

## МОДЕЛІ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО СТРУКТУРИЗАЦІЇ РОБІТ ПРИ РОЗРОБЦІ ПРИКЛАДНИХ ПРОГРАМНИХ СИСТЕМ

*І.А. Стрелов, П.П. Ігнатенко*

Інститут програмних систем НАН України  
03187, Київ, проспект Академіка Глушкова, 40,  
тел.: +380 (44) 266 1540, факс: +380 (44) 266 6263  
e-mail: ignat@isofts.kiev.ua

Розглянуто задачі оптимізації розподілу робіт за етапами проекту створення ППС та вибору оптимального набору робіт для окремого етапу проекту. Запропоновано підхід до вирішення таких задач на основі методу „гілок та меж” та алгоритми вирішення. Дано обґрунтування оптимальності рішень і скінченності алгоритмів.

Problems of distributing tasks among iterations of a software application development project and choosing optimal set of tasks for a single iteration are examined. An approach and algorithms based on “branches and borders” method are proposed for these problems. Proofs of solutions optimality and algorithms finiteness are given.

При застосуванні методики оцінювання економічних характеристик розробки або модифікації прикладної програмної системи (ППС) [1], виникає необхідність у прийнятті керівництвом ряду рішень щодо управління розробкою проекту [2].

У названій методиці авторами запропоновано комплексний підхід до оцінювання економічних характеристик проектів розробки ППС, що поєднує методи FPA (Function Point Analysis, аналіз одиниць функціональності) [3] та СОСОМО (COConstructive COSt MOdel, конструктивна модель вартості) [4].

Водночас прийняття організаційних рішень, тісно пов'язаних з одержанням необхідних економічних характеристик у процесі проектування у методиці лишається без достатнього обґрунтування.

Важливими рішеннями, що суттєво впливають на тривалість та вартість розробки ППС є структуризація робіт на етапи з урахуванням договірних вимог замовника проекту та обмежень щодо наявності в організації необхідних ресурсів.

Для підтримки прийняття організаційних рішень у методиці запропоновано використовувати експертну оцінку щодо розбиття проекту на підпроекти, що є підвищенням об'єктивності та зниження похибки оцінки за рахунок того, що менші проекти, або легше піддаються оцінюванню, або вже мають добре відому та чітко визначену оцінку.

У роботі пропонуються математичні моделі, що дозволяють обґрунтовано прийняти рішення щодо структуризації робіт по створенню ППС, обґрунтовано розбивати великі складні задачі на підзадачі тощо.

### 1. Задача визначення складу робіт поточного етапу створення ППС

Нехай задані роботи  $\{R_j\}$ ,  $j: \overline{1, n}$ , час виконання яких  $\{t_j\}$ ,  $j: \overline{1, n}$ .  $T$  – загальний час  $U$ , який призначений для виконання робіт щодо доопрацювання етапу проекту. При цьому  $\sum_i t_i \gg T$ . Задані також  $\{C_j\}$ ,  $j: \overline{1, n}$ , які визначають вартість (важливість) робіт  $\{R_j\}$ . Необхідно виконати на етапі  $U$  підмножину робіт  $\{R_j\}$  з максимальною сумою  $\{C_j\}$ .

**Формальна постановка задачі.** Задані  $\{R_j\}$ ,  $j: \overline{1, n}$ ,  $\{t_j\}$ ,  $j: \overline{1, n}$ ,  $\{C_j\}$ ,  $j: \overline{1, n}$ ,  $T$ . При цьому можна вважати  $T > 0$ ,  $t_j > 0$ ,  $C_j > 0$ . Введемо булеву змінну  $x_j$ , що дорівнює 1, якщо робота  $R_j$  виконується на етапі  $U$ , та 0, в іншому випадку.

Серед усіх векторів  $\langle X_j \rangle$ , що задовольняють умовам

$$\sum_{j=1}^n t_j x_j \leq T, \quad (1)$$

$$x_j = 0 \vee 1 \quad (2)$$

необхідно знайти вектор  $\langle X_j \rangle$ , що доставляє функціоналу

$$\sum_{j=1}^n C_j x_j \quad (3)$$

максимальне значення. Це – задача про ранець [5].

Для її вирішення пропонується адаптивний алгоритм, що базується на ідеї методу „гілок та меж” в модифікації А. Ленд і А. Дойг [6]. Його особливістю є використання ефективного прийому для вирішення допоміжної задачі.

**Опис алгоритму.** Розглянемо модель

Знайти 
$$\max \sum_{j=1}^n C_j x_j \quad (4)$$

при обмеженнях 
$$\sum_{j=1}^n t_j x_j \leq T \quad (5)$$

$$0 \leq x_j \leq 1, \quad (6)$$

де  $T > 0$ ,  $t_j > 0$ ,  $C_j > 0$ . Будемо називати цю задачу допоміжною, або  $\Gamma$ -задачею.

Її рішення може бути одержане таким чином. Якщо змінні  $x_j$  записані в порядку співвідношення  $\frac{C_j}{T_j}$ :

$$\frac{C_1}{T_1} \geq \frac{C_2}{T_2} \geq \dots \geq \frac{C_n}{T_n},$$

то покладаємо:

$$x_1 = \min\left(1, \frac{T}{t_1}\right),$$

$$x_2 = \min\left(1, \frac{T - t_1 x_1}{t_2}\right),$$

...

$$x_j = \min\left(1, \frac{T - \sum_{k=1}^{j-1} t_k x_k}{t_j}\right).$$

Рішення цієї  $\Gamma$ -задачі вибираємо в якості оцінки („оцінки зверху”) функції-критерію вихідної задачі.

Позначимо  $\Delta^i$  – значення прогнозу функції-критерію задачі (1) – (3) на  $i$ -й ітерації алгоритму,  $\{x_j\}^i$  – краще зі знайдених рішень;  $F^i(x_j^0)$  – значення функціоналу, яке доставляє оптимальне рішення  $i$ -ї допоміжної задачі (4)–(6).

На першій ітерації список задач, які підлягають рішенню містить одну допоміжну задачу (4)–(6). На довільній ітерації  $i$  виконуються наступні кроки.

**Крок 1.** Перевірка: список допоміжних задач, які підлягають рішенню, пустий? Якщо так – крок 11, ні – крок 2.

**Крок 2.** Вибрати  $i$ -ту допоміжну задачу і викреслити її із списку задач, що підлягають рішенню.

**Крок 3.** Вирішити вибрану допоміжну задачу.

**Крок 4.** Перевірка:  $F^i(x_j^0) > \Delta^i$ ? Якщо ні – крок 5, так – крок 6.

**Крок 5.** Прийняти  $\Delta^{i+1} = \Delta^i$ , перехід на крок 1.

**Крок 6.** Перевірка: умови (2) виконуються? Так – крок 7, ні – крок 8.

**Крок 7.** Прийняти  $\Delta^{i+1} = F^i(x_j^0)$ ,  $\{x_j\} = \{x_j^0\}$ ,  $j: \overline{1, n}$ , повернутися на крок 1.

**Крок 8.** Вибрати  $\{x_{j_0}^0\} \neq 0 \vee 1$ .

**Крок 9.** Додати дві допоміжні задачі в список задач, які підлягають рішенню. Вони ідентичні  $i$ -й допоміжній задачі, але в одній з них додано умову  $x_{j_0}^0 = 1$ , а в іншій  $x_{j_0}^0 = 0$ .

**Крок 10.** Прийняти  $\Delta^{i+1} = \Delta^i$ , перехід на крок 1.

**Крок 11.** Кінець алгоритму.

Скінченність даного алгоритму є наслідком скінченності множини варіантів рішення задач (1)–(3).

Позначимо  $\delta_n = \langle x_1, x_2, \dots, x_n \rangle$  – довільне рішення розглядуваних задач, а  $\delta_l = \langle x_1, x_2, \dots, x_l \rangle$  – фрагмент їх рішення ( $l < n$ ). Для запропонованого алгоритму дійсні наступні твердження.

**Твердження 1.** Нехай  $\Delta(\delta_n^0)$  – значення функції-критерію оптимального рішення задачі (4)–(6),  $F(\delta_n)$  – функція-критерій задачі (1)–(3). Тоді  $F(\delta_n) \leq \Delta(\delta_n^0)$ .

Для доказу цього твердження покажемо спочатку, що для будь-якого рішення  $\delta_n$  і фрагменту рішення  $\delta_l$ , задовольняючого умови (2) задачі (1)–(3) виконується нерівність

$$F(\delta_n) \leq F(\delta_l) + \sum_{\{\tilde{x}_j^0\}} C_j x_j^0,$$

де  $\{\tilde{x}_j^0\}$  – оптимальний розв'язок наступної допоміжної задачі:

$$\max \sum_{j \in \delta_l} C_j x_j \tag{7}$$

при умовах

$$\sum_{j \in \delta_l} t_j x_j \leq T(\delta_l), \tag{8}$$

$$0 \leq x_j \leq 1. \tag{9}$$

Нехай умови (2) виконуються для  $\delta_n$ , тоді

$$F(\delta_n) = F(\delta_l) + \sum_s T_s,$$

де

$$s = \{j: j \in \delta_n\} \setminus \{j: j \in \delta_l\}.$$

З іншого боку, якщо умови виконуються лише для фрагменту рішення  $\delta_l$ , маємо:

$$F(\delta_n) = F(\delta_l) + \sum_{j=l+1}^n C_j x_j.$$

Оскільки  $\{\tilde{x}_j^0\}$  забезпечує максимум рішення задачі (7)–(9) і  $\{\tilde{x}_j^0\} \in \delta_n \setminus \delta_l$ , то

$$\sum_s C_s \leq \sum_{\{\tilde{x}_j^0\}} C_j x_j^0.$$

Отже, твердження леми вірне, так як

$$\Delta(\delta_n) = F(\delta_l) + \sum_{\{\bar{x}_j^0\}} C_j x_j^0.$$

**Твердження 2.** Якщо на довільній ітерації  $i$  розглядуваного алгоритму список  $r$ -задач  $\{x_1, x_2, \dots, x_n\}$ , які підлягають вирішенню, пустий, то  $\{x_j\}^i$  – оптимальне рішення задачі (1)–(3), а відповідне йому  $\Delta$  – максимальне значення функціоналу.

Справедливість твердження 2 випливає із коректності побудови в алгоритмі оцінок варіантів рішень (твердження 1) і того очевидного факту, що описаний алгоритм охоплює всю множину варіантів рішень задачі (1)–(3).

## 2. Загальна задача визначення складу робіт етапів створення ППС

Нехай задана сукупність всіх робіт  $\{r_j\}$ ,  $j: \overline{1, n}$  за виконанням проекту ППС з оцінками часу виконання  $\{t_j\}$ ,  $j: \overline{1, n}$ . При цьому множину робіт проекту необхідно розбити на  $m$  підетапів  $\{E_i\}$ ,  $i: \overline{1, m}$ , час виконання яких визначено  $\{T_i\}$ ,  $i: \overline{1, m}$ .

Задана матриця  $C_{ij}$  з елементами, які характеризують вартість виконання роботи  $j$  на етапі  $i$ .

Задана також матриця  $k_{ij}$  з елементами, які характеризують коефіцієнти щодо часу виконання роботи  $j$  на етапі  $i$ .

Як відомо, серед робіт  $\{r_j\}$ ,  $j: \overline{1, n}$  можуть бути роботи незалежні та такі, що виконуються за умови виконання всіх попередніх робіт, згідно графа, яким описано план-графік цих робіт. Тобто, якщо ми хочемо почати виконання не з незалежної роботи, а з роботи, яка стоїть у даний момент останньою у ланцюжці залежних робіт, ми повинні за допомогою коефіцієнтів  $k_{ij}$  забезпечити відповідний час для виконання цих робіт, а за допомогою  $C_{ij}$  забезпечити значення її вартості.

**Формальна постановка задачі.** Задані вектори  $\{r_j\}$ ,  $j: \overline{1, n}$ ;  $\{t_j\}$ ,  $j: \overline{1, n}$ ;  $\{E_i\}$ ,  $i: \overline{1, m}$ ;  $\{T_i\}$ ,  $i: \overline{1, m}$ ; а також матриці  $C_{ij}$ ,  $i: \overline{1, m}$ ,  $j: \overline{1, n}$ ;  $k_{ij}$ ,  $i: \overline{1, m}$ ,  $j: \overline{1, n}$ .

Введемо змінну  $x_{ij}$ , що дорівнює  $t_j$ , якщо робота  $r_j$  виконується на етапі  $E_i$ , та 0, в іншому випадку.

Необхідно серед матриць  $x_{ij}$ , які задовольняють умовам

$$\sum_i x_{ij} = t_j, \quad j: \overline{1, n}, \quad (10)$$

$$\sum_i k_{ij} x_{ij} \leq T_i, \quad i: \overline{1, m}, \quad (11)$$

$$x_{ij} = t_j \vee 0, \quad (12)$$

знайти матрицю  $x_{ij}$ , яка доставляє мінімальне значення функціоналу

$$\sum_i \sum_j c_{ij} x_{ij}. \quad (13)$$

Умова (12) означає, що розрив виконання однієї роботи на різні етапи забороняється.

Задачі, що вирішуються за допомогою цієї моделі розподілу, мають наступну особливість: кількість етапів значно менше кількості робіт ( $m \ll n$ ). Це суттєво для запропонованого далі алгоритму, оснований, як і для попередньої моделі, на ідеї методу „гілок та меж” [6].

**Опис алгоритму.** В запропонованому алгоритмі допоміжною задачею є:

$$\text{Знайти} \quad \min \sum_i \sum_j c_{ij} x_{ij} \quad (14)$$

при обмеженнях

$$\sum_i x_{ij} = t_j, \quad j: \overline{1, n}, \quad (15)$$

$$\sum_i k_{ij} x_{ij} \leq T_i, \quad i: \overline{1, m}, \quad (16)$$

$$x_{ij} \geq 0. \quad (17)$$

При цьому допускаємо, що  $T_i > 0$ ,  $t_j > 0$ ,  $k_{ij} > 0$ ,  $c_{ij} > 0$ .

Це відома розподільча задача лінійного програмування (R-задача). Ефективність алгоритмів, заснованих на ідеї методу „гілок та меж” у розглядуваній модифікації сильно залежить від наявності ефективного алгоритму вирішення цієї R-задачі. Такий машинно-орієнтований алгоритм запропоновано в [7].

Використовуючи значення функціоналу оптимального рішення цієї допоміжної задачі в якості „оцінки знизу” значення функції-критерії задачі (10)–(13), алгоритм її вирішення може бути описано наступним чином.

На першій ітерації алгоритму список задач, що підлягають вирішенню, містить одну допоміжну задачу (14)–(17).

Позначимо  $\Delta^i$  – значення прогнозу функції-критерію задачі (10)–(13) на  $i$ -й ітерації;  $\bar{x}_{ij}^i$  – краще із знайдених рішень;  $L^i(x_{ij}^0)$  – значення функціоналу, яке доставляє оптимальний розв’язок рішення  $i$ -ї R-задачі.

На довільній ( $i$ -й) ітерації алгоритму виконуються такі кроки:

**Крок 1.** Перевірка: список R-задач, які підлягають вирішенню, пустий? Так – крок 12, ні – крок 2.

**Крок 2.** Вибрати  $i$ -ту R-задачу і викреслити її із списку задач, які підлягають вирішенню.

**Крок 3.** Перевірка: вибрана R-задача має допустиме рішення? Так – крок 4, ні – крок 1.

**Крок 4.** Вирішити вибрану R-задачу,  $x_{ij}^0$  – її розв’язок.

**Крок 5.** Перевірка:  $\Delta^{i+1} > L^i(x_{ij}^0)$ ? Так – крок 7, ні – крок 6.

**Крок 6.** Прийняти  $\Delta^{i+1} = \Delta^i$  і перейти на крок 1.

**Крок 7.** Перевірка: умови (12) виконуються? Так – крок 8, ні – крок 9.

**Крок 8.** Прийняти  $\Delta^{i+1} = L^i(x_{ij}^0)$ , зафіксувати  $x_{ij}^0$ , перейти на крок 1.

**Крок 9.** Вибрати  $x_{i_0j_0} \neq d_j \vee 0$ .

**Крок 10.** Додати дві задачі до списку R-задач, які підлягають вирішенню. Вони ідентичні  $i$ -й задачі, але в одній з них додано умову  $x_{i_0j_0} = d_j$ , а в іншій –  $x_{i_0j_0} = 0$ .

**Крок 11.** Прийняти  $\Delta^{i+1} = \Delta^i$  і перейти на крок 1.

**Крок 12.** Кінець алгоритму.

Скінченність даного алгоритму витікає із скінченності множини варіантів рішень задачі (10)–(13).

Для нього дійсні наступні твердження.

**Твердження 3.** Нехай  $\Delta(x_{ij}^0)$  – значення функції-критерію для оптимального рішення допоміжної задачі (14)–(17).  $L(x_{ij})$  – функція-критерій задачі (10)–(13).

Тоді 
$$L(x_{ij}) \geq \Delta(x_{ij}^0).$$

Справедливість того, що функція-критерій оптимального рішення запропонованої R-задачі є „оцінкою знизу” для функції-критерію основної задачі може бути, так же як і в попередньому випадку строго доведена. Не приводячи в цій роботі строгого доведення ми лише зауважимо, що справедливість твердження 3 зумовлена рішенням з вибору допоміжної задачі (R-задачі).

**Твердження 4.** Якщо рішення основної задачі існує і список R-задач, які підлягають вирішенню, на  $i$ -му кроці алгоритму пустий, то зафіксоване алгоритмом рішення  $x^0$  оптимальне, а відповідне йому  $\Delta$  – мінімальне значення функціоналу  $L(x)$ .

Дійсно, за допомогою вибраної оцінки  $\Delta$  в алгоритмі відкидаються всі ті варіанти рішення, для яких  $L^i(x_{ij}^0)$  більше чи дорівнює зафіксованому  $\Delta^i$  (кроки 5 - 6).

Обґрунтованість такого відсіву варіантів рішень дає твердження 3, з якого витікає також, що коли на  $i$ -й ітерації алгоритму вирішені всі допоміжні задачі із списку задач та їх рішення задовольняють умові (12), то розглядати варіанти, що лишилися, нема сенсу, оскільки всі вони будуть відкинуті на кроках 5-6.

Оптимум не буде досягнуто за допомогою описаного алгоритму лише в тому випадку, коли не існує жодного допустимого рішення основної задачі (10)–(13).

## Висновки

Оптимізація структуризації робіт щодо виконання проектів по створенню ППС з урахуванням кваліфікації кадрового потенціалу організації є важливою складовою процесу керування розробкою.

Впровадження пропонованих методів підтримки прийняття рішень дозволяє більш точно прогнозувати потребу в людських та часових ресурсів організації, що у галузі розробки ППС має велике значення. Внаслідок цього можна точніше як планувати процеси виконання окремого проекту, так і узгоджувати між собою розробку декількох проектів, які виконуються паралельно. Це дає змогу скорочувати цикли розробки ППС, що є суттєвим для конкурентоспроможності підприємств у сучасних умовах.

1. Стрелов І.А., Ігнатенко П.П. Підхід до оцінювання економічних характеристик проектних рішень при розробці, модифікації та реінжинірингу програмних систем // Пробл. програм. – 2004. – № 1. – С. 38–51.
2. Исследование операций в 2-х т. Т.2 Модели и применения / Под ред. Дж. Моудера, С. Елмаграби. – Пер. с англ. – М.: Мир, 1981. 677 с.
3. KPMG, 2001 – KPMG Consulting, Inc., Mark II Function Point Analysis Guide // [www.kpmg.co.uk/kpmg/uk](http://www.kpmg.co.uk/kpmg/uk)
4. CSE, 1999 – Center for Software Engineering, "COCOMO II Model Definition Manual", Computer Science Department, USC Center for Software Engineering, 1999.
5. Вагнер Г. Основы исследования операций / Пер. с англ. – М.: Мир, 1972. – 2. – 488 с.
6. Land A.H., Doig A.G. An Automatic Method of Solving Discrete Programming Problems. – *Econometrica*, 1960. – 28. – P. 496 – 520.
7. Ігнатенко П.П., Опанович М.И., Шкурба В.В. Использование списковых структур в распределительной задаче. В сб. Системотехника. – Киев: ИК АН УССР, 1970. - № 2. - С. 15–36.