 – Імовірності покриття проектів

Надлишкове покриття проекту по-різному впливає на збіжність стохастичної гри, залежно від варіантів залученння агентів для виконання проектів. Типові реалізації функції усереднених у часі програшів для проекту Π_1 подані на рис. 8 для різних імовірностей покриття. Графік 1 отримано для $p_{\text{cov}}(\Pi_1) = 0.64$, а графік 2 – для $p'_{\text{cov}}(\Pi_1) = 0.768$, обчислених для $q_{1,k} = q = 0.8$, $k = 1..K$.

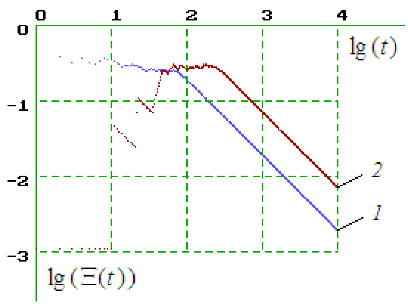

Рисунок 8 – Вплив надлишкового покриття проекту на збіжність стохастичної гри

Як видно на рис. 7 надлишкове покриття проектів через залучення додаткових агентів може забезпечити зростання імовірності покриття затребуваних онтологій проектів наявними онтологіями агентів. Однак, як показано на рис. 8, для випадкових графів це може призвести до погіршення показників збіжності ігрового методу за рахунок залежності проектів від одних і тих же агентів.

Результатом залучення однакових додаткових агентів є зростання зв'язності графа проектів, який у випадку відмов агентів розпадається на більшу кількість підграфів, у ньому частіше з'являються ізольовані вершини, що послаблює умови збіжності стохастичної гри розфарбовування випадкового графа. Збільшення порядку графа, що еквівалентно збільшенню кількості проектів, також призводить до сповільнення збіжності стохастичної гри.

DOI 10.15588/1607-3274-2022-1-14

Залучення надлишкових агентів до виконання проектів можна обмежити додатковим критерієм мінімізації вартості виконання проектів.

ВИСНОВКИ

За результатами проведеного дослідження можна зробити такі висновки:

1) розв'язано задачу планування послідовності виконання проектів на основі самонавчального стохастичного ігрового методу розфарбовування графа;

2) завдяки своїм адаптивним властивостям, стохастичний ігровий метод можна застосувати для розфарбовування випадкового графа, який враховує імовірності участі агентів у виконанні проектів;

3) результатом навчання стохастичної гри є асимптотично правильно розфарбований випадковий граф, який дозволяє визначити послідовність призначення персоналу для виконання проектів;

4) збіжність стохастичного ігрового методу забезпечується збалансованим співвідношенням його параметрів при дотриманні фундаментальних обмежень стохастичної апроксимації;

5) зростання порядку графа, зв'язності графа, імовірностей відмов агентів призводять до зростання кількості кроків, необхідних для збіжності стохастичної гри розфарбовування графа;

6) метод стохастичної гри розфарбовування графа можна застосувати для розв'язування подібних задач, що формулюються в умовах неповної інформації, наприклад, складання різноманітних розкладів, розпаралелювання алгоритмів, класифікації, кластеризації даних та інших;

7) розглянута стохастична гра має також самостійне значення як модель глобальної самоорганізації станів розподіленої системи, проявом якої є правильне розфарбування випадкового графа в умовах невизначеності на основі опрацювання локально зібраних даних.

ПОДЯКИ

Роботу виконано в рамках держбюджетних тем «Методи та засоби функціонування систем підтримки прийняття рішень на основі онтологій» (ID:839 2017-05-15 09:20:01 (2459-315)) та «Система підтримки прийняття рішень розпізнавання мультиспектральних образів на основі технологій машинного навчання та онтологічного підходу» (ID:0120U102203). Дослідження провадилось в межах спільних наукових досліджень кафедри інформаційних систем та мереж НУ «Львівська політехніка» на тему «Дослідження, розроблення і впровадження інтелектуальних розподілених інформаційних технологій та систем на основі ресурсів баз даних, сховищ даних, просторів даних та знань з метою прискорення процесів формування сучасного інформаційного суспільства». Наукові дослідження провадилися також в рамках ініціативної тематики досліджень кафедри ІСМ НУ «Львівська політехніка» на тему «Розроблення інтелектуальних розподілених систем на основі онтологічного підходу з метою інтеграції інформаційних ресурсів».

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Virtual Communities: Concepts, Methodologies, Tools and Applications / Information Resources Management Association (USA). – Vol. 1–4. – Hershey : IGI Global, 2011. – 2930 p. DOI: 10.4018/978-1-60960-100-3.
2. Heagney J. Fundamentals of Project Management / J. Heagney. – HarperCollins Focus, 2018. – 240 p.
3. Asdre K. The harmonius coloring problem is NP-complete for interval and permutation graphs / K. Asdre, K. Ioannidou, S. D. Nikolopoulos // Discret Applied Mathematics. – 2007. – Vol. 155. – P. 2377–2382.
4. NP-completeness of local colorings of graphs / Z. Li, E. Zhu, Z. Shao, J. Xu // Information Processing Letters. – 2018. – Vol. 130. – P. 25–29.
5. Keet C. M. An Introduction to Ontology Engineeing, v.1.5 / C. M. Keet. – University of Cape Town, South Africa. – 2020. – 306 p. – Access mode: https://people.cs.uct.ac.za/~mkeet/0Ebook/.
6. Кравець П. О. Ігрова модель онтологічної підтримки проектів / П. О. Кравець, В. В. Литвин, В. А. Висоцька // Radio Electronics, Computer Science, Control. – 2021. – Vol. 1, No. 1. – P. 172–183. DOI: 10.15588/1607-3274-2021-1-17 .
7. Chen B.-S. Stochastic Game Strategies and their Applications / B.-S. Chen. – CRC Press, 2019. – 610 p.
8. Panik M. J. Linear Programming and Resource Allocation Modeling / M. J. Panik. – Wiley, 2018. – 448 p.
9. Markis S. Allocation of Manufacturing Tasks to Humans and Robots / S. Markis // Cooperating Robots for Flexible Manufacturing. – Springer, 2021. – P. 373–380.
10. Wang Z. Machine learning-based intermittent equipment sheduling model for flexible production process / Z. Wang, Y. Man // Application of Artificial Intelligence in Process Systems Enineering. – 2021. – P. 473–495.
11. Yang G. Resource allocation algorithm and job sheduling of virtual manufacturing workshop / G. Yang, S. Chen // Academic Journal of Manufacturing Engineering. – 2020. – Vol. 18, No. 2. – P. 155–161.
12. Yeganeh F.T. A multi-objective optimazation approach to project sheduling with resiliency criteria under uncertain activity duration / F. T. Yeganeh, S. H. Zegordi // Annals of Operations Research. – 2020. – Vol. 285, No. 1. – P. 161–196.
13. Liu W. Production sheduling and equipment matching of flexible workshops based on multi-objective and multi-process hybrid optimization algorithm / W. Liu // Academic Journal of Manufacturing Engineering. – 2020. – Vol. 18, No. 4. – P. 158–163.
14. Катренко А. В. Дослідження операцій : підручник / А. В. Катренко. – Львів : Магнолія Плюс, 2006. – 549 p.
15. The Reseach of Task Assignment Based on Ant Colony Algorithm / [Z. Wang, S. Li, Y. Wang, S. Li] // International Conference on Mechatronics and Automation. – IEEE Publisher, 2009. – P. 2334–2339. DOI: 10.1109/ICMA.2009.5246570.
16. Sahu A. Solving the Assignment Problem using Genetic Algorithm and Simulated Annealing / A. Sahu, R. Tapadar // IAENG International Journal of Applied Mathematics, IJAM_36_1_7. – 2007. – Vol. 36:1. – P. 1–4.
17. Zais M. A graph coloring approach to the deployment shedaling and unit assignment problem / M. Zais, M. Laguna // Journal of Sheduling. – 2016. – Vol. 19. – P. 73–90.
18. Zhu A. A neural network approach to dynamic task assignment of multirobots / A. Zhu, S. X. Yang // IEEE transactions on neural networks. – 2006. – Vol. 17, No. 5. – P. 1278–1287.

DOI 10.15588/1607-3274-2022-1-14

19. Wu S. S. Heuristic algorithm for task assignment and sheduling in a processor network / S. S. Wu, D. Sweeting // Parallel Computting. – 1994. – Vol. 20, Issue 1. – P. 1–14.
20. Chartrand G. Chromatic Graph Theory / G. Chartrand, P. Zhang. – Chapman and Hall/CRC, 2019. – 525 p.
21. Saoub K. R. Graph Theory. An Introduction to Proofs, Algorithms, and Applications / K. R. Saoub. – Chapman and Hall/CRC, 2021. – 437 p.
22. Monasson R. On the Analysis of Backtrack Procedures for the Colouring of Random Graphs / R. Monasson // Lect. Notes Phys. – 2004. – Vol. 650. – P. 235–254.
23. Christofides N. Graph theory: an algorithmic approach / N. Christofides. – New York : Academic Press, 1975. – 400 p.
24. Bincy A. K. Graph Coloring and its Real Time Applications an Overview / A. K. Bincy, B. J. Presitha // International Journal of Mathematics And its Applications. – 2017. – Vol. 5, Issue 4-F. – P. 845-849.
25. Denysenko O. Overview of graph coloring methods and algorithms / O. Denysenko // Міжнародний мультидисциплінарний науковий журнал «ΛΌГОΣ. Мистецтво наукової думки». – 2019. – No. 7. – P. 27–32.
26. Lima A. M. Exact Algorithms for the Graph Coloring Problem / A. M. Lima, R. Carmo // Revista de Informatica Teórica e Aplicada. – RITA. – 2018. – Vol. 25, No. 04. – P. 57–73.
27. Gupta S. Greedy Graph Coloring Algorithm Based on Depth First Search / S. Gupta, D. P. Singh // International Journal on Emerging Technologies. – 2020. – Vol. 11, No. 2. – P. 854–862.
28. A genetic algorithm for total graph coloring / [A. Dey, A. Agarwal, P. Dixit et al.] // Journal of Intelligent & Fuzzy Systems. – 2019. – Vol. 37, No. 6. – P. 7831–7838.
29. Philipsen W. J. M. Graph coloring using neural networks / W. J. M. Philipsen, L. Stok // IEEE International Symposium on Circuits and Systems. – 1991. – P. 1597–1600.
30. A chaotic binary salp swarm algorithm for solving the graph coloring problem / [Y. Meraihi, A. Ramdane-Cherif, M. Mahseur, D. Achelia] // International Symposium on Modelling and Implementation of Complex Systems. – 2018. – P. 106–118.
31. Dowsland K. A. An improved ant colony optimization heuristic for graph colouring / K. A. Dowsland, J. M. Thompson // Discrete Applied Mathematics. – 2008. – Vol. 156, No. 3. – P. 313–324.
32. Blum A. Multiagent Graph Coloring: Pareto Efficiency, Fairness and Individual Rationality / A. Blum, J. S. Rosenschein // Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence. – 2008. – P. 24–29.
33. Panagopoulou P. N. A Game Theoretic Approach for Efficient Graph Coloring / P. N. Panagopoulou, P. G. Spirakis // In: Hong S. H., Nagamochi H., Fukunada T. (eds). Algorithm and Computation. ISAAC 2008. Lecture Notes in Computer Science. – Springer, Berlin, Heidelberg, 2008. – Vol. 5369. DOI: 10.1007/978-3-540-92182-0_19.
34. Кравець П. О. Ігрова модель хроматичного розфарбовування графів / П. О. Кравець // Комп'ютерні системи проектування. Теорія і практика: Вісник НУ «Львівська політехніка». – 2008. – № 626. – С. 63–74.
35. Райгородский А. М. Модели случайных графов / А. М. Райгородский. – Москва : МЦНМО, 2011. – 136 с.
36. Frieze A. Introduction to random graphs / A. Frieze, M. Karoński. – Cambridge University Press, 2016. – 478 p.
37. Ungureanu V. Pareto-Nash-Stackelberg Game and Control Theory: Intelligent Paradigms and Applications / V. Ungureanu. – Springer, 2018. – 343 p.
38. Назин А. В. Адаптивный выбор вариантов: Рекуррентные алгоритмы / А. В. Назин, А. С. Позняк. – Москва : Наука, 1986. – 288 с.
39. Kushner H. Stochastic Approximation and Recursive Algorithms and Applications / H. Kushner, G. G. Yin. – Springer Science & Business Media, 2013. – 417 p.
40. Benveniste A. Adaptive Algorithms and Stochastic Approximations / A. Benveniste, M. Metivier, P. Priouret. – Springer Science & Business Media, 2012. – 365 p.
41. Neogy S. K. Mathematical Programming and Game Theory / S. K. Neogy, R. B. Bapat, D. Dubey. – Springer, 2018. – 226 p.
42. Кравець П. О. Ігрова самоорганізація системи агентів з індивідуальним оцінюванням стратегій / П. О. Кравець // Комп'ютерні системи та мережі: Вісник НУ «Львівська політехніка». – 2005. – № 546. – С. 75–85.
43. Білова Т. Г. Метод оцінки ступеню структурної близькості зв'язних неорієнтованих графів / Т. Г. Білова, І. О. Побіженко // Обробка інформації в складних технічних системах. – 2017. – Вип. 1, № 47. – С. 9–12. – DOI: 10.30748/soi.2017.147.02.

Стаття надійшла до редакції 11.10.2021.
Після доробки 08.12.2021.

УДК 004.[852+94]; 519.837.3

МОДЕЛИРОВАНИЕ ИГРОВОЙ ЗАДАЧИ НАЗНАЧЕНИЯ ПЕРСОНАЛА ДЛЯ ВЫПОЛНЕНИЯ ИТ-ПРОЕКТОВ НА ОСНОВЕ ОНТОЛОГИЙ

Кравец П. А. – канд. техн. наук, доцент, доцент кафедры «Информационные системы и сети», Национальный университет «Львовская политехника», Украина.

Литвин В. В. – д-р техн. наук, профессор, заведующий кафедры «Информационные системы и сети», Национальный университет «Львовская политехника», Украина.

Высоцка В. А. – канд. техн. наук, доцент, доцент кафедры «Информационные системы и сети», Национальный университет «Львовская политехника», Украина.

АННОТАЦИЯ

Актуальность. В этой статье описано решение игровой задачи назначения персонала для работы над проектами на основе онтологического подхода. Суть задачи состоит в следующем. Существует потребность в создании команд для выполнения нескольких проектов. Каждый проект задается набором необходимых онтологических знаний. Для выполнения проектов менеджеры привлекают квалифицированных специалистов (агентов), способности которых также задаются наборами онтологий. Состав команд должен быть таким, чтобы объединенные онтологии их агентов покрывали множества онтологий соответствующих проектов. Каждый агент с определенными вероятностями может принять последовательную участие в выполнении нескольких проектов. Одновременная работа агента над различными проектами не допускается. Необходимо определить порядок выполнения проектов и соответствующий ему порядок назначения персонала.

Целью исследования является разработка математической модели стохастической игры, рекуррентных марковских методов для ее решения, алгоритмического и программного обеспечения, проведение компьютерного эксперимента, анализ результатов и выработкой рекомендаций по их практическому применению.

DOI 10.15588/1607-3274-2022-1-14

Метод. Для планирования выполнения проектов использовано стохастический игровой алгоритм раскраски неориентированного случайного графа. Для этого количество вершин графа принято равным количеству проектов. Ребрами соединены те вершины графа проектов, для выполнения которых привлечено одного и того же агента. С учетом восстановительных отказов агентов связи между вершинами графа динамически изменяются. Необходимо достичь правильной раскраски случайного графа. Тогда проекты с одинаково раскрашенными вершинами графа могут быть выполнены параллельно, а проекты с различными цветами вершин – последовательно.

Результаты. В статье построена математическая модель стохастической игры и обучаемый марковский метод для ее решения. Каждая вершина графа контролируется игроком. Чистыми стратегиями игрока являются элементы палитры цветов. После выбора цвета своей вершины каждый игрок вычисляет текущий проигрыш как относительное количество одинаковых цветов в локальном множестве соседних игроков. Цель игроков заключается в минимизации функций средних проигрышей. Марковский рекуррентный метод обеспечивает адаптивный выбор цветов вершин случайного графа на основе динамических векторов смешанных стратегий, значения которых зависят от текущих проигрышей игроков. Результатом стохастической игры является асимптотически правильно раскрашенный случайный граф, когда каждому ребру начального детерминированного графа будут соответствовать в среднем разные цвета вершин.

Выводы. Проведен компьютерный эксперимент, который подтвердил сходимость стохастической игры для задачи раскраски случайного графа. Это дало возможность определить порядок назначения персонала для выполнения проектов.

КЛЮЧЕВЫЕ СЛОВА: проект, агент, онтология, назначение персонала, раскраска случайного графа, стохастическая игра, марковский рекуррентный метод, адаптация, самообучение.

UDC 004.[852+94]; 519.837.3

MODELLING GAME TASK OF ASSIGNING STAFF TO PERFORM IT-PROJECTS BASED ON ONTOLOG IES

Kravets P. – PhD, Associate Professor of Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, Ukraine.

Lytvyn V. – Dr. Sc., Professor, Head of Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, Ukraine.

Vysotska V. – PhD, Associate Professor of Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, Ukraine.

ABSTRACT

Context. This article describes how to solve the game problem of assigning staff to work on projects based on an ontological approach. The essence of the problem is this. There is a need to create teams to carry out several projects. Each project is defined by a set of necessary ontological knowledge. To implement projects, managers invite qualified specialists (agents), whose abilities are also defined by sets of ontologies. The composition of the teams should be such that the combined ontologies of their agents cover the set of ontologies of the respective projects. Each agent with a certain probability can take part in the implementation of several projects. Simultaneous work of the agent on different projects is not allowed. It is necessary to determine the order of project implementation and the corresponding order of personnel appointment.

Objective of the study is to develop a mathematical model of stochastic game, recurrent Markov methods for its solution, algorithmic and software, computer experiment, analysis of results and development of recommendations for their practical application.

Method. A stochastic game algorithm for coloring an undirected random graph was used to plan project execution. To do this, the number of vertices of the graph is taken equal to the number of projects. The edges of the project graph for which the same agent is invited are connected by edges. Due to the recovery failures of agents, the connections between the vertices of the graph change dynamically. It is necessary to achieve the correct coloring of the random graph. Then projects with the same colored vertices of the graph can be executed in parallel, and projects with different colors of vertices – in series.

Results. The article builds a mathematical model of a stochastic game and a self-learning Markov method for its solution. Each vertex of the graph is controlled by the player. The player's pure strategies are the elements of the color palette. After selecting the color of their own top, each player calculates the current loss as a relative number of identical colors in the local set of neighboring players. The goal of the players is to minimize the functions of average losses. The Markov recurrent method provides an adaptive choice of colors for the vertices of a random graph based on dynamic vectors of mixed strategies, the values of which depend on the current losses of players. The result of a stochastic game is an asymptotically correctly colored random graph, when each edge of the initial deterministic graph will correspond on average to different colors of vertices.

Conclusions. A computer experiment was performed, which confirmed the convergence of the stochastic game for the problem of coloring a random graph. This made it possible to determine the procedure for appointing staff to implement projects.

KEYWORDS: project, agent, ontology, staff assigning, coloring of random graph, stochastic game, Markovian recursive method, adaptation, self-learning.

REFERENCES

1. Virtual Communities: Concepts, Methodologies, Tools and Applications, *Information Resources Management Association (USA*), Vol. 1–4. Hershey, IGI Global, 2011, 2930 p. DOI: 10.4018/978-1-60960-100-3.
2. Heagney J. Fundamentals of Project Management. HarperCollins Focus, 2018, 240 p.
3. Asdre K., Ioannidou K., Nikolopoulos S. D. The harmonius coloring problem is NP-complete for interval and permutation graphs, *Discret Applied Mathematics*, 2007, Vol. 155, pp. 2377–2382.
4. Li Z., Zhu E., Shao Z., Xu J. NP-completeness of local colorings of graphs, *Information Processing Letters*, 2018, Vol. 130, pp. 25–29.

DOI 10.15588/1607-3274-2022-1-14

5. Keet C. M. An Introduction to Ontology Engineeing, v.1.5. University of Cape Town, South Africa, 2020, 306 p. Access mode: https://people.cs.uct.ac.za/~mkeet/0Ebook/.
6. Kravets P., Lytvyn V., Vysotska V. Game model of ontological project support (in ukrainian), *Radio Electronics, Computer Science, Control*, 2021, Vol. 1, No. 1, pp. 172–183. DOI: 10.15588/1607-3274-2021-1-17.
7. Chen B.-S. Stochastic Game Strategies and their Applications. CRC Press, 2019, 610 p.
8. Panik M. J. Linear Programming and Resource Allocation Modeling. Wiley, 2018, 448 p.
9. Markis S. Allocation of Manufacturing Tasks to Humans and Robots, *Cooperating Robots for Flexible Manufacturing*. Springer, 2021, pp. 373–380.
10. Wang Z., Man Y. Machine learning-based intermittent equipment sheduling model for flexible production process, *Application of Artificial Intelligence in Process Systems Enineering*, 2021, pp. 473–495.
11. Yang G., Chen S. Resource allocation algorithm and job sheduling of virtual manufacturing workshop, *Academic Journal of Manufacturing Engineering*, 2020, Vol. 18, No. 2, pp. 155–161.
12. Yeganeh F. T., Zegordi S. H. A multi-objective optimazation approach to project sheduling with resiliency criteria under uncertain activity duration, *Annals of Operations Research*, 2020, Vol. 285, No. 1, pp. 161–196.
13. Liu W. Production sheduling and equipment matching of flexible workshops based on multi-objective and multi-process hybrid optimization algorithm, *Academic Journal of Manufacturing Engineering,* 2020, Vol. 18, No. 4, pp. 158-163.
14. Katrenko A.V. Operations research. Manual (in ukrainian). Lviv, Magnolia Plus, 2006, 549 p.
15. Wang Z., Li S., Wang Y., Li S. The Reseach of Task Assignment Based on Ant Colony Algorithm, *International Conference on Mechatronics and Automation, IEEE Publisher,* 2009, pp. 2334–2339. DOI: 10.1109/ICMA.2009.5246570.
16. Sahu A., Tapadar R. Solving the Assignment Problem using Genetic Algorithm and Simulated Annealing, *IAENG International Journal of Applied Mathematics, IJAM_36_1_7,* 2007, Vol. 36:1, pp. 1–4.
17. Zais M., Laguna M. A graph coloring approach to the deployment shedaling and unit assignment problem, *Journal of Sheduling*, 2016, Vol. 19, pp. 73–90.
18. Zhu A., Yang S. X. A neural network approach to dynamic task assignment of multirobots, *IEEE transactions on neural networks,* 2006, Vol. 17, No. 5, pp. 1278–1287.
19. Wu S. S., Sweeting D. Heuristic algorithm for task assignment and sheduling in a processor network, *Parallel Computting*, 1994, Vol. 20, Issue 1, pp. 1–14.
20. Chartrand G., Zhang P. Chromatic Graph Theory. Chapman and Hall/CRC, 2019, 525 p.
21. Saoub K. R. Graph Theory. An Introduction to Proofs, Algorithms, and Applications. Chapman and Hall/CRC, 2021, 437 p.
22. Monasson R. On the Analysis of Backtrack Procedures for the Colouring of Random Graphs, *Lect. Notes Phys*, 2004, Vol. 650, pp. 235–254.
23. Christofides N. Graph theory: an algorithmic approach / N. Christofides. New York, Academic Press, 1975, 400 p.
24. Bincy A. K., Presitha B. J. Graph Coloring and its Real Time Applications an Overview, *International Journal of Mathematics And its Applications*, 2017, Vol. 5, Issue 4-F, pp. 845–849.
25. Denysenko O. Overview of graph coloring methods and algorithms, *International Multidisciplinary Science Journal «ΛΌΓΟΣ. The art of scientific thought»*, 2019, No. 7, pp. 27–32.
26. Lima A. M., Carmo R. Exact Algorithms for the Graph Coloring Problem, *Revista de Informatica Teórica e Aplicada*, RITA, 2018, Vol. 25, No. 04, pp. 57–73.
27. Gupta S., Singh D. P. Greedy Graph Coloring Algorithm Based on Depth First Search, *International Journal on Emerging Technologies*, 2020, Vol. 11, No. 2, pp. 854–862.
28. Dey A., Agarwal A., Dixit P., Long H. V., Werner F., Pal T., Son L. H. A genetic algorithm for total graph coloring, *Journal of Intelligent & Fuzzy Systems*, 2019, Vol. 37, No. 6, pp. 7831–7838.
29. Philipsen W. J. M., Stok L. Graph coloring using neural networks, *IEEE International Symposium on Circuits and Systems,* 1991, pp. 1597–1600.
30. Meraihi Y., Ramdane-Cherif A., Mahseur M., Achelia D. A chaotic binary salp swarm algorithm for solving the graph coloring problem, *International Symposium on Modelling and Implementation of Complex Systems*, 2018, pp. 106–118.
31. Dowsland K. A., Thompson J. M. An improved ant colony optimization heuristic for graph colouring, *Discrete Applied Mathematics,* 2008, Vol. 156, No. 3, pp. 313–324.
32. Blum A., Rosenschein J. S. Multiagent Graph Coloring: Pareto Efficiency, Fairness and Individual Rationality, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 2008, pp. 24–29.
33. Panagopoulou P. N., Spirakis P. G. A Game Theoretic Approach for Efficient Graph Coloring, *In: Hong S.H., Nagamochi H., Fukunada T. (eds). Algorithm and Computation. ISAAC 2008. Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 2008, Vol. 5369, DOI: 10.1007/978-3-540-92182-0_19.
34. Kravets P. O. Game model of chromatic coloring of graphs (in ukrainian), *Computer Design Systems. Theory and Practice: Bulletin of the National University of Lviv Polytechnic*, 2008, No. 626, pp. 63–74.
35. Raigorodskii A. M. Models of random graphs (in russian). Moscow, Moscow Center for Continuous Mathematical Education (MCCME), 2011, 136 p.
36. Frieze A., Karoński M. Introduction to random graphs. Cambridge University Press, 2016, 478 p.
37. Ungureanu V. Pareto-Nash-Stackelberg Game and Control Theory: Intelligent Paradigms and Applications. Springer, 2018, 343 p.
38. Nazin A. V., Poznyak A. S. Adaptive Choice of Variants: Recurrence Algorithms (in russian). Moscow, Science, 1986, 288 p.
39. Kushner H., Yin G. G. Stochastic Approximation and Recursive Algorithms and Applications. Springer Science & Business Media, 2013, 417 p.
40. Benveniste A., Metivier M., Priouret P. Adaptive Algorithms and Stochastic Approximations. Springer Science & Business Media, 2012, 365 p.
41. Neogy S. K., Bapat R. B., Dubey D. Mathematical Programming and Game Theory. Springer, 2018, 226 p.
42. Kravets P. A. Game self-organization of agents system with individual estimation of strategies (in ukrainian), *Computer systems and networks: Bulletin of the National University of Lviv Polytechnic*, 2005, No. 546, pp. 75–85.
43. Bilova T. G., Pobezhenko I. A. Method of estimation the degree of structural proximity of bound non-oriented graphs (in ukrainian), *Processing of information in complex technical systems,* 2017, Issue 1, No. 47, pp. 9–12. DOI: 10.30748/soi.2017.147.02.

DOI 10.15588/1607-3274-2022-1-14