

ВИКОРИСТАННЯ МЕТОДУ ОДНОРІДНИХ ІНТЕРВАЛІВ У ЗАДАЧАХ ОПТИМІЗАЦІЇ ПЛАНУВАННЯ

Інститут комп'ютерних технологій
Національного авіаційного університету

Запропонована методика пошуку однорідних інтервалів. Вона може використовуватись при оптимізації планування задач, де процес функціонування системи може бути описаний шляхом завдання розкладу (календарного плану, часового графіку тощо). Об'єктом дослідження в таких задачах виступають події, область допустимих значень яких задається інтервальними проміжками

Вступ

В ряді оптимізаційних задач об'єктом дослідження виступають певні події, що мають бути реалізовані в деякий момент часу протягом заданого проміжку. Завдання оптимізації полягає у визначенні найбільш оптимального розподілу цих подій у заданому часовому просторі [1].

Зазвичай, при вирішенні таких задач весь термін, протягом якого можуть бути реалізовані всі задані процеси, розбивається на певну кількість рівних інтервалів [2]. Крок вибирається з урахуванням тривалості кожної події і підбирається в кожному конкретному випадку окремо. Цей метод не досить зручний та може призвести до невиправданого збільшення обсягів розрахунків.

Такі принципи закладено, наприклад, при розробці програмних продуктів фінансового аналізу „Альт-Інвест”, *Project Expert*, *Comfar*.

В даній статті пропонується метод однорідних інтервалів. В цьому випадку весь час, що розглядається, буде поділений на нерівні інтервали. Так як на певних проміжках часу дві або більше подій можуть відбуватись одночасно, то потрібно виділити однорідні інтервали, протягом кожного з них у будь-який момент часу може реалізовуватись один і той же набір подій.

Кожна з подій може розглядатись на одному з однорідних інтервалів, на які буде розбитий відповідний початковий проміжок часу, протягом якого вона може бути реалізована.

Постановка задачі

Нехай на проміжку часу T повинні бути реалізовані n деяких процесів, кожен з яких складається з m подій nm C . Кожна з цих подій може відбутись у деякий момент часу, що належить певному початковому інтервалу P_{nm} , заданому найменш раннім $t_{nm}^{поч}$ та найпізнішим k_{in} nm можливими строками:

$$P_{nm} = [t_{nm}^{поч}, t_{nm}^{кін}]$$

Причому завжди має виконуватись вимога:

$$t_{nm}^{поч} \leq t_{nm}^{кін}$$

На рис. 1 представлена діаграма, що ілюструє приклад розміщення у часовому просторі трьох абстрактних процесів, що складаються з подій C_{nm} , кожній з яких відповідає окремий початковий інтервал P_{nm} .

Способи поєднання інтервалів

Попарне поєднання деяких з інтервалів може мати наступний вигляд [3]:

1) пересікання подій C_{nm} і $C_{n'm'}$ (інтервали $P_{2,1}$ та $P_{3,1}$ на рис. 1):

$$t_{nm}^{поч} < t_{n'm'}^{поч} < t_{nm}^{кін} < t_{n'm'}^{кін}$$

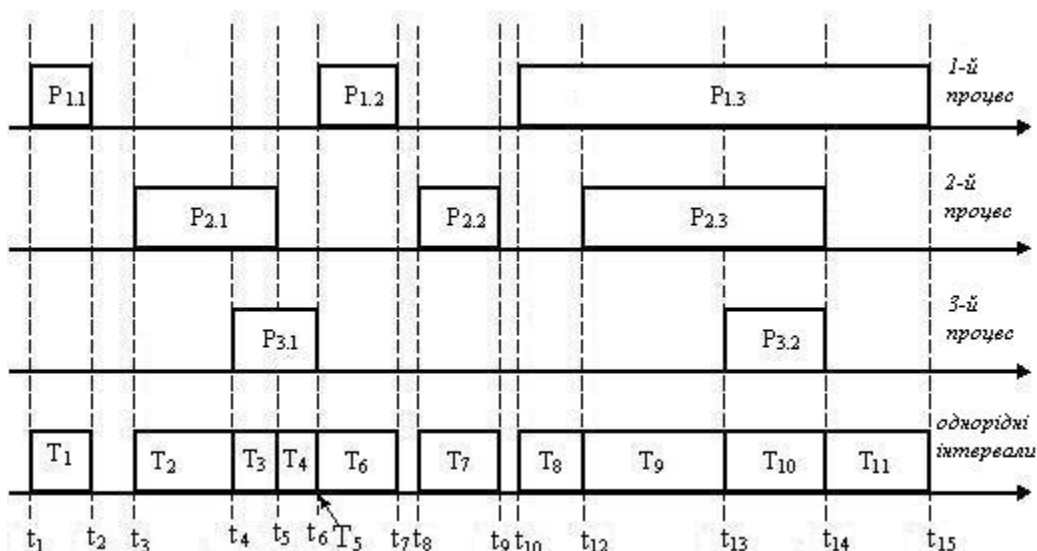


Рис. 1 Часова діаграма демонструє приклад розбиття початкових інтервалів P_{nm} на однорідні T_n

У випадку, коли початкова точка, але не кінцева, наступного інтервалу належить проміжку попереднього, то з врахуванням спільного проміжку утворюються три однорідні інтервали. Інтервал перетину буде закритим, а сусідні з ним інтервали будуть напіввідкриті в точках, що обмежують спільний проміжок:

$$T_i = [t_{n,m}^{поч}, t_{n',m'}^{поч}],$$

$$T_{i+1} = [t_{n',m'}^{поч}, t_{n,m}^{кін}],$$

$$T_{i+2} =]t_{n,m}^{кін}, t_{n',m'}^{кін}].$$

Початкові інтервали будуть розбиті на відповідні їм новоутворені однорідні інтервали (1):

$$\begin{cases} P_{nm} = T_i \oplus T_{i+1} \\ P_{n'm'} = T_{i+1} \oplus T_{i+2} \end{cases}, \quad (1)$$

2) пересікання в одній точці - початкова точка одного інтервалу є одночасно кінцевою точкою іншого (інтервали $P_{3,1}$ та $P_{1,2}$ на рис. 1):

$$t_{nm}^{кін} = t_{n'm'}^{поч}$$

У цьому випадку, спільна точка буде виключена з обох початкових інтервалів і

буде утворювати собою один окремий однорідний інтервал:

$$T_i = [t_{n,m}^{поч}, t_{n,m}^{кін}],$$

$$T_{i+1} = [t_{n,m}^{кін}, t_{n,m}^{кін}] = [t_{n',m'}^{поч}, t_{n',m'}^{поч}],$$

$$T_{i+2} =]t_{n',m'}^{поч}, t_{n',m'}^{кін}].$$

Відповідність однорідних інтервалів початковим буде аналогічна попередньому випадку (1).

3) поглинання одного інтервалу іншим без спільних меж) – один інтервал повністю належить іншому і відсутнє співпадіння між собою відповідних граничних точок обох інтервалів (інтервали $P_{1,3}$ та $P_{3,2}$ на рис. 1 :

$$t_{nm}^{поч} < t_{n'm'}^{поч}, \quad t_{n'm'}^{кін} < t_{nm}^{кін}$$

За таких обставин однорідні інтервали будуть мати межі:

$$T_i = [t_{n,m}^{поч}, t_{n',m'}^{поч}],$$

$$T_{i+1} = [t_{n',m'}^{поч}, t_{n',m'}^{кін}],$$

$$T_{i+2} =]t_{n',m'}^{кін}, t_{n,m}^{кін}].$$

Менший початковий інтервал залишить свої межі, а більший – буде розділений на три нові:

$$\begin{cases} P_{n,m} = T_i \oplus T_{i+1} \oplus T_{i+2} \\ P_{n',m'} = T_{i+1} \end{cases}, \quad (2)$$

4) поглинання з однією спільною межею (інтервали $P_{2,3}$ та $P_{3,2}$ на рис. 1):

$$(t_{nm}^{поч} = t_{n'm'}^{поч}) \oplus (t_{n'm'}^{кін} = t_{nm}^{кін})$$

При цьому будуть утворені два однорідні інтервали:

$$T_i = [t_{nm}^{поч}, t_{n'm'}^{поч}],$$

$$T_{i+1} = [t_{n'm'}^{поч}, t_{n'm'}^{кін}].$$

Як і в попередньому випадку (2), менший початковий інтервал залишиться без змін, але більший – буде розділений на два нові:

$$\begin{cases} P_{n,m} = T_i \oplus T_{i+1} \\ P_{n',m'} = T_{i+1} \end{cases}. \quad (3)$$

Формування проміжних матриць

На основі цих вхідних даних формуються наступні матриці [4]:

B – містить початкові значення всіх інтервалів P_{nm} :

$$P_{nm}^i = \begin{cases} [t_{n,m}^{поч}, R_{nm}(1)], & i = 1 \\ [R_{nm}(i), R_{nm}(i+1)], & \text{при } 1 < i < k_{nm}, (i+1) = k_{nm} \\ [R_{nm}(i), R_{nm}(i+1)], & \text{при } 1 < i < k_{nm}, (i+1) < k_{nm} \\ [R_{nm}(i), t_{nm}^{кін}], & \text{при } i = k_{nm} \end{cases}. \quad (4)$$

При пересіканні в одній точці вона виділяється як самостійний інтервал. Перелік цих елементів міститься в множині W . Потрібно виключити ці елементи в усіх новосформованих інтервалах (4), де вони включені як початкові, або якщо вони збігаються зі значенням $t_{кін}^n$ для будьякої з подій:

$$B_{ij} = t_{ij}^{поч}, \quad i = \overline{1, n}, \quad j = \overline{1, m}.$$

E – містить кінцеві значення всіх інтервалів P_{nm} :

$$E_{ij} = t_{ij}^{кін}, \quad i = \overline{1, n}, \quad j = \overline{1, m}.$$

D – містить всі значення, що належать множинам B та E :

$$D = B \cup E.$$

Дані множини мають бути відсортовані в порядку збільшення, всі повторні значення вилучаються.

В послідовність W заноситься перетин множин B та E :

$$W = B \cap E$$

Для кожної події C_{nm} потрібно виокремити з множини D відрізок R_{nm} , що обмежується відповідним інтервалом P_{nm} :

$$R_{nm} \in D, \quad t_{nm}^{поч} \leq x \leq t_{nm}^{кін}, \quad x \in R_{nm}$$

Якщо на цьому відрізку більш ніж два елементи, тобто між початковим та кінцевим значеннями інтервалу P_{nm} розміщені проміжні елементи, то даний інтервал буде розбитий на k_{nm} нових:

$$k_{nm} = |R_{nm}| - 1.$$

Формування однорідних інтервалів

При перетині двох інтервалів нові інтервали будуть сформовані наступним чином:

$$P_n^{k+1} =]a, b[\quad \text{при } a \in W,$$

$$P_n^{k+1} = [c, d[\quad \text{при } d = t_{кін}^n, \quad d \in W.$$

При цьому остаточна кількість однорідних інтервалів v_{nm} буде відповідно збільшена:

$$v_{nm} = k_{nm} + 1.$$

Загальний алгоритм пошуку однорідних інтервалів

1. $i := 1, k_{nm} = |R_{nm}| - 1, v = k_{nm}$.
2. For $i = 1, \dots, k_{nm}$ do
 - 2.1. if $i := 1$ then
 - if $t_{nm}^{noc} \notin W$ then

$$P_{nm}^i = [t_{nm}^{noc}, R_{nm}(i)];$$
 else $P_{nm}^i = [t_{nm}^{noc}, R_{nm}(i)], v = v + 1,$

$$P_{nm}^v = [t_{nm}^{noc}, t_{nm}^{noc}];$$
 - 2.2. if $1 < i < k_{nm}, (i + 1) < k_{nm}$
 - if $R_{nm}(i) \notin W$ then

$$P_{nm}^i = [R_{nm}(i), R_{nm}(i + 1)];$$
 else $P_{nm}^i = [R_{nm}(i), R_{nm}(i + 1)], v = v + 1,$

$$P_{nm}^v = [R_{nm}(i), R_{nm}(i)];$$
 - 2.3. if $1 < i < k_{nm}, (i + 1) = k_{nm}$ then
 - if $R_{nm}(i) \notin W$ then
 - if $R_{nm}(i + 1) \notin W$ then

$$P_{nm}^i = [R_{nm}(i), R_{nm}(i + 1)];$$
 else $P_{nm}^i = [R_{nm}(i), R_{nm}(i + 1)],$

$$v = v + 1,$$

$$P_{nm}^v = [R_{nm}(i + 1), R_{nm}(i + 1)];$$
 - 2.4. if $1 < i < k_{nm}, (i + 1) = k_{nm}$ then
 - if $R_{nm}(i) \in W, R_{nm}(i + 1) \notin W$
 then $P_{nm}^i = [R_{nm}(i), R_{nm}(i + 1)],$

$$v = v + 1, P_{nm}^v = [R_{nm}(i), R_{nm}(i)];$$
 else
 - if $R_{nm}(i) \notin W, R_{nm}(i + 1) \in W$
 then $P_{nm}^i = [R_{nm}(i), R_{nm}(i + 1)],$

$$v = v + 1,$$

$$P_{nm}^v = [R_{nm}(i + 1), R_{nm}(i + 1)]$$
 else $P_{nm}^i = [R_{nm}(i), R_{nm}(i + 1)],$

$$v = v + 1, P_{nm}^v = [R_{nm}(i), R_{nm}(i)],$$

$$v = v + 1,$$

$$P_{nm}^v = [R_{nm}(i + 1), R_{nm}(i + 1)];$$
 - 2.5. if $i = k_{nm}$ then
 - if $t_{nm}^{kih} \notin W$ then

$$P_{nm}^i = [R_{nm}(i), t_{nm}^{kih}]$$

$$\text{else } P_{nm}^i = [R_{nm}(i), t_{nm}^{kih}], v = v + 1,$$

$$P_{nm}^j = [t_{nm}^{kih}, t_{nm}^{kih}].$$

Початковий інтервал буде розбитий на відповідні однорідні інтервали, поєднані виключаючим або:

$$P_{nm} = P_{nm}^1 \oplus \dots \oplus P_{nm}^i, \quad i = \overline{1, v_{nm}}$$

Висновки

Результатом описаних вище дій є набір однорідних інтервалів $P_{nm}^i, i \in \overline{1, v_{nm}}$, що позначають проміжки часу, протягом яких в будь-який момент може відбутись один і той же набір подій.

Для кожної з подій визначено перелік інтервалів, протягом одного з яких вона може бути реалізована. Подальша задача оптимізації полягає у виборі шляхом послідовного перебору одного з цих буденінтервалів як самого пріоритетного.

На відміну від класичного розбиття на рівні інтервали всього терміну, що досліджується, даний метод, забезпечує поглиблення кроку до найменшої одиниці часу, встановленої умовою конкретної задачі, при цьому кількість варіантів перебору K буде мінімальною:

$$K = \prod_{i=1}^n \prod_{j=1}^m v_{ij}.$$

Описаний в даній статті метод однорідних інтервалів може використовуватись в комбінаторних задачах, у яких розглядається певний набір подій, області допустимих значень яких зазначено інтервально, і в результаті необхідно максимально звузити ці проміжки.

Список літератури

1. Аоки М. Введение в методы оптимизации. – М.: Наука, 1977. – 344 с.
2. Конвей Р.В., Максвелл В.Л., Миллер Л.В. Теория расписаний. – М.: Наука, 1975. – 359 с.
3. Поляк Э. Численные методы оптимизации. – М.: Мир, 1974. – 376 с.
4. Кофман А., Анри-Лабордер А. Методы и модели исследования операций. Целочисленное программирование. – М.: Мир, 1997. 423 с.