

Тема № 2. Методи прийняття рішень за умов апріорної визначеності

Зміст

Методи багатокритеріальної оптимізації	2
Метод дерева рішень [15]	8
Детермінований метод Белмана	12

Методи багатокритеріальної оптимізації

Задача багатокритеріальної оптимізації (1.4) є некоректною, тому що у загальному випадку не забезпечує визначення єдиного оптимального рішення з допустимої множини X . Цю некоректність можна усунути шляхом регуляризації задачі, тобто введенням деякої додаткової інформації, математичних співвідношень або правил, що дають змогу забезпечити вибір єдиного рішення. При реалізації евристичного підходу джерелом регуляризаційної інформації є особа, що приймає рішення, (ОПР) на інтуїтивному рівні.

Формальний (конструктивний) підхід ґрунтується на визначенні формальних правил вибору єдиного рішення із області компромісів. При цьому можливі два випадки.

1. Рішення ранжирують за порядком спадання чи зростання якості на множині компромісу X^c або, у загальному випадку, на всій допустимій множині X , тобто визначають строгий $x_1 \succ x_2 \succ \dots \succ x_n$, або нестрогий $x_1 \succ x_2 \leftrightarrow x_3 \succ x_4 \succ \dots \succ x_n$ порядок, де \succ і \leftrightarrow є знаками відповідно переваги й еквівалентності. Потім знаходять екстремальний (крайній) елемент ряду.
2. Безпосередньо визначають екстремальне рішення $x^* \succ (\forall x \in X)$.

Загальний підхід до розв'язання цієї проблеми полягає в трансформації багатокритеріальної задачі в однокритеріальну зі скалярним критерієм. Існує декілька способів трансформації багатокритеріальних оптимізаційних задач в однокритеріальні. Розглянемо основні з них [12].

Метод головного критерію базується на виділенні головного критерію і переведенні всіх інших критеріїв у обмеження. Для цього проводиться аналіз конкретних особливостей багатокритеріальної задачі, із множини частинних критеріїв вибирається один – найважливіший, який береться за єдиний критерій оптимізації. Для кожного з інших частинних критеріїв призначається граничне значення, нижче якого він не може опускатися. Отже, всі частинні критерії, крім одного, перетворюються в обмеження, що

додатково звужують множину допустимих рішень X . Тоді вихідна багатокритеріальна задача (1.4) перетворюється в однокритеріальну[^]

$$x^* = \arg \mathop{extr}_{x \in X} k^*(x), \quad k_i(x) \geq (\leq) k_{\text{інг}}(x), \quad i = \overline{1, n-1}, \quad (2.1)$$

де $k^*(x)$ - оптимізаційний скалярний критерій; $k_{\text{інг}}(x)$ – найгірші допустимі значення частинних критеріїв–обмежень; знак " \geq " використовується для критеріїв, що потрібно максимізувати, а знак " \leq " мінімізувати. Вибір головного (оптимізаційного) критерію і рівнів обмежень $k_{\text{інг}}(x)$ для всіх інших критеріїв є суб'єктивною операцією, здійснюваною експертами або ОПР. Слід зазначити, що можна розглянути кілька різних варіантів і порівняти результати.

Реалізуючи розглянутий метод, потрібно звертати особливу увагу на те, щоб допустима множина рішень, яка задана частинними критеріями-обмеженнями, не виявилась порожньою.

Функціонально-вартісний аналіз. Вихідна множина частинних критеріїв $K = \{k_i(x)\}, i = \overline{1, n}$, що за визначенням досить повно характеризує ефективність допустимих рішень $x \in X$ у цілому, розбивається на дві підмножини: $K_{\Phi} = \{k_j(x)\}, j = \overline{1, m}$ і $K_B = \{k_l(x)\}, l = \overline{1, L}; m + L = n$. Перша група критеріїв K_{Φ} характеризує функціональну якість рішення, тобто ступінь досягнення цілі системи, що аналізується, а друга група K_B – витрати, потрібні для досягнення цілі. З кожної із зазначених підмножин виділяється один головний критерій; позначимо їх відповідно K_{Φ}^* і K_B^* , а інші частинні критерії переводяться в обмеження. Одержуємо оптимізаційну задачу з двома скалярними критеріями, яку треба звести до однокритеріальної. У функціонально-вартісному аналізі використовуються такі оптимізаційні критерії.

1. Якщо обидва критерії K_{Φ}^* і K_B^* , мають однакову вимірність або їх можна звести до однакової вимірності, то використовується узагальнений оптимізаційний критерій

$$K_1(x) = K_{\Phi}^*(x) - K_B^*(x) \quad (2.2)$$

а оптимальне рішення визначається за схемою

$$x_1^* = \max_{x \in X} [K_\Phi^* - K_B^*(x)], \quad (2.3)$$

де X – звужена додатковими обмеженнями область допустимих рішень. Критерій $K_1(x)$ може бути інтерпретований як прибуток системи.

2. Якщо критерії K_Φ^* і K_B^* , мають різну вимірність, то використовується критерій

$$K_2(x) = \frac{K_\Phi^*(x)}{K_B^*(x)}, \quad (2.4)$$

а оптимальне розв'язання має вигляд:

$$x_2^* = \max_{x \in X} \left[\frac{K_\Phi^*(x)}{K_B^*(x)} \right]. \quad (2.5)$$

Критерій $K_2(x)$ являє собою нормований на одиницю витрат ефект системи.

В окремому випадку для зведення двокритеріальної задачі до однокритеріальної у функціонально-вартісному аналізі використовується принцип головного критерію. Тоді один із двох розглянутих критеріїв K_Φ^* або K_B^* перетворюється в додаткове обмеження, і оптимізаційні задачі мають відповідно вигляд:

$$x_3 = \min_{\substack{x \in X \\ K_\Phi^* \geq K_\Phi^\Pi}} K_B^*(x); \quad (2.6)$$

$$x_4 = \max_{x \in X^*} K_{\Phi}^*(x), \quad (2.7)$$

$$K_B \leq K_B^{\Pi}$$

де $K_{\Phi}^{\Pi}, K_B^{\Pi}$ – допустимі рівні відповідних критеріїв.

Функціонально-вартісний аналіз застосовується при дослідженні соціально-економічних і технічних систем, які мають однакову якісну залежність ефекту K_{Φ}^* від витрат K_{Φ}^* , що описується логістичною (іноді її називають *S*-подібною) кривою.

У загальному випадку ця залежність широко використовується для аналізу *життєвого циклу систем*, що починається в момент зародження ідеї (принципу) створення системи і містить у собі етапи реалізації, інтенсивного розвитку, морального старіння і завершується заміною її новою, більш прогресивною системою. При цьому поряд із задачами оцінювання і порівнювання ефективності різних варіантів рішень на всіх етапах (розроблення, створення, модернізація, реінжиніринг) життєвого циклу виникають специфічні завдання стратегічного планування розвитку, тобто визначення моментів і розмірів витрат на нове розроблення, модернізацію існуючих систем, заміну їх новими тощо.

Метод послідовної оптимізації (лексикографічного впорядкування). Ідея цього методу полягає в трансформації багатокритеріальної оптимізаційної задачі в упорядковану послідовність однокритеріальних. Для цього всі частинні критерії упорядковуються в послідовності спадання важливості, тобто встановлюється лінійний порядок

$$k_1 \succ k_2 \succ \dots \succ k_n. \quad (2.8)$$

У цій послідовності розв'язуються однокритеріальні оптимізаційні задачі за кожним частинним критерієм. Метод послідовної оптимізації зводиться до правила впорядкування слів за алфавітом при створенні словників, тому його іноді називають методом лексикографічного упорядкування рішень.

Відповідно до принципу послідовної оптимізації з рішень $u \in X$, $\nu \in X$ перше більш переважно, тобто $u \succ \nu$, якщо виконуються умови:

$$k_j(u) = k_j(\nu), \quad k_i(u) > k_i(\nu), \quad j = \overline{1, i-1}, \quad i = \overline{1, n}. \quad (2.9)$$

Звідси найкраще рішення визначається за наступною схемою. На першому кроці з вихідної множини допустимих рішень X виділяється підмножина x_1^* рішень, які є еквівалентними (рівноцінними) за першим (найважливішим) критерієм. Для цього розв'язується однокритеріальна оптимізаційна задача

$$x_1^* = \arg \max_{x \in X} k_1(x). \quad (2.10)$$

Якщо множина x_1^* містить понад одне рішення, переходимо до наступного етапу, тобто розв'язуємо задачу вибору еквівалентних рішень за другим за важливістю критерієм, але вже з множини x_1^* :

$$x_2^* = \arg \max_{\substack{x \in X \\ x \in x_1^*}} k_2(x). \quad (2.11)$$

У загальному випадку

$$x_i^* = \arg \max_{\substack{x \in X \\ x \in x_{i-1}^*}} k_i(x), \quad i = \overline{1, n}.$$

Оптимізація продовжується доти, поки на l -му кроці не дістанемо єдине рішення або не вичерпаються всі критерії. Якщо всі частинні критерії вичерпані, але єдине рішення не отримано, формуються додаткові критерії.

Отримане єдине рішення береться якнайкраще (оптимальне). Якщо треба ранжувати всю множину рішень $x \in X$ то отримане найкраще рішення виключається з X , і на множині рішень, що залишилися, повторюється вищеописана процедура. Потім визначаємо друге за якістю розв'язання і так далі, поки не будуть упорядковані всі розв'язання.

Коли вже на перших кроках оптимізації знайдено єдине рішення, усі наступні частинні критерії втрачають сенс. Тут може бути застосована *схема поступки*. Відповідно до цієї схеми ОПР призначає допустимий рівень зниження $\Delta k_i(x)$ частинного критерію, за яким отримано вищезазначене єдине рішення порівняно з екстремальним значенням цього критерію

$$x_{i+1}^* = \arg \max_{x \in X} [k_{i+1}(x)], \quad i = \overline{1, n-1} \quad (2.12)$$

за умови $k_i(x) > k_i(x_i^*) - \Delta k_i(x)$.

Вибір розміру поступки є евристичною операцією, яка здійснюється ОПР і залежить від особливостей задачі оптимізації. Залежність $k_i(x_i^*) = f(\Delta k_{i-1}(x))$ має неявний характер і у більшості випадків не є апіорно передбаченою. Тому для вибору значення поступки $\Delta k_i(x)$ часто використовуються людино-машинні процедури, коли ОПР призначає деякий ряд значень поступки і, аналізуючи результати оптимізації для кожного з них, приймає рішення про доцільний рівень поступки.

Метод послідовної оптимізації широко застосовується в ситуаціях, коли ОПР має у своєму розпорядженні тільки якісну інформацію про важливість частинних критеріїв.

Метод формування узагальненого скалярного критерію, що враховує всі різномірні частинні критерії. У цьому випадку єдиний скалярний критерій K формується як функціонал частинних критеріїв

$$K = F[k_i(x)], \quad i = \overline{1, n}. \quad (2.13)$$

Це найбільш загальний і універсальний підхід до розв'язання задачі багатокритеріальної оптимізації, відомий як *проблема багатфакторного оцінювання*.

Метод дерева рішень [15]

Розглянемо багатокрокову задачу прийняття рішень, що може розглядатися, наприклад, при максимізації прибутку або мінімізації витрат. Нехай одержання прибутку реалізується на кожному кроці процесу прийняття рішень, а потім ці прибутки додаються. Модель такої багатокрокової задачі прийняття рішень може бути подано у вигляді орієнтованого графа, що називається деревом рішень і вказує шлях із заданої множини початкових вершин у задану множину його кінцевих вершин. При цьому з кожною вершиною графа асоціюється деякий стан S_i , в якому знаходиться об'єкт прийняття рішень, а дуги, що виходять з вершини, відповідають можливим переходам з одного стану в інший залежно від рішень, що приймаються. На рис. 2.1 наведено приклад детермінованого дерева рішень. Тут зліва знаходяться заштриховані початкові вершини, а справа – кінцеві вершини. При цьому кожна гілка графа має свою вагу – дійсне число, що враховує витрати на перехід в інший стан. Основне завдання полягає в оптимальному виборі початкової вершини із множини допустимих та оптимізації шляху із неї в будь-яку допустиму кінцеву вершину. Задача оптимізації допустимого шляху полягає в мінімізації сумарних витрат і адекватна задачі вибору мінімального шляху в графі. У частковому випадку множини допустимих початкових і кінцевих вершин можуть бути одноелементними.

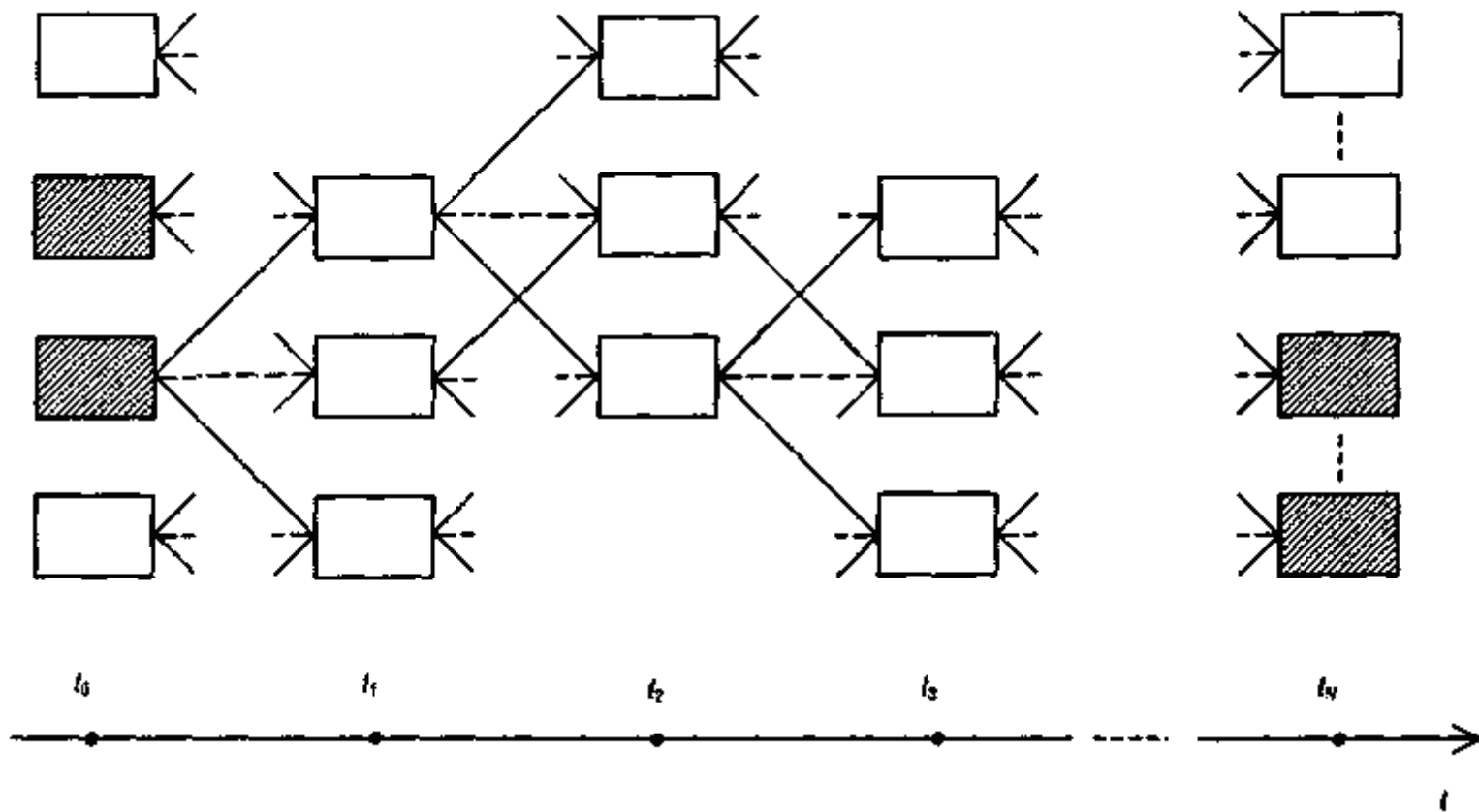


Рис. 2.1. Дерево рішень за умов визначеності

У наведеному прикладі граф містить тільки так звані основні, або "розв'язувальні" вершини (рис. 2.2). При цьому вважається, що система (об'єкт прийняття рішень) знаходиться у визначеному фазовому стані S_i , число яких скінчено. Із вершини S_i виходить декілька дуг графа, кожна з яких відповідає різним рішенням, що можуть прийматися в даному стані. Вибір конкретної альтернативи d_j призводить до переходу системи в нову розв'язувальну вершину (новий стан).

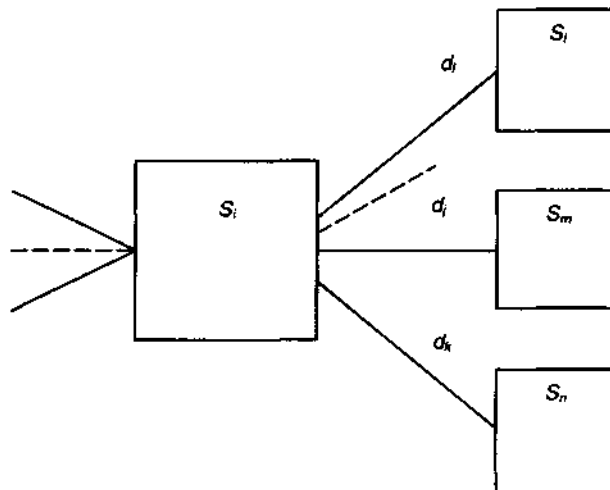


Рис. 2.2. "Розв'язувальні" вершини

Якщо вибір конкретного рішення d_j визначає не новий стан системи, а задає деяку лотерею на множині можливих нових станів (щільність розподілу ймовірностей), то має місце ймовірнісний зв'язок між вершинами графа (рис. 2.3).

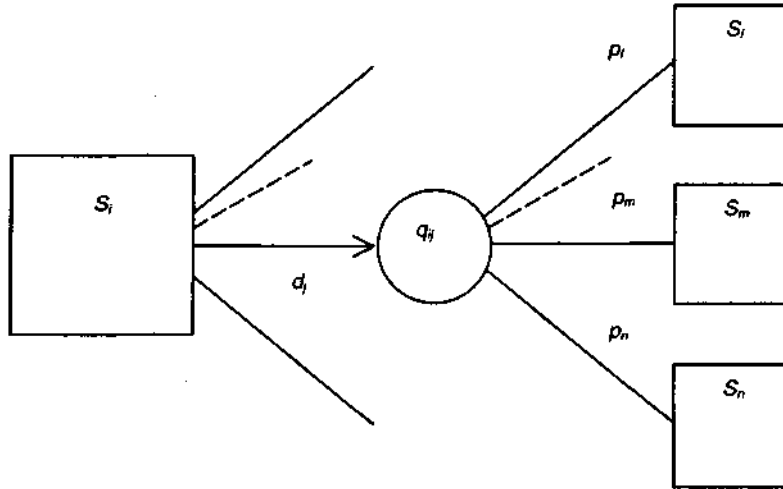


Рис. 2.3. Імовірнісний зв'язок вершин

У ймовірнісному дереві рішень (рис. 2.3) після вибору d_j ми попадаємо в деяку випадкову допоміжну вершину $q_{i,j}$ і далі переходимо в один із можливих для даного кроку станів S_1, \dots, S_m відповідно до заданих ймовірностей p_1, \dots, p_m , де $\sum_{j=1}^m p_j = 1$.

У випадку апріорної невизначеності структура рис. 2.3 зберігається, але стрілки, що виходять із вершини $q_{i,j}$ вже не будуть мати вагових коефіцієнтів, оскільки відповідні ймовірності відсутні.

Детермінований метод Беллмана

Розглянемо конкретний приклад детермінованого дерева рішень (рис. 2.4) [15].

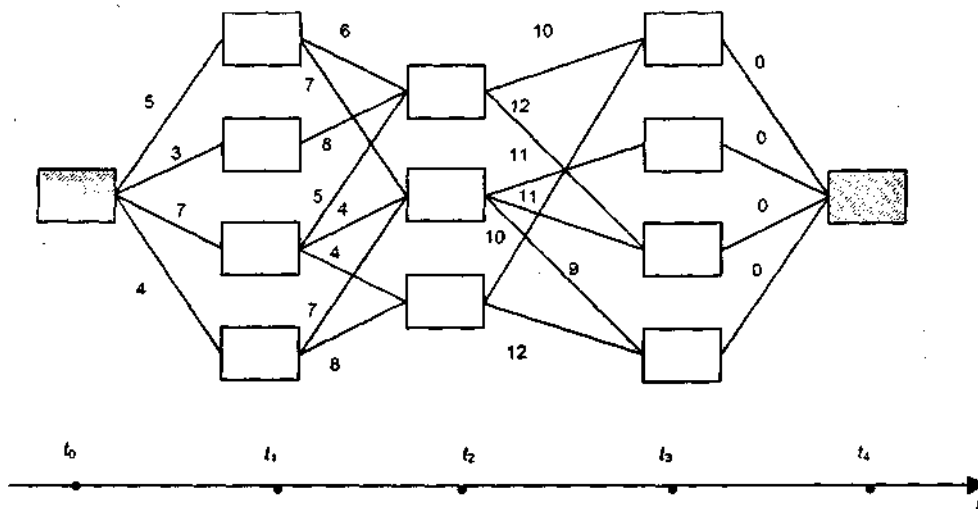


Рис. 2.4. Дерево рішень із заданими локальними витратами

На рис. 2.4 кінцевими є всі чотири вершини, що відповідають моменту часу t_3 . Але шляхом введення фіктивної вершини для t_4 вдалося звести задачу до графа з однією кінцевою вершиною. Числа у дуг графа означають трудомісткість (витрати), що забезпечують перехід від однієї "розв'язувальної" вершини до іншої. Головне завдання полягає у виборі оптимального в сенсі сумарних витрат шляху, який з'єднає початкову та кінцеву вершини.

Основна ідея методу Беллмана складається із двох моментів:

1 Завдання пошуку оптимального шляху починає розв'язуватися з кінця.

2 Як початкова вершина послідовно розглядаються всі без винятку вершини графа при одній і тій самій кінцевій вершині.

Реалізуємо метод Беллмана для показаного на рис. 2.4 графа. Будемо просуватися по графу справа-наліво, відмічаючи всередині кожного квадратика-вершини числа, які дорівнюють найменшим із можливих сумарних витрат при просуванні із поточної початкової вершини по оптимальному шляху, та зазначаючи оптимальні напрямки руху з кожної вершини за допомогою стрілок. Наприклад, якщо має місце ситуація, зображена на рис. 2.5,а, де вершини 2, 3, 4 вже помічено в правому верхньому куті квадратика, і треба помітити вершину 1, то будемо розмірковувати так. Якщо як початкову взяти вершину 1, то пошук оптимального шляху із неї зводиться до порівняння трьох чисел: $5+10=15$, $4+10=14$, $4+9=13$. Найменше із цих чисел – 13, і тому саме це число записується у вершину 1, а стрілкою з'єднуються вершини 1 і 4 (рис. 2.3.2 б) і відповідно оптимальний маршрут з вершини 1 проходить через вершину 4.

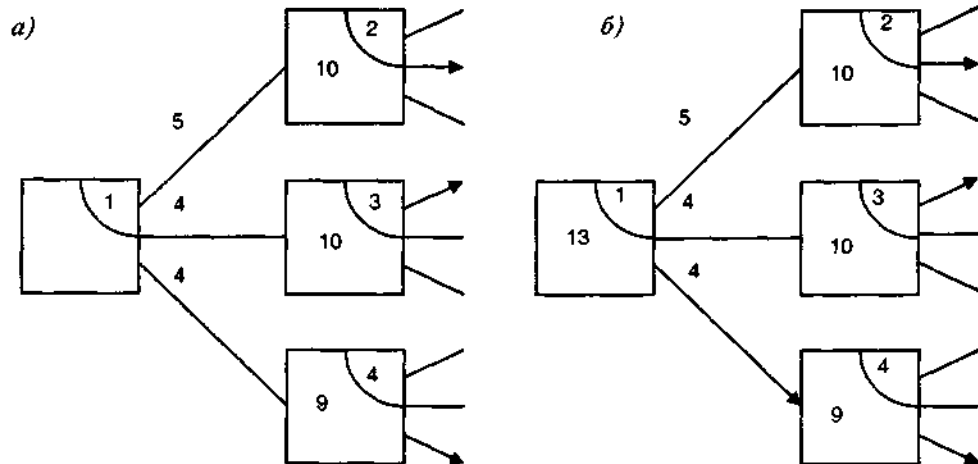


Рис. 2.5. Метод динамічного програмування Беллмана

Просуваючись справа-наліво і послідовно обробляючи вертикальні шари вершин, розмічується весь граф (рис. 2.6).

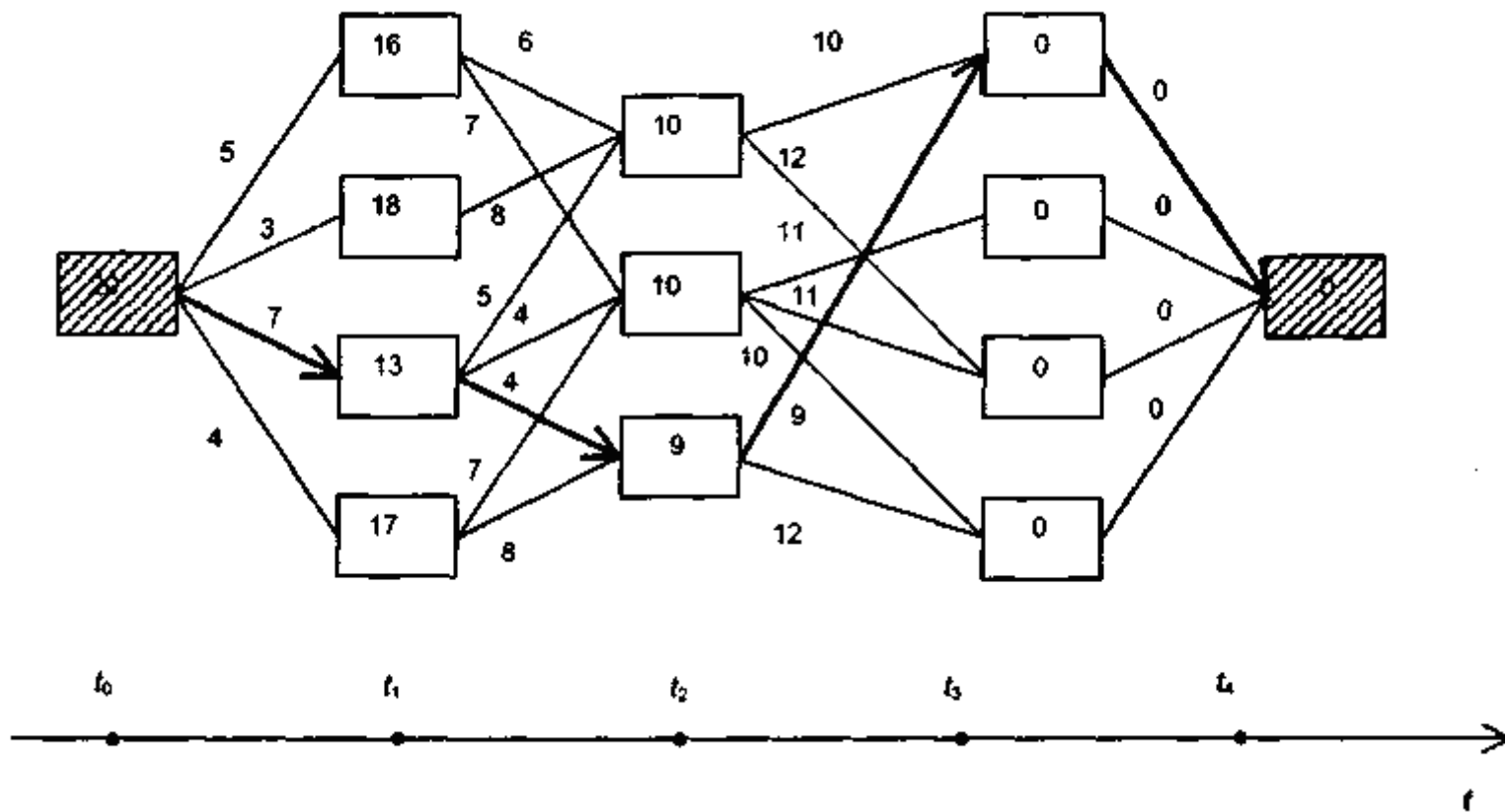


Рис. 2.6. Розмічений граф

Для відновлення оптимального шляху в графі (2.4) тепер достатньо пройти вже зліва-направо у напрямку стрілок, починаючи із вже поміченої початкової вершини (оптимальний шлях на рис. 2.6 виділено жирними стрілками). Отримане на етапі розмітки графа число 20 означає мінімальні можливі витрати. Оскільки оптимальний шлях може бути не єдиним, то за методом Беллмана одночасно знаходяться всі можливі оптимальні шляхи. При цьому варто розуміти, що глобально оптимальна стратегія не є суперпозицією локально оптимальних стратегій. Наприклад, при спробі побудови оптимального шляху на графі (рис. 2.6) при просуванні зліва-направо і виборі кожного разу стрілок з найменшою вагою слід очікувати невдачу, оскільки отримаємо результат: $3+8+10=21$, що більше 20.