Аналіз складних проектів у встановлений термін на основі потокових моделей з обмеженою пропускною здатністю

Будівельне виробництво характеризується високим рівнем розподілу праці, складністю виконуваних проектів, великою кількістю порівнюваних альтернативних варіантів. Отже тут формується головне завдання - вибрати з альтернативних варіантів ефективний. В цьому випадку зручно користуватися моделями.

Для проведення подальшої роботи скористаємося такими поняттями як «моделювання» і «модель».

Модель - це деякий матеріальний або віртуальний об'єкт чи явище, який заміщає оригінальний об'єкт або явище, зберігаючи тільки деякі важливі його властивості, наприклад, в процесі пізнання (споглядання, аналізу і синтезу) або конструювання. Іншими словами, це аналогічні об’єкти, тобто ті, що достатньою мірою повторюють основні властивості модельованого об'єкту або явища (об’єкту-прототипу).

Моделювання - дослідження об'єктів пізнання на їх моделях; побудова і вивчення моделей реально існуючих предметів, процесів або явищ з метою отримання пояснень цих явищ, а також для прогнозу явищ, що цікавлять дослідника.

Виробничий процес можна представити у вигляді віртуальної, описової або графічної моделі.

Вимоги до складання виробничих моделей:

1) подібність об’єкту-прототипу;

2) простота і наочність сприйняття;

3) зручність для проведення аналізу;

4) віддзеркалення повного переліку робіт, послідовності їх виконання і взаємозв'язку.

Будь-яку виробничу модель можна зобразити за допомогою лінійних графіків, циклограм і сітьових моделей. Особливу зацікавленість викликає останній спосіб відображення виробничих моделей, бо має ряд переваг в раціональності його використання:

1) оптимальне відображення послідовності виконання складного проекту;

2) забезпечення керівника і виконавців всією необхідною інформацією для ухвалення управлінських рішень;

3) встановлення чіткого взаємозв'язку робіт;

4) відображення наочної технологічної послідовності;

5) аналіз ходу будівництва у просторі та часі;

6) об'єднання в одній моделі всього комплексу робіт;

7) надання можливості широко використовувати обчислювальні програмні комплекси для розрахунку.

Графічне відображення будівельно-монтажних робіт у вигляді сітьової моделі наведене на рис. 1.2.

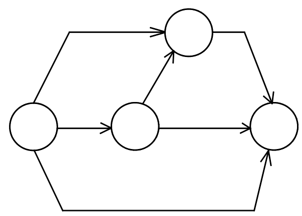

Рисунок 1.2. Графічне відображення будівельно-монтажних робіт у вигляді сітьової моделі

Сітьова модель - це така організаційно-технологічна модель, яка відображає комплекс робіт, операцій і подій, пов'язаних з реалізацією деякого будівельного проекту в технологічній і логічній послідовності і зв'язку. У основі побудови сітьової моделі полягає теорія графів. **Теорія графів** - це розділ дискретної математики, що вивчає властивості графів.

У загальному розумінні «**граф**» - це сукупність об'єктів із зв'язками між ними, об'єкти являють собою вершини (вузли) графа, а зв'язки - дуги (ребра).

Для різних сфер застосування види графів можуть розрізнятися спрямованістю, обмеженнями за кількістю зв'язків і додатковими даними про вершини або ребра.

Вузлами графу (кружками) зображені події, тобто моменти початку або закінчення кожної роботи, її стан. Ребрами графу (стрілками - дугами) зображені процеси (операції), вістрям стрілки показаний напрям в часі від початку до кінця роботи. Напрям стрілок в сітьовій моделі визначає зв'язок між подіями в часі. Подія, в яку входить стрілка (наступна подія), не може відбутися без звершення попередньої події, з якого стрілка виходить.

Отже, виходячи з вищесказаного, в реалізації будівельних проектів перед фахівцями встає питання - вибрати з альтернативних варіантів найбільш ефективний. Цей вибір ми проводимо за показником критерію оптимальності.

Критерій оптимальності – це ознака, за якою порівнюються і оцінюються варіанти певних дій для досягнення поставленої мети. Якщо процес вибору рішень, описати функцією, шукані змінні якої є допустимими і такими, що описують рух до мети, то таку функцію називають **цільовою**, а рішення - **оптимальним**. Таким чином, встановити оптимальне рішення означає визначити екстремум функції.

Критеріям оптимальності притаманний ряд важливих характеристик [5, 21]:

1) критерій оптимальності повинен вимірювати справжню ефективність системи;

2) критерій оптимальності повинен виражатися за кількісними одиницями виміру;

3) критерій оптимальності для задачі, яка вирішується, повинен бути один (в різних задачах можуть бути також і часні критерії, але обов’язково підпорядковані загальному критерію ефективності);

4) значення критерію оптимальності повинен визначатися достатньо точно без надмірних витрат часу та додаткових засобів;

5) критерій оптимальності повинен забезпечувати врахування всіх існуючих сторін;

6) критерій оптимальності повинен мати фізичний сенс, який робить його зрозумілим та відчутним, а також полегшує порівняння ідеальної та реальної характеристик.

Щоб встигнути закінчити будівництво об'єкту в заданий термін $T_{зад} <= T_n$ слід всі роботи, які входять до складу проекту, виконувати з певною швидкістю, узгодженою з кінцевою метою, заданим терміном введення об'єкту в експлуатацію.

Залучення ресурсів пов'язане з додатковими витратами і збільшенням змінності виробництва. Звідси формується одна з **основних цілей** організації та управління будівельного виробництва - мінімізувати залучення ресурсів з дотриманням термінів реалізації проекту.

Друга назва нашої задачі за ім'ям вчених – задача Форда-Фалкерсона [34].

Розглянемо граф $G(U, A)$,

де G – позначення графа,

U - множина вузлів (подій) графа,

A - множина операцій (дуг, робіт) $(i, j) \in A$.

Кожна робота (операція) будівельного проекту характеризується тривалістю виконання (тривалістю реалізації) - t_{ij} та інтенсивністю виробництва N_{ij}, де $(i, j) \in A$.

Серед таких умов справедлива залежність:

$$N_{ij} \cdot x_{ij} = Q_{ij}, \tag{1.1}$$

де Q_{ij} – трудомісткість роботи $(i, j) \in A$, яка залежить від обсягу (i = 1,2,..., n-1; j -2,3,..., n);

n - кількість вузлів (подій) в сітьовій моделі;

$\boldsymbol{x}$ – це пошукова величина оптимальної тривалості t_{ij} виконання кожної роботи (операції) для того, щоб встигнути завершити будівництво у зазначений термін.

По кожній роботі $(i,j) \in A$ відома мінімальна інтенсивність - N_{ij}^{D}, якій відповідає тривалість D_{ij} і d_{ij} - тривалість, що відповідає прискореній (максимальній) концентрації використання ресурсів N_{ij}^{d} (максимальна інтенсивність).

Сформулюємо математичну модель задачі.

Дана сітьова модель (D_{ij}, T^{D}), по роботі $(i,j) \in A$ відомо d_{ij} – тривалість максимальної концентрації ресурсів, та C_{ij} - "вартість" скорочення роботи на одиницю часу.

Скорочення тривалості виконання робіт $(i,j) \in A$ на величину $\Delta x_{ij} = D_{ij} - x_{ij}$ може бути забезпечено залученням додаткових ресурсів, тобто за рахунок збільшення інтенсивності виробництва і визначається:

$$\Delta N_{ij} = c_{ij} \cdot \Delta x_{ij}. \tag{1.2}$$

Задля досягнення мети потрібно визначити роботи $(i,j) \in A$, які необхідно прискорити, а також роботи, для яких необхідно зберегти нормальну тривалість D_{ij}. Іншими словами, потрібно знайти таке рішення (x_{ij}, T_n), яке мінімізує цільову функцію:

$$L(x) = \sum_{(i,j)\in A} \Delta N_{ij} = \sum_{(i,j)\in A} C_{ij}(D_{ij} - x_{ij}) \to min, \tag{1.3}$$

де $\sum_{(i,j)\in A} \Delta N_{ij}$ - сумарне залучення додаткових ресурсів для скорочення терміну будівництва.

Множину вузлів (подій) можна визначити як U=(1, 2,..., n), де вузол 1 (n=1) позначає початок реалізації проекту, а вузол n - закінчення.

Обмеження у вирішенні задачі наступні:

$$T_i - T_j + x_{ij} \le 0 \text{ для всіх } (i,j) \in A, \tag{1.4}$$

$$-T_1 + T_n \le T_з \tag{1.5}$$

$$d_{ij} \le x_{ij} \le D_{ij}, \text{ для всіх } (i,j) \in A, \qquad (1.6)$$

де T_i (T_j) - ранній термін звершення подій реалізації проекту,

$T_з$ – заданий термін на будівництво об'єкту (реалізації проекту).

Умова (1.4) відображає нерозривність мережі та $T_j = max(T_i + t_{ij})$. Умова (1.5) висловлює вимогу не перевищення заданого терміну будівництва, тобто в оптимальному рішенні значення критичного шляху $T_n \in T_{кр}$ не повинне перевищувати заданого терміну будівництва (реалізації проекту). Обмеження (1.6) визначається технологією і організацією виробництва всіх операцій $(i,j) \in A$.

Наша мета – визначити невідомі значення тривалості кожної операції x_{ij}. Для їх визначення ставимо задачу (1.3) ***L(x)***. Вид цільової функції (1.3) і обмеження мають лінійну залежність, тому сформульована задача є параметричною задачею лінійного програмування.

Для вирішення поставленої задачі потрібно перевірити вирішуваність при встановленому терміні будівництва $T_{зад}$. Використаємо для цього наступний прийом. Припустимо, що невідома тривалість дорівнює тривалості з максимальною інтенсивністю виробництва $x_{ij} = d_{ij}$, тоді критичний шлях позначимо $T^d_{кр}$. Якщо $T_{зад} \ge T^d$, то задача має рішення, у протилежному випадку рішення немає. Якщо припустити, що $x_{ij} \le D_{ij}$, то критичний шлях визначиться як $T^D_{кр}$. Таким чином нам необхідне дотримання наступної умови:

$$T^d \le T_{зад} \le T^D \qquad (1.7)$$

У подальшому рішенні визначаємо для кожного значення T_n з сегменту [$T^d \div T^D$] мінімуму функції:

$$L(x) = \sum_{(i,j)\in A} c_{ij}(D_{ij} - x_{ij}) = \left(\sum_{(i,j)\in A} c_{ij}D_{ij} - \sum_{(i,j)\in A} c_{ij}x_{ij} \right) \to min. \qquad (1.8)$$

При поставлених умовах (1.4) ÷ (1.6) задача являє собою параметричну задачу лінійного програмування. Наведена економіко-математична модель

(ЭММ) еквівалентна наведеній нижче задачі лінійного програмування з максимізацією функції мети.

Враховуючи, що в формулі (1.8)

$$\sum_{(i,j)\in A} c_{ij} D_{ij} \to const, \tag{1.9}$$

замінимо цільову функцію вихідної задачі на іншу функцію:

$$L(x) = \sum_{(i,j)\in A} c_{ij} x_{ij} \to max, \tag{1.10}$$

яка приймала б максимальне значення та відповідала умовам:

$$T_i - T_j + x_{ij} \le 0 \text{ для всіх } (i,j)\in A, \tag{1.11}$$

$$-T_1 + T_n \le T_{зад} \tag{1.12}$$

$$x_{ij} \le D_{ij} \text{ для всіх } (i,j)\in A, \tag{1.13}$$

$$-x_{ij} \le -d_{ij} \text{ для всіх } (i,j)\in A, \tag{1.14}$$

У постановці (1.10) ÷ (1.14) задача може бути вирішена універсальним симплекс-методом, який використовується для вирішення екстремальних задач лінійного програмування. Такі методи громіздкі і їх застосування доцільно тільки тоді, коли спеціальні методи виявляються недостатніми. Нижче наведемо методику приведення такої задачі до канонічного (стандартного) виду (дивись п. 1.2).

У нашому випадку пропонується прогресивний підхід, заснований на теорії двоїстості лінійного програмування в умовах доповнюючої нежорсткості. Ці властивості відображені у теоремах двоїстості [2, 3].

<u>Перша теорема двоїстості:</u> якщо одна із двоїстих задач має оптимальне рішення, то і інша також має оптимальне рішення, причому оптимальні значення цільових функцій прямої і двоїстої задач співпадають, тобто $maxF(x) = minF(y)$ або для нашої задачі $maxL(x) = minZ(f)$.

<u>Друга теорема двоїстості або теорема доповнюючої нежорсткості:</u> якщо хоч би одне оптимальне рішення одної з двоїстих задач обертає i-е обмеження цієї задачі в строгу нерівність, то i-а компонента (тобто x_i або y_i) кожного оптимального рішення другої двоїстої задачі дорівнює нулю.

Якщо ж i-а компонента хоч би одного оптимального рішення одної з двоїстих задач позитивна, то кожне оптимальне рішення іншої двоїстої задачі обертає i-е обмеження в строгу рівність.

У постановці (1.10) ÷ (1.14) задача має вигляд, аналогічний задачі мінімальної вартості проекту, тобто визначення оптимального потоку, а також володіє значною перевагою в обчислюваннях, має економічне і фізичне тлумачення, що дуже важливо в практичному застосуванні.

Підійдемо до вирішення двоїстої задачі. Досліджується задача, для якої у відповідність обмежень (1.11) ÷ (1.14) ставляться невід'ємні змінні f_{ij}, V, γ_{ij}, δ_{ij}, які називаються двоїстими. Вони наведені (перераховані) в такому ж порядку, в якому вводилися обмеження в модель.

Двоїста задача до $(1.10) \div (1.14)$ формулюється наступним чином. Необхідно мінімізувати цільову функцію:

$$L(f) = \left(T \cdot V + \sum_{(i,j) \in A} D_{ij} \cdot \gamma_{ij} - \sum_{(i,j) \in A} d_{ij} \cdot \delta_{ij} \right) \to min \qquad (1.15)$$

при умовах:

$$f_{ij} + \gamma_{ij} - \delta_{ij} = c_{ij} \quad \text{для } (i,j) \in A, \qquad (1.16)$$

$$\sum_j f_{ij} - V = 0 \qquad i=1, \qquad (1.17)$$

$$\sum_i (f_{ij} - f_{ji}) = 0 \quad \text{для всіх } i = 2,....,n-1, \qquad (1.18)$$

$$-\sum_i f_{in} + V = 0 \qquad i=n, \qquad (1.19)$$

$$f_{ij}, \gamma_{ij}, \delta_{ij} \geq 0 \qquad \text{для всіх } (i,j) \in A. \qquad (1.20)$$

Двоїста задача (1.15) ÷ (1.20) сформульована у відповідності з правилами теорії лінійного програмування. Двоїсті обмеження є рівняннями, оскільки змінні в основній задачі в явному вигляді не обмежені за знаком.

На основі математичної структури двоїсті змінні f_{ij}, які відповідають x_{ij} в прямій задачі, розглядаються як потоки в сіті з обмеженою пропускною здатністю.

Для оптимального рішення повинні виконуватися наступні результати:

$$T_i - T_j + x_{ij} < 0 \text{, то } f_{ij} = 0$$

$$T_i - T_j + x_{ij} = 0 \text{, то } f_{ij} > 0$$

$$\left.\begin{array}{r} \text{якщо } x_{ij} = D_{ij} \text{, то } \gamma_{ij} > 0 \\ \text{якщо } x_{ij} = d_{ij} \text{, то } \delta_{ij} > 0 \\ \text{якщо } x_{ij} < D_{ij} \text{, то } \gamma_{ij} = 0 \\ \text{якщо } x_{ij} > d_{ij} \text{, то } \delta_{ij} = 0 \end{array}\right\} \qquad (1.21)$$

Двоїсті змінні γ_{ij}, δ_{ij} не можуть бути одночасно позитивними, так як $D_{ij} \neq d_{ij}$. В обмеженні (1.16) значення γ_{ij} и δ_{ij} визначаються таким чином:

$$\left.\begin{array}{r} \gamma_{ij} = c_{ij} - f_{ij} \text{, при } \delta_{ij} = 0; \\ \delta_{ij} = f_{ij} - c_{ij} \text{, при } \gamma_{ij} = 0. \end{array}\right\} \qquad (1.22)$$

Тому $\gamma_{ij} = max(0, c_{ij} - f_{ij})$, при $\delta_{ij} = 0$; $\delta_{ij} = max(0, f_{ij} - c_{ij})$, при $\gamma_{ij} = 0$.

При дослідженнях всіх можливих значень f_{ij}, γ_{ij}, δ_{ij} можна виділити три випадки:

$$\left.\begin{array}{l} 1.\ \gamma_{ij} > 0; \delta_{ij} = 0; 0 \le f_{ij} \le c_{ij}; x_{ij} = D_{ij} \\ 2.\ \gamma_{ij} = 0, \delta_{ij} = 0, f_{ij} = c_{ij}, d_{ij} \le x_{ij} \le D_{ij} \end{array}\right\}$$

(1.23)

$$3.\ \gamma_{ij} = 0, \delta_{ij} > 0, f_{ij} > c_{ij}, d_{ij} = x_{ij}$$

На основі потокового алгоритму послідовно для кожного випадку визначаються f_{ij} та $T_i(T_j)$, які задовольняють умовам оптимальності:

1. $0 < f_{ij} < c_{ij}$ та $T_i - T_j + D_{ij} = 0$; $f_{ij} = 0$ та $T_i - T_j + D_{ij} < 0$

при $a_{ij}^{/} = 0$ (1.24)

2. $f_{ij} = c_{ij}$ та $T_i - T_j + x_{ij} = 0$, $d_{ij} \le x_{ij} \le D_{ij}$ при $\bar{x}_{ij} = 0$ (1.25)

3. $c_{ij} < f_{ij} < \infty$ та $T_i - T_j + d_{ij} = 0$ при $a_{ij}^{//} = 0$ (1.26)

В умовах (1.24) – (1.26) використані наступні позначення::

$$a_{ij}^{/} = T_i - T_j + D_{ij} \text{ - резерв критичності;}$$

$$a_{ij}^{//} = T_i - T_j + d_{ij} \text{ - резерв скорочення;} \qquad (1.27)$$

$$\left.\begin{aligned} \bar{x}_{ij} &= T_i - T_j + x_{ij} \\ x_{ij} &= min(D_{ij}, T_j - T_i) \end{aligned}\right\} \text{ - невідомі змінні}$$

Послідовно визначаються $\boldsymbol{f_{ij}}$ та $\boldsymbol{T_i}$ ($\boldsymbol{T_j}$), які відповідають умовам оптимальності для значень $\boldsymbol{T_n}$, які зменшуються, після чого розраховуємо невідомі змінні.

Алгоритм вирішення задачі

Основна мета організації і управління будівництвом - це мінімізація залучення ресурсів для дотримання термінів реалізації проекту, іншими словами нам потрібно скоротити тривалість проекту і укластися у виділену суму грошей.

Для виконання подальших досліджень, розглянемо наступні позначення основних складових сітьової моделі (рис. 1.3).

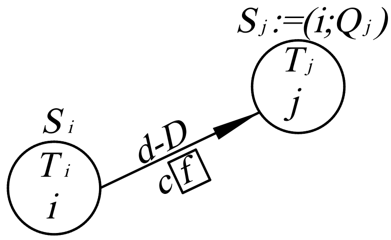

Рисунок 1.3. Елемент сітьової моделі

i, j – подія (її номер) тобто момент початку чи закінчення кожної роботи (особливість позначення: перша подія – 101, друга подія – 102, третя подія – 103 і т.д.).

$\boldsymbol{T_i}$, $\boldsymbol{T_j}$ – терміни звершення подій ***i, j*** (час настання події).

D_{ij} - тривалість мінімальної інтенсивності виробництва $N^{D}{}_{ij}$;

d_{ij} - тривалість максимальної інтенсивності виробництва N_{ij}^{d};

c_{ij} - вартість скорочення роботи на одиницю $T_{з}$;

f_{ij} - потік по дузі (роботі) (*i;j*);

S*i*; ***Sj*** **–** коди подій ***i, j*** відповідно. Позначення коду події розташовується над подією та складається з двох частин:

1) перша частина – номер (код) попередньої події, з боку якого виконувалося кодування ***j***-ї події:
2) друга частина ***Qj*** – визначається за спеціальними правилами згідно алгоритму.

Отже, загальний вигляд коду події ***j*** такий:

$$S_j{:=}(i;\ Q_j) \qquad (1.28)$$

Введемо наступні позначення:

i^*, j^- - пряма дуга (пряме кодування) – це робота, початкова подія якої закодована, а кінцева – ні;

i^-, j^* - зворотна дуга (зворотне кодування) – це робота, початкова подія якої не має коду, а кінцева – закодована.

Подальшу роботу з вирішення задачі виконуватимемо за допомогою алгоритму (рис. 3).

Алгоритм починає роботу з максимальної тривалості проекту при $t_{ij} = D_{ij} \to T^D$ (див. рис. 1.3 блок №3) та на кожній ітерації оцінюються додаткові витрати, за допомогою яких досягається деяке скорочення критичного шляху $\boldsymbol{T_{кр}}$ на значення $\boldsymbol{\Delta T_i}$.

Алгоритм складається з трьох основних кроків:

1) перший крок – перевірка можливості скорочення заданої тривалості проекту, тобто дотримання вихідної умови $T^d \le T_{зад} \le T^D$;
2) другий крок – здійснення процедури кодування подій для модифікації в сіті потоків, які відповідають двоїстій задачі;

3) третій крок – скорочення тривалості проекту, якщо на другому кроці алгоритму досягається непрорив сіті, тобто кінцева ***n***-а подія коду не отримала.

В якості вихідних даних приймаємо $\boldsymbol{T_1 = 0}$, $\boldsymbol{T_j = max\,(T_j + D_{ij})}$ (див. рис. 1.3 блок №3) всі дугові потоки $\boldsymbol{f_{ij}}$ можуть бути прийняті нульовими або дорівнюватимуть якому-небудь іншому значенню. Але тут важливе забезпечення допустимості початкового варіанту, а воно буде тоді, коли задовольниться умова збереження потоку в сіті. Нульовий вектор $\boldsymbol{f_{ij}}$, автоматично забезпечує допустимість. Цьому правилу ми слідуватимемо при вирішенні задачі.

У нашій задачі вузол №1 (початкова подія) має постійну позначку (0; ∞).

Розглянемо процедуру кодування подій (див. рис. 1.3 блоки №4, 6). Розділимо її на дві частини:

1) пряме кодування - збільшення потоку уздовж прямих дуг (робіт);

2) зворотне кодування - тобто зменшення потоку уздовж зворотних дуг.

<u>Правила кодування прямих дуг</u> (i^*, j^-):

1) розглянемо наступну подію ***j*** у складі роботи ***(i, j)***, яка не має коду (тобто поки що незакодована), при цьому попередня подія ***i*** має код (закодована подія). Для роботи ***(i, j)*** виконуються умови $a'_{ij} = 0$ та $f_{ij} < c_{ij}$. Тоді наступна подія ***j*** отримує код $(+i, Q_j)$, де друга частина коду визначається за формулою:

$$Q_j = min(Q_i, c_{ij} - f_{ij}). \tag{1.29}$$

Потік f_{ij} можна збільшити на мінімальне значення, яке знаходиться між величиною потоку в ***i***-й події і величиною, необхідної для досягнення величини c_{ij}, яка обмежує умови потоку;

2) розглянемо наступну подію ***j*** у складі роботи ***(i, j)***, яка не має коду (тобто поки що незакодована), при цьому попередня подія ***i*** має код (закодована подія). Для роботи ***(i, j)*** виконується умова $a''_{ij} = 0$. Тоді наступна

подія ***j*** отримує код $(+i, Q_i)$, тобто друга частина коду події ***j*** дорівнює другій частині коду події ***i***:

$$Q_i = Q_j; \tag{1.30}$$

3) розглянемо наступну подію j у складі роботи (i, j), яка не має коду (тобто поки що незакодована), при цьому попередня подія i має код (закодована подія). Для роботи (i, j) умови з першого та другого правил не виконуються, тоді подія j коду не отримує, тобто залишається незакодованою.

У випадку неможливості кодування події j за трьома вищевказаними правилами можна використати правила зворотних дуг.

<u>Правила кодування зворотних дуг</u> (i^-, j^*):

1) розглянемо попередню подію ***i*** у складі роботи ***(i, j)***, яка не має коду (тобто поки що незакодована), при цьому наступна подія ***j*** має код (закодована подія). Якщо для роботи ***(i, j)*** виконуються умови $a'_{ij} = 0$ та $f_{ij} > 0$, тоді попередня подія ***i*** отримує код $(-j, Q_i)$, де друга частина коду визначається за формулою:

$$Q_i = min(Q_j, f_{ij}). \tag{1.31}$$

З ***i***-ї події до ***j***-ї можна направити зустрічний потік, розмір якого обмежується потоком, який вже є в ***j***. Призначення зустрічного потоку полягає в зменшенні розміру потоку f_{ij}.

2) розглянемо попередню подію ***i*** у складі роботи ***(i, j)***, яка не має коду (тобто поки що незакодована), при цьому наступна подія ***j*** має код (закодована подія). Якщо для роботи ***(i, j)*** виконуються умови $a''_{ij} = 0$ та $f_{ij} > c_{ij}$, тоді попередня подія ***i*** отримує код $(-j, Q_i)$, де друга частина коду визначається за формулою:

$$Q_i = min(Q_j, f_{ij} - c_{ij}). \tag{1.32}$$

3) розглянемо попередню подію ***i*** у складі роботи ***(i, j)***, яка не має коду (тобто поки що незакодована), при цьому наступна подія ***j*** має код

(закодована подія). Для роботи ***(i, j)*** умови з першого та другого правил не виконуються, тоді подія ***i*** коду не отримує, тобто залишається незакодованою.

В результаті застосування процедури кодування кінцева ***n***-а подія (остання подія в сітьовій моделі) може отримати позначку «прорив сіті» або залишитися без неї «непрорив сіті».

У разі «**прориву сіті»** необхідно змінити потік уздовж критичного шляху на величину другої частини коду останньої події (1.33) (див. рис. 1.3, блок №5) у напрямі першої частини коду. Після досягнення початкової події починаємо (повторюємо) процедуру кодування.

$$f_{ij}^{нов} = f_{ij}^{ст} \pm Q_n \tag{1.33}$$

У разі «**непрориву сіті»** (коли ми не можемо закодувати кінцеву подію) застосовуємо процедуру зміни термінів звершення подій.

Визначаємо дві непересічні безлічі подій, які утворюють дуги. Можливі наступні стани дуг: (i^*, j^-); (i^-, j^*); (i^-, j^-); (i^*, j^*). Розглядаємо тільки ті роботи, які мають закодовану та незакодовану події, вони утворюють як мовилося вище такі види дуг:

i^*, j^- - пряма дуга (пряме кодування) - це робота, початкова подія якої закодована, а кінцева немає коду.

i^-, j^* - зворотна дуга (зворотне кодування) - це робота, початкова подія якої не має код, а кінцева має.

<u>Для кожної такої дуги визначаємо:</u>

- резерви критичності a'_{ij};

- резерви скорочення a''_{ij};

- значення Δ_1 і Δ_2

$$\left.\begin{aligned} \Delta_1 &= min\left|-a'\right|, \\ \Delta_2 &= min\left|-a''\right|; \end{aligned}\right\} \tag{1.34}$$

- величину скорочення тривалості критичного шляху (1.35):

$$\Delta T_n = min(\Delta_1, \Delta_2), \tag{1.35}$$

де n - це номер ітерації.

- значення нових термінів звершень для подій, які не отримали коду за формулою:

$$T^{нов} = T^{стар} - \Delta T_n \tag{1.36}$$