Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики
Кафедра дослідження операцій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему

Оптимальне визначення найкращого об'єкта. Знаходження порогового рівня

Виконав студент 2-го курсу
Римар Олексій Сергійович

Науковий керівник:
професор, доктор фізико-математичних наук
Закусило Олег Каленикович

Робота заслухана на засіданні кафедри дослідження операцій та рекомендована до захисту в ЕК, протокол № 8 від 3 травня 2023р.

Завідувач кафедри ДО проф. Іксанов О.М.

Київ - 2023

ЗМІСТ

Вступ

Задача про секретаря або *Задача про перебірливу наречену* утворює важливий клас задач, які вивчає теорія ігор та теорія оптимальної зупинки, є модельним прикладом для дослідження оптимального прийняття рішень в умовах невизначеності та послідовного вибору.

Основна мета полягає в тому, щоб вибрати найкращого кандидата з набору, проте ми не знаємо який кандидат є найкращим доки не порівняємо його з іншими кандидатами. Важливою умовою є те, що після відхилення кандидата ми не можемо повернутися до нього, тобто наше рішення є остаточним.

Проблема секретаря має застосування у різних галузях, наприклад :

Найм та підбір персоналу: проблему можна застосовувати у випадках, коли компанія шукає кандидата на позицію і хоче вибрати найкращого кандидата з пулу заявок. Знання оптимальної стратегії допомагає визначити оптимальну кількість кандидатів для співбесіди перед прийняттям рішення.

Пошук квартири або будинку: при пошуку нового місця проживання принципи проблеми про секретаря можуть використовуватися для керування процесом прийняття рішень, а саме допомагають визначити, скільки варіантів потрібно розглянути перед прийняттям остаточного рішення про оренду або купівлю нерухомості.

Вибір постачальників або постачальників послуг:в бізнесовому закупівельному процесі можна застосовувати принципи задачі про секретаря при виборі постачальників. Компанії можуть мати потребу оцінити кілька варіантів і вирішити, коли припинити дослідження альтернатив та обрати найкращого.

Вибір проектів або інвестицій: принципи проблеми секретаря можуть застосовуватися до вибору проектів або прийняття рішень щодо процесу інвестування. Вони допомагають визначити, скільки потенційних проектів або можливостей для інвестування треба оцінити перед розподілом ресурсів на найкращий варіант.

Це лише кілька прикладів того, знання оптимальної стратегії вирішення проблеми про секретаря можуть застосовуватися у різних сферах прийняття рішень. Проблема підкреслює важливість прийняття оптимальних рішень при обмеженій інформації та невизначеності.

У практичному застосуванні можуть бути варіації задачі секретаря, залежно від конкретної ситуації. Наприклад, можуть бути додані фактори, такі як, выдсутнысть пам'яті про кандидатів, обмеження часу їх існування, невідома кількість кандидатів та інші.

Основні результати

В данній дипломній роботі розглядаються модифіковані версії цієї задачі, в яких об'єкти вважаються смертними. Така постановка задачі має більше прикладне значення ніж класична, тому що як і в реальному житті кандидати, до яких ще не дійшла черга, можуть бути недоступні або кандидат, якого ми вважали до цього кращим, міг зникнути після його перегляду.

Необхідно дослідити зміни в постановці задачі, а саме : як будуть виглядати перехідні ймовірності ланцюга Маркова, функція виграшу, ціна гри та опорна множина. Також оцінити, що відбувається з оптимальним моментом зупинки(першим потраплянням в опорну множину) шляхом експериментів.

Уявлення про математику як суху та теоретичну науку розширюється завдяки ролі експериментів у сучасній математиці. Вони допомагають відкривати нові ідеї, розробляти гіпотези та навіть доводити теореми. Створення моделей та проведення чисельних експериментів можуть підказати математикам потенційні шаблони, тенденції чи закономірності, які не завжди очевидні. Математичні гіпотези часто формулюються на основі спостережень, інтуїції та експериментальних даних. Використовуючи обчислювальні методи. Математики можуть перевірити їх на великих обсягах даних або застосувати статистичні методи для їхнього підтвердження або спростування.

Задача про 4 фарби є хорошим прикладом використання експериментів для вирішення математичних проблем. Твердження проблеми, яке було сформульоване ще в 1878р., досить просте: будь-яку карту на прощині чи сфері можна розфарбувати не більше ніж чотирма різними кольорами (фарбами) так, щоб будь-які дві області зі спільною границею мали різний колір. при цьому на самі області накладається умова зв'язності та неточковості границі.

Вирішена задача була тільки через століття, а саме у 1976р. Кен Аппел та Вольфган Хакен із Університету Іллінoіса перші використали підхід, який опирався на комп'ютерні обрахунки для вирішення складних математичних задач. Для вирішення цієї проблеми була побудована програма, яка перевіряла цю властивість на певному наборі карт, жодна з яких не включала в себе карту меншого розміра, яка могла би послужити в якості контрприкладу. Математики дійшли до висновку, що не існує найменшого контрприкладу, бо тоді він би знаходився серед карт, тому контрприкладу не існує взагалі.

Розрахунки, які проводила програма неможливо перевірити вручну, але все ж результат у вигляді висновків з експерименту був отриманий, і зрештою прийнятий.

Для цілей цієї роботи, була побудована програма, яка надає результати на основі яких ми зможемо робити висновки про характер змін в постановці задачі.

Результати подаються у вигляді графіків, таблиць, припущень та самої програми обрахунку.

Огляд відомих результатів для класичної постановки задачі найкращого вибору

Припустимо, у нас є набір з n кандидатів, яких ми розглядаємо для прийняття на роботу як секретаря. Ці кандидати проходять співбесіди послідовно, один за одним, випадковим чином. Після кожної співбесіди нам потрібно відразу прийняти рішення про те, чи прийняти кандидата або почати знайомство з наступним.

Основна мета полягає в тому, щоб вибрати найкращого кандидата з набору. Проте, ми не знаємо, який кандидат є найкращим, доки не порівняємо його з іншими кандидатами. Важливою умовою є те, що після відхилення кандидата ми не можемо повернутися до нього, тобто наше рішення є остаточним.

В [1] було наведене наступне означення ціни гри $v(x)$: ціна гри v - найменша зі всіх ексцесивних функцій, більших або рівних функції f. Також доведене існування такої мажоранти для всіх функцій f.

Також наведені умови яким повинна задовольняти ціна гри, а саме :

$$\left.\begin{aligned} v(x) &\geq \sum_{y\in E} p(x,y)v(x) \\ v(x) &\geq f(x) \\ v(x) &\geq 0 \end{aligned}\right\} (x \in E)$$

, де $f(x)$ - функція виграшу.

Далі в [1] вводиться поняття *опорної множини* Γ, як множини станів x, в яких функція виграшу $f(x)$ дорівнює свої ексцесивній мажоранті $v(x)$. Таккож робиться припущення, що оптимальною стратегією в данних умовах буде зупинитися в момент τ першого потрапляння в опорну множину Γ.

Як відомо, класична задача зводиться до оптимальної зупинки ланцюга Маркова зі станами $1, 2, ..., n$, функцією виграшу $f(x) = \frac{k}{n}$ та перехідними ймовірностями

$$p(k,l) = \begin{cases} \frac{k}{l(l-1)}, & \text{при } l > k \\ 0, & \text{при } l \leq k \end{cases}$$

Опорна множина:

$$\Gamma = \{k : f(k) = v(k)\}$$

.

Тоді для класичної постановки задачі умови для $v(k)$ ціна гри набудуть вигляду :

$$v(k) \geq \tfrac{k}{n},$$

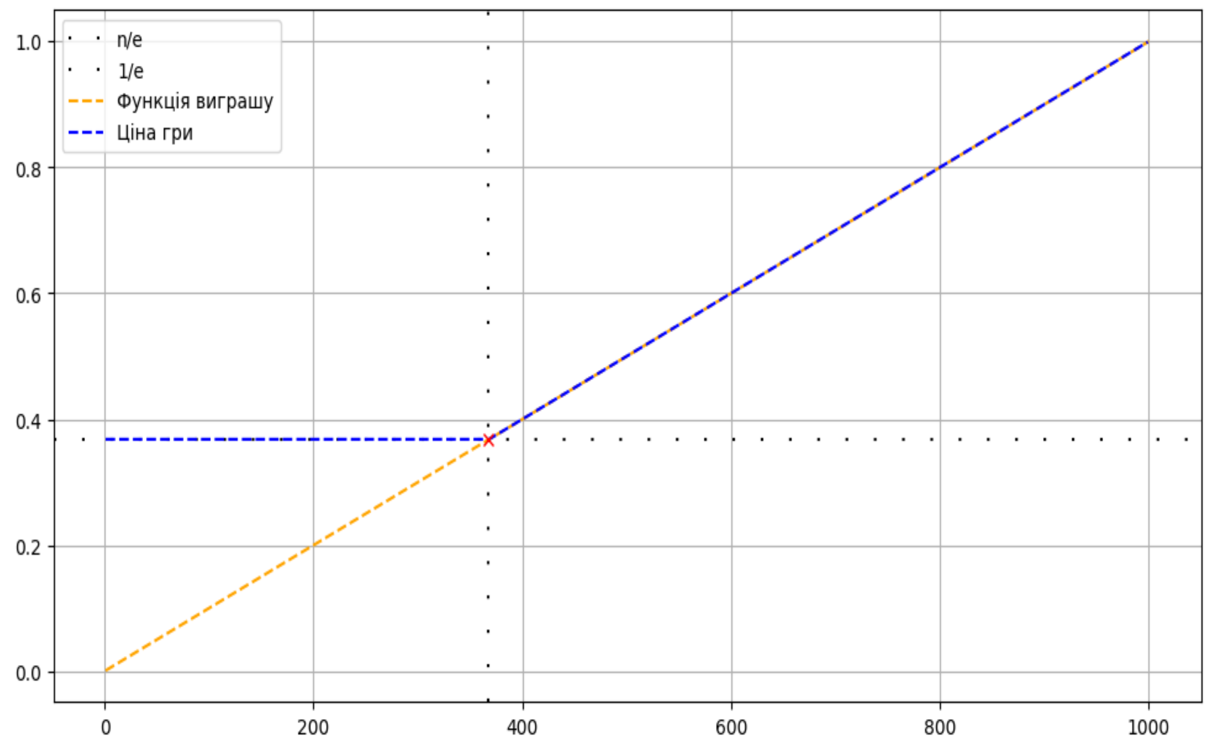
$$v(k) \geq \sum_{l=k+1}^{n} \frac{k}{l(l-1)} v(l)$$

$k = 1, 2, ..., n$. Тоді для $v(l)$ для $l > k$ вже відомо, то

$$v(k) = max\{\frac{k}{n}, \sum_{l=k+1}^{n} \frac{k}{l(l-1)} v(l)\}$$

Оптимальна стратегія для класичної задачі відома - при великих n потрібно пропустити приблизно перших $\frac{n}{e}$ претендентів (так зване правило 37 відсотків), а потім зупинитися на першому елементі, який кращому за найкращий з елементів з набору перших $\frac{n}{e}$.

Зокрема в [2] наведені розрахунки за схожою, до тої яка була описана вище, схемою.



Модифікація класичної задачі зі смертними об'єктами

Познайомимося ближче з першою модифікацією класичної задачі, а саме, коли об'єкти смертні. У цій модифікованій задачі, як і раніше, у нас є набір з n кандидатів. Ми розглядаємо їх послідовно, один за одним, і повинні вирішити, чи прийняти об'єкт як кращий, або продовжити з пошук найкращого. Проте, на відміну від класичної версії, в цій модифікованій задачі кожен кандидат може зникнути з початкового набору кандидитів до того як черга дійде до нього.

Відлік життя кожного з n кандидатів починається незалежно від інших в початковий момент часу. На кожному кроці ймовірність того , що елемент зникне складає α, тобто тривалість життя кожного об'єкта є випадковою величиною з геометричним розподілом з параметром $1 - \alpha$.

Необходно розглянути зміни, які відбулись в побудові опорної множини, функції виграшу, ціни гри та ймовірностей переходу ланцюга Маркова при переході до нових умов.

Почнемо з перехідних ймовірностей. Врахувавши те, що аби перейти на новий елемент потрібно також взяти до уваги факт його доступності, ймовірності переходу набудуть такого вигляду :

$$p(k,l) = \frac{k}{l(l-1)}\alpha^{l-1}, k \geq 0, k < l \leq n, \alpha \in (0,1).$$

Опорна множина, функція виграшу та ціна гри не зазнають змін у нових умовах.

Проведемо експерименти, для знаходження порогового рівня. Будемо відслідковувати момент першого потрапляння до опорної множини. Результати будуть наведені в якості графіків, на яких буде побудована ціна гри та варіюватися параметри кількості претендентів та ймовірності їх зникнення. Також побудована більш детальна таблиця з більш гнучкими значеннями параметрів.

Перейдемо до результатів :

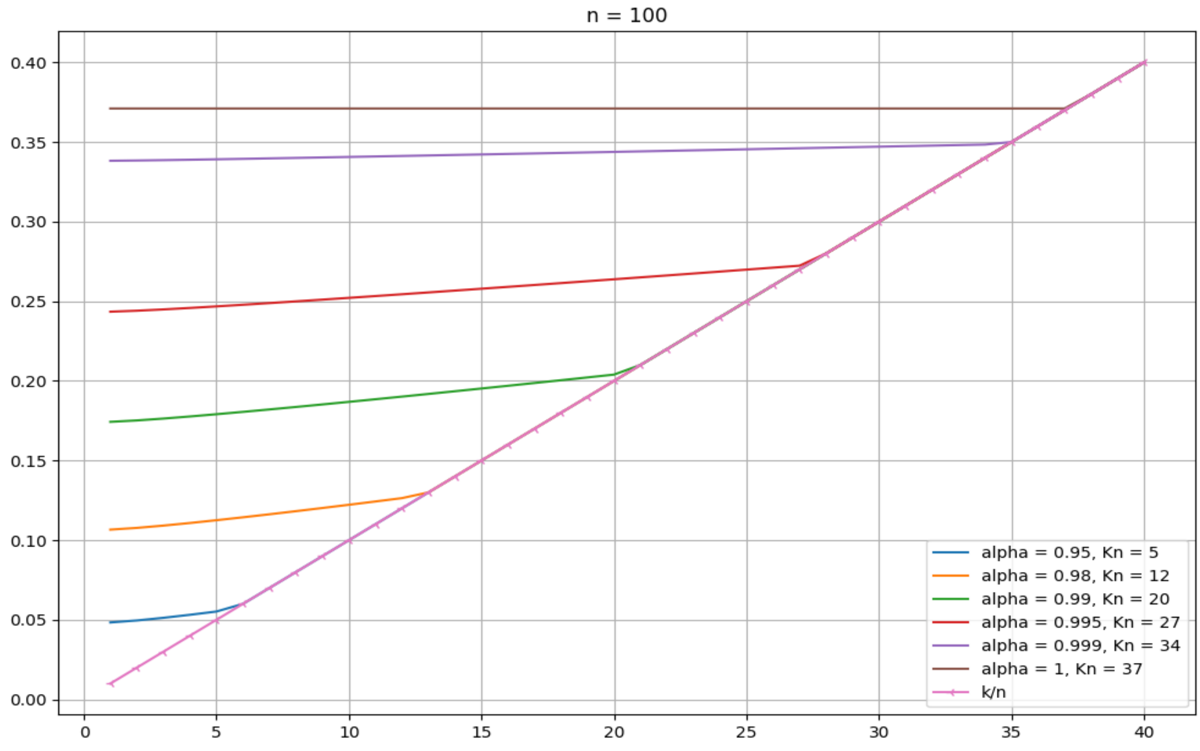


Рис.1 Графік ціни гри для n = 100 об'єктів

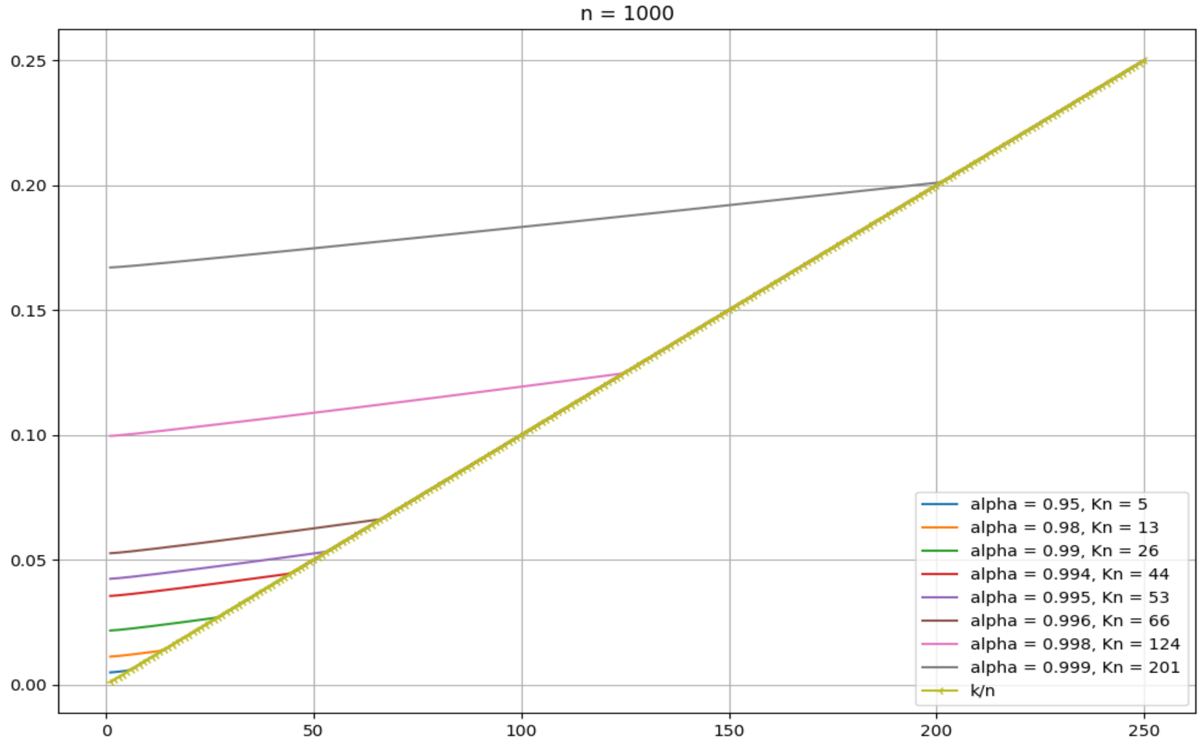


Рис.2 Графік ціни гри для n = 1000 об'єктів

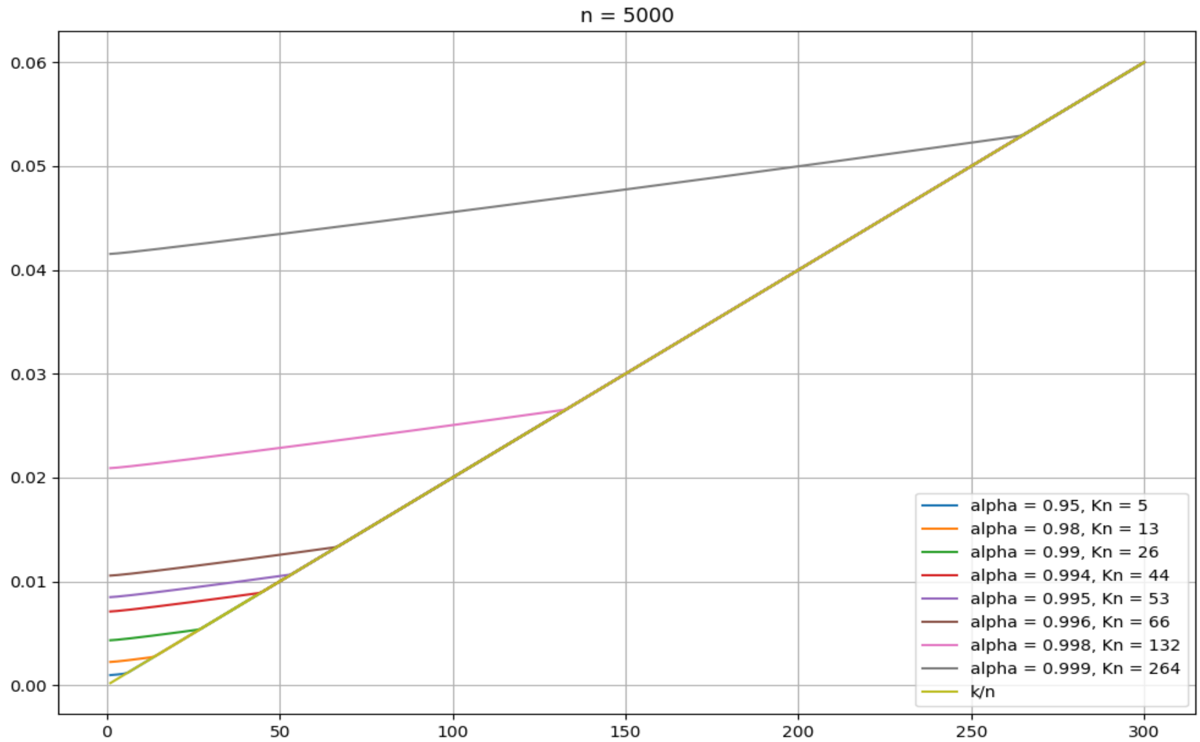


Рис.3 Графік ціни гри для n = 5000 об'єктів

На Рис.1, Рис.2 та Рис.3 представлені результати для $n = 100, n = 1000, n = 5000$ відповідно. Порівнюючи Рис.1 та Рис.2, можна побачити, що для менших α перший момент потрапляння в опорну множину не змінився.

Далі, порівнюючи Рис.1, Рис.2 та Рис.3 можна помітити, що кількість значень α для яких число k_n не змініилося - зросло, що може наштовхувати на висновок, що при збільшенні кількості об'єктів, вплив числа n кількості об'єктів зменшується.

Глянемо на таблицю в якій на перетині занчень α та n стоїть момент першого потрапляння в опорну множину. Сама таблиця знаходиться в додатку.

Зі значень першого потрапляння в опорну множину, наведених в таблицях з додатку для першої модифікації, бачимо, що теорія про те, що при збільшенні кількості об'єктів, вплив числа n кількості об'єктів зменшується - справджується.

Розглянемо також графіки, які показують зміну моменту зупинки від n для сталих α

Та графік першого потрапляння в опорну множину від α для сталих n

Цей графік демонстурє, що при менших α для будь-яких n пороговий рівень майже не залежить від n. При $\alpha \to 1, k_n \to \frac{n}{e} \to \infty$.

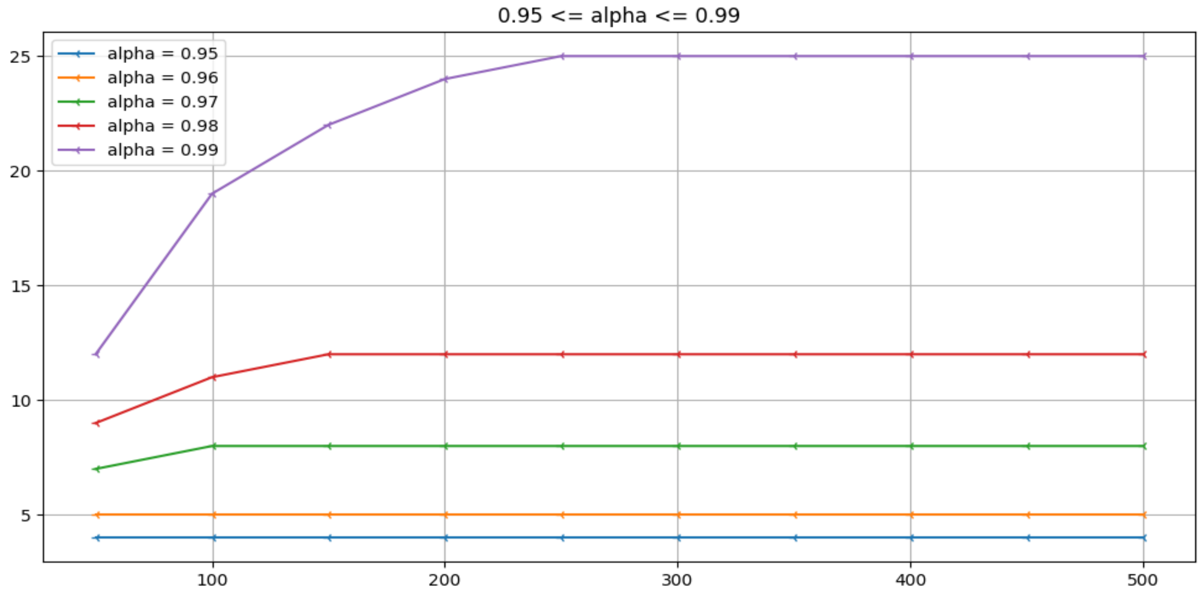


Рис.4 Перше потрапляння в опорну множину від n для сталих $0.95 \leq \alpha \leq 0.99$

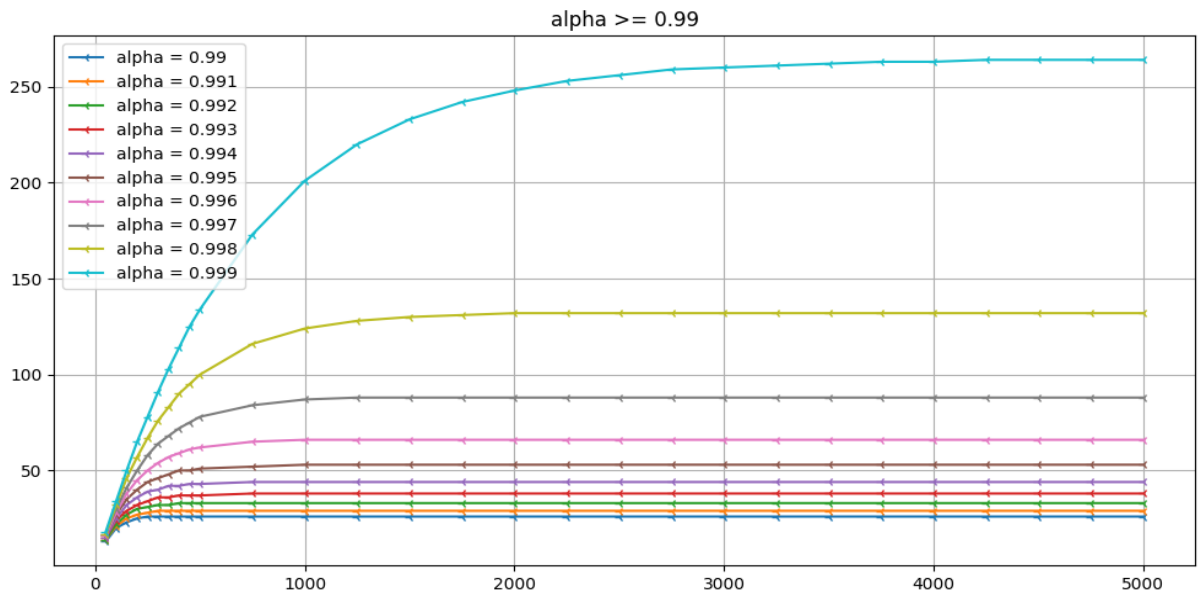


Рис.5 Перше потрапляння в опорну множину від n для сталих $\alpha \geq 0.99$

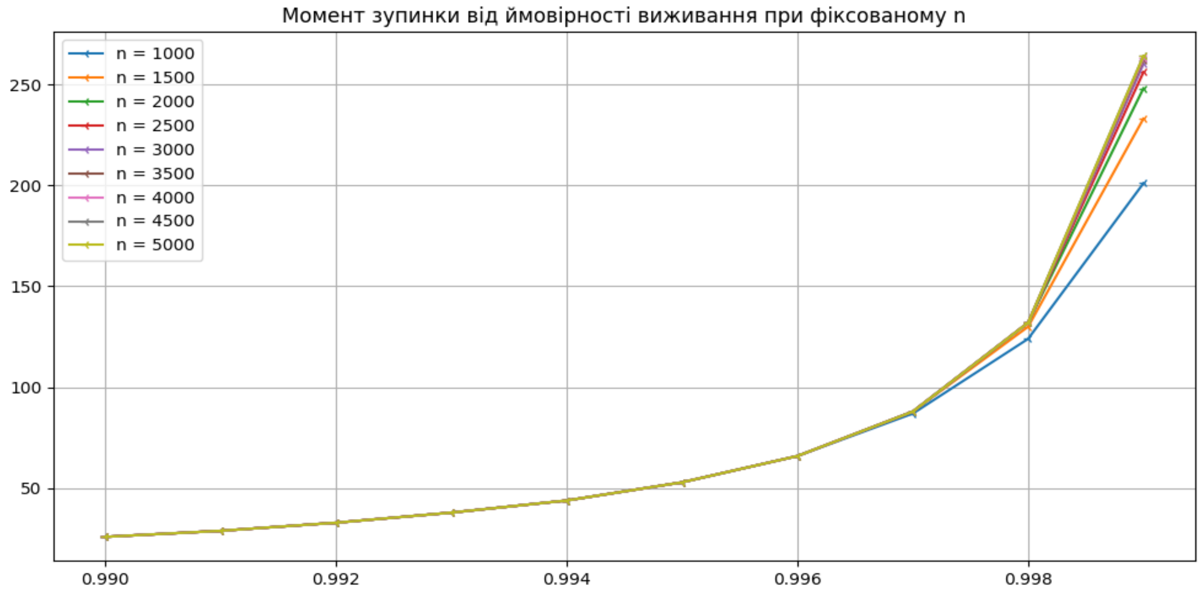


Рис.6 Перше потрапляння в опорну множину від α для сталих n

Модифікація класичної задачі зі смертними об'єктами та поверненням

Перейдемо до другої модифікації класичної задачі, а саме, тепер дозволимо повертатися до об'єкта, який вважається найкращим на певному кроці. Оскільки в таких умовах оптимальна стратегія дуже проста, а саме : переглянути всі об'єкти, запам'ятати найкращий та повернутися до нього, накладемо додаткову умову, що об'єкти є сметрними і відлік їх життя починається з моменту їх перегляду. В цій задачі, як і раніше, у нас є набір з n кандидатів. Ми розглядаємо їх послідовно, один за одним, і повинні вирішити, чи прийняти той об'єкт який є кращим в певний момент часу, або продовжити пошук найкращого. Оскільки об'єкти смертні, потрібно зважати на те, що зафіксувавши об'єкт як "найкращий", що чим далі від нього ми пішли тим менша ймовірність того, що він ще доступний.

Як вже було згадано, відлік життя кожного з n кандидатів починається після того як об'єкт потрапив на "співбесіду". На кожному кроці ймовірність того, що елемент зникне складає α, тобто тривалість життя кожного об'єкта є випадковою величиною з геометричним розподілом з параметром $1 - \alpha$.

Як і раніше, задача може бути зведена до пошуку оптимальної зупинки лінцюга Маркова. Сам ланцюг описаний в [3] та має вигляд $(x(k) = (N(k), B(k)), k \geq 0)$, де $N(k)$ - кількість об'єктів переглянутих на момент к, $B(k)$ - номер об'єкта, який є найкращим серед переглянутих на момент k. Також в Лемі 1 в [3] доводиться, що послідовність $(x(k), k \geq 0)$ - однорідний ланцюг Маркова з множиною станів $((u, j), 1 \leq u \leq n, 1 \leq j \leq u)$, в якій стани (n, j) є поглинаючими. Крок з непоглинаючих станів відбувається за схемою $(u, j) \to (u+1, j)$ або $(u, j) \to (u+1, j+1)$. Перехідні ймовірності :

$$p((u, j), (u+1, j)) = \frac{u}{u+1}$$
$$p((u, j), (u+1, j+1)) = \frac{1}{u+1}$$

Моментом оптимальної зупинки, як і раніше, може служити момент першого потрапляння в опорну множину, яка матиме вигляд :

$$\Gamma = \{(u, j) : v(u, j) = f(u, j)\}$$

, де $f(u, j)$ - функція виграшу, а $v(u, j) = max\{f(u, j), Pv(u, j)\}$ - Ціна гри. В нових умовах для функції виграшу потрібно брати до уваги ймовірність того, що об'єкт до якого ми пповернемося в разі зупинки, може вже не існувати, тому $f(u, j) = \alpha^{u-j} \frac{u}{n}$.

Необходно розглянути зміни, які відбулись в побудові опорної множини, функції виграшу, ціни гри та ймовірностей переходу ланцюга Маркова при переході до нових умов.

Почнемо з перехідних ймовірностей. Врахувавши те, що аби перейти на новий елемент потрібно також взяти до уваги факт його доступності, ймовірності переходу набудуть такого вигляду :

$$p(k,l) = \frac{k}{l(l-1)}\alpha^{l-1}, k \geq 0, k < l \leq n, \alpha \in (0,1).$$

Опорна множина, функція виграшу та ціна гри не зазнають змін у нових умовах.

Проведемо експерименти, для знаходження порогового рівня. Будемо відслідковувати момент першого потрапляння до опорної множини. Результати будуть наведені в якості графіків, на яких буде побудована ціна гри та варіюватися параметри кількості претендентів та ймовірності їх зникнення. Також побудована більш детальна таблиця з більш гнучкими значеннями параметрів.

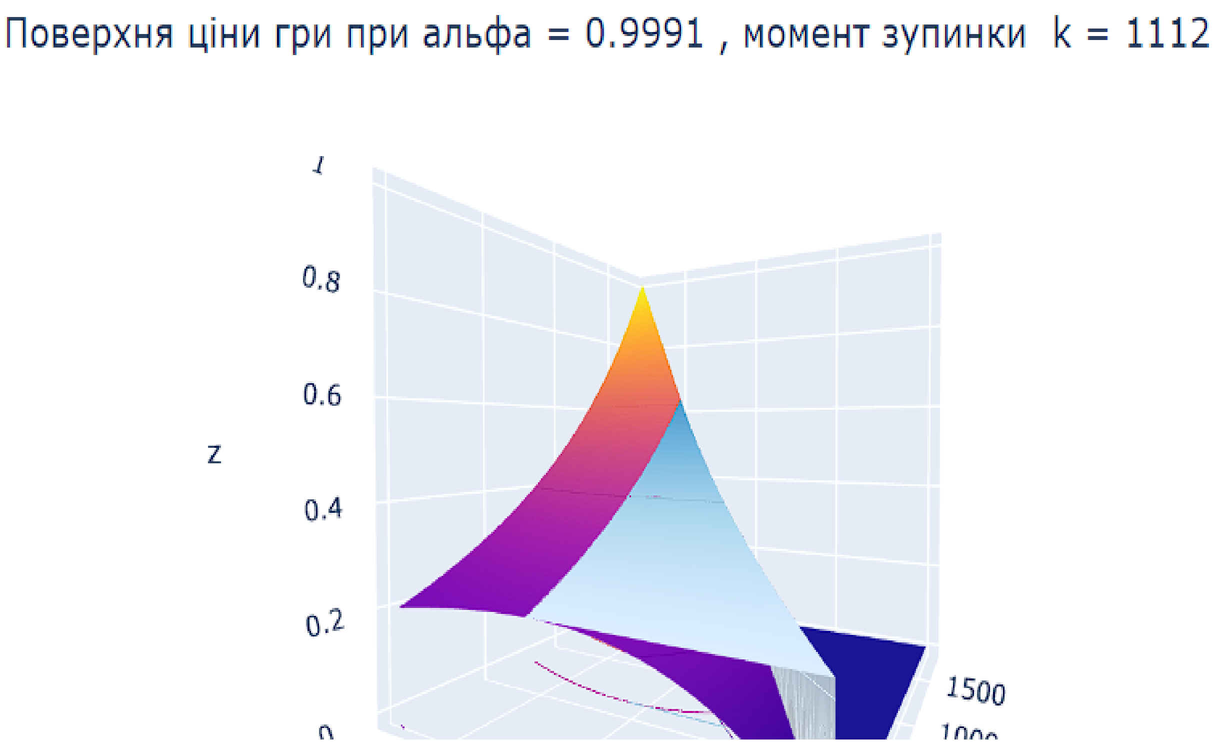


Рис.7 Графік ціни гри для n = 1800 об'єктів, $\alpha = 0.9991$

На Рис.5 та Рис.6 побудований для прикладу графік ціни

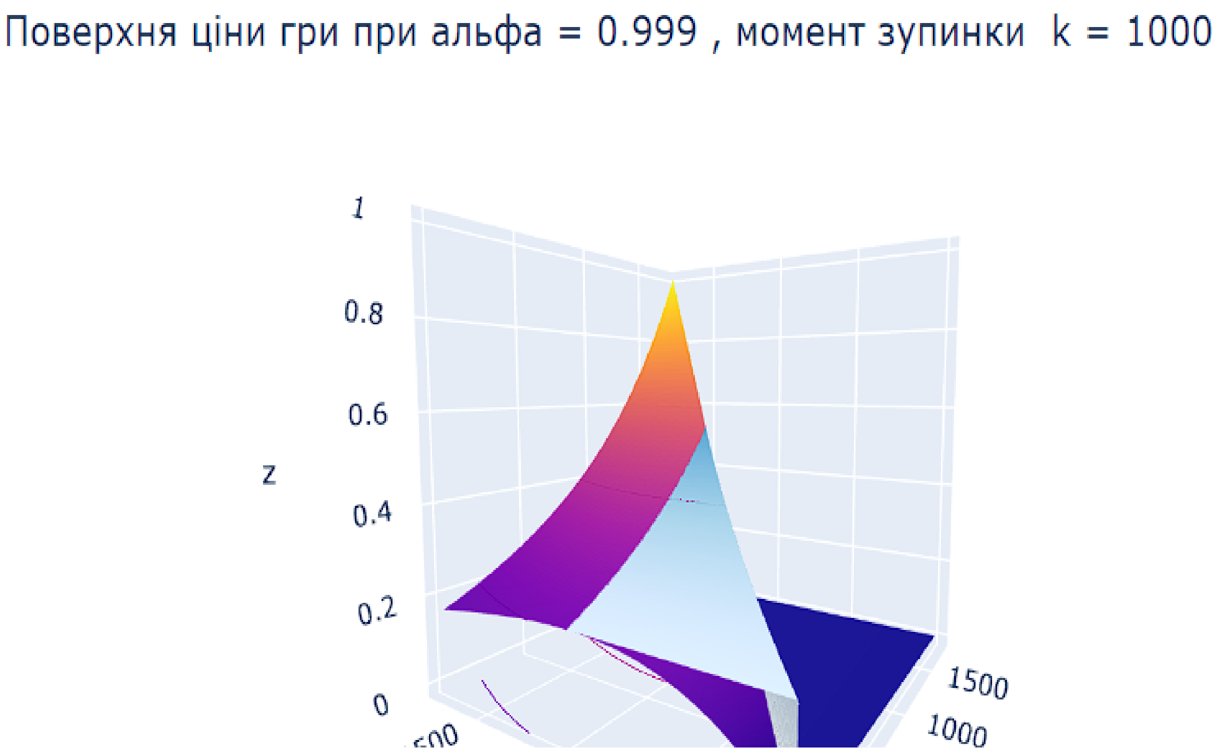


Рис.8 Графік ціни гри для n = 1800 об'єктів, $\alpha = 0.999$

гри, кольорова частина якого відповідає опорній множині при заданих параметрах.

Як і в минулій задачі виникає припущення, що при збільшенні n числа об'єктів, моменту зупинки від стає більш залежним від α ймовірності виживання ніж від n та майже не демонструє видимого збільшення.

Зі значень першого потрапляння в опорну множину, наведених в таблицях з додатку для другої модифікації, Рис.9 та Рис.10 можемо бачити, що теорія про те, що при збільшенні кількості об'єктів, вплив числа n кількості об'єктів зменшується - підтверджується. При $\alpha \to 1$ оптимальна стратегія буде наближатися до стратегії "пропустити всіх". При $\alpha \to 0$ Стратегія буде - зупинити прегляди як умога раніше.

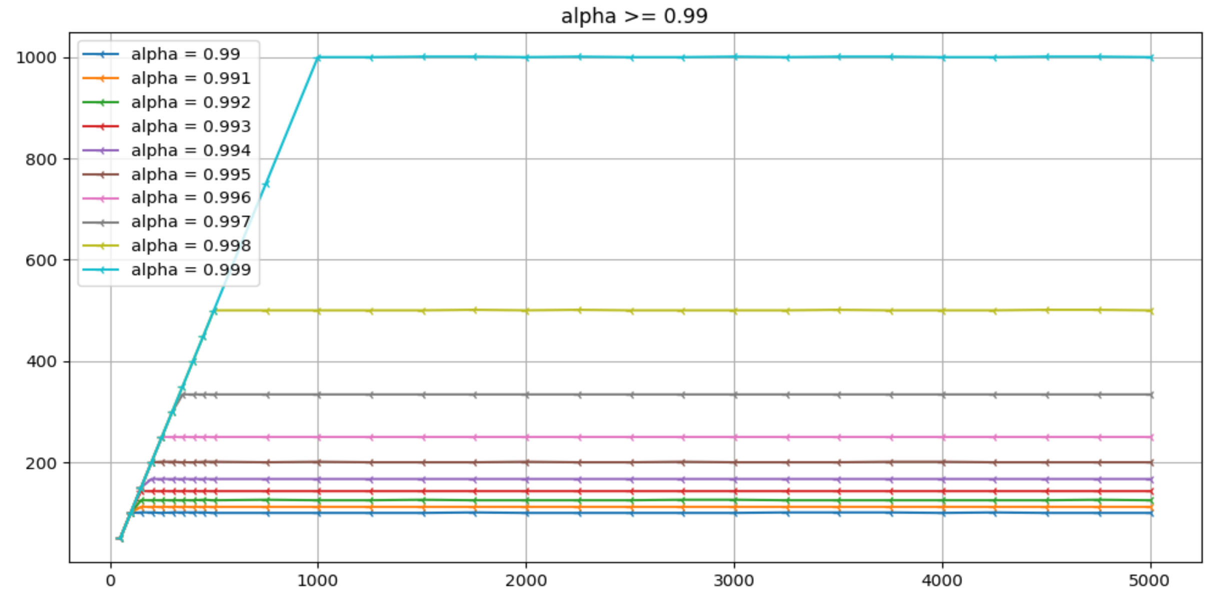


Рис.9 Перше потрапляння в опорну множину від n для сталих $0.95 \leq \alpha \leq 0.99$

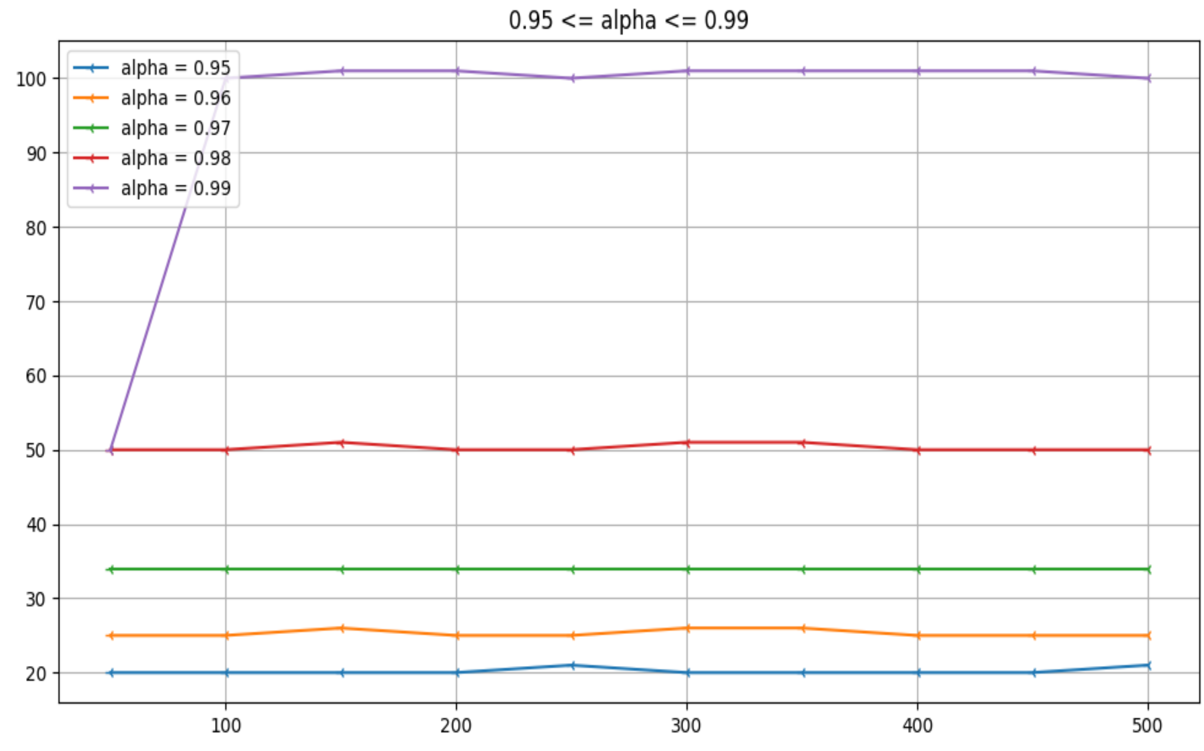


Рис.10 Перше потрапляння в опорну множину від n для сталих $\alpha \geq 0.99$

Висновки

В данній дипломній роботі розглядаються модифіковані версії задачі про секретаря (перебірливу наречену), а саме модифікації в яких об'єкти є смертними та відлік їх життя починається або з початку плину часу, або з моменту їх розгляду. Були проведені чисельні експерименти та висловлені припущення про характер залежності моменту зупинки від показників n та α кількості об'єктів та ймовірності виживання на кожному кроці відповідно. Як було видно в обох модифікаціяїх момент зупинки буде більш залежним від ймовірності виживання ніж від кількості об'єктів. Результати представлені у вигляді графіків, таблиці та реалізації обчислень.

Додаток 1. Модифікація перша

Таблиця значень для першої модифікації

alpha	50	100	150	200	250	300	350	400	450	500
0.10	1	1	1	1	1	1	1	1	1	1
0.20	1	1	1	1	1	1	1	1	1	1
0.30	1	1	1	1	1	1	1	1	1	1
0.40	1	1	1	1	1	1	1	1	1	1
0.50	1	1	1	1	1	1	1	1	1	1
0.60	1	1	1	1	1	1	1	1	1	1
0.70	1	1	1	1	1	1	1	1	1	1
0.80	1	1	1	1	1	1	1	1	1	1
0.90	2	2	2	2	2	2	2	2	2	2
0.91	3	3	3	3	3	3	3	3	3	3
0.92	3	3	3	3	3	3	3	3	3	3
0.93	4	4	4	4	4	4	4	4	4	4
0.94	4	4	4	4	4	4	4	4	4	4
0.95	5	5	5	5	5	5	5	5	5	5
0.96	6	6	6	6	6	6	6	6	6	6
0.97	8	9	9	9	9	9	9	9	9	9
0.98	10	12	13	13	13	13	13	13	13	13
0.99	13	20	23	25	26	26	26	26	26	26

alpha	50	100	150	200	250	300	350	400	450	500
0.990	13	20	23	25	26	26	26	26	26	26
0.991	14	21	25	27	28	29	29	29	29	29
0.992	14	22	27	30	31	32	32	33	33	33
0.993	15	24	29	32	34	36	36	37	37	37
0.994	15	25	32	36	39	40	42	42	43	43
0.995	16	27	35	40	44	46	48	50	50	51
0.996	16	28	38	45	50	54	57	59	61	62
0.997	17	30	41	50	58	64	68	72	75	78
0.998	17	32	46	57	67	76	83	90	95	100
0.999	18	34	50	65	78	91	103	114	125	134

alpha	250	500	750	1000	1250	1500	1750	2000	2250	2500
0.990	25	25	25	25	25	25	25	25	25	25
0.991	27	28	28	28	28	28	28	28	28	28
0.992	30	32	32	32	32	32	32	32	32	32
0.993	33	36	37	37	37	37	37	37	37	37
0.994	38	42	43	43	43	43	43	43	43	43
0.995	43	50	51	52	52	52	52	52	52	52
0.996	49	61	64	65	65	65	65	65	65	65
0.997	57	77	83	86	87	87	87	87	87	87
0.998	66	99	115	123	127	129	130	131	131	131
0.999	77	133	172	200	219	232	241	247	252	255

Додаток 2. Модифікація друга

Таблиця значень для другої модифікації

alpha	50	100	150	200	250	300	350	400	450	500
0.10	2	2	2	2	2	2	2	2	2	2
0.20	2	2	2	2	2	2	2	2	2	2
0.30	2	2	2	2	2	2	2	2	2	2
0.40	2	2	2	2	2	2	2	2	2	2
0.50	2	2	2	2	2	2	2	2	2	2
0.60	3	3	3	3	3	3	3	3	3	3
0.70	4	4	4	4	4	4	4	4	4	4
0.80	5	5	6	5	6	6	6	5	5	6
0.90	10	10	11	10	10	11	11	10	11	10
0.91	12	12	12	12	12	12	12	12	12	12
0.92	13	13	13	13	13	13	13	13	13	13
0.93	15	15	15	15	15	15	15	15	15	15
0.94	17	17	17	17	17	17	17	17	17	17
0.95	20	20	20	20	21	20	20	20	20	21
0.96	25	25	26	25	25	26	26	25	25	25
0.97	34	34	34	34	34	34	34	34	34	34
0.98	50	50	51	50	50	51	51	50	50	50
0.99	50	100	101	101	100	101	101	101	101	100

alpha	250	500	750	1000	1250	1500	1750	2000	2250	2500
0.990	100	100	100	100	100	100	101	100	100	100
0.991	112	112	112	112	112	112	112	112	112	112
0.992	125	125	126	125	125	126	125	125	125	125
0.993	143	143	143	143	143	143	143	143	143	143
0.994	167	167	167	167	167	167	167	167	167	167
0.995	201	201	200	201	200	200	200	201	200	200
0.996	250	250	250	250	250	250	250	250	250	250
0.997	250	334	334	334	334	334	334	334	334	334
0.998	250	500	500	500	500	500	501	500	501	500
0.999	250	500	750	1000	1000	1001	1001	1000	1001	1000

Додаток 3. Реалізація програми

```python
import numpy as np
import pandas as pd
import sys
import plotly.graph_objects as go



sys.setrecursionlimit(100000)
```

Реалізована програма на мові програмування Python.

```
class V_solver:
    '''
    Клас за допомогою якого проводилися всі обрахунки.
    Методами класу є функції які повертають значення в
    залежності від свого призначення
    '''
    def __init__(self, n, alpha):
        self.n = n
        # Число "претиндентів"
        self.alpha = alpha
        # Ймовірність виживання
        self.values = [np.NaN for i in range(n)]
        # Список значень v(k) першої модифыкацій
        self.matrix = np.zeros([n, n])
        # Матриця v(i,j) другої модифікації
        self.Kn = 0
        # Момент зупинки
        self.support_matrix = np.zeros([n, n])
        # Опорна множина другої модифікації

    1 usage
    def p_mod(self, k, m):
        '''
        Перехідні ймовірності класичної задачі
        '''
        return (k / (m * (m - 1))) * pow(self.alpha, m - 1) if m > k else 0
```

```
def p(self, k, m):
    '''
    Перехідні ймовірності модифікованої першої задачі
    '''
    return k / (m * (m - 1)) if m > k else 0


1 usage
def v_true(self, k):
    '''
    Ціна гри класичної задачі
    '''
    if k >= self.n:
        return k / self.n

    if self.values[k - 1] > 0:
        return self.values[k - 1]
    if sum([l for l in range(k, self.n)]) < 1:
        self.values[k - 1] = k / self.n
    else:
        p_val = sum([self.v_true(l) * self.p(k, l) for l in range(k + 1, self.n + 1)])
        self.values[k - 1] = max(k / self.n, p_val)
        if self.values[k - 1] == k / self.n:
            self.Kn = k
    return self.values[k - 1]
```

```python
def v_mod(self, k):
    '''
    Ціна гри модифікованої задачі
    '''
    if k >= self.n:
        return k / self.n

    if self.values[k - 1] > 0:
        return self.values[k - 1]
    p_val = sum([self.v_mod(l) * self.p_mod(k, l) for l in range(k + 1, self.n + 1)])
    self.values[k - 1] = max(k / self.n, p_val)
    if self.values[k - 1] == k / self.n:
        self.Kn = k
    return self.values[k - 1]
```

```
def v_mod_matrix(self):
    '''
    Ціна гри модифікованої задачі з поверненням
    '''
    for i in range(self.n - 1, -1, -1):
        if (i == self.n - 1) or (
                self.f(i + 1, 1) >= self.matrix[i + 1][0] * (i + 1) / (i + 2) + self.matrix[i + 1][1] / (i + 2)):
            for j in range(i, -1, -1):
                f_i_j = self.f(i + 1, j + 1)
                self.matrix[i][j] = f_i_j
                self.support_matrix[i][j] = f_i_j
                self.Kn = i+1
        else:
            for j in range(i, -1, -1):
                f_i_j = self.f(i + 1, j + 1)
                self.matrix[i][j] = max(f_i_j,
                                        self.matrix[i + 1][j] * (i + 1) / (i + 2) + self.matrix[i + 1][j + 1] / (
                                                i + 2))
                if self.matrix[i][j] == f_i_j:
                    self.support_matrix[i][j] = f_i_j
```

```
mat1 = V_solver(1800, 0.995)
mat1.v_mod_matrix()
data = [go.Surface(x=[x for x in range(1, mat1.n+1)], y=[y for y in range(1, mat1.n+1)], z=mat1.matrix,
                   colorscale='Blues'),
        go.Surface(x=[x for x in range(1, mat1.n+1)], y=[y for y in range(1, mat1.n+1)],
                   z=v_base_matrix(mat1.n, mat1.alpha))]


fig = go.Figure(data=data)
print(f'Kn = {min([i+1 if mat1.support_matrix[i][0] > 0 else 1000000 for i in range(mat1.n)])}')
fig.data[0].name = 'v(i,j)'
fig.data[1].name = 'f(i,j)'
fig.update_traces(contours_z=dict(show=True, usecolormap=True,
                                  highlightcolor="limegreen", project_z=True))

fig.update_layout(title=f'Поверхня ціни гри при альфа = {mat1.alpha} , момент зупинки  k = {mat1.Kn}', autosize=False,
                  scene_camera_eye=dict(x=1.87, y=0.88, z=-0.64),
                  width=800, height=400,
                  margin=dict(l=65, r=50, b=65, t=90))


fig.show()
```

Бібліоґрафія

[1] Е.Б. Дынкин, А.А. Юшкевич., *Теоремы и задачи о процессах Маркова.*. М., Наука, 1967. ст. 111-116

[2] Yang, M.C.K., *Recognizing the Maximum of a Random Sequence Based on Relative Rank with Backward Solicitation* on. J. Appl. Prob. 11, 1974, ст.504-512

[3] Zakusilo O.K., *Optimal choice of the best object with possible returning to previously observed* Theory of Stochastic Processes. Vol. 10(26), no.3–4, 2004, pp.142–149.