

Зміст

1 Введення	3
1.1 Побудова математичної моделі	4
1.2 Класифікація моделей	6
2 Лінійне програмування	9
2.1 Типові проблеми	10
2.2 Геометричний сенс	16
2.3 Симплекс-метод	27
2.4 Цілочисельне лінійне програмування	37
2.5 Проблеми оптимального вибору	37
3 Нелінійне програмування	39
4 Динамічне програмування	41

Розділ 1

Введення

Говорячи про «дослідження операції», ми маємо на увазі пошук рішення якоїсь проблеми, пов'язаної із реальним життям. Дослідження полягає в тому, щоби знайти якнайкращий спосіб представлення реального світу чисельно: у вигляді матриць, векторів або функціональних залежностей. Тоді, використовуючи математичний апарат, відповідь на задачу можна знайти із будь-якою точністю. Після цього її потрібно інтерпретувати вже у «людських» поняттях.

Мистецтво чисельного опису світу називається побудовою **математичної моделі**. Така потреба виникала настільки часто, що з отриманих результатів можна виділити загальні підходи та ідеї, і використовувати їх на інших, ще не досліджених проблемах (операціях).

Цей посібник надає приклади багатьох популярних задач, що можуть бути розв'язані з допомогою лінійного, нелінійного або динамічного програмування, а також аналіз теорії, що стоїть за ними. Автор сподівається, що це дозволить сформувати хороше розуміння дослідження операцій, а також набути інтуїцію, яку потім можна використовувати при

розв'язуванні нових, невідомих досі проблем.

1.1 Побудова математичної моделі

Логічно, що з початком аналізу будь-якої задачі, варто виділити ті дані, які подаються як факт – тобто, на них неможливо вплинути, чи змінити будь-як. Це те, біля чого можна написати «дано», і далі подати перелік змінних з присвоєними їм значеннями. Це називається **параметрами операції**, або **некерованими змінними**.

Якщо якісь дані описують одну і ту саму характеристику різних предметів, цілком доцільно об'єднати їх у вектор. Надалі використовуватиметься така нотація: $\mathbf{a} = \langle a_1, \dots, a_n \rangle$. В тих випадках, коли подані дані є двовимірними, чи відображають деякі залежності, можна використовувати матриці. Наприклад, якщо в умові сказано, що *відомо час перевезення вантажу між кожним із обласних центрів*, то їх можна пронумерувати, а дані помістити, скажімо, в матрицю \mathbf{T} , де кожен елемент t_{ij} позначатиме час перевезення з i -ого міста в j -те.

Опісля маємо з'ясувати, які ж дані в задачі залишились невідомими, і їх потрібно знайти – себто, визначити на основі відомих. Вони називаються **керувальними параметрами**, або ж **керованими змінними**. Присвоєння цим змінним конкретних значень і є вирішенням проблеми.

У будь-яких задачах, які читач міг бачити до цього, зазвичай потрібно було з'ясувати значення одного невідомого параметра (деякого x), або навіть кількох. При дослідженні операцій може так траплятись, що невідомою буде ціла матриця, або вектор.

Якщо існує таке рівняння, що явно пов'язувало б некеровані змінні з керованими, то розв'язок можна знайти **аналітично**, тобто, обчислити звичайними арифметичними операціями. Наприклад, якщо відомо, що

пропускна спроможність каналу передачі становить n Мб/с, і потрібно з'ясувати, скільки максимально x Мб інформації можна передати за t секунд, то зрозуміло, що відповідь може бути знайдена з формули $x = tn$.

Однак бувають такі проблеми, для яких неможливо – або принаймні дуже складно – скласти будь-яке рівняння, тому що у них взаємодія між собою велика кількість змінних, кожна з яких по-своєму впливає на оптимальність результату в цілому. Тоді розв'язування може нагадувати намагання збалансувати складну систему тягарців: як тільки потягнути за один, одразу ж починають рух інші. Відповідь на такі проблеми найчастіше можна знайти **алгоритмічно**, на кожному з кроків покращуючи оптимальність результату. Власне, саме такий спосіб і використовується найчастіше для дослідження операцій.

Для того, щоби розв'язати задачу алгоритмічно, прийнятою практикою є побудова (опис) деякої множини G , що містить всі можливі допустимі розв'язки задачі. Звичайно, перед цим потрібно визначити, в якому вигляді ми взагалі шукатимемо розв'язок. Скажімо, якщо треба знайти дві змінні $x_1, x_2 \in \mathbb{R}$, то розв'язком буде вектор $\mathbf{X} = \langle x_1, x_2 \rangle$, і множина розв'язку – $G \subset \mathbb{R}^2$. Іноді рішенням проблеми може бути матриця, чи складніші багатовимірні об'єкти.

Наостанок, для того, щоб оцінити оптимальність будь-якого розв'язку, прийнято вводити **цінову функцію** (інша назва – **коефіцієнт ефективності**), що встановлює чисельну оцінку якості $\varphi : G \rightarrow \mathbb{R}$. Формулювання функції обирається залежно від умови.

Наприклад, якщо невідомі змінні x_1, x_2 встановлюють кількість товару двох видів, яку виготовлятиме завод, а c_1, c_2 – їхню ціну відповідно, то коефіцієнт ефективності можна визначити як прибуток від всього товару загалом: $\varphi(\mathbf{X}) = c_1x_1 + c_2x_2$. Тоді знайти такий \mathbf{X} , що забезпечував би максимальний прибуток, означає знайти екстремум φ .

1.2 Класифікація моделей

Запропонована структура математичної моделі дозволяє описати будь-яку операцію чисельно, зводячи розв'язування до пошуку такого $X \in G$, щоби досягнути мінімуму або максимуму φ . Зрозуміло, що розв'язувати такі проблеми перебором, щонайменш неефективно. На щастя, для цього існують спеціальні методи, проте вони гарантують знаходження адекватного розв'язку лише для якогось конкретного типу математичної моделі. Тому важливо розрізняти, які типи існують взагалі, і вміти визначити їх для моделі своєї задачі. Далі приведено класифікацію математичних моделей за характерною ознакою.

	Так	Ні
Рішення проблеми знаходиться розв'язком рівняння.	аналітична	алгоритмічна
Хоча б одна з величин є випадковою величиною.	стохастична	детермінована
Кожен з кроків алгоритму розв'язку покращує рішення, знайдене на попередньому.	статична	
На кожному з кроків алгоритму розв'язку приймається найоптимальніше рішення.		динамічна
Хоча б одна з керованих змінних може приймати тільки цілі значення.	дискретна	недискретна

Розділ 2

Лінійне програмування

Одночасно простим для розуміння та корисним на практиці є математичний апарат розв'язку задач лінійного програмування (ЛП), з яким читач познайомиться у цьому розділі.

- У параграфі [2.1 Типові проблеми](#) читач дізнається про деякі типові проблеми ЛП та отримає інтуїтивне розуміння того, як варто формувати ОДР і цінкову функцію.
- У параграфі [2.2 Геометричний сенс](#) описано графічний метод розв'язку проблем ЛП для $\mathbf{X} \in \mathbb{R}^2$.
- Параграф [2.3 Симплекс-метод](#) описує симплекс-метод розв'язування ЛП для n -вимірних проблем ($\mathbf{X} \in \mathbb{R}^{\geq 2}$).
- У параграфі [2.4 Цілочисельне лінійне програмування](#) читач дізнається про принцип границь та галузей для розв'язування цілочисельної задачі ЛП.

- У параграфі 2.5 Проблеми оптимального вибору описано постановку та розв'язування ЛП задачі при $\mathbf{X} \in \{0, 1\}^n$.

Задачу можна назвати лінійною тоді, коли цінова функція є функцією першого степеня (2.1), а область допустимих значень може бути описана лінійними рівностями або нерівностями (2.2).

$$\varphi(\mathbf{X}) = \sum_{j=1}^n c_j x_j = \mathbf{cX}^T, \quad \mathbf{c} = \langle c_1, \dots, c_n \rangle \quad (2.1)$$

$$G = \left\{ \langle x_1, \dots, x_n \rangle : \sum_{j=1}^n a_{ij} x_j \begin{array}{|c|} \leq, \geq \text{ або } = \\ \hline \end{array} b_i \quad \forall i = \overline{1, m} \right\} \quad (2.2)$$

2.1 Типові проблеми

Приведені далі постановки задач є типовими. Вони отримали свою назву від проблеми, у вирішенні якої найкраще використовувати такі означення. Якщо їх проаналізувати, то можна отримати відпрацьований математичний апарат для розв'язання більш специфічних задач.

2.1.1 Максимізація прибутку

Фабрика виробляє продукцію n видів. Для неї потрібна певна кількість сировини m видів, при чому її є в обмеженій кількості. Відомо можливий прибуток від реалізації кожного з видів продукції. Якщо існує якийсь оптимальний розподіл всієї сировини на виробництво певних видів продукції так, щоби отримати якнайбільший прибуток, то яким буде такий розподіл?

Якщо між двома некерованими змінними існує відношення, то доцільно описати його у вигляді матриці. Тому нехай в \mathbf{A} міститимуться елементи a_{ij} , кожен з яких означатиме витрату сировини i -ого ви-

ду на виготовлення продукції j -ого виду. Ще одна некерована змінна $\mathbf{b} = \langle b_1, \dots, b_m \rangle$ позначатиме кількість відповідної сировини кожного виду, доступної на фабриці. Прибуток від реалізації можна визначити вектором $\mathbf{c} = \langle c_1, \dots, c_n \rangle$.

Керованою змінною є вектор $\mathbf{X} = \langle x_1, \dots, x_n \rangle$, кожен елемент якого позначає кількість продукції, яку має виготовити фабрика – ці дані варто змінювати для досягнення оптимальності.

Тоді умовою допустимості розв'язку буде перш за все виконання обмежень на доступну кількість сировини, та деякі очевидні твердження – наприклад, те, що ця кількість має бути невід'ємною (2.3).

$$\mathbf{X} \in G \Leftarrow \left\{ \begin{array}{l} \sum_{j=1}^n a_{ij}x_j \leq b_i \quad \forall i = \overline{1, m} \\ x_j \geq 0 \quad \forall j = \overline{1, n} \end{array} \right\} \quad (2.3)$$

Питання для роздумів

Чи можна поставити умову так, щоби знайдені x_i обов'язково повинні були би бути цілими числами?

В означенні (2.3) формулювання $\sum_{j=1}^n a_{ij}x_j$ позначає суму використаної сировини певного типу на виготовлення ресурсів всіх можливих видів. Відповідно « $\leq b_i$ » позначає, що цією сировини можна використати не більше певної кількості. Умова $\forall i$ означає, що така нерівність повинна виконуватись для всіх видів сировини.

Цінову функцію потрібно вводити так, щоби вона визначала прибутовість прийнятого рішення. Тоді оптимальним розв'язком буде такий, що гарантує досягнення її максимуму (2.4).

$$\varphi(\mathbf{X}) = \sum_{j=1}^n c_j x_j \rightarrow \max_{\mathbf{X} \in G} \quad (2.4)$$

Питання для роздумів

Якщо існує така задача, для якої потрібно вводити кілька цінових функцій, і оптимізувати їх одночасно, то якою буде умова такої задачі? Чи існує якийсь універсальний спосіб поєднання кількох оцінок оптимальності в одну?

2.1.2 Мінімізація витрат

Розглянемо задачу, подібну тій, що поставлена у розділі 2.1.1. *Фабрика отримує сировину n видів, з якої виготовляє товар за m різними технологіями. Кожна з технологій потребує певну комбінацію сировини у різних кількостях. Разом з тим, кожна з технологій спричиняє в атмосферу викид певної кількості кілограм вуглецю. Для кожного виду сировини визначена доступна кількість, яку можна використати для виробництва. Потрібно оптимізувати виробництво так, щоби отримати якомога більше готової продукції, але водночас зменшити кількість шкідливих викидів.*

Що в цій задачі уже відомого? Найперше – відношення технології та сировини, яку вона потребує. Запишемо це в матрицю \mathbf{A} , в якій кожен елемент a_{ij} позначатиме витрату сировини i -ого типу за j -ою технологією. Викиди різних технологій можна зберігати у векторі $\mathbf{c} = \langle c_1, \dots, c_m \rangle$, а кількість доступної сировини кожного виду – як $\mathbf{b} = \langle b_1, \dots, b_n \rangle$.

Що потрібно знайти? Потрібно знайти кількість товару, яка буде виготовлена за кожною з технологій. Тому позначимо її через вектор $\mathbf{X} = \langle x_1, \dots, x_m \rangle$. При пошуку значення різних x_j доведеться вирішити дилему: користуватись технологіями, які потребують меншу кількість сировини, але спричиняють більше викидів, чи навпаки?

Будь-який розв'язок \mathbf{X} буде допустимим, якщо задовольнятиме дві

умови: використано ресурсів не більше, ніж доступно взагалі, і жоден з x_j не є від'ємним. Тому ознака допустимості розв'язку є такою самою, як і (2.3) – ми використовуємо ті самі назви змінних, тому формула не втрачає свого змісту. Тепер формулювання $\sum_{j=1}^n a_{ij}x_j$ позначатиме кількість сировини певного виду, використаної загалом, за якими технологіями б не відбувалось виробництво.

Допоки розв'язок допустимий, він використовуватиме лише доступну кількість сировини. Однак різні розв'язки все ще спричиняють різний викид вуглецю, тому оптимальність кожного з них визначатимемо саме за цим параметром. Цінова функція знову така сама, як і в (2.4), проте цього разу її доцільно мінімізувати.

$$\varphi(\mathbf{X}) = \mathbf{c}\mathbf{X}^T \rightarrow \min_{\mathbf{X} \in G} \quad (2.5)$$

Тепер, після ознайомлення з двома задачами, важливо розуміти, що не існує жодних правил, як варто складати формулювання ОДР чи цінової функції, так само як і неможливо передбачити всі можливі «типові задачі». Формулювання будь-якої математичної моделі залежить винятково від області дослідження та винахідливості дослідника.

2.1.3 План перевезень

Досі ми розглядали задачі, де невідомою керованою змінною був вектор – тобто, кількість чогось різних видів, або оптимальна конфігурація різних характеристик. Чи можливо тепер за допомогою лінійного програмування розв'язати якусь проблему, що встановлюватиме конфігурацію відношення? Тобто, цього разу невідоме \mathbf{X} буде двовимірним – матрицею.

Компанія має у своєму розпорядженні n зерносховищ та m постачальників зерна. Кожне зерносховище може зберігати певну кількість тон зерна, купленого у різних постачальників. З ними укладено контракт, за яким компанія зобов'язується викупити щонайменш певну кількість тон за раз. Відомі затрати на перевезення зерна від кожного постачальника до кожного сховища – скажімо, час, відстань, паливе тощо. Потрібно визначити, скільки зерна між різними постачальниками та відповідними сховищами доцільно перевезти, щоби отримати якомога менші витрати, та перевезти якомога більше вантажу.

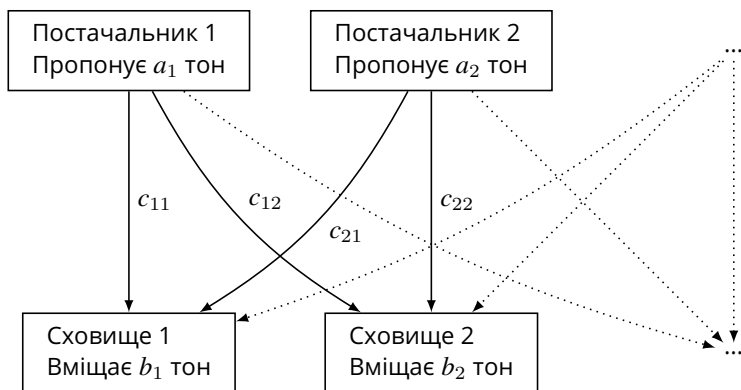


Рис. 2.1: Кожен з постачальників пропонує $\mathbf{a} = \langle a_1, \dots, a_m \rangle$ тон, а кожне зі сховищ може зберігати до $\mathbf{b} = \langle b_1, \dots, b_n \rangle$. Затрати на перевезення між ними задаються матрицею $\mathbf{C} \in \mathbb{R}^{m \times n}$.

Отже, умову проблеми можна зобразити у вигляді графа, що на Рис. 2.1. Зрозуміло, що некерованими змінними є \mathbf{a} , \mathbf{b} та \mathbf{C} . Тоді розв'язком буде матриця \mathbf{X} , кожен елемент x_{ij} якої задаватиме «оптимальну» кількість тон, яку варто перевезти між i -им постачальником та j -им сховищем. Саме така конфігурація, що буде задана цією матрицею і гарантуватиме найменші витрати та якнайбільше заповнення сховищ.

Ця проблема є класичним випадком транспортної задачі, і для її розв'язку існують набагато ефективніші методи, про які можна дізнатись у наступних розділах. Попри це, розв'язок саме лінійним програмуванням є досить цікавим.

Тепер ОДР повинно мати такі обмеження:

- потрібно вивезти весь товар, що надають постачальники;
- до кожного зі сховищ не можна привезти більше, ніж воно може вмістити.

Тоді умову допустимості розв'язку можна записати як 2.6.

$$\mathbf{X} \in G \Leftrightarrow \left\{ \begin{array}{l} \sum_{i=1}^m x_{ij} = a_i \quad \forall j = \overline{1, n} \\ \sum_{j=1}^n x_{ij} \leq b_j \quad \forall i = \overline{1, m} \end{array} \right\} \quad (2.6)$$

Тепер для того, щоби порахувати затрати при поточному плані перевезення, достатньо перемножити значення c_{ij} з матриці затрат на значення x_{ij} . Оптимальним буде той розв'язок, у якому це значення найменше, тому вводимо цінову функцію (2.7), і вимагаємо її мінімізації.

$$\varphi(\mathbf{X}) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min_{\mathbf{X} \in G} \quad (2.7)$$

Питання для роздумів

Нехай матриця C відображає не затрати на перевезення, а максимальну пропускну здатність маршруту. Скажімо, в день по дорозі між двома пунктами можна перевезти не більше, ніж c_{ij} тон вантажу. Потрібно знайти таке завантаження X різних доріг, щоби перевезти якомога більше вантажу. Допускається ситуація, коли не весь вантаж буде вивезено від постачальників. Як сформулювати ОДР та цінову функцію?

2.2 Геометричний сенс

Двовимірні та (іноді) тривимірні випадки лінійного програмування можна розв'язувати графічно. Розуміння геометричних процесів, які відбуваються при відшукуванні розв'язку допоможуть при розв'язуванні багатовимірних задач.

2.2.1 Двовимірний випадок

Розгляньмо випадок, коли вектор X складається з двох координат x_1 та x_2 (тобто, $X \in \mathbb{R}^2$). В ідеальному випадку множина ОДР $G \in \mathbb{R}^2$ утворює на площині деяку замкнену область, як показано на Рис. 2.2. Зрозуміло, що коли обрати будь-яку точку з неї, то такі координати $\langle x_1, x_2 \rangle$ є допустимим розв'язком. Допустимих розв'язків існує безкінечна кількість (оскільки ми оперуємо множиною дійсних чисел, яка сама по собі є континуумом), але найоптимальнішим буде той, для якого цінова функція $\varphi(X)$ визначає якнайбільше число. Як можна знайти такий розв'язок графічно?

Проаналізуймо цінову функцію. В загальному випадку вона матиме

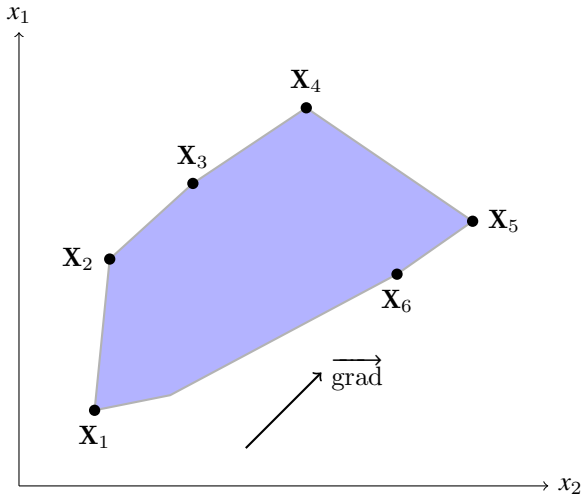


Рис. 2.2: ОДР утворює опуклий багатокутник на площині.

вигляд $\varphi = c_1x_1 + c_2x_2$, тобто утворюватиме площину в тривимірному просторі з координат $\langle x_1, x_2, \varphi \rangle$. Це можна уявити як на Рис. 2.3. Тоді допустимі розв'язки лежатимуть всередині проекції ОДР на площину цінкової функції. З рисунку добре видно, в яку сторону вона зростає, тож тепер зрозуміло, що точка X_1 є допустимим розв'язком з мінімально можливим φ , а X_5 – з максимальним φ . Тоді обидві точки будуть розв'язками задачі на мінімізацію та максимізацію цінкової функції відповідно.

Чи існує більш практичний спосіб, без потреби відмальовування тривимірних площин? Оскільки φ – лінійна функція, то за будь-яких x_1 та x_2 вона є неперервною, і зростає завжди однаково. Тобто, при будь-яких аргументах вектор градієнту завжди буде однакової довжини та спрямований в однаковому напрямку. Це означає, що на звичайному двовимірному малюнку достатньо позначити вектор $\overrightarrow{\text{grad}} \varphi$ (як і показано на Рис. 2.2). Тоді розв'язком задачі на максимізацію буде точка, що лежить якнайдалі

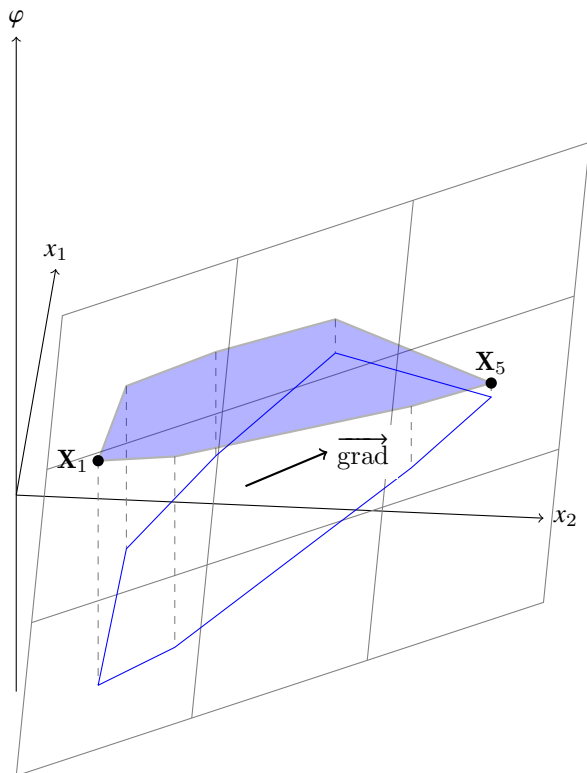


Рис. 2.3: Погляд на ОДР з іншої перспективи. Пунктиром позначено проєкцію G на φ .

у напрямку цього вектору. Аналогічно при мінімізації оптимальна точка розв'язку лежатиме у протилежному напрямку.

Продемонстрованим прикладом також можна пояснити випадок, коли цінова функція не є площиною, а має якийсь складніший вигляд, хоч це вже не буде задачею ЛП. Проекція G на φ сформує набір точок $\langle x_1, x_2, \varphi \rangle$, а оптимальним розв'язком буде та, що матиме найбільшу або найменшу координату φ .

Питання для роздумів

Яким чином потрібно модифікувати метод градієнтного спуску, щоб знайти оптимальний розв'язок при якій завгодно функції φ та замкненій ОДР? Чи ефективніше використати якийсь інший алгоритм?

Питання для роздумів

Чи можна системою нелінійних рівнянь описати таку ОДР, щоб утворився неопуклий багатокутник?

Висновки

- оптимальний розв'язок може бути розташований в одній з кутових точок ^a;
- кутова точка є оптимальним розв'язком, якщо жодна з сусідніх точок не є оптимальнішою.

^a(англ.) *extreme point*

2.2.2 Розв'язування задачі

Розглянемо деяку специфічну задачу. Броварня виготовляє ель та пиво. Виробництво кожного з них вимагає певної комбінації кукурудзи, хмелю та ячмінного солоду на 1 умовну одиницю (бочку, партію тощо). Кількість ресурсів обмежена. Кожен продукт має свою ціну. Параметри виробництва показано на Рис. 2.4. Потрібно розрахувати оптимальну кількість виробництва двох продуктів, щоб отримати якнайбільший прибуток.

Некеровані змінні можна позначити вектором $\mathbf{a} = \langle 220, 5, 540 \rangle$ (кількість доступних ресурсів на складі), $\mathbf{b} = \langle 213, 485 \rangle$ (прибуток від кожного

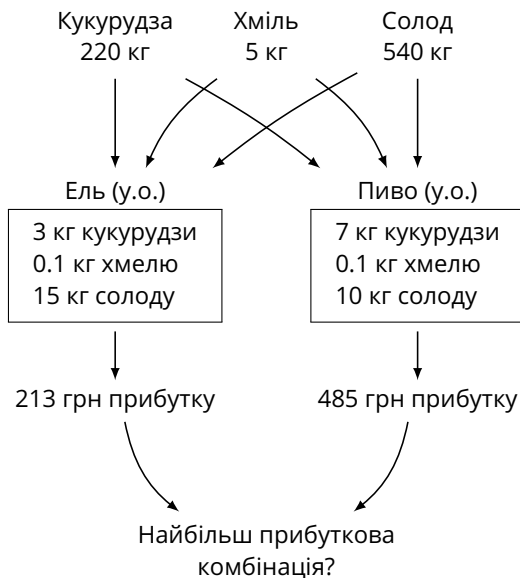


Рис. 2.4: Запаси ресурсів, параметри виробництва та прибутковість елю й пива.

з товарів) та матрицею \mathbf{A} (2.8), де a_{ij} позначатиме витрату i -ого ресурсу на j -ий продукт. Вартість кожного з продуктів визначатиме вектор $\mathbf{c} = \langle 213, 485 \rangle$.

$$\mathbf{A} = \begin{bmatrix} 3 & 0.1 & 15 \\ 7 & 0.1 & 10 \end{bmatrix} \quad (2.8)$$

Тоді керованою змінною є вектор $\mathbf{X} = \langle x_1, x_2 \rangle$ координати якого позначають скільки бочок елю та пива потрібно виготовити відповідно. Якщо розв'язок лежить в ОДР, мусять виконуватись такі умови:

- кількість використаних кукурудзи, хмелю та солоду не повинна перевищувати доступної;

- не можна виготовити від'ємну кількість продукції.

Кількість, використаної кукурудзи обраховується як $3x_1 + 7x_2$ (тобто, залежить від того, скільки виготовляється обох видів продукту). Оскільки маємо на неї обмеження, можна скласти нерівність $3x_1 + 7x_2 \leq 220$, що повинна виконуватись для допустимості розв'язку. Аналогічно складаються решту нерівностей, що утворюють ОДР задачі (2.9).

$$\mathbf{X} \in G \Leftrightarrow \left\{ \begin{array}{l} 3x_1 + 7x_2 \leq 220 \\ 0.1x_1 + 0.1x_2 \leq 5 \\ 15x_1 + 10x_2 \leq 540 \\ x_1, x_2 \geq 0 \end{array} \right. \quad (2.9)$$

Разом з тим оптимальність будь-якої комбінації x_1 та x_2 доцільно визначати за кількістю прибутку, що вона приносить. Тому цінова функція набуде вигляду (2.10).

$$\varphi(\mathbf{X}) = \mathbf{X}\mathbf{c}^T = 213x_1 + 485x_2 \rightarrow \max_{\mathbf{X} \in G} \quad (2.10)$$

На Рис. 2.5 показано ОДР та градієнт цінової функції. З отриманого рисунку видно, що точка з ОДР, найдалі розташована у напрямку градієнта (а значить у напрямку зростання оптимальності), міститься в координатах $x_1 = 21.067$ та $x_2 = 22.4$. Це і є оптимальним розв'язком задачі.

Питання для роздумів

Яким був би оптимальний розв'язок задачі, якби знак « \leq » в ОДР замінити на « $<$ »?

Питання для роздумів

Якби x_1 та x_2 визначали б не умовні одиниці, а вимірювалися б у бочках (тобто, тепер $\mathbf{X} \in \mathbb{Z}^2$), яким чином розв'язок такої задачі можна знайти графічно?

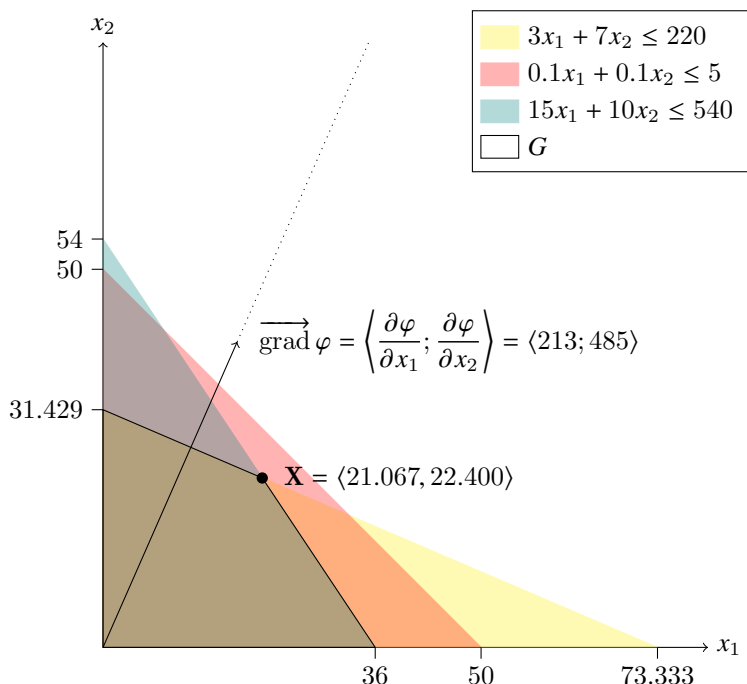


Рис. 2.5: ОДР задачі на двовимірній площині, що утворюється трьома нерівностями та обмеженнями на додатність x_1 і x_2 .

Як видно, графічний спосіб розв'язку досить зручний і простий. Однак коли результат потрібно отримати з високою точністю, таке розв'язування стає дуже складним та непрактичним, не кажучи вже про те, що реальні задачі рідко коли бувають двовимірними. Проте потрібно віддати належне – графічний спосіб ідеальний для унаочнювання даних.

Питання для роздумів

Чи можна якимось чином зменшити вимірність проблеми $\mathbf{X} \in \mathbb{R}^{>2}$, щоби скористатись графічним способом розв'язування?

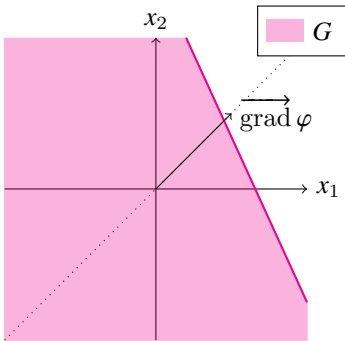


Рис. 2.6: ОДР складається з однієї нерівності, тому є необмеженою зі сторони від'ємних x_1 та x_2 . Тут можна розв'язати задачу на максимізацію, однак розв'язком мінімізації буде точка $\langle -\infty, -\infty \rangle$.

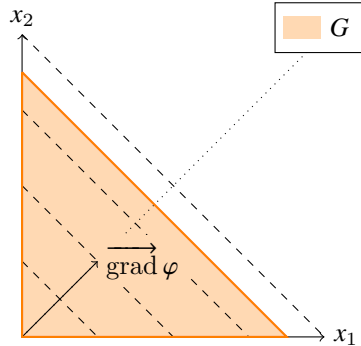


Рис. 2.7: Іноді неможливо визначити точку, що лежить найдалі за напрямком градієнта, оскільки всі точки на деякій лінії мають однакову оптимальність. В такому разі вони всі є оптимальними, і не існує єдиного \mathbf{X} .

2.2.3 Крайнощі

Описані раніше випадки, взагалі кажучи, є ідеальними, а із задачею нам просто пощастило.

Найперша проблема, яку іноді можна отримати – ОДР не є замкнутою, як показано на Рис. 2.6. В такому разі з тієї сторони, з якої ОДР необмежена, єдиного розв'язку не існує, позаяк до нього можна наблизуватись безкінечно. Якщо така відповідь є допустимою, розв'язком можна вважати точку $\langle \pm\infty, \pm\infty \rangle$, однак найчастіше подібний результат просто не є адекватним.

Іноді оптимальних розв'язків існує *безкінечна кількість*. Подібне формулювання може заплутати, і створити враження, ніби оптимальними

є взагалі всі можливі розв'язки, хоча це не так. Така ситуація продемонстрована на Рис. 2.7. У відповіді до цієї задачі доречно просто описати закон розрахунку оптимальних x_1 та x_2 .

Може так трапитись, що нерівності, з яких складається ОДР, ніде не перетинаються одночасно, як показано на Рис. 2.8. В такому разі ОДР порожня, а отже жодного оптимального розв'язку не може існувати за визначенням.

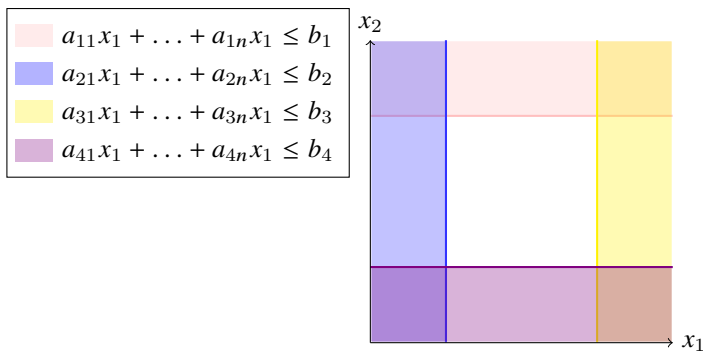


Рис. 2.8: ОДР складається з 4 нерівностей, однак не існує жодної такої області, де вони всі перетиналися б одночасно. Тому $G = \emptyset$.

2.2.4 Тривимірний випадок

Якщо у двовимірному просторі набір лінійних нерівностей може відтинати багатокутник, то у тривимірній області G ми отримаємо багатогранник. Приклад такої фігури показано на Рис. 2.9. В цьому випадку читачу може бути складно уявити цінкову функцію $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}$, позаяк вона є чотиривимірною, однак сенс залишається той самий: якщо перелік всіх точок фігури посортувати за тим, наскільки «далеко» вони розташовані в напрямку вектора-градієнта (а він тривимірний), то це одночасно буде

перелік, посортований за зростанням оптимальності. Можна ще уявити це як зростання температури, яскравості чи будь-якої іншої зручної характеристики в певному напрямку.

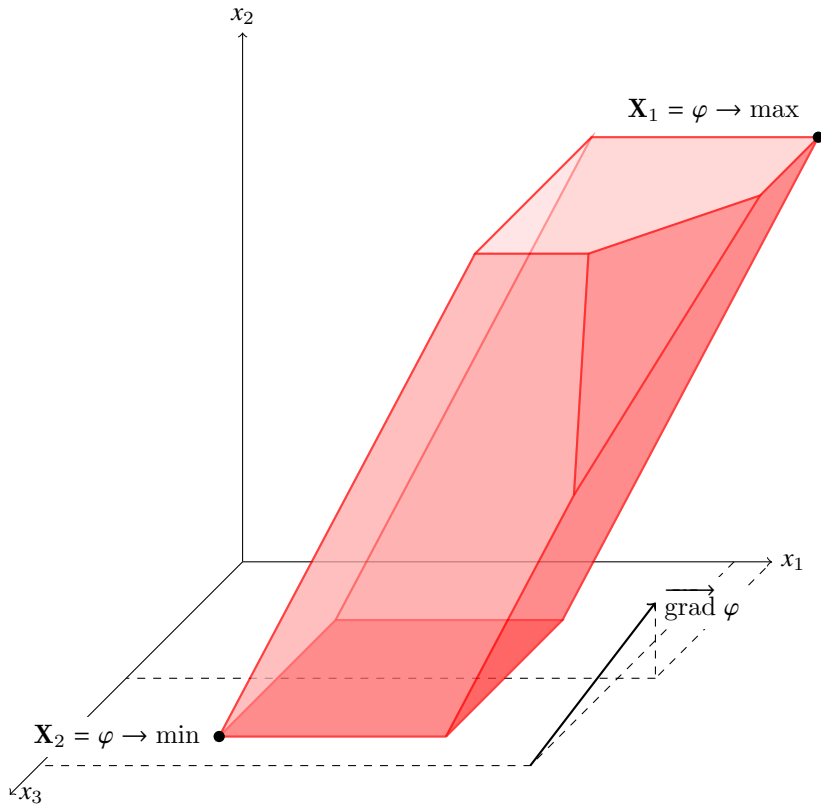


Рис. 2.9: Оскільки така фігура має 7 граней, можна зробити висновок, що вона утворена 7-ома нерівностями. Будь-яка точка $\langle x_1, x_2, x_3 \rangle$ всередині неї є допустимим розв'язком. Зважаючи на напрямок вектора градієнту, точка X_2 є розв'язком задачі на мінімізацію, а X_1 – на максимізацію цінкової функції.

Хоча отримана фігура досить проста на вигляд, малювати площини та визначати їхні перетини від руки насправді дуже складно. У вищих вимірностях проблема стає практично нерозв'язуваною. Тому хороший дослідник потребує більш точних аналітичних методів.

2.3 Симплекс-метод

З висновків на с. 19 може з'явитись така ідея:

- оптимальний розв'язок потрібно шукати в кутових точках – для цього їх потрібно спочатку знайти (отримати список координат);
- опісля обраховуємо значення цінової функції φ у всіх точках;
- обираємо точку з найбільшим φ , тобто найбільшою оптимальністю – розв'язок знайдено;
- якщо якісь дві суміжні точки мають однакову оцінку ефективності, то весь відрізок між ними визначає оптимальні розв'язки (тобто, їх безліч).

Читач, який колись пробував конструювати власні алгоритми, може заперечити, що така ідея є рішенням «в лоб» (бо кутових точок може бути дуже багато), і подібні дії обов'язково призведуть до катастрофічного зростання часової та просторової складності алгоритму. Це справедливе зауваження, тому спробуємо трохи покращити метод:

- спочатку обираємо деяку одну заздалегідь відому кутову точку, яку називатимемо **опорним планом**¹ – наприклад $\mathbf{X}_0 = \langle 0, 0, \dots \rangle$;

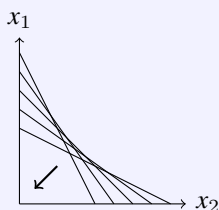
¹(англ.) *basic feasible solution*

- оцінюємо значення φ у сусідніх точках, і рухаємось до тієї, яка має найбільшу оптимальність;
- якщо обидві сусідні точки мають менше значення оптимальності, поточне рішення і є розв'язком задачі.

Що ж, вийшло набагато краще. Власне, описана ідея і лежить в основі симплекс-методу. З'ясуймо ще деякі моменти.

Що, як початок координат не є опорним планом? Якщо всі обмеження ОДР містять знак « \leq » або « $<$ », і одночасно $\forall i : x_i \geq 0$, цього ніколи не станеться. В інших випадках, звичайно, точка $\langle 0, 0, \dots \rangle$ може і не бути допустимим розв'язком. Тоді для розв'язування задачі потрібно буде вводити **штучний базис**². З цією технікою читач ознайомиться в наступних розділах

Так стається тому, що під яким би кутом не була б нахилена межа обмеження « \leq », область допустимих значень завжди буде спрямована до початку координат. Тому за умови дотатності всіх x_i неможливо створити таку ОДР, щоби вона виключала точку $\langle 0, 0, \dots \rangle$.



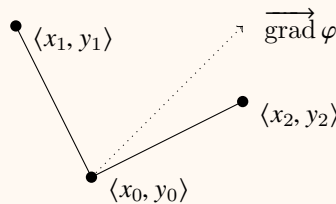
Як оцінити оптимальність сусідньої точки? Припустимо, що цінова функція має вигляд $\varphi = 10x_1 + 20x_2$. На першій ітерації ми перебуваємо в координатах $\mathbf{X}_0 = \langle 0, 0 \rangle$, а дві сусідні точки – $\mathbf{X}_A = \langle 0, 5 \rangle$ і $\mathbf{X}_B = \langle 5, 0 \rangle$. Тепер, якщо ми рухатимемось до \mathbf{X}_A , то змінюватиметься лише координати

²(англ.) *artificial basis*

ната x_2 , а $x_1 = \text{const}$. Аналогічно при переході до X_B змінюється лише x_1 . Тобто при виборі напрямку потрібно відповісти на таке питання: зміна якої зі змінних призводить до найбільшого зростання φ ? Очевидно, що якщо одночасно збільшувати обидва аргументи, то доданок $20x_2$ вносить більший вклад у зростання значення φ , ніж $10x_1$ – це можна з'ясувати і простими логічними міркуваннями, і за допомогою апарату математичного аналізу, порівнявши $\frac{\partial \varphi}{\partial x_1}$ з $\frac{\partial \varphi}{\partial x_2}$. Висновок такий: перехід до X_B дасть найбільше з можливих зростань значення цінової функції, тому оптимальний розв'язок на цій ітерації стане $X = \langle 0, 5 \rangle$.

Питання для роздумів

Нехай на поточній ітерації ми перебуваємо в точці $\langle x_0, y_0 \rangle$. Наступні сусідні точки – $\langle x_1, y_1 \rangle$ і $\langle x_2, y_2 \rangle$, а цінова функція задана як $\varphi = xc_1 + yc_2$. Чи можна за рисунком оцінити, перехід до якої точки призведе до найшвидшого зростання φ ? Чи можна також зробити це аналітично, не обраховуючи значення φ ? Якщо можна, то як? Чи працюватиме це для багатовимірних просторів?



Як оцінити оптимальність точок, перехід до яких вимагає зміну кількох координат одночасно? Якщо читач вже має відповідь на попереднє питання для роздумів, у нього, ймовірно, з'явилось кілька формул з використанням кутів між векторами або щось. Однак для цього прийнято використовувати дещо складніший з першого погляду, але простіший за обчисленнями спосіб. Головна ідея полягає в тому, щоби перехід до

кожної сусідньої точки завжди вимагав збільшення лише однієї змінної цінкової функції. В такому разі найоптимальніший випадок завжди можна буде знайти за допомогою частинного диференціювання. Згодом читач помітить, що з цього можна вивести загальні формули, ото ж розв'язок ЛП буде абстрагованим від реального геометричного зображення ОДР. Вони працюватимуть для просторів будь-якої вимірності, і не вимагатимуть операцій складніших за ділення або множення.

Отож, при розв'язування задачі симплекс-методом прийнято використовувати так звані **люзові змінні**³. Вони перетворюють нерівності виду $f(x_1, x_2, \dots) \leq b_i$ у рівності $f(x_1, x_2, \dots) + s_i = b_i$. Тоді ОДР задачі можна записати так:

$$\begin{aligned} a_{11}x_1 + \dots + a_{1m}x_n + s_1 &= b_1 \\ a_{21}x_1 + \dots + a_{2m}x_n + s_2 &= b_2 \\ &\vdots \end{aligned}$$

Або в матрично-векторному вигляді:

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + I \begin{bmatrix} s_1 \\ \vdots \\ s_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix},$$

де I – одинична матриця. Якщо позначити усі матриці через відповідні змінні, отримаємо ще коротший запис:

$$AX^T + Is = b.$$

Постановка задачі лінійного програмування, яка записана лише за допомогою рівностей, називається **канонічною**⁴.

³(англ.) *slack variables* для нерівностей зі знаком « \leq », *surplus variables* для нерівностей « \geq ».

⁴(англ.) *canonical form*

Щоби зрозуміти сенс люзових змінних, прослідкуймо за таким перетворенням:

$$\begin{aligned} f(x_1, x_2, \dots) + s_i &= b_i \Rightarrow \\ s_i &= b_i - f(x_1, x_2, \dots). \end{aligned} \quad (2.11)$$

Якщо b_i позначає обмеження на доступні ресурси, а f – функція, яка обчислює значення використаних, то логічно, що s_i позначатиме кількість невикористаних ресурсів. Отже, якщо ми змінюватимемо якесь x_i , то для того, щоби отримані рівняння справджувались, потрібно буде одразу перераховувати $s_i \forall i$. Так само й навпаки: якщо змінити якесь s_i , потрібно одночасно перерахувати решту змінних.

Це досить очевидно: якщо, наприклад, виробництво одиниці продукту x_1 вимагає 2 одиниці ресурсу s_1 та 3 одиниці s_2 , то нові значення змінних будуть $s_1 - 2\Delta x_1$ та $s_2 - 3\Delta x_1$ відповідно. Тут Δx позначає зміну x .

Проаналізуймо двовимірну задачу з ціновою функцією $\varphi = c_1 x_1 + c_2 x_2$ та ОДР як на Рис. (2.10), щоби зрозуміти геометричний сенс люзових змінних. Уявімо, що в процесі пошуку оптимального розв'язку алгоритм проходить маршрут $\mathbf{X}_0 \rightarrow \mathbf{X}_A \rightarrow \mathbf{X}_B$. Спробуймо змоделювати його поведінку.

Крок 1. Перебуваючи в \mathbf{X}_0 , ми можемо перейти до \mathbf{X}_A або \mathbf{X}_C . Перехід до обох цих точок вимагає зміну лише або x_1 або x_2 . Отже для оцінки оптимальності потрібно мати запис цінової функції у цих змінних, а тоді порівняти $\frac{\partial \varphi(x_1, x_2)}{\partial x_1}$ та $\frac{\partial \varphi(x_1, x_2)}{\partial x_2}$. Оскільки φ лінійна функція, то для порівняння частинних похідних потрібно лише порівняти коефіцієнти біля відповідних змінних – c_1 та c_2 . Припустимо, що $c_2 > c_1$, тому

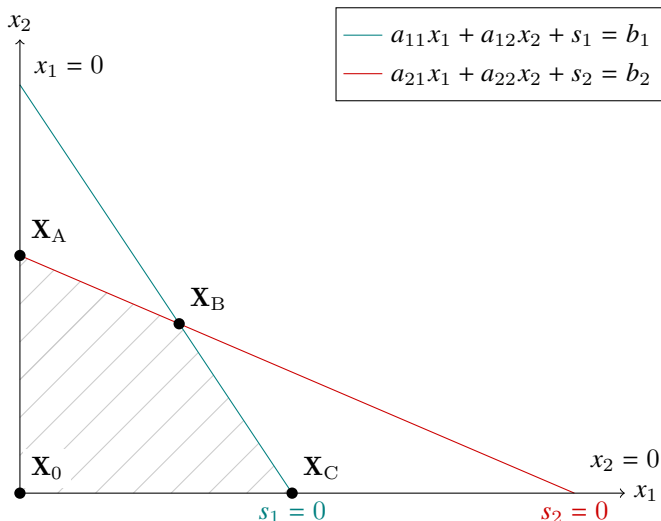


Рис. 2.10: Заштрихована область позначає ОДР. При руху по осі x_1 змінна x_2 залишається постійною. Аналогічно з віссю x_2 . А от при руху по лінії одного з рівнянь незмінними залишаються s_1 або s_2 , коли ж x_1 та x_2 одночасно змінюються.

найвигідніше збільшувати x_2 . Отже збільшуємо цю змінну доки не отримаємо координату точки X_A . Опісля – перераховуємо значення s_1 та s_2 . В цьому випадку x_1 та x_2 називаються **вільними змінними**, а s_1 та s_2 – **базисними змінними**⁵.

Крок 2. Точка X_A має дві сусідні: X_0 та X_B . Перехід до кожної з них вимагає зміну лише або x_2 або s_2 . Тому потрібно отримати цінкову функцію, виражену у цих змінних. Це зробити зовсім нескладно. Візьмімо з ОДР задачі рівняння, яке містить s_2 . Його можна переписати так:

⁵(англ.) *nonbasis variables* та *basis variables* відповідно

$$a_{21}x_1 + a_{22}x_2 + s_2 = b_2 \Rightarrow$$

$$x_1 = \frac{b_2 - s_2 - a_{22}x_2}{a_{21}}.$$

Тепер, підставляючи x_1 в φ , отримуємо запис цінкової функції виражений лише у змінних x_2 та s_2 :

$$\varphi = c_1x_1 + c_2x_2 =$$

$$c_1 \frac{b_2 - s_2 - a_{22}x_2}{a_{21}} + c_2x_2$$

$$c_1a_{21}(b_2 - s_2 - a_{22}x_2) + a_{21}c_2x_2 =$$

$$c_1a_{21}b_2 - c_1a_{21}s_2 - c_1a_{21}a_{22}x_2 + a_{21}c_2x_2.$$

В цьому випадку ми кажемо, що змінна s_2 **вийшла з базису**⁶, а x_1 – **увійшла в базис**⁷ замість неї.

Отримана функція завжди буде лінійною, тому з неї можна легко отримати частинні похідні $\frac{\partial \varphi(s_2, x_2)}{\partial s_2} = c_1a_{21}$ та $\frac{\partial \varphi(s_2, x_2)}{\partial x_2} = c_1a_{21}a_{22} + a_{21}c_2$. Припустимо, що порівнявши ці значення, ми з'ясували, що найвигодніше збільшувати s_2 . Це потрібно робити доти, доки ми не досягнемо межі ОДР. Якщо отриманий розв'язок $\langle x_1, x_2 \rangle$ не буде оптимальним, потрібно знову оцінити, яку змінну варто замінити в базисі, отримати новий запис φ , і продовжувати змінювати вільні змінні.

Звичайно, якщо ми вже переконались, що $\varphi(\mathbf{X}_0) < \varphi(\mathbf{X}_A)$, то досить очевидно, що від \mathbf{X}_A залишається перейти лише до \mathbf{X}_B – тобто, немає сенсу повертатись назад. Такі висновки мають хороше практичне значення, однак у комп'ютерній реалізації ми не надто збільшимо ефективність, запам'ятовуючи вже пройдені маршрути.

⁶(англ.) *leaving basis variable*

⁷(англ.) *entering basis variable*

Питання для роздумів

Маємо постановку ЛП з n змінними та m рівняннями, тобто $A \in \mathbb{R}^{n \times m}$. Нехай поточний базис – $\langle x_1, x_2, \dots, x_n \rangle$. З нього виходить змінна x_7 , а натомість входить s_5 . Якщо пронумерувати всі рівняння ОДР, то з якого з них потрібно виразити x_7 , щоби потім підставити формулювання з s_5 у цінovu функцію? А якби з базису виходила змінна x_{10} , і замість неї входила б s_{28} ? Чи можна вивести якусь загальну формулу нових коефіцієнтів при змінних у цінovій функції з потрібним базисом?

Спробуймо покроково розв'язати просту задачу з умовою (2.12), щоби подивитись, чи це дійсно спрацює. Для наглядності працюватимемо спочатку з графікою.

$$\begin{array}{ll}
 \text{оптимізувати} & \varphi = 2x_1 + 3x_2 \rightarrow \max \\
 \text{з обмеженнями} & 2x_1 + x_2 \leq 4; \\
 & x_1 + 3x_2 \leq 4.
 \end{array} \tag{2.12}$$

2.3.1 Графічне розв'язування

Ітерація 1. Стаємо в точку $x_1 = x_2 = 0$. Оскільки вона є допустимим розв'язком, і ми починаємо алгоритм з неї, її справедливо назвати **опорним планом**⁸. Поточний стан алгоритму продемонстровано на Рис. ??.

⁸(англ.) *basic feasible solution*

Рядок ціни задає коефіцієнти для цінової функції:

$$\varphi = 2x_1 + 3x_2 + 0s_1 + 0s_2$$

Для збільшення x_1 на 1 потрібно зменшити s_1 на $a_{11} = 2$.
Так само для решти чисел таблиці.

Аналогічно,

$$x_2 + 1 \rightarrow s_2 - a_{22}.$$

Рядок змінних відображає усі змінні, задіяні в обчисленнях.

Якщо збільшимо x_2 , то наскільки щонайбільш, враховуючи доступні s_1 та s_2 ?

$$\text{межа} = \frac{\text{план}}{a_{ij}}$$

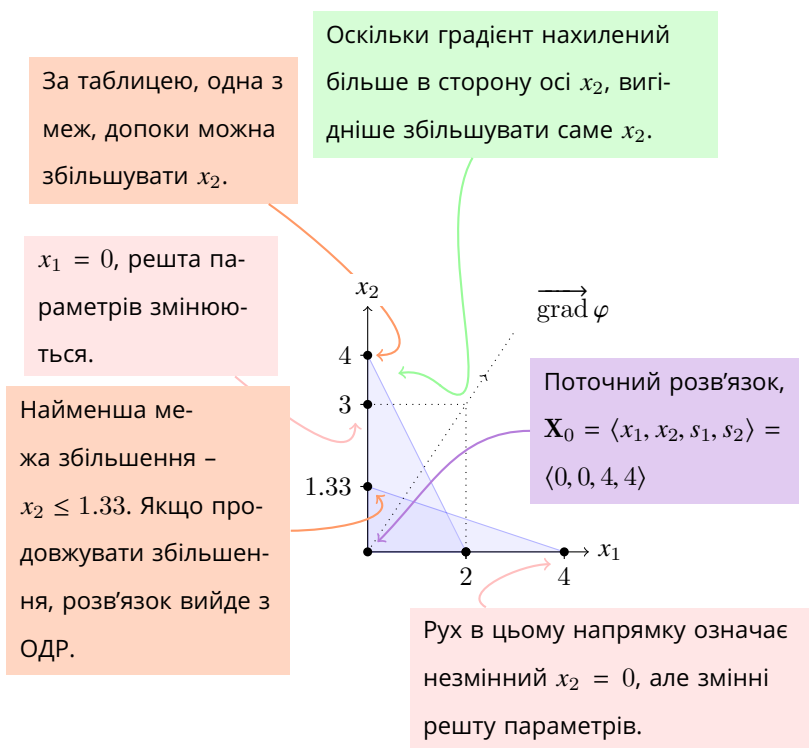
Поточне значення s_1 .

		Ціна					
		2	3	0	0		
		Змінні				План	Межа
		x_1	x_2	s_1	s_2		
Базис	s_1	2	1	1	0	4	4
	s_2	1	3	0	1	4	1.33
Ризик		0	0	0	0		
Вигода		2	3	0	0		

Значення втраченої вигоди від прийнятого рішення. На поточному кроці – нульовий рядок.

Базисні змінні – це ті, що не дорівнюють нулю, і не підлягають збільшенню чи зменшенню.
Отже змінюватимемо x_1 або x_2 .

На поточному кроці немає жодного ризику, тому вигода дорівнює ціні. Оскільки це найбільше значення в рядку, найвигідніше збільшувати x_2 .



2.4 Цілочисельне лінійне програмування

2.5 Проблеми оптимального вибору

Припустимо таку ситуацію, коли ми маємо набір різних рішень, з яких складається розв'язок проблеми. Наприклад, *складне обчислення можна розпаралелити на n процесорів, кожен з яких має свою продуктивність операцій/с та потужність у ватах. Скільки і які процесори потрібно увімкнути, щоби продуктивності вистачило на виконання деякого обчислення у вказаний час, але спожити щонайменше енергії?* Тоді набір всіх можливих рішень

$$P = \{ \text{увімкнути процесор 1, увімкнути процесор 2, \dots,} \\ \text{увімкнути процесор } n \}$$

Відповідно, будь-який розв'язок задачі буде лише підмножиною P . Однак такими формулюваннями оперувати дуже не дуже зручно, тому краще ввести характеристичний вектор \mathbf{X} , кожен елемент x_i якого позначатиме, чи прийняте деяке i -те можливе рішення. Так, якщо $x_i = 0$, то i -ий процесор варто вимкнути, але якщо $x_i = 1$ – то навпаки.

Формально такий випадок можна записати як $\mathbf{X} \in \{0, 1\}^n$, або простіше – $\forall i : x_i \in \{0, 1\}$.

Розділ 3

Нелінійне програмування

Розділ 4

Динамічне програмування