Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Кафедра обчислювальної математики

Кваліфікаційна робота на здобуття ступеня бакалавра

за спеціальністю 113 – «Прикладна математика» з теми:

Моделювання процесу розподілення потоків ресурсів

Виконав студент 4-го курсу ОС «Бакалавр»
Тітімець Артем Олександрович ____________

Науковий керівник:

кандидат фіз.-мат. наук, асистент
Оноцький В’ячеслав Валерійович ____________

Засвідчую, що в цій роботі
немає запозичень з праць
інших авторів без
відповідних посилань.

Студент ____________

Роботу розглянуто й допущено
до захисту на засіданні кафедри
обчислювальної математики

«29» травня 2023 р.,

протокол № 8

Завідувач кафедри
проф. Сергій Ляшко ____________

Київ-2023

РЕФЕРАТ

Обсяг роботи 25 сторінок, 12 джерел посилання, 4 ілюстрації,1 таблиця,2 додатки.

Ключові слова:

ПОТІК ПАЛЬМА, МЕТОД РОЗШИРЕННЯ, РОЗШИРЕНА ЗАДАЧА, РОЗВ'ЯЗОК ЗАДАЧІ ОПТИМІЗАЦІЇ, МЕТОД МНОЖНИКІВ ЛАГРАНЖА, КРИТЕРІЙ ОПТИМАЛЬНОСТІ.

Об'єкт дослідження – задача оптимального розподілення ресурсів між потоками.

Мета роботи – дослідити метод розширення множини допустимих розв'язків як основоположний алгоритм у моделюванні процесу розподілення потоків ресурсів. Розглянути його узагальнення на різні класи задач.

Результати: проведено експерименти з реалізації алгоритмів методу розширення для лінійної та нелінійної задач розподілу ресурсів. Наведено ілюстрації їх виконання.

Робота складається зі вступу, теоретичної та практичної частин. Наприкінці зроблено висновки. У вступі описана важливість та актуальність теми, а також надана історична довідка про вчених, що її опановували. Власне теоретична частина поділяється на розділів та   підрозділів. Крім змістової частини, було введено додатки А та Б. Додаток А містить код, з допомогою якого отримано результати практичної частини, додаток Б – опорні таблиці для теоретичного блоку.

Зміст

Глосарій

Наведемо список основних визначень, на які будемо опиратися в роботі при аналізі різноманітних математичних явищ. Використаємо для цього джерела [1], [11] і [12]

Визначення 1: *Випадковою величиною* називають деяку величину, котра може набути заздалегідь невідомого (проте лише одного) числового значення, отримання якого є результатом випробувань або спостережень за явищами (наприклад, підкидання грального кубика).

Визначення 2: *Випадковим процесом* називають випадкову величину, яка залежить від часу.

Визначення 3: *Подією* називають фактичний наслідок проведення випробування, експерименту або спостереження.

Визначення 4: *Законом розподілу* випадкової величини називають відповідність між можливими значеннями випадкової величини та ймовірністю їх прийняття.

Визначення 5: Випадкові величини називаються *незалежними*, якщо їхні закони розподілу не залежать від набутих одна одною значень.

Пальмівський процес є прикладом випадкового процесу з розміщенням точок на прямій. Складається із послідовності незалежних випадкових величин, які відповідають моментам приходу подій в часовій послідовності.

Структура пальмівського процесу має гілкоподібну форму. Події або точки на прямій випадково розташовуються у вигляді гілок та можуть розповсюджуватися в різних напрямках, створюючи густіші або розріджені області точок на прямій.

Визначення 6: *Потоком однорідних подій* називають випадкову послідовність *подій, розташовану в порядку неспадання моментів часу.*

Визначення 7: *Ординарним потоком однорідних подій* називають випадковий процес, який описує послідовність незалежних подій, що виникають з однаковою частотою за певний інтервал часу.

Визначення 8: *Потоком Пальма* або *потоком з обмеженою післядією* називають ординарний потік однорідних подій, для якого проміжки часу *T1, T2…* між деякими подіями є незалежними випадковими величинами.

Вступ

Не використовуючи математичний апарат, сьогодні майже неможливо описати поведінку й перспективи будь-якої економічної системи або соціального явища. Навіть якщо деякий спосіб це зробити існує, то він не може бути пояснений з точки зору науки, а отже – не може бути прийнятий людством. Математика є інструментом, який реалізовує постановку, вивчення і розв’язання конкретних проблем, що виникають у виробництві, сільському господарстві, сфері послуг тощо.

Розв’язуючи такого роду проблеми, людина спрямовує зусилля на пошук такого розв’язку, який би найкраще задовольняв умову поставленої задачі. Такий розв’язок називають *оптимальним.* Оптимальний розв’язок, як правило, є найбільшим або найменшим значенням досліджуваної величини. Задача, метою якої є знаходження максимуму чи мінімуму (далі – екстремуму) деякої функції (тут – *цільової функції*) в обмеженій деякими рівняннями та/або нерівностями області скінченновимірного векторного простору, називається *задачею оптимізації.* [6]

Очевидно, що за тисячолітню історію розвитку математичної науки була досліджена, винайдена і реалізована значна кількість методів розв’язання задач оптимізації. В цій роботі буде наведений приклад реалізації методів оптимізації для розв’язку задачі *моделювання процесу розподілу потоків ресурсів.*

Чимало всесвітньо відомих вчених досліджували питання оптимізації в цілому та власне розподіл ресурсів. Якщо говорити про історію вивчення оптимізації як явища, то одними з перших були вивчені і описані *задачі лінійного програмування. Симплекс-метод*, який сьогодні є ключовим у цій галузі був запропонований двічі у різні часи. Першим це зробив Жан Фур’є 1820 року, а потім – Джордж Данціг 1947 року.

Відомий радянський вчений Леонід Канторович зробив значний внесок до створення *методу потенціалів* для розв’язування *транспортної задачі*, вперше сформульованої Френком Хітчкоком у 1941 році. Згодом дана задача про знаходження плану транспортування вантажу з деякої точки відправлення споживачеві отримає також ім'я *задача Монжа-Канторовича.*

Вже у середині п'ятдесятих років двадцятого століття видається велика кількість статей та праць з *нелінійного* (а саме – з *квадратичного програмування),* зокрема роботи Гаррі Марковіца. В цей же період для розв'язування задач нелінійного програмування було запропоновано *методи градієнтного спуску*. Власне, суть градієнтного спуску полягає у випадковому виборі точки, а далі у здійсненні розрахунку напрямку найшвидшого спадання цільової функції (з використанням градієнта функції в обраній точці). Подальше завдання полягає в обчисленні нових значень функції, здійснюючи рух в обраний бік.

Якщо ж говорити безпосередньо про питання розподілу ресурсів, то тут значну роль відіграє внесок Леоніда Гурвіца, який у 2007 році отримав Нобелівську премію за створення концепції механізму розподілу ресурсів. Однією з найважливіших робіт стосовно цієї теми вважаються написані Канторовичем 1939 року «Математичні методи організації та планування виробництва». Там саме і було розглянуто задачу оптимального розподілу ресурсів у виробничій системі. Дана праця дала поштовх до стрімкого розвитку оптимізації в цілому [11].

Серед численних практичних проблем, які виникають насамперед в економіці, однією з найпоширеніших є саме розподіл ресурсів, зокрема тих, які можна охарактеризувати стохастичними матеріальними та інформаційними потоками. В промисловості найчастіше це потоки деталей, сировини, замовлень товару та інші. Дуже часто ці потоки є випадковими не тільки в часі, а й в інших параметрах, наприклад об'ємі. Ці параметри можуть бути як дискретними, так і неперервними, а також багатовимірними. Виділимо конкретні проблеми розподілу ресурсів, з якими можна зіткнутися у повсякденності. Наприклад, відправка товару одного виробника у різні магазини або ж роздача запитів для водіїв однієї служби таксі. Звісно, таких прикладів можна навести і більше, але наведені вище є одними з найбільш поширених в наш час [1].

1.Метод розширення множини допустимих розв’язків

Для розв’язування задач розподілу ресурсів між паралельними об’єктами в [4] запропонований *метод розширення* множини допустимих значень роз в’язків, враховуючи можливість (але не обов’язковість) виродженості матриці обмежень. За таким підходом розв’язок вихідної задачі визначається переходом з точки, котра буде розв’язком допоміжної задачі з розширеною множиною допустимих значень. Розглянемо його. Для цього введемо деяку задачу розподілу виробництва. Нехай маємо скінченну кількість локацій, на яких можуть розгорнути свою діяльність виробники однотипної продукції. Також даний список покупців, що зацікавлені в цих товарах. Кількість виготовленого продукту для кожного виробника необмежена. Відомими є вартість розгортання виробництва на кожній з локацій та витрати на задоволення попиту всіх замовників окремо. Ці дані необхідні для того щоб розподілити виробників за локаціями і прив’язати покупців до виробника так, щоб мінімізувати витрати на виготовлення і транспортування готового товару.

Задля побудови математичної моделі описаної задачі, введемо такі позначення:

m – можлива кількість місць для розгортання виробництва, i – порядковий номер відповідного місця (виробника), $i \in I = (1, 2, \ldots, m)$;

n – кількість покупців товару, для кожного з яких заданий порядковий номер $j \in J = (1, 2, \ldots, n)$;

Величина $b_j = 1, \forall j \in J$ визначає попит покупця на запропонований продукт. Для спрощення вважається, що всі значення попиту рівні одиниці;

c_i – вартість розгортання виробництва на i-тому місці;

c_{ij} – вартість задоволення попиту j-того покупця i-тим виробником (місцем виробництва);

y_{ij} – частка виготовленого продукту, котру i-тий виробник доставляє до j-того покупця;

x_i – величина, що вказує, чи розгорнуте на i-тому місці виробництво. Вона дорівнює одиниці, коли виробництво розгорнуте, інакше – рівна нулю.

Математично задачу поставимо таким чином. Нехай дана цільова функція:

$$F(x,y)=\sum_{i=1}^{m}c_i x_i+\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}y_{ij}\to \min. \tag{1.1}$$

На неї накладається система обмежень виду:

$$\sum_{j=1}^{n}y_{ij}\le x_i,\, i=\overline{1,m}, \tag{1.2}$$

$$\sum_{i=1}^{m}y_{ij}=1,\ j=\overline{1,n}, \tag{1.3}$$

$$x=\{0,1\},\, y_{ij}\ge 0,\, i=\overline{1,m},\ j=\overline{1,n}. \tag{1.4}$$

При фіксації значень величини x_i запропонована задача оптимізації прийме вигляд:

$$F=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}y_{ij}\to \min, \tag{1.5}$$

$$\sum_{j=1}^{n}y_{ij}\le 1,\, i=\overline{1,m}, \tag{1.6}$$

$$\sum_{i=1}^{m}y_{ij}=1,\ j=\overline{1,n}, \tag{1.7}$$

$$y_{ij} \in [0,1], i = \overline{1,m},\ j = \overline{1,n}. \tag{1.8}$$

Введемо розширену задачу, котру можна отримати, відкинувши з вихідної задачі обмеження (1.6):

$$F = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} y_{ij} \to \min, \tag{1.9}$$

$$\sum_{i=1}^{m} y_{ij} = 1,\ j = \overline{1,n}, \tag{1.10}$$

$$y_{ij} \in [0,1], i = \overline{1,m},\ j = \overline{1,n}. \tag{1.11}$$

Встановімо зв'язок між множинами роз в'язків Y вихідної та Y^p розширеної задач відповідно. Виходячи з умов (1.5)-(1.8) та (1.9)-(1.11) очевидним є наступне *твердження*:

Твердження 1: Множина допустимих розв'язків вихідної задачі є підмножиною для множини розв'язків розширеної задачі, тобто $Y \subset Y^p$ Справедливість цього твердження випливає із структури досліджуваних задач.

Проте оптимальні розв'язки цих двох задач збігаються, якщо задовольняється таке *твердження*:

Твердження 2: Оптимальні розв'язки вихідної та розширеної задач збігаються тоді і лише тоді, коли:

1) Множини допустимих розв'язків обох задач еквівалентні;
2) Оптимальний розв'язок розширеної задачі належить множині Y , тобто $y^{p*} \in Y$.

Докладні обґрунтування цих тверджень можна дослідити в [8], [9],[10].

Якщо розв'язок розширеної задачі (1.9)-(1.11) не належатиме множині Y, тобто $y^{p*} \notin Y$, то $\sup F > F^p$. Тоді для пошуку оптимальних розв'язків вихідної задачі використовують, наприклад, такий підхід:

- скористатися розширеною задачею (1.9)-(1.11) для встановлення ефективних обмежень вихідної задачі. Це значно зменшить її розмір, а отже допоможе запобігти значній кількості помилок при обчисленнях.
- розв'язок допоміжної задачі y^p корисний для прямого переходу до розв'язку основної задачі. Для подальших потреб, введемо множину підзадач задачі (1.9)-(1.11) таким чином:

$$F_j = \sum_{i=1}^{m} c_{ij} y_{ij} \to \min \tag{1.12}$$

$$\sum_{i=1}^{m} y_{ij} = 1 \tag{1.13}$$

$$y_{ij} \in [0,1], i = \overline{1,m}, j = \overline{1,n}. \tag{1.14}$$

Суть методу розширення полягає в тому, що розв'язок задачі (1.5)-(1.8) отримується прямим переходом до його оптимального розв'язку з точки, котра є розв'язком розширеної задачі (1.9)-(1.11). Оскільки значення цільової функції розширеної задачі в оптимальній точці є нижньою границею можливих значень цільової функції задачі (1.5)-(1.8), будь-який перехід від розв'язку задачі (1.9)-(1.11) до іншої точки $y \in Y$ погіршить значення цільової функції. Тобто, він означатиме спуск від $-F^p$ до деякого іншого значення цільової функції.

Нова точка, отримана внаслідок такого спуску буде оптимальною, якщо рух в цьому напрямку призведе до найменшої зміни значення цільової функції порівняно з іншими напрямками. Враховуючи вищесказане, можна сформувати загальну схему розв'язку задачі розподілу об'єктів методом розширення:

1. Знайти розв'язок розширеної задачі (1.9)-(1.11).

2. Перевіряємо, чи узгоджується отриманий розв'язок із обмеженням (1.6). Якщо так, то він оптимальний. Інакше – переходимо до кроку 3.
3. Обираємо напрямок і крок спуску.
4. Переходимо до нового розв'язку.

Метод розширення, як і решта алгоритмів оптимізації, вимагає конкретного застосування, яке залежить від типу математичної моделі поданої задачі. В наступному пункті опишемо алгоритм розширення детальніше, а саме розглянемо послідовність дій, яку необхідно застосувати при спуску до нового розв'язку.

1.1 Розв'язування задачі розподілу виробництва методом розширення з урахуванням необхідності спуску

Припустимо, що знайдений розв'язок розширеної задачі не задовольняє всі обмеження основної задачі і слід знайти перехід до іншого розв'язку:

$$y = y^p + h$$

Тут крок спуску h приймає за значення одиницю, якщо змінна індексу відповідає початку спуску, -1, якщо вона відповідає кінцю спуску і нуль в решті випадків.[4]

Виходячи із пропозицій стосовно оптимального розв'язку неперервної лінійної задачі розподілу ресурсів, сформульованих та доведених в [8], можемо підтвердити таку *лему* про вибір напрямку спуску в дискретній задачі розподілу ресурсів:

Лема 1: Точка $y = y^p + hj^*kl$ є розв'язком вихидної задачі розподілу виробництва (1.5)-(1.8) тоді і лише тоді, коли параметри j^*, k, l визначаються з умови:

$$(j^*, k, l) = \max_{j \in J_i} \left\{ \frac{\left[\min\limits_{(k,l) \in N_b^j} \left(c_{jl} - c_{jk}^p \right),\ j = \overline{1, n} \right]}{c_{j^*k}} \right\}. \qquad (1.1.1)$$

Тут J_i – множина індексів підзадач (1.12)-(1.14), чий розширений розв'язок потрібно змінити.

На основі загальної структури розв'язку та наведеної леми можемо сформулювати покроковий алгоритм розв'язування вихідної задачі (1.5)-(1.8):

Крок 1: Знайти розв'язки розширеної задачі (1.9)-(1.11).

Крок 2: Перевірити, чи задовольняє знайдений розв'язок обмеження (1.6). Якщо так, то перейти до *кроку 8.*

Крок 3: Сформувати множину індексів обмежень (1.6), котрі були порушені у розв'язку розширеної задачі за формулою:

$$I_i = \left\{ k \mid s_i = \sum_{j=1}^{n} y_{ij} > 1, i = \overline{1, m} \right\}.$$

Крок 4: Знайти можливі напрямки спуску з умови:

$$N_b = \bigcup_{j=1}^{n} N_b^j, N_b^j = \left\{ (k,l) \mid \min\left[c_{jl} - c_{jk}^p \mid s_i = 0 \right] \right\}.$$

Крок 5: Визначити множину індексів підзадач (1.12)-(1.14), в межах якої здійснюватиметься перехід до нового розв'язку.

$$J_H = \left\{ j \mid \min(c_{jl} - c_{jk}^p) \right\}.$$

У випадку, коли ця умова визначає лише одну з підзадач (1.12)-(1.14), слід перейти до *кроку 7*.

Крок 6: Визначити індекс *j* підзадачі, який треба змінити в першу чергу для переходу до нового розв'язку. Виконати це з умови:

$$j^* = \max\left\{ \frac{c_{j^*l} - c_{j^*k}^p}{c_{j^*k}^p} \right\}.$$

Крок 7: Перейти до нового розв'язку *j*-тої підзадачі (1.12)-(1.14)

$$y_{ij} = y_{ij}^{p} + h, h = \begin{cases} -1, i = l \\ 1, i = k \\ 0, else \end{cases}.$$

Обчислити $s_i, i = \overline{1, m}$. Перейти до *кроку 2.*

Крок 8: Обчислити значення цільової функції *F*.

2.Узагальнення методу розширення для інших класів задач розподілу ресурсів

В подальшому наведений вище метод розширення буде узагальнений для інакшого класу задач розподілу ресурсів, котрі будуть формуватися у виді випадкових потоків зі стохастичними об’ємами ресурсів.

Метою наступних досліджень буде побудова імітаційної моделі й алгоритмів процесу надходження ресурсів та оптимального їх розподілу з урахуванням пріоритетів, що накладаються економічними або технологічними умовами.

Нехай маємо деякий ресурс *s* із множини S, який може бути охарактеризований як визначеними, так і випадковими параметрами, для яких закони розподілу задані. Ресурс *s* в моменти часу розподіляється між *m* паралельними об’єктами. В [1] наведений рисунок (рис.1) що ілюструє процес потоку ресурсів:

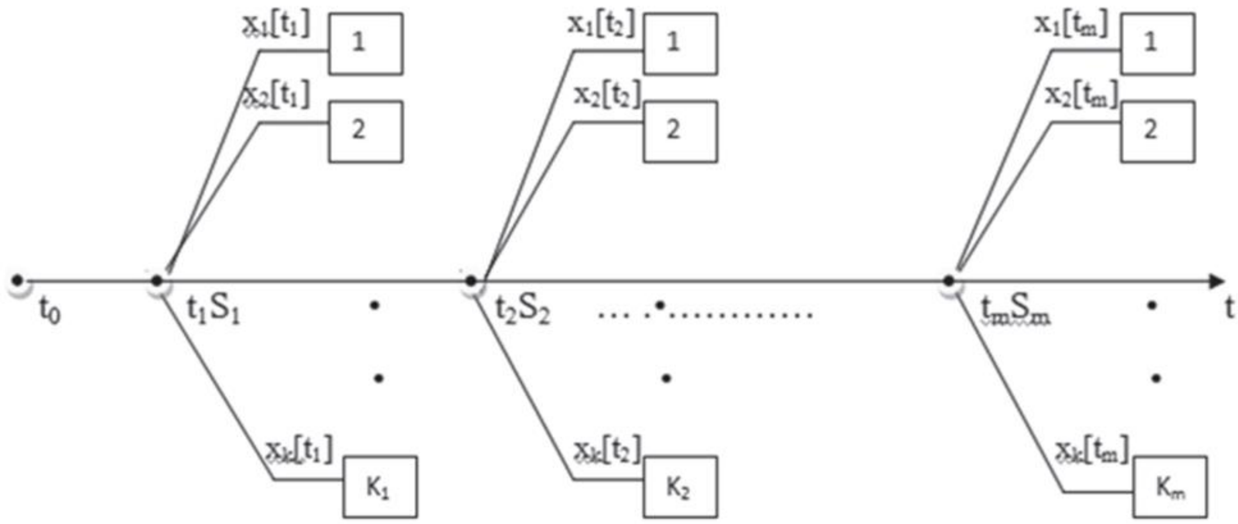

Рис.1

Якщо звернути увагу на сучасні економічні відносини, то зрозуміло, що потоки надходжень ресурсів (матеріалу, сировини чи замовлень) є формалізованими у вигляді *пальмівських процесів* надходжень із великою кількістю випадкових закономірностей, які характеризують об’ємні показники ресурсів. При їх розподілі виникає низка проблем не тільки з економічними пріоритетами, а й з математичною реалізацією цих розподілів. Система моделювання і розподілення потоків ресурсів між паралельними об'єктами складається з таких етапів:

- моделювання пальмівських процесів надходження ресурсів;
- моделювання і оптимізація розподілу об'ємів ресурсів для кожного моменту надходження;
- аналіз результатів і внесення необхідних змін у вихідні дані імітаційної моделі, допоки результати не стануть бажаними.

2.1 Потоки Пальма та моделювання пальмівських процесів надходжень ресурсів.

З терміном «потік Пальма», а також з рядом суміжних математичних визначень можна ознайомитись в *глосарії*. При моделюванні потоку ресурсів використовуватимемо саме властивості потоків Пальма, тому важливо розуміти їх природу. Як приклад потоку Пальма наведемо рух колони автівок. Нехай усі автомобілі у колоні рухаються із однаковою швидкістю і намагаються триматися на заданій відстані від автомобіля, котрий прямує попереду. Оскільки багатьма випадковими чинниками тут знехтувати не можна, ця задана відстань фактично втримується неточно. Відповідно, моменти часу перетину кожним автомобілем встановленої межі *T1, T2…* будуть незалежними випадковими величинами і утворюватимуть потік Пальма.

Часто потік Пальма є вихідним потоком систем масового обслуговування. Це сформульовано у відповідній *теоремі:[12]*

Теорема 1 *(Пальма):* Нехай до деякої системи масового обслуговування надходить потік заявок типу Пальма. Заявка, для якої усі канали будуть зайнятими, отримує відмову (не обслуговується). Якщо при цьому час обслуговування має експоненційний закон розподілу, то потік заявок, що не були обслужені теж є потоком типу Пальма.

На практиці заявки, що отримують відмову отримують перенаправлення до іншої системи масового обслуговування, тобто утворюють для цієї системи вхідний потік. Саме тому ця теорема є такою важливою.

Перейдімо безпосередньо до моделювання потоків ресурсів [1]. Нехай множина $T = \{t_j\}$ утворює стаціонарний (тобто, такий, що не змінюється з часом) потік Пальма із заданою щільністю $\varphi(\tau)$ інтервалів між елементами з початком у другому інтервалі потоку *T.* Аби визначити моменти надходжень t_j використовують стандартну формулу:

$$t_j = t_{j-1} + \tau_{j,}\, j = 1, 2..., n,$$

де τ_j – інтервали між елементами множини *T.*

Функції щільності $\varphi(\tau)$ недостатньо, щоб змоделювати потік Пальма, оскільки функція щільності першого інтервалу зазвичай відмінна від $\varphi(\tau)$, тобто $\varphi_1(\tau) \neq \varphi(\tau)$. Значення $\varphi_1(\tau)$ може бути обчислене за формулою:

$$\varphi_1(\tau_1) = \lambda\left(1 - \int_0^{\tau_1} \varphi(\tau)d\tau\right), \tag{2.1.1}$$

де λ – інтенсивність потоку, яка вказує, скільки надходжень відбувається за одиницю часу. Значення інтервалів τ_j між елементами ланцюга T можуть бути визначені методом оберненої функції моделювання неперервних випадкових величин, принцип якого можна сформулювати такою *теоремою:* [7]

Теорема 2: Для довільної функції розподілу $F(\tau)$ випадкова величина $\eta = F(\xi)$ буде розподіленою рівномірно на відрізку *[0,1].*

Доведення. Припустимо, що для деяких обраних випадкових чином чисел $\xi_1, \xi_2, ..., \xi_n$ задана функція розподілу

$$F(\tau) = P(\xi < \tau) \tag{2.1.2}$$

Виходячи із цього, сформуємо послідовність випадкових чисел, що матиме вигляд $\eta_1 = F(\xi_1), \eta_2 = F(\xi_2), ..., \eta_n = F(\xi_n)$. Усі значення $F(\xi)$ знаходяться в межах *[0,1]*, отже $\eta \in [0,1]$. З'ясуймо, який вигляд має функція $G(y) = P(\eta < y)$:

$$G(y) = P(\eta < y) = P(F(\xi) < y) = P\left(F^{-1}\left(F(\xi)\right) < F^{-1}(y)\right) = P\left(\xi < F^{-1}(y)\right).$$

З урахуванням рівності (2.1.2) маємо:

$$G(y) = P\left(\xi < F^{-1}(y)\right) = F\left(F^{-1}(y)\right) = y.$$

Випадкова величина $\eta = F(\xi)$ рівномірно розподілена на відрізку *[0,1]*, бо її функція розподілу прийняла вигляд $G(y) = y$. *Теорему доведено.*

Отже, процедура моделювання неперервної випадкової величини, що має функцію розподілу $F(\tau)$ передбачає розв'язування рівняння виду $F(\tau) = z$. Тут z є випадковою величиною з рівномірним на інтервалі *[0,1)* розподілом. Звідси знаходимо $\tau = F^{-1}(z)$.

У випадках, коли щільність $\varphi(\tau)$ інтервалів між елементами потоку *T* є неперервною випадковою величиною та її закон розподілу є одним із стандартних, то при моделюванні інтервалів τ_j доцільно користуватися формулами, короткий перелік яких наведений в *Таблиці 1*(див. *додаток Б*). [1]

2.2 Розподіл ресурсів з урахуванням малих відмінностей в паралельних процесах

Нехай для довільного елемента множини $T = \{t_j\}$ задача розподілу ресурсів сформульована таким чином:[1]

- Необхідно максимізувати заданий показник системи

$$\max F(t_j) = f(x), \tag{2.2.1}$$

враховуючи систему обмежень:

- технологічних

$$g(x) \le s(t_j); \tag{2.2.2}$$

- обмежень рівноваги об'ємних показників ресурсів

$$Ex = s_m(t_j); \tag{2.2.3}$$

- обмежень деяким проміжком

$$V \le x \le W, \tag{2.2.4}$$

де *E* – одиничний *n*-вимірний вектор-рядок, *x* – *n*-вимірний вектор, *f(x), g(x)* – неперервно диференційовані векторні функції. Їх можна задати, наприклад, так:

$$f(x) = f_0(x) + \varepsilon f_1(x);$$

$$g(x) = g_0(x) + \varepsilon g_1(x),$$

де малий скалярний параметр ε вводиться, аби зробити функції $f_0(x), g_0(x)$ співвимірними (тобто, залежними від однакового набору змінних). Ці функції характеризують усереднені параметри паралельних об'єктів з функціями $f_1(x), g_1(x)$, що відповідають малим відмінностям цих об'єктів.

Відповідно до наведеного вище *методу розширення* допоміжна розширена задача може бути введена, відкидаючи із вихідної «незручні» обмеження (2.2.2). Тож, для розглядуваного випадку вона прийме вигляд:

$$\max F(t_j) = f(x), \tag{2.2.5}$$

$$Ex = s_m(t_j); \tag{2.2.6}$$

$$V \le x \le W, \tag{2.2.7}$$

За допомогою двох *тверджень*, наведених у *розділі 1* може бути встановлений зв'язок між множинами допустимих розв'язків відповідно вихідної (2.2.1)-(2.2.4) та розширеної (2.2.5)-(2.2.7) задач.

Загальна структура розв'язування задачі розподілу ресурсів між паралельними потоками буде схожою на алгоритм розв'язку задачі розподілу об'єктів методом розширення, про яку йшла мова у *попередньому розділі* [4]. Фактично послідовність дій буде ідентичною, тільки в даному випадку додається ще один пункт на початку. Тепер схема має такий вигляд:

1. Змоделювати елементи ланцюга *T* (див. *пункт 2.1*)
2. Розв'язати розширену задачу (2.2.5)-(2.2.7).
3. Перевірити отриманий розв'язок за обмеженням (2.2.2) вихідної задачі. Якщо перевірку пройдено, то кінець алгоритму. Інакше – перейти до наступного кроку.
4. Обрати напрям і крок спуску.
5. Перейти до нового розв'язку.

Розв'язок, який буде отриманий після спуску вважається оптимальним, якщо спуск за обраним напрямком стане причиною найменшої зміни значення цільової функції у порівнянні з іншими напрямками.

2.3. Алгоритм розподілу паралельних потоків із загальним ресурсом

В [1] і [2] детально пояснено алгоритм знаходження оптимального розподілу загального ресурсу, використовуючи лінійну задачу:

$$\max F(t_j) = cx, \tag{2.3.1}$$

$$\left(A_0 + \varepsilon A_1\right) x \le s(t_j); \tag{2.3.2}$$

$$Ex = s_m(t_j); \tag{2.3.3}$$

$$V \le x \le W, \tag{2.3.4}$$

де c – n-вимірний вектор-рядок. Матриці A_0, A_1 тут є відповідниками функцій $g_0(x), g_1(x)$, про які йшлося у *попередньому пункті.*

Як відомо з [3] розв'язок такої задачі, отриманий методами лінійного програмування є нестійким і неточним. Тому на основі загального, наведеного вище алгоритму, можемо застосувати метод розширення, виконавши послідовність кроків:

Крок 1: Ввести вхідні дані.

Крок 2: Використати формулу (2.1.1) для знаходження функції щільності $\varphi_1(\tau_1)$.

Крок 3: Методом оберненої функції отримати значення інтервалів τ_j між елементами ланцюга *T*.

Крок 4: Обчислити значення $\{t_j\}$ за формулою:

$$t_j = t_{j-1} + \tau_j,\ j = 1, 2\ldots, n.$$

Крок 5: Обчислити значення потоків розподіленого ресурсу $s(t_j), s_m(t_j)$

Крок 6: Знайти розв'язок задачі (2.3.1), (2.3.3), (2.3.4).

Крок 7: Перевірити отримане на оптимальність за умовою (2.3.2). Якщо перевірку не пройдено, то перейти до наступного кроку.

Крок 8: Обчислити прирости коефіцієнтів цільової функції, а також матриці $A = A_0 + \varepsilon A_1$:

$$\Delta c_{j,j+p} = c_j - c_{j+p},\ j = 1, 2, \ldots, n,\ p = 1, 2\ldots, n - j;$$

$$\Delta a^t_{j,j+p} = f_{t,j+p},\ j = 1, 2\ldots, n,\ p = 1, 2\ldots, n - j, t \in I_H.$$

Далі визначити можливі напрями спуску, представлені як об'єднання множин:

$$N_B = N_B^{dir} \cup N_B^{rev},$$

де

$$N_B^{dir} = \left(\left(j, j+p\right) | \Delta c_{j,j+p} \geq 0, \Delta a^t_{j,j+p} > 0 \forall t \in I_H\right);$$

$$N_B^{rev} = \left(\left(j+p, j\right) | \Delta c_{j,j+p} \geq 0, \Delta a^t_{j,j+p} < 0 \forall t \in I_H\right).$$

Тут *t* – номер обмеження (2.3.2), яке не відповідає розв'язку x^*;

I_H – множина обмежень (2.3.2), які не відповідають розв'язку x^*.

Якщо $N_B = 0$, то система обмежень вихідної задачі є несумісною (тобто, не має коренів).

Крок 8: Визначити напрямки спуску з умови

$$\alpha_t = \max_{t \in I_H} \min_{(j,j+p) \in N_B} \left\{ \frac{\Delta c_{j,j+p}}{\Delta a_{j,j+p}^t} \left(S_t^p - S_t \right) \right\}.$$

Крок 9: Обчислити значення кроку спуску

$$h_{kl} = \frac{S_t^p - S_t}{\Delta a_{kl}^t}.$$

Крок 10: Перевірити умову

$$h_{kl} \le \min \left\{ x_k^p - V_k, W_l - x_l^p \right\}.$$

Якщо умова виконується, то далі слід перейти до *кроку 12.* При значенні

$h_{kl} = 0$ не розглядати даний напрям і перейти до *кроку 8.*

Крок 11: Покласти

$$h_{kl} = \min \left\{ x_k^p - V_k, W_l - x_l^p \right\}.$$

Крок 12: За формулою

$$x_j = \begin{cases} x_j^p - h_{kl},\ j = k \\ x_j^p + h_{kl},\ j = l\ , \\ \quad x_j^p, else \end{cases}$$

віднайти новий розв’язок для задачі (2.3.1)-(2.3.4) та здійснити його перевірку на оптимальність.

2.4 Розподіл ресурсів з урахуванням малих відмінностей між паралельними процесами. Випадок нелінійності задачі

Продовжимо досліджувати метод розширення та його застосування. За приклад нелінійних задач візьмемо задачі, котрі мають квадратичну цільову функцію. Використаємо модель, роз’яснену в [1]:

$$\max F = \left(c\text{x} + \text{x}^T D\text{x}\right); \tag{2.4.1}$$

$$\left(A_0 + \varepsilon A_1\right) x \le S; \tag{2.4.2}$$

$$E\text{x} = S_m. \tag{2.4.3}$$

Оскільки суть задачі в максимізації цільової функції, то припускається, що матриця D є від'ємно визначеною (тобто, всі її власні числа є від'ємними). Розширена задача в цьому випадку має вигляд:

$$\max F^e = \left(c\text{x} + \text{x}^T D\text{x}\right); \tag{2.4.4}$$

$$\lambda\text{x} = S_m. \tag{2.4.5}$$

Дану розширену задачу без урахування обмеження (2.4.2) з матрицею близькою до виродженої можна розв'язати методом *множників Лагранжа.* Коротко нагадаймо його суть [11]. Нехай для функції багатьох змінних $F(x_1, x_2, ..., x_n)$ треба знайти найбільше або найменше значення. При цьому накладається s умов виду:

$$g_i\left(x_1, x_2, ..., x_n\right) = 0, i = 1, 2, ...s.$$

Спочатку будуємо *функцію Лагранжа* з s множниками λ_i:

$$L\left(x_1, x_2, ..., x_n, \lambda_1, \lambda_2, ..., \lambda_s\right) = F\left(x_1, x_2, ..., x_n\right) - \sum_{i=1}^{s} \lambda_i g_i\left(x_1, x_2, ..., x_n\right).$$

Оптимальним аргументом функції $F(x_1, x_2, ..., x_n)$ буде розв'язок системи рівнянь

$$\frac{\partial L\left(x_1, x_2, ..., x_n, \lambda_1, \lambda_2, ..., \lambda_s\right)}{\partial x_i} = 0, i = 1, 2..., n,$$

$$\frac{\partial L\left(x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2, \ldots, \lambda_s\right)}{\partial \lambda_j} = g_j\left(x_1, x_2, \ldots, x_n\right),\ j = 1, 2\ldots, s.$$

Рівнянь та змінних у системі налічується $n+s$ штук.

Якщо застосувати вищенаведений метод до розширеної задачі (2.4.4)-(2.4.5), то дійсно,

$$L\left(\mathrm{x}, \lambda\right) = F(\mathrm{x}^e) - \lambda\left(\sum x_i - S_m\right);$$

$$\frac{\partial L}{\partial x_i} = \frac{\partial F\left(\mathrm{x}^e\right)}{\partial x_i} - \lambda = 0; \qquad (2.4.6)$$

$$\frac{\partial L}{\partial \lambda} = \sum x_i - S_m = 0. \qquad (2.4.7)$$

Розв'язавши систему рівнянь (2.4.6)-(2.4.7) отримаємо x^e, λ .

Розкладаючи функцію цілі в ряд Тейлора в околі точки x^e, отримаємо:

$$F\left(\mathrm{x}\right) = F\left(\mathrm{x}^e + h\right) + \nabla F\left(\mathrm{x}^e\right)h + \frac{1}{2}h^T H h, \qquad (2.4.8)$$

де $\nabla F(\mathrm{x}^e)$ – оператор Гамільтона (векторний диференціальний оператор з частковими похідними функції $F(\mathrm{x}^e)$; H– матриця Гессе (деяка квадратна матриця, елементами для якої служать часткові похідні функції $F(\mathrm{x})$).

Для формули (2.4.8) окремо розглянемо другий і третій доданки:

$$\nabla F(\mathrm{x}^e)h = \left(\frac{\partial F}{\partial x_1},\ldots,\frac{\partial F}{\partial x_k},\ldots,\frac{\partial F}{\partial x_l},\ldots,\frac{\partial F}{\partial x_n}\right)\times\begin{pmatrix}0\\ \vdots\\ -h_{kl}\\ \vdots\\ h_{kl}\\ \vdots\\ 0\end{pmatrix} =$$

$$= 0+\cdots+\left(-\frac{\partial F\left(\mathrm{x}^e\right)}{\partial x_k}+\frac{\partial F\left(\mathrm{x}^e\right)}{\partial x_l}\right)\times h_{kl}+\cdots+0.$$

Крім цього, виходячи з логіки (2.4.6) можемо стверджувати, що

$$\frac{\partial F\left(\mathrm{x}^e\right)}{\partial x_k} = \frac{\partial F\left(\mathrm{x}^e\right)}{\partial x_l} = \lambda.$$

Тоді $\nabla F(\mathrm{x}^e)h = 0;$

$$\frac{1}{2}h^T H h = \frac{1}{2}\left(0,-h_{kl}\cdots,h_{kl}\cdots 0\right)\times$$

$$\times\begin{pmatrix} 2d_{11} & \cdots & d_{1k} & \cdots & d_{1l} & \cdots & d_{1n}\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ d_{k1} & \cdots & d_{kk} & \cdots & d_{kl} & \cdots & d_{kn}\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ d_{l1} & \cdots & d_{lk} & \cdots & d_{ll} & \cdots & d_{ll}\\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\ d_{n1} & \cdots & d_{nk} & \cdots & d_{nl} & \cdots & d_{nn}\end{pmatrix}\times\begin{pmatrix}0\\ \vdots\\ -h_{kl}\\ \vdots\\ h_{kl}\\ \vdots\\ 0\end{pmatrix} =$$

$$= \left(d_{kk}+d_{ll}-d_{kl}\right)h_{kl}^2.$$

Робимо висновок, що

$$F\left(\mathrm{x}^e\right) = F(\mathrm{x}^e)+\left(d_{kk}+d_{ll}-d_{kl}\right)h_{kl}^2.$$

Для досліджуваної задачі маємо лінійні обмеження, тому

$$F\left(\mathrm{x}^{e}\right)=F(\mathrm{x}^{e})-\left(d_{kl}-d_{kk}-d_{ll}\right)\frac{\left(S_t^e-S_t\right)^2}{\left(a_{tk}-a_{tl}\right)^2}.$$

Тут t – порядковий номер обмеження (2.4.2), котре не відповідає розв'язку x^{e*}

k – індекс, що відповідає тому елементу вектора x^{e*} з якого здійснюється спуск;

l – індекс, що відповідає елементу x^{e*}, до якого виконується спуск;

Тепер можемо сформулювати *лему* про вибір напрямку спуску в нелінійній задачі розподілу ресурсів, подібно до *леми 1*.

Лема 2: Точка $\mathrm{x}=\mathrm{x}^{e}+h$ буде розв'язком для задачі (2.4.1)-(2.4.3) тоді і лише тоді, коли параметри t,k,l визначаються такою умовою:

$$\beta=\max_{t\in I_H}\min_{(j,j+p)\in N_B}\left\{\left(d_{kl}-d_{kk}-d_{ll}\right)\frac{\left(S_t^e-S_t\right)^2}{\left(a_{tk}-a_{tl}\right)^2}\right\}.$$

Аби довести цю лему, достатньо скористатися схемою доведення аналогічного твердження для випадку лінійної задачі. Ґрунтовно алгоритм доведення поданий в [8].

Квадратична задача розподілу ресурсів має шість основних *кроків* розв'язування:

Крок 1: Розв'язати розширену задачу (2.4.4)-(2.4.5). Покласти

$$\mathrm{x}^{*}=\mathrm{x}^{e},S_t^{*}=S_t^{e}$$

Крок 2: Перевірити отриманий розв'язок на допустимість за обмеженням (2.4.2). Якщо він не виявиться допустимим, то перейти до *кроку 3.*

Крок 3: Обчислити такі значення:

$$\gamma_{kl}=d_{kl}-d_{kk}-d_{ll},k=1,2,..,n,l=k+m,\forall m=1,2,...,n-k,$$

а далі з наступної умови визначити, які напрямки можливі для спуску:

$$N_B = \{(k,l) \mid \gamma_{kl} > 0\}.$$

Крок 4: Визначити напрями спуску:

$$\beta = \max_{t \in I_H} \min_{(j, j+p) \in N_B} \left\{ \gamma_{kl} \frac{\left(S_t^e - S_t\right)^2}{\left(a_{tk} - a_{tl}\right)^2} \right\}.$$

Крок 5: Обчислити потрібний крок спуску

$$h_{k^* l^*} = \frac{S_t^e - S_t}{a_{tk^*} - a_{tl^*}}.$$

Крок 6: Перехід до нового розв'язку і здійснення його перевірки, передбаченої *кроком 2.*

3.Практична частина

Оскільки всі теоретичні аспекти задач розподілення ресурсів вже викладені, в наступному розділі розглянемо власне процес розв'язування конкретних прикладів та його програмну реалізацію мовою програмування *Python.* Розв'яжемо дві з досліджуваних вище задач – лінійну і нелінійну. Наприкінці візуалізуємо результати.

3.1 Лінійний випадок

На *рис.2* можемо спостерігати, як відбувається розподілення деякого ресурсу в m паралельних потоків, де досягнення ресурсом певних етапів виробництва визначається випадковим чином. Рисунок взятий із джерела [2].

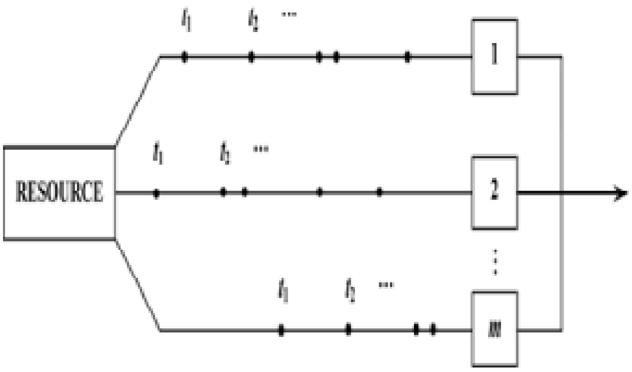

Рис.2

Отже, використовуючи дану схему, наведені вище алгоритми, можемо сформулювати конкретний приклад задачі типу (2.3.1)-(2.3.4):

Нехай математична модель задачі полягає у максимізації функції [2]:

$$F(x) = 5.3x_1 + 5.1x_2 + 5x_3 + 4.8x_4 \to \max \tag{3.1.1}$$

з системою обмежень:

$$3x_1 + 3.1x_2 + 3.2x_3 + 2.8x_4 \le s_1(t_j), \tag{3.1.2}$$

$$2.1x_1 + 1.9x_2 + 2.2x_3 + 1.9x_4 \le s_2(t_j), \tag{3.1.3}$$

$$x_1 + x_2 + x_3 + x_4 = s_3(t_j), \tag{3.1.4}$$

$$1 \le x_1 \le 5$$ (3.1.5)

$$1 \le x_2 \le 4$$

$$1 \le x_3 \le 5$$

$$1 \le x_4 \le 5$$

Вхідні параметри задачі: $t_0 = 9, m_\tau = 20, \sigma_\tau = 12$. Два останніх параметри є математичним сподіванням та, відповідно, середнім квадратичним відхиленням інтервалів τ_j. Самі інтервали τ_j мають показниковий розподіл.

Код програми, яка розв'язує цю задачу та ілюструє результат продемонстровано в *додатку А*. Після завершення роботи програма видаватиме такий результат:

```
Оптимальний розв'язок:
[2.1849 3.0411 1.8767 2.8973]
Значення функції цілі:
50.38
```

Додатково отримали графік цільової функції (рис. 3)

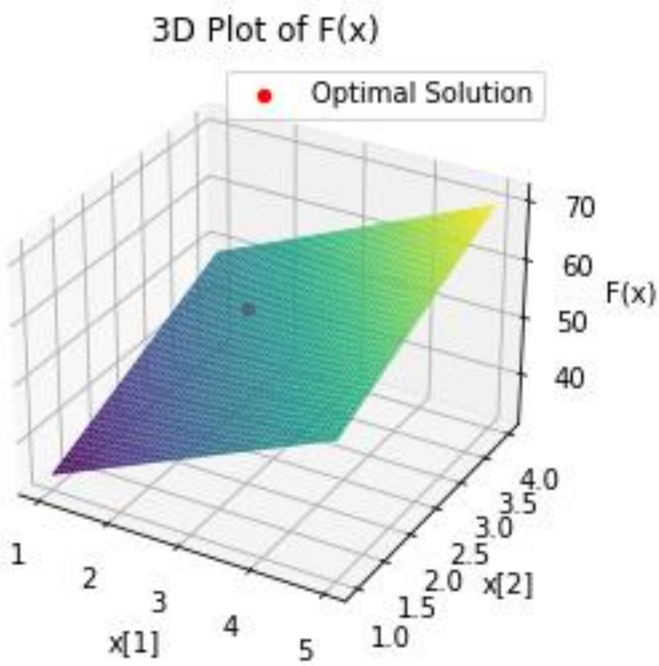

Рис.3

3.2 Нелінійний випадок

Розв'яжемо нелінійну задачу розподілу ресурсів (2.4.1)-(2.4.3) використовуючи наведений у *пункті 2.4* алгоритм. Покладемо у (2.4.1) такі параметри:

$$c = (1, 2, 3), D = \begin{pmatrix} -2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & -1 \end{pmatrix};$$

та введемо обмеження:

$$\begin{aligned} x_1 + x_3 &\le 4; \\ x_2 + x_3 &\le 3; \end{aligned} \qquad (3.2.1)$$

$$x_1 + x_2 + x_3 = 4.46. \qquad (3.2.2)$$

Після виконання алгоритму (*див. додаток A)* отримали такий розв'язок, значення цільової функції та її графік (рис.4):

```
Optimal solution: [1.67 2.31 0.48]
Objective value: 7.37
```

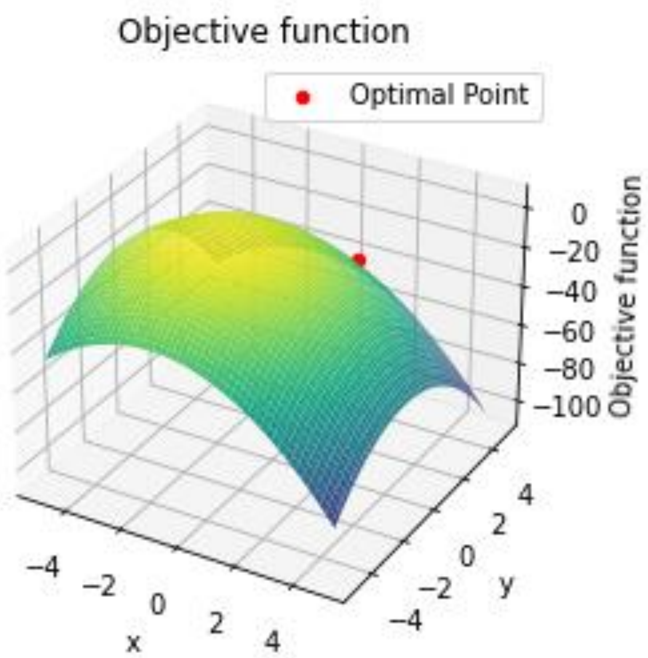

Рис.4

Висновки

У цій роботі досліджено процес розподілення потоків ресурсів. Розглянуто метод розширення множини допустимих розв’язків як основний підхід до розв’язання задач оптимізації досліджуваного процесу. Продемонстровано теоретичний механізм роботи методу, наведені критерії оптимальності знайденого розв’язку розглядуваної задачі, а також запропонований алгоритм реалізації методу.

Метод було запропоновано для розгляду задля його подальшого узагальнення на клас задач, для яких потоки ресурсів будуть формуватися випадковим чином. Отже, виникла потреба описати моделювання цих потоків за допомогою методу обернених функцій моделювання неперервних випадкових величин.

На основі методу розширення, були складені алгоритми для розв’язування лінійної задачі оптимального розподілу із загальним ресурсом та нелінійної задачі розподілу ресурсів між паралельними потоками.

У практичній частині було продемонстровано та проілюстровано роботу цих алгоритмів на конкретних прикладах. Швидкість їх збіжності варіюється в залежності від обраної цільової функції, кількості обмежень та методу, за яким обрано розв’язувати розширену задачу.

Нині проблема розподілу потоків ресурсів є популярним об’єктом дослідження, оскільки задачі, пов’язані з нею дотичні до багатьох повсякденних сфер діяльності людини. Активне вивчення і освоєння теми почалося всередині XX ст., проте узагальнення існуючих методів оптимізації процесу та створенні нових є нагальним питанням прикладної математики і сьогодні.

Використана література

1. Шукаев Д.Н. Прикладные методы оптимизации: учебник. – М.: Издательский дом Академии Естествознания, 2017. – 212 с.
2. Shukayev D.N., Kim E.R., Shukayev M., KozhamkulovaZh. Modeling allocation of parallel flows with general resource. Proceeding of the 22st IASTED International Conference Modelling and simulation (MS 2011) July 4-7, 2011 Calgary, Alberta, CANADA, s. 110-117.
3. Таха Х. Введение в исследование операций. – М.: Мир, 1985.
4. Shukayev D.N., Kim E.R. Extension method in location problem with discrete objects. Proceedings of the 21st IASTED International Conference “Modelling and Simulation (MS 2010)”, Banff, Alberta, Canada, 2010, P. 270-274.
5. Сікора Я. Б. Методи оптимізації. Навчально-методичний посібник для студентів напряму 6.040302 Інформатика*. – Житомир: Вид-во ЖДУ ім. Івана Франка, 2012. – 82 с.
6. Жалдак М.І., Триус Ю.В. Основи теорії і методів оптимізації: Навчальний посібник. - Черкаси: Брама-Україна, 2005. - 608 с.
7. Махней О. В. Математичне моделювання : навчальний посібник / О. В. Махней. — Івано-Франківськ : Супрун В. П., 2015. — 372 с.
8. T.E. Esbatyrov, D.N. Shukayev, B.D. Hisar, Distribution of loads in parallel systems by means of expansion, Mathematical, algorithmic and technical support of DCS, Tashkent, 1988, 153-160.
9. D.N. Shukayev, Optimization of resources allocation processes in parallel structure systems, Presentation of International scientific and technical conference FEIIC, Almaty, 1999, 185-192.
10. D.N. Shukayev, Computer Simulation (Almaty: KazNTU, 2004).

Інтернет-ресурси

11. https://uk.wikipedia.org
12. https://matica.org.ua

Додаток А

Код програми, яка розв’язує задачу з *пункту 3.1:*

```
import numpy as np

# Вхідні дані
c = np.array([5.3, 5.1, 5, 4.8])
A = np.array([[3, 3.1, 3.2, 2.8],
              [2.1, 1.9, 2.2, 1.9],
              [1, 1, 1, 1]])
s = np.array([30.1, 20, 10])
V = np.array([1, 1, 1, 1])
W = np.array([5, 4, 5, 5])
t0 = 9
lambda_val = 0.2
m_tau = 20
sigma_tau = 12

# Кількість змінних
n = len(c)

# Ініціалізація змінних
x = np.zeros(n)
t = np.zeros(n)
tau = np.zeros(n)
z = np.zeros(n)
s_t = np.zeros(n)
s_m = np.zeros(n)

# Функція щільності phi
def phi(tau_val):
```

```
    return lambda_val * (1 - np.exp(-lambda_val *
tau_val))


# Розрахунок моментів tau
t[0] = t0
for j in range(1, n):
    tau[j] =
phi(np.random.exponential(scale=1/lambda_val))


for j in range(1, n):
    t[j] = t[j-1] + tau[j]


# Розрахунок значень s(t) та s(m)
for j in range(n):
    z[j] = np.random.normal(loc=m_tau, scale=sigma_tau)
    s_t[j] = 1 / phi(z[j])
    s_m[j] = 1 / phi(t[j])


# Розширена задача
eps = 1e-6
A_extended = A.copy()
A_extended[0] += eps * A[1] * x[1]
A_extended[1] += eps * A[0] * x[0]


# Розв'язок розширеної задачі
optimal_solution = np.linalg.lstsq(A_extended, s,
rcond=None)[0]


# Перевірка оптимальності розв'язку
if np.all(np.matmul(A, optimal_solution) <= s):
    solution = optimal_solution
```

```
else:
    # Крок спуску
    indices = np.where(np.matmul(A, optimal_solution) >
s)[0]
    t_indices = np.where(s_t > s_m)[0]
    alpha_t = np.max([np.min([(c[j] - np.dot(A[j],
optimal_solution)) / np.dot(A[j], (s_t - s_m)[t_indices])
for j in indices])])

    # Обчислення кроку спуску h
    h = np.minimum(np.minimum(x - V, W - x), alpha_t *
(s_t - s_m))

    # Оновлення розв'язку
    solution = np.where(indices[:, None] == np.arange(n),
optimal_solution - h, optimal_solution)

# Округлення розв'язку
solution = np.round(solution, 4)
objective_value = np.round(np.dot(c, solution), 2)

# Виведення результатів
print("Оптимальний розв'язок:")
print(solution)
print("Значення функції цілі:")
print(objective_value)
```

Вивід графіка цільової функції:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Оптимальний розв'язок
```

```
x_optimal = np.array([2.1849, 3.0411, 1.8767, 2.8973])

# Функція цілі
def F(x):
    return 5.3*x[0] + 5.1*x[1] + 5*x[2] + 4.8*x[3]

# Значення x1, x2
x1 = np.linspace(1, 5, 100)
x2 = np.linspace(1, 4, 100)

X1, X2 = np.meshgrid(x1, x2)
Z = F([X1, X2, x_optimal[2], x_optimal[3]])

# Побудова тривимірного графіка
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.plot_surface(X1, X2, Z, cmap='viridis')
ax.set_xlabel('x[1]')
ax.set_ylabel('x[2]')
ax.set_zlabel('F(x)')
ax.set_title('3D Plot of F(x)')

# Виділення оптимальної точки
ax.scatter(x_optimal[0], x_optimal[1], F(x_optimal),
color='r', label='Optimal Solution')

plt.legend()
plt.show()
```

Аналогічні фрагменти коду наведемо для *пункту 3.2:*

```
import numpy as np


# Функція для перевірки нерівності обмежень
def inequality_constraint(x, A, S):
    return np.dot(A, x) - S


# Головна функція для розв'язку квадратичної задачі
розподілу ресурсів
def solve_quadratic_resource_allocation(c, D, A, S, E,
S_m, V, W):
    n = c.shape[0]


    # Крок 1: Встановити значення множників Лагранжа
    lambda_val = S_m / np.dot(S_m, S_m)


    # Початкові значення
    x = np.ones(n)
    S_t = S_m


    while True:
        # Крок 2: Перевірити допустимість розв'язку
        inequality_res = inequality_constraint(x, A, S)
        if np.all(inequality_res <= 0):
            return x


        # Крок 3: Обчислити значення γ_kl та визначити
напрямки спуску
        gamma = np.zeros((n, n))
        N_B = []


        for k in range(n):
```

```
            for l in range(k + 1, n):
                gamma[k, l] = D[k, l] - D[k, k] - D[l, l]
                if gamma[k, l] > 0:
                    N_B.append((k, l))

        # Крок 4: Знайти напрямки спуску β
        beta = float('inf')

        for t in range(len(E)):
            for (k, l) in N_B:
                a_tk = E[t, k]
                a_tl = E[t, l]
                if a_tk - a_tl != 0:
                    S_t_e = S_m[t]
                    beta = min(beta, gamma[k, l] * (S_t_e - S_t[t]) ** 2 / (a_tk - a_tl) ** 2)

        # Крок 5: Обчислити крок спуску h_(k^* l*)
        h_star = np.zeros((n, n))
        for (k, l) in N_B:
            a_k = E[:, k]
            a_l = E[:, l]
            if np.sum(gamma[k, l] * (a_k - a_l) ** 2) != 0:
                h_star[k, l] = beta * np.sum((a_k - a_l) * (S_t - S_new)) / np.sum(gamma[k, l] * (a_k - a_l) ** 2)
                h_star[l, k] = -h_star[k, l]

        # Оновлення розподілу ресурсів
        S_new = S_t - np.dot(h_star, E.T)
```

```
        # Крок 6: Оновлення розв'язку x
        x_new = np.zeros(n)
        for i in range(n):
            x_new[i] = max(0, x[i] + V[i] + W[i] *
(S_new[i] - S_t[i]) / lambda_val[i])

        # Перевірка на збіжність
        if np.all(x_new == x) and np.all(S_new == S_t)
and np.all(inequality_res <= 0):
            return x_new

        # Оновлення розв'язку
        x = x_new
        S_t = S_new

# Параметри задачі
c = np.array([1, 2, 3])
D = np.array([[-2, 0, 0], [0, -3, 0], [0, 0, -1]])
A = np.array([[1, 0, 1], [0, 1, 1]])
S = np.array([4, 3])
E = np.array([[1, 1, 1]])
S_m = np.array([4.46])
V = np.array([0, 0, 0])
W = np.array([2, 3, 1])

# Розв'язок задачі
result = solve_quadratic_resource_allocation(c, D, A, S,
E, S_m, V, W)
print("Optimal solution:", result)

# Обчислення значення цільової функції
```

```
obj_value = np.dot(c, result)
print("Objective value:", obj_value)
```

Вивід графіка:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Цільова функція
def objective_function(x, c, D):
    return 0.5 * np.dot(np.dot(x.T, D), x) - np.dot(c, x)

# Параметри задачі
c = np.array([1, 2, 3])
D = np.array([[-2, 0, 0], [0, -3, 0], [0, 0, -1]])

# Діапазон значень x і y
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
X, Y = np.meshgrid(x, y)

# Значення цільової функції для кожної комбінації x і y
Z = np.zeros_like(X)
for i in range(len(x)):
    for j in range(len(y)):
        Z[i, j] = objective_function(np.array([X[i, j],
Y[i, j], X[i, j]]), c, D)

# Побудова тривимірного графіка цільової функції
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(X, Y, Z, cmap='viridis')
```

```
# Настройка відображення
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('Objective function')
ax.set_title('Objective function')

# Позначення оптимальноі точки
result = np.array([1.67, 2.31, 0.48])
optimal_point = objective_function(result, c, D)
ax.scatter(result[0], result[1], optimal_point,
color='red', label='Optimal Point')
ax.legend()

# Виведення графіка
plt.show()
```

Додаток Б

Закон розподілу	Щільність	Моделююча формула
Ґаусівський	$\varphi(\tau)=\frac{1}{\sigma_\tau\sqrt{2\pi}}e^{\frac{(\tau-m_\tau)^2}{2\sigma_\tau^2}},-\infty<\tau<\infty.$	$\tau=m_\tau+\sigma_\tau\left(\sum_{i=1}^{12}z_i-6\right).$
Рівномірний	$\varphi(\tau)=\frac{1}{b-a},\tau\in[a,b].$	$\tau=a+z(b-a).$
Показниковий	$\varphi(\tau)=\lambda e^{-\lambda\tau},\tau\ge 0$	$\tau=-\frac{1}{\lambda}\ln z.$
Лінійний	$\varphi(\tau)=\lambda\left(1-\frac{\lambda}{2}\tau\right),\tau\in\left(0,\frac{2}{\lambda}\right].$	$\tau=\frac{2}{\lambda}(1-\sqrt{z}).$
Гамма-розподіл	$\varphi(\tau)=\frac{\alpha^k}{(k-1)!}\tau^{(k-1)}e^{-\alpha\tau},\alpha>0,k>0,\tau\ge 0.$	$-\frac{1}{\alpha}\ln(z_1\times z_2\times\ldots\times z_k).$

Таблиця 1

Відгук наукового керівника на кваліфікаційну роботу бакалавра з теми:

«Моделювання процесу розподілення потоків ресурсів»

студента 4-го курсу ОС «Бакалавр»

факультету комп'ютерних наук та кібернетики

Київського національного університету імені Тараса Шевченка

Тітімця Артема Олександровича

Рецензована робота присвячена моделюванню процесу розподілення потоків ресурсів шляхом побудови і розв'язування задач оптимізації. Розподіл ресурсів активно досліджується останнім часом, наприклад у роботі "Dynamic Resource Allocation in Wireless Networks with Heterogeneous Services" Лі Чжана та інших (2019).

Цікавою є реалізація методу розширення множини допустимих розв'язків для задач розподілу ресурсів. На практиці метод може бути узагальнений для різних класів задач оптимізації. Це є додатковою мотивацією для дослідження теми.

У роботі студентом було сформульовано кроки методу розширення для двох різнотипних задач розподілу ресурсів, а також проведений експеримент з комп'ютерної реалізації методу. Недоліком роботи є відсутність обчислень, пов'язаних зі швидкістю збіжності досліджуваного алгоритму. Також бракує прикладів конкретного застосування наведених методів для оптимізації виробництва. Проте загальне враження від роботи є непоганим.

Вважаю, що кваліфікаційна робота студента Артема Тітімця відповідає вимогам, які висуваються до бакалаврських робіт, і заслуговує на оцінку «добре», а її автор заслуговує на присвоєння кваліфікації бакалавра.

Асистент кафедри обчислювальної математики

факультету комп'ютерних наук та кібернетики

Київського національного університету

імені Тараса Шевченка,

кандидат фізико-математичних наук, асистент В'ячеслав ОНОЦЬКИЙ

Рецензія на кваліфікаційну роботу бакалавра з теми:

«Моделювання процесу розподілення потоків ресурсів»

студента 4-го курсу ОС «Бакалавр»

факультету комп'ютерних наук та кібернетики

Київського національного університету імені Тараса Шевченка

Тітімця Артема Олександровича

Розподілення ресурсів — одна з ключових сфер застосування методів оптимізації. Сьогодні ця тема є особливо популярною насамперед через постійне зростання обсягів даних, з якими доводиться працювати виробникам і людству в цілому. За останні роки обсяги даних інтенсивно зростають у різних галузях, таких як бездротові мережі, хмарні обчислення, Інтернет речей та соціальні медіа. Розподілення ресурсів стає важливим, оскільки потрібно ефективно управляти цими великими обсягами даних та забезпечувати доступ до ресурсів для різних користувачів і пристроїв.

Метою кваліфікаційної роботи Артема ТІТІМЦЯ є дослідження методу розширення множини допустимих розв'язків як основоположного алгоритму в моделюванні процесу розподілення потоків ресурсів, а також його узагальнення на різні класи задач. Загальна ідея методу полягає у введені для розглядуваної задачі оптимізації допоміжної задачі (розширеної), яка б не враховувала «проблемні» обмеження, через які пошук оптимального розв'язку значно ускладнюється. Далі знайдений одним із методів оптимізації розв'язок розширеної задачі перевіряють на задоволення всіх вихідних обмежень. За умови, коли не всі обмеження задовольняються, здійснюється спуск до інших розв'язків доти, доки обмеження не будуть задовільнені.

Робота має теоретико-експериментальний характер. Студентом викладені основні кроки різних модифікацій методу розширення, а також здійснені обчислювальні експерименти з задачами оптимізації лінійної та квадратичної функцій. Чітко прослідковується градація складності алгоритмів розв'язку зі зміною класів задач. Всі алгоритми мають фактично однакову теоретичну основу, однак відповідно до типу задачі впроваджуються додаткові кроки, наприклад моделювання пальмівських процесів в лінійному випадку.

Вважаю, що кваліфікаційна робота студента Артема ТІТІМЦЯ відповідає вимогам, які висуваються до бакалаврських робіт, і заслуговує на оцінку «добре», а її автор заслуговує на присвоєння кваліфікації бакалавра.

Доцент кафедри дослідження операцій

факультету комп'ютерних наук та кібернетики

Київського національного університету імені Тараса Шевченка,

кандидат фізико-математичних наук, доцент Роман ЯКИМІВ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

СИСТЕМА ЗАПОБІГАННЯ ТА ВИЯВЛЕННЯ АКАДЕМІЧНОГО ПЛАГІАТУ

Довідка про оригінальність кваліфікаційної роботи за освітнім рівнем бакалавр

UNICHECK

Ім'я користувача:
Оноцький В'ячеслав ФКомпНаук

Дата перевірки:
02.06.2023 11:40:35 EEST

Дата звіту:
02.06.2023 11:44:14 EEST

ID перевірки:
1015388259

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100002816

Назва документа: ТітімецьАртемОлександрович

Кількість сторінок: 30 Кількість слів: 4948 Кількість символів: 37007 Розмір файлу: 1.05 MB ID файлу: 1015053140

1.72% Схожість

Найбільша схожість: 0.49% з Інтернет-джерелом (http://elartu.tntu.edu.ua/bitstream/123456789/17880/4/OMM_meod_p..

1.54% Джерела з Інтернету 32 .... Сторінка 32

1.13% Джерела з Бібліотеки 70 .... Сторінка 32

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Експертна оцінка роботи науковим керівником :

Робота виконана студентом Тітімцем самостійно і не містить суттєвих запозичень без відповідних посилань

Науковий керівник: ____________________ ___Оноцький В.В.________________
(підпис) (ПІБ)

Оператор: ____________________ Оноцький В.В.______________
(підпис) (ПІБ)