Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра математики

Курсова робота

на тему: «Дифузія знань і технологій»

Виконала: студентка 3-го року
навчання, спеціальності 113
"Прикладна математика"

Харченко Олександра Сергіївна

Керівник курсової роботи:
к. ф.-м. н. Чорней Р.К.

Київ 2021

ЗМІСТ

ВСТУП

За останні десятиліття економіка зазнає великих змін. Виробництво та конкуренція на ринку зростає швидкими темпами, впроваджуються інновації. Нові технології в бізнесі допомагають підприємцям розширяти свій бізнес, підвищувати прибуток та виводити компанії на новий рівень. Наразі завдання з дослідження та оптимізації виробничих процесів є особливо актуальними.

Процес, під час якого знання та технології поширюються з плином часу, називається «дифузією знань» [1, с. 25]. Таке розповсюдження відбувається між організаціями для комунікації та впровадження інновацій. Без ефективного обміну ідеями та технологіями компанії не матимуть змоги розвиватися. Для розширення бізнесу, підприємствам також слід вивчати конкурентів та заздалегідь оцінювати вплив прийнятих ними рішень.

Метою дослідження є розглянути використання марковських процесів в економіці. Зокрема, при визначенні стратегії економічного розвитку підприємства. Дослідити наслідки впровадження тих чи інших технологій.

Об'єктом дослідження є дифузії технологій.

Предметом дослідження є задачі управління та оптимізації підприємства із застосуванням полів Маркова.

Для досягнення поставленої в роботі мети використано методи дослідження операцій із застосуванням теорії ймовірності.

Дана робота складається з 2 розділів. У першому розділі представлено основні означення та твердження, що необхідні для кращого розуміння роботи. Також у цьому розділі детально описана економічна задача вибору стратегії підприємства. Матеріал 1-го розділу має реферативний характер, підрозділи 1.2 та 1.3 спираються на дослідження П. С. Кнопова, що опубліковані в статті про застосування полів Маркова в економіці [2, с. 3923-3931]. У другому розділі наводиться приклад задачі, описаної у теоретичній частині роботи.

РОЗДІЛ 1. Теоретична частина

1.1 Дослідження марковських ланцюгів та основні поняття теорії випадкових процесів

Основні поняття теорії марковських ланцюгів ввів у 1906 видатний російський математик А.А. Марков (1856-1922). Він вперше почав вивчати схеми залежних випробувань, а також пов'язаних із ними сум випадкових величин і отримав ряд основоположних результатів [3, с. 6]. Теорія марковських процесів широко застосовується в різних областях наук таких, як: інженерія, фізика, біологія та економіка.

Введемо поняття «випадкової величини» та «випадкового процесу», які необхідні для кращого розуміння теорії марковських процесів.

Випадкова величина – це вимірна числова функція $\xi = f(\omega)$ $(f\colon \Omega \to \mathbb{R})$ елементарної події [4, с. 44].

Нехай кожному t з деякого числового інтервалу поставлена у відповідність випадкова величина $\xi(t)$. Тоді сукупність $\{\xi(t)\}$ цих випадкових величин називається випадковим процесом. Зазвичай $\xi(t)$ називають значенням випадкового процесу в момент часу t [4, с. 119].

Нехай проводиться серія випробувань з номерами $0,1 \dots$ та можливими наслідками $x_1, x_2, \dots x_n$. Позначимо ξ_n – наслідок n-го випробування. Послідовність $\{\xi_n,\ n \in Z+\}$ називається ланцюгом Маркова, якщо за відомого результату n-го випробування, результат $(n+1)$-го не залежить від наслідків 0-го, $\dots$, $(n-1)$-го випробувань [4, с. 105]:

$$P(\xi_{n+1} = x_j / \xi_0 = x_{i_0}, \dots, \xi_{n-1} = x_{i_{n-1}}, \xi_n = x_i) = P(\xi_{n+1} = x_j / \xi_n = x_i)$$

Іншими словами: при фіксованому теперішньому, кожен наступний стан об'єкта є результатом попереднього і є джерелом наступного стану, при цьому інші стани до уваги не приймаються.

Розрізняють два різновиди випадкових процесів, які відбуваються в системі: з дискретним і неперервним часом.

Випадковий процес $x(t)$ називається процесом з дискретним часом, якщо система, у якій він відбувається, може змінювати свої стани тільки в моменти часу $t_1, t_2, t_3, \dots t_n$, кількість яких є зліченною [5, с. 15].

Випадковий процес $x(t)$ називається процесом з неперервним часом, якщо переходи системи зі стану в стан можуть відбуватися у будь-який момент часу t спостережуваного періоду континуальної потужності [5, с. 15].

У даній роботі зупинимося на процесах з дискретним часом.

1.2 Випадкові поля Маркова

Введемо деякі поняття, що будуть необхідні в подальшому. Матеріал цього підрозділу є реферативним, використовуються теоретичні положення роботи П.С. Кнопова [2, с. 3923-3924].

Нехай $\Gamma = (V, B)$ – деякий скінченний граф із множиною вершин V і множиною ребер B. Позначимо через (k, j) ребро графа, що з'єднує вершини k та j. Околом вершини k є множина вузлів $N(k) = \{j: (k, j) \in B,\ k \in V\}$, тобто це вершини, що з'єднані ребром, які виходять з k. Повним околом вершини k позначимо множину $\widetilde{N}(k) = N(k) \cup \{k\}$.

Нехай $X_i = \{x_i^1, \dots, x_i^n\}$ – це простір станів вершини $i \in V$ та $X = \prod_{i \in V} X_i$ – простір станів всієї системи. Для кожної підмножини $K \subset V$ послідовність $\{x_k,\ k \in K\}$ буде позначатися як x_k.

Випадкова величина ξ, яка визначена на V та приймає значення в X, називається випадковим полем на V. Ймовірнісна міра, що відповідає цьому полю, також часто позначається як ξ.

У багатьох задачах економічного характеру випадкове значення ξ залежить від часу t. У цьому випадку запишемо $\xi = \xi^t$, щоб підкреслити часову залежність. Якщо t приймає дискретні значення $t = 0,1, \dots$, тоді запис ξ_t^k позначатиме граничну зміну для часу t і вузла k деякого векторного процесу $\xi = (\xi^t : t = 0,1 \dots)$ [6, с. 53].

Використаємо такі позначення:

$$P\{\xi_k = x_k, k = K\} = \xi(x_k),$$

де $P\{\xi_k = x_k, k = K/\xi_j;\ j \in J \backslash K\} = \xi(x_k/x_{J-K})$, індекс k величини ξ_k підкреслює, що ця змінна пов'язана з вершиною k.

Введемо деякі поняття властивостей Маркова:

Означення 1.1. Випадкова величина ξ на (Γ, X) називається випадковим полем Маркова, якщо для всіх $K \subset V,\ J \subset V$ справедлива така рівність:

$$\xi(x_K/x_{J-K}) = \xi(x_K/X_{N(K)}),$$

де $N(K) = \{N(k), k \in K\}$.

Це означення передбачає, що поведінка випадкового поля на деякому підграфі K графу Γ при фіксованому значенні поля в околі $N(k)$, не залежить від поведінки поля поза підграфом K.

Якщо поле ξ залежить від часу, тоді необхідно розширити поняття сусідньої точки для $k \in V$. Передбачається, що окіл $\widetilde{N}(k)$ точки k включає в себе точку *k.* У цьому випадку скористаємося наступним визначенням властивості Маркова для ξ^t.

Означення 1.2. Випадкове поле ξ^t на (Γ, X) називається випадковим полем Маркова, якщо для всіх $K \subset I \subset V$ маємо:

$$P\{\xi^{t+1}{}_K = x_k/\xi^t{}_I\} = P\{\xi^{t+1}{}_K = x_k/\xi^t{}_{\widetilde{N}(k)}\}.$$

Ймовірність переходу у вершину k залежить тільки від попереднього стану її повного околу. У цьому випадку поле можна розглядати як звичайний марковський процес, стан якого в кожен момент часу є випадковим полем. Більш точним є уявлення про марковське поле як процес Маркова з взаємодією, окремим випадком якого є марковське поле у значенні Означення 1.2.

Для таких процесів характерна задача: нехай система складається зі скінченної кількості частинок, розвиток яких, описується незалежними ланцюгами Маркова. На шляху розвитку цієї системи встановлюється певний зв'язок та накладаються деякі обмеження. Тоді еволюція кожної частинки не є марковською, хоча еволюція всієї системи має марковську властивість [2, с. 3924].

1.3. Прикладні задачі: економічна постановка

Для прикладу розглянемо економічну задачу. Матеріал цього підрозділу також має реферативний характер, використовуються положення роботи П.С. Кнопова [2, с. 3924-3931].

Розглянемо раніше визначений граф $\Gamma = (V, B)$ з визначеним околом точок. Множина вершин – це множина компаній, а множина ребер – множина взаємодій між ними. Нехай $X_i = \{x,..., x_i^{n_i}\}$ – множина станів елементів x_i, де кожній організації відповідає деякий кінцевий набір значень. У кожному стані множини X_i i-та компанія може приймати той чи інший стандарт або технологію. Покладемо $X = \prod_{i=V} X_i$. Нехай на (Γ, X) буде визначено або випадкове поле ξ, або ймовірність міра, що відповідає полю ξ.

Припустимо, що рішення x, прийняте k-ою організацією у кожен момент часу, залежить від рішень, прийнятих іншими організаціями в попередні моменти. Властивість Маркова у значенні Означення 1.2. передбачає, що це рішення залежить лише від рішень, прийнятих у попередній момент часу сусідніми організаціями до k. Можливі стани системи можна заздалегідь визначити, а перехід системи з одного стану в інший відбувається «стрибком», майже миттєво.

Розглянемо випадкові поля Маркова, які мають наступну властивість:

$$P\{\xi_k^{t+1} = x_k / \xi^t = y\} = \prod_{k \in K} P\{\xi_k^{t+1} = x_k / {\xi^t}_{\tilde{N}(k)} = y_{\tilde{N}(k)}\} \ (1)$$

Співвідношення (1) передбачає, що для відомих компонент ξ^t, компоненти ξ_k^{t+1} поля Маркова ξ^{t+1} залежать лише від ${\xi^t}_{\tilde{N}(k)}$ і стани компонент ${\xi^t}_i$ змінюються незалежно один від одного.

Позначимо через Q функцію переходу розглянутого поля Маркова, а через q – його компоненти:

$$q_{N(k)}(x_k) = P\{\xi_k^{t+1} = x_k / {\xi^t}_{N(k)} = x_{N(k)}\}$$

Вважатимемо, що $q_{N(k)}(x_k)$ є незалежним від t.

Припустимо, що i-та компанія має продукцію двох типів, і нехай r_{ij} є витратами продажу j-го виду продукції i-ої компанії протягом одного періоду часу. Нехай f_{ij} будуть ймовірностями того, що i-та компанія вирішить вивести на ринок j-й тип товару, $j = 1, 2$. Тоді середні витрати i-ої компанії за один період часу дорівнюють:

$$Er_1 = f_{i1}\, r_{\mathrm{i}1} + f_{i2}\, r_{\mathrm{i}2} \qquad (2)$$

Як і у випадку зі звичайними ланцюгами Маркова, можна продемонструвати, що величина (2) дорівнює середнім витратам за одиницю часу системи, що працює в стаціонарному режимі. Кінцеві ймовірності f_{ij} залежать від перехідних ймовірностей розглянутого процесу, які, в свою чергу, залежать від стратегії, що приймає та чи інша компанія.

Представимо задачу в більш загальному вигляді. Як і раніше, припускаємо, що i-та компанія виготовляє продукцію двох типів, тобто $x_i = \{0,1\}$, і в кожний момент часу рішення, прийняте цією компанією, залежить від стану всієї системи та від рішення, прийнятого в попередній момент часу. За один період часу для i-ої компанії надається множина можливих рішень U_i, які для простоти вважаються незалежними від часу.

Припустимо, що множина рішень U_i складається з двох елементів $U_i = \{\alpha_1, \alpha_2\}$. Кожного моменту часу рішення приймається відповідно до перехідної ймовірності, що відповідає α_1 або α_2. Якщо задається функція витрат залежно від множини U_i, то задача полягає у виборі такого рішення, при якому функція досягає свого оптимального значення. Сформулюємо задачу в більш строгому вигляді.

Ми досліджуємо стохастичну систему ξ^t з дискретним часом $t = 0, 1, \ldots$ та множиною станів X. Стохастична система ξ^t визначається графом

$\Gamma = (V, B)$, де V вважається скінченним. Нехай $\Delta = \{\Delta^t\}$ – множина можливих рішень. Вважатимемо, що у кожен момент часу t, множина можливих рішень складається з K елементів та є скінченною. Тоді ймовірності переходу ξ визначаються так:

$$P\{\xi_S^{t+1} = x/\xi^0, \Delta_i^0 \dots \xi^t = y, \Delta_i^t = k\} = P\{\xi_S^{t+1} = k/\xi^t = y, \Delta_i^t = k\} =$$

$$= \prod_{j\in S} P\{\xi_j^{t+1} = x_j/\xi_{\tilde{N}(j)}^t = y_{\tilde{N}(j)}, \Delta_i^t = k\} = q_{yx}^S(k).$$

Покладемо $q_{yx}(k) = P\{\xi^{t+1} = x/\xi^t = y, \Delta_i^t = k\}$, якщо $S = V$. Очевидно, що

$$\sum_{x\in X} q_{yx}(k) = 1, \qquad y \in X.$$

Позначимо за D_{yk} ймовірність того, що у поточному стані y, буде прийняте k-е рішення, тобто:

$$D_{yk} = P\{\Delta_i^t = k/\xi^0, \Delta_i^0, \dots, \xi^t = y\}, \text{ де}$$

$$\sum_{k=1}^{K} D_{yk} = 1$$

Покладемо:

$$P_{yx} = \sum_{k=1}^{K} D_{yk} * q_{yx}(k)$$

$$\text{та } \varphi_{Txk}(y) = \frac{1}{T+1}\sum_{t=0}^{T} P\{\xi^t = x, \Delta_i^t = k/\xi^0 = y\}.$$

$\varphi_{Txk}(y)$ – це критерій якості керування, тобто це очікуване значення ймовірності того, що з початкового стану y, ми отримаємо стан x у момент часу t, в якому будемо приймати рішення k.

Означимо $W_{xk}(i)$ як витрати i-ої компанії, якщо $\xi^t = x$, і компанія приймає рішення k ($\Delta_i^t = k$). Тоді середні очікувані витрати з часом T дорівнюють:

$$Q_{Ti}(y) = E\frac{1}{T+1}\sum_{t=0}^{T}\Delta_t^i = \sum_{x\in X}\sum_{k=1}^{K}W_{xk}(i)*\varphi_{Txk}(y) \quad (3)$$

Задача полягає в тому, щоб знайти таке рішення з послідовності Δ_i^t при $t = 0{,}1\ldots$, яка б мінімізувала $\lim_{T\to\infty} sup\, Q_{Ti}(y)$. Подібно до випадку звичайних ланцюгів Маркова з скінченною множиною станів і можливих рішень, можна показати, що оптимальне рішення досягається серед ланцюгів Маркова, для яких справедливим є наступне співвідношення:

$$\lim_{T\to\infty}\varphi_{Txk}(y) = \Pi_x D_{xk}, \quad x \in \mathrm{X}, \qquad k = 1, \ldots, K,$$

де Π_x – це ергодичні ймовірності, які задовольняють рівняння:

$$\sum_{y\in X}\mathrm{P}_{yx} * \Pi_y = \Pi_y,$$

$$\sum_{y}\Pi_y = 1$$

Нехай $z_{xk} = \Pi_{xk}D_{xk}$ (4). Тоді, підставляючи значення $(4) \to (3)$, ми отримаємо таку задачу лінійного програмування: знайти

$$min\sum_{x}\sum_{k}W_{xk}(i) * z_{xk} \quad (5)$$

при умові, що

$$z_{xk} \geq 0,$$

$$\sum_{k=1}^{K} z_{xk} = \sum_{y \in X} \sum_{k=1}^{K} z_{yk} * q_{yx}(k), \qquad x \in X \quad (6)$$

$$\sum_{y \in X} \sum_{k=1}^{K} z_{yk} = 1. \ (7)$$

Введемо такі позначення:

$$U_0 = \frac{1}{\sum_{k=1}^{K} z_{yk}}, U_{xk} = \frac{z_{xk}}{\sum_{k=1}^{K} z_{xk}} \ (8)$$

Тоді задача (5) – (7) буде еквівалентною такій: знайти

$$min \sum_{x \in X} \sum_{k=1}^{K} W_{xk} * U_{xk} \quad (9)$$

за умов:

$$U_0 \geq 0, \qquad U_{xk} \geq 0$$

$$\sum\nolimits_{k=1}^{K} U_{xk} = \sum_{y \in X} \sum_{k=1}^{K} U_{yk} * q_{yx}(k),$$

$$\sum_{x \in X} \sum\nolimits_{k=1}^{K} U_{xk} = U_{0,} \quad \sum_{k=1}^{K} U_{xk} = 1$$

Після знаходження U_{xk} та мінімізації (9), знайдемо оптимальний рішення, яке має прийняти компанія, коли система перебуває в тому чи іншому стані:

$$D_{xk} = \frac{U_{xk}}{\sum_{k}^{K} U_{xk}} \quad (10)$$

Отже, ми розглянули приклад задач оптимізації та управління в галузі математичної економіки.

РОЗДІЛ 2. Практична частина

Розглянемо приклад задачі, у якому докладно розпишемо процес взаємодії трьох компаній.

Нехай $\Gamma = (V, B)$ – це скінченний граф із множиною вершин $V = 3$ і множиною ребер $B = 2$. Множина вершин V – це множина організацій, а множина ребер B – множина взаємодій між ними.

У нашому випадку перша компанія взаємодіє з другою та третьою. Зафіксуємо центр керування, що буде знаходиться у першій вершині.

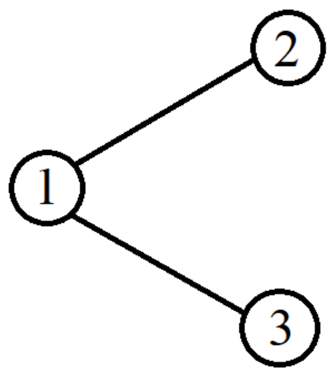

Рис 1. Взаємодія компаній

У певний момент часу 1-а компанія має прийняти рішення щодо підтримки нової технології на основі того, в якому стані перебуває система. Тоді стан першої компанії залежатиме від стану 2-ої та 3-ої компанії.

Нехай $X = \{0, 1\}$ – множина станів кожної компанії. $K = \{1, 2\}$ – множина прийнятих рішень.

Розпишемо ймовірності переходу для всіх вершин:

Для 1-ої вершини: $P(\xi_1/\xi_1, \xi_2, \xi_3, \Delta_1)$

Для 2-ої вершини: $P(\xi_2/\xi_1, \xi_2)$

Для 3-ої вершини: $P(\xi_3/\xi_1, \xi_3)$

Отже, представлено три компанії, кожна з яких у певний момент часу може перебувати у двох станах і приймати рішення $\{1,2\}$. Тоді загальна кількість змінних у нашій задачі дорівнюватиме $2^3 * 2 = 16$. Тобто $Z = \{z_1, z_2, z_3 \dots z_{15}, z_{16}\}$.

Маємо такі перехідні ймовірностей системи:

$$\mathrm{P}(\xi_s/\xi_s, \Delta_1) = \prod P(\xi_1\ /\xi_S, \Delta_1) * P(\xi_2\ /\xi_1, \xi_2, \Delta_1) * P(\xi_3\ /\xi_1, \xi_3, \Delta_1)$$

У теоретичній частині ми показали, що наша задача зводиться до задачі лінійного програмування. Цільова функція описується рівнянням (5), а обмеження – рівностями (6) – (7). Для розв’язання задачі задамо чисельні значення витрат компанії W в залежності від станів, в яких перебуває система, та перехідні ймовірності (таблиця 1).

	W(111)	w(100)	W(010)	W(001)	W(110)	W(101)	W(011)	W(000)
	79	84	48	82	46	29	16	11
(1,1,1) прийнято рішення 1	0.683	0.01	0.02	0.215	0.048	0.003	0.02	0.001
(1,1,1) прийнято рішення 2	0.289	0.57	0.085	0.019	0.004	0.006	0.016	0.011
(1,0,0) прийнято рішення 1	0.693	0.145	0.009	0.029	0.048	0.056	0.005	0.015
(1,0,0) прийнято рішення 2	0.88	0.034	0.015	0.016	0.013	0.029	0.008	0.005
(0,1,0) прийнято рішення 1	0.569	0.18	0.011	0.15	0.067	0.001	0.001	0.021
(0,1,0) прийнято рішення 2	0.064	0.441	0.027	0.018	0.048	0.001	0.001	0.4
(0,0,1) прийнято рішення 1	0.421	0.274	0.162	0.015	0.096	0.007	0.013	0.012
(0,0,1) прийнято рішення 2	0.358	0.166	0.19	0.156	0.034	0.03	0.065	0.001
(1,1,0) прийнято рішення 1	0.2	0.133	0.198	0.016	0.385	0.021	0.032	0.015
(1,1,0) прийнято рішення 2	0.016	0.021	0.287	0.147	0.484	0.002	0.001	0.042
(1,0,1) прийнято рішення 1	0.112	0.015	0.036	0.139	0.187	0.325	0.012	0.174
(1,0,1) прийнято рішення 2	0.052	0.098	0.158	0.636	0.014	0.024	0.013	0.005
(0,1,1) прийнято рішення 1	0.119	0.2	0.095	0.155	0.045	0.292	0.052	0.042
(0,1,1) прийнято рішення 2	0.185	0.089	0.089	0.116	0.309	0.068	0.098	0.046
(0,0,0) прийнято рішення 1	0.118	0.356	0.025	0.07	0.163	0.082	0.027	0.159
(0,0,0) прийнято рішення 2	0.164	0.329	0.147	0.09	0.024	0.008	0.184	0.054

Таблиця 1

За формулою (5) запишемо цільову функцію задачі

$$F \ = \ 79z_1 + 79z_2 + 84z_3 + 84z_4 + 48z_5 + \cdots + 11z_{15} + 11z_{16} \rightarrow min$$

Запишемо обмеження за формулами (6) та (7):

$$\begin{cases} z_1 + z_2 + z_3 + \cdots + z_{15} + z_{16} = 1 \\ z_1 + z_2 = 0.683z_1 + 0.289z_2 + 0.693z_3 + \cdots + 0.164z_{16} \\ z_3 + z_4 = 0.01z_1 + 0.57z_2 + 0.145z_3 + \cdots + 0.329z_{16} \\ z_5 + z_6 = 0.02z_1 + 0.085z_2 + 0.009z_3 + \cdots + 0.147z_{16} \\ \ldots \\ z_{15} + z_{16} = 0.001z_1 + 0.011z_2 + 0.015z_3 + \cdots + 0.054z_{16} \end{cases}$$

Спростимо отриману систему:

$$\begin{cases} z_1 + z_2 + z_3 + \cdots + z_{15} + z_{16} = 1 \\ -0.317z_1 - 0.711z_2 + 0.693z_3 + \cdots + 0.164z_{16} = 0 \\ 0.01z_1 + \cdots - 0.855z_3 - 0.966z_4 + \cdots + 0.329z_{16} = 0 \\ 0.02z_1 + \cdots - 0.989z_5 - 0.973z_6 + \cdots + 0.147z_{16} = 0 \\ 0.215z_1 + \cdots - 0.985z_7 - 0.844z_8 + \cdots + 0.09z_{16} = 0 \\ 0.048z_1 + \cdots - 0.615z_9 - 0.516z_{10} + \cdots + 0.024z_{16} = 0 \\ 0.003z_1 + \cdots - 0.675z_{11} - 0.976z_{12} + \cdots + 0.008z_{16} = 0 \\ 0.02z_1 + \cdots - 0.948z_{13} - 0.902z_{14} + \cdots + 0.184z_{16} = 0 \\ 0.001z_1 + \cdots - 0.841z_{13} - 0.946z_{14} = 0 \end{cases}$$

За допомогою симплекс-методу розв'яжемо цю задачу лінійного програмування в Microsoft Excel.

Цільова функція

z1	z2	z3	z4	z5	z6	z7	z8	z9	z10	z11	z12	z13	z14	z15	z16	Z
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
79	79	84	84	48	48	82	82	46	46	29	29	16	16	11	11	

Система обмежень

-0.317	-0.711	0.693	0.88	0.569	0.064	0.421	0.358	0.2	0.016	0.112	0.052	0.119	0.185	0.118	0.164	0	=	0
0.01	0.57	-0.855	-0.966	0.18	0.441	0.274	0.166	0.133	0.021	0.015	0.098	0.2	0.089	0.356	0.329	0	=	0
0.02	0.085	0.009	0.015	-0.989	-0.973	0.162	0.19	0.198	0.287	0.036	0.158	0.095	0.089	0.025	0.147	0	=	0
0.215	0.019	0.029	0.016	0.15	0.018	-0.985	-0.844	0.016	0.147	0.139	0.636	0.155	0.116	0.07	0.09	0	=	0
0.048	0.004	0.048	0.013	0.067	0.048	0.096	0.034	-0.615	-0.516	0.187	0.014	0.045	0.309	0.163	0.024	0	=	0
0.003	0.006	0.056	0.029	0.001	0.001	0.007	0.03	0.021	0.002	-0.675	-0.976	0.292	0.068	0.082	0.008	0	=	0
0.02	0.016	0.005	0.008	0.001	0.001	0.013	0.065	0.032	0.001	0.012	0.013	-0.948	-0.902	0.027	0.184	0	=	0
0.001	0.011	0.015	0.005	0.021	0.4	0.012	0.001	0.015	0.042	0.174	0.005	0.042	0.046	-0.841	-0.946	0	=	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	=	1

У результаті отримаємо такі коефіцієнти:

Цільова функція																
z1	z2	z3	z4	z5	z6	z7	z8	z9	z10	z11	z12	z13	z14	z15	z16	Z
0.45802	0	0.10165	0	0	0.07688	0	0.15271	0	0.10929	0.02633	0	0	0.02382	0.05131	0	67.6705
79	79	84	84	48	48	82	82	46	46	29	29	16	16	11	11	

Після оптимізації скористаємося формулою (8) і знайдемо оптимальний контроль:

		U
1	(1,1,1) прийнято рішення 1	1
2	(1,1,1) прийнято рішення 2	0
3	(1,0,0) прийнято рішення 1	1
4	(1,0,0) прийнято рішення 2	0
5	(0,1,0) прийнято рішення 1	0
6	(0,1,0) прийнято рішення 2	1
7	(0,0,1) прийнято рішення 1	0
8	(0,0,1) прийнято рішення 2	1
9	(1,1,0) прийнято рішення 1	0
10	(1,1,0) прийнято рішення 2	1
11	(1,0,1) прийнято рішення 1	1
12	(1,0,1) прийнято рішення 2	0
13	(0,1,1) прийнято рішення 1	0
14	(0,1,1) прийнято рішення 2	1
15	(0,0,0) прийнято рішення 1	1
16	(0,0,0) прийнято рішення 2	0

У станах системи 1,3,6,8,10,11,14,15 компанії краще прийняти рішення 1, а у станах 2,4,5,7,9,12,13,16 – рішення 2.

Також у Python був написаний код, за допомогою якого було вирішено дану задачу див. додаток *A*. Відповіді з Excel зійшлись.

```
Коефіцієнти:
[0.45801731 0.         0.10165059 0.         0.         0.07688123
 0.         0.15270706 0.         0.1092881  0.02632642 0.
 0.         0.02381608 0.05131321 0.        ]
Оптимальний контроль:
[1. 0. 1. 0. 0. 1. 0. 1. 0. 1. 1. 0. 0. 1. 1. 0.]
```

Рис 2. Вивід результату з Python

У додатку Б записаний алгоритм для розв’язку будь-якої задачі, що має однакову структуру доданої, де користувач вводить інформацію самостійно.

ВИСНОВКИ

У курсовій роботі було досліджено використання марковських процесів при визначенні стратегії економічного розвитку підприємства. Зокрема, була розглянута задача взаємодії трьох компаній, в результаті якої, компанія-керівник може оцінювати яким чином їй краще впроваджувати нові технології. Розв'язок цієї задачі зводиться до задачі лінійного програмування.

У подальшому можна досліджувати більш складні задачі, в яких кількість компаній та взаємодій між ними буде більшою.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] Роджерс, Еверетт М. Диффузія інновацій / Пер. з англ. Василя Старка. – К.: Вид. дім «Києво-Могилянська академія», 2009. – 591 с.

[2] Knopov P. S. Markov fields and their applications in economics. Journal of Mathematical Sciences / Knopov. // Journal of Mathematical Sciences. – 1999. – №97. – С. 3923–3931.

[3] Казаков В. А. Введение в теорию марковских процессов и некоторые радиоехнические задачи / В. А. Казаков. – Москва, 1973. – 232 с.

[4] Чорней Р. К. Елементи теорії ймовірностей і випадкових процесів / Р. К. Чорней. – Київ, 2018. – 136 с.

[5] Голосков О. Є. Основи теорії експоненціальних систем / О. Є. Голосков, А. О. Голоскова, Є. О. Мошко. – Харків, 2017. – 310 с.

[6] Chornei R. K. Control of spatially structured random processes and random fields with applications / R. K. Chornei, H. Daduna, P. S. Knopov. – Boston/Dordrecht/London: Springer. – 261 c.

Додаток А

Програмний код для розглянутого прикладу з практичної частини.

```
import numpy as np
from scipy.optimize import linprog
n = 8
# an array that contains all costs of the company in each condition
w = np.array([79, 84, 48, 82, 46, 29, 16, 11])
# a matrix that contains all the transition probabilities of the system
matrix = np.array([
    [0.683, 0.01, 0.02, 0.215, 0.048, 0.003, 0.02, 0.001],
    [0.289, 0.57, 0.085, 0.019, 0.004, 0.006, 0.016, 0.011],
    [0.693, 0.145, 0.009, 0.029, 0.048, 0.056, 0.005, 0.015],
    [0.88, 0.034, 0.015, 0.016, 0.013, 0.029, 0.008, 0.005],
    [0.569, 0.18, 0.011, 0.15, 0.067, 0.001, 0.001, 0.021],
    [0.064, 0.441, 0.027, 0.018, 0.048, 0.001, 0.001, 0.4],
    [0.421, 0.274, 0.162, 0.015, 0.096, 0.007, 0.013, 0.012],
    [0.358, 0.166, 0.19, 0.156, 0.034, 0.03, 0.065, 0.001],
    [0.2, 0.133, 0.198, 0.016, 0.385, 0.021, 0.032, 0.015],
    [0.016, 0.021, 0.287, 0.147, 0.484, 0.002, 0.001, 0.042],
    [0.112, 0.015, 0.036, 0.139, 0.187, 0.325, 0.012, 0.174],
    [0.052, 0.098, 0.158, 0.636, 0.014, 0.024, 0.013, 0.005],
    [0.119, 0.2, 0.095, 0.155, 0.045, 0.292, 0.052, 0.042],
    [0.185, 0.089, 0.089, 0.116, 0.309, 0.068, 0.098, 0.046],
    [0.118, 0.356, 0.025, 0.07, 0.163, 0.082, 0.027, 0.159],
    [0.164, 0.329, 0.147, 0.09, 0.024, 0.008, 0.184, 0.054]])
```

```
# making the transformation (5)
k = 0
for i in range(0, 8, 1):
    k = 0
    if i == 0:
        matrix[i][i] = matrix[i][i] - 1
        matrix[i + 1][i] = matrix[i + 1][i] - 1
        k = k + 1
    else:
        matrix[k + 2 * i][i] = matrix[k + 2 * i][i] - 1
        matrix[k + 2 * i + 1][i] = matrix[k + 2 * i + 1][i] - 1
newRow = np.zeros(16)
# adding the condition (6)
for i in range(0, 16):
    newRow[i] = 1
newMatrix = matrix.transpose()
newMatrix = np.vstack([newMatrix, newRow])
print(newMatrix)
n = 9
m = 16
A = []
for i in range(n):
    K = []
    for j in range(m):
        K.append(newMatrix[i][j])
```

```
    A.append(K)
# Simplex method
obj = [79., 79., 84., 84., 48., 48., 82., 82., 46., 46., 29., 29., 16., 16., 11., 11.]
lhs_eq = A
rhs_eq = [0., 0., 0., 0., 0., 0., 0., 0., 1.]
opt = linprog(obj, A_eq=lhs_eq, b_eq=rhs_eq, method='Simplex')
# Print coefficients
print('Коефіцієнти:')
print(opt.x)
xx = opt.x
xp = np.zeros(16)
for i in range(0, 16, 2):
    xp[i] = xx[i] / (xx[i] + xx[i + 1])
    xp[i + 1] = xx[i + 1] / (xx[i] + xx[i + 1])
print('Оптимальний контроль:')
print(xp)
```

Додаток Б

Програмний код для розв’язку розглянутої задачі, де головна компанія взаємодіє з n-ою кількістю компаній, які користувач вводить самостійно. Також користувач має ввести інформацію про витрати компанії у різних станах та перехідні ймовірності.

```
import numpy as np

from scipy.optimize import linprog

v = int(input('Введіть кількість компаній:'))

n = 2 ** (v+1)

m = 2 ** v

# remember the initial value of n and m

nn = n

mm = m

# initializing an empty array that contains all costs of the company in each
condition

w = np.zeros(m)

for i in range(m):

    # taking costs input from the user

    z = input('{0}{1}{2}'.format('Введіть', i + 1, ' витрати компанії: '))

    w[i] = int(z)

# initializing an empty matrix that contains all the transition probabilities of the
system

matrix = []

for i in range(n):

    # taking row input from the user

    row = list(map(float, input('{0}{1}{2}'.format('Введіть ', i + 1, '
рядок:')).split()))

    # appending the 'row' to the 'matrix'
```

```
    matrix.append(row)
print(matrix)
# making the transformation (5)
for i in range(0, m, 1):
    k = 0
    if i == 0:
        matrix[i][i] = matrix[i][i] - 1
        matrix[i + 1][i] = matrix[i + 1][i] - 1
        k = k + 1
    else:
        matrix[k + 2 * i][i] = matrix[k + 2 * i][i] - 1
        matrix[k + 2 * i + 1][i] = matrix[k + 2 * i + 1][i] - 1
newRow = np.zeros(n)
# adding the condition (6)
for i in range(0, n):
    newRow[i] = 1
numpy_array = np.array(matrix)
transpose = numpy_array.T
newMatrix = transpose.tolist()
newMatrix.append(newRow)
print('Matrices')
print(newMatrix)
n = m + 1
m = nn
print(n)
print(nn)
A = []
```

```
# prepare the matrix for the simplex method
for i in range(n):
    K = []
    for j in range(m):
        K.append(newMatrix[i][j])
    A.append(K)
# preparing objective function1
obj = np.zeros(m)
d = 0
for i in range(n-1):
    obj[d] = w[i]
    obj[d+1] = w[i]
    d +=2
# use Simplex method
lhs_eq = A
rhs_eq =np.zeros(n)
for i in range (n):
    if i !=n-1:
        rhs_eq[i]=0.
    else:
        rhs_eq[i] = 1.
opt = linprog(obj, A_eq=lhs_eq, b_eq=rhs_eq, method='Simplex')
print(opt.x)
xx = opt.x
xp = np.zeros(m)
for i in range(0, m, 2):
    xp[i] = xx[i] / (xx[i] + xx[i + 1])
```

```
    xp[i + 1] = xx[i + 1] / (xx[i] + xx[i + 1])
print(xp)
```