УДК 519.854.2

АЛГОРИТМ ГІЛОК ТА МЕЖ ДЛЯ ЗАДАЧІ ФОРМУВАННЯ КОМАНДИ ТА ПОШУКУ ІНВЕСТОРІВ ДЛЯ СТАРТАП-ПРОЕКТІВ

Кравченко Є. І.

Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”, Україна, Київ

Розглянута задача формування команди для стартап-проекту. Побудована математична модель задачі визначення кандидатів в команду проекту, що мають найбільший досвід роботи та вимагають найменшу зарплатню. Задача зведена до задачі про рюкзак. Описаний метод та обґрунтована доцільність використання алгоритму гілок та меж для розв’язання задачі.

Ключові слова: старап-проект, команда проекту, задача про рюкзак, метод гілок та меж.

Кравченко Е. И. Алгоритм ветвей и границ для задачи формирования команды и поиска инвесторов для стартап-проектів / Национальный технический университет Украины "Киевский политехнический институт имени Игоря Сикорского", Украина, Киев

Рассмотрена задача формирования команды для стартап-проекту. Построена математическая модель задачи определения кандидатов в команду проекта, которые имеют наибольший опыт работы и требуют наименьшую зарплату. Задача сведена к задаче о рюкзаке. Описан метод и обоснована целесообразность использования алгоритма ветвей и границ для решения задачи.

Ключевые слова: старап-проект, команда проекта, задача о рюкзаке, методе ветвей и границ.

Kravchenko E. I. Algorithm of branches and boundaries for the task of forming a team and investors searching for startup projects / National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, Kiev.

The task of forming a team for a startup project is considered. A mathematical model of the problem of determining candidates for the project team has been constructed. The candidates must have the most experience and require the least salary. The problem is reduced to the task of the backpack. A method is described and the expediency of using the algorithm of branches and boundaries for solving the problem is justified.

Key words: startup-project, project team, task of the backpack, branch and borders method

Вступ. Жодний стартап не перетвориться з ідеї у продукт без команди, що реалізує цю ідею, та інвесторів, що фінансуватимуть усі витрати. На сьогоднішній день існує багато стартап-проектів, де тільки для пошуку команди та інвесторів необхідно здійснити багато зусиль. Тому було б доцільно автоматизувати ці процеси та створити веб-застосування, однією з функцій якого було б формування команди та пошук інвесторів для стартап-проектів.

Мета статті. Метою цієї роботи є обґрунтування доцільності автоматизації процесу формування команди стартап-проектів та пошуку інвесторів за допомогою веб-застосування, цілями якого є:

- полегшення формування команди для стартапу;
- оптимізація відбору кандидатів на участь у проекті відповідно до потрібних проекту компетенцій;
- полегшення пошуку інвесторів для стартапу.

Ці цілі пропонується досягти, вирішуючи такі задачі: збір заявок кандидатів на участь у стартапі; аналіз заявок кандидатів на участь у стартапі; формування пулу потенційних інвесторів проекту.

Постановка задачі. Нехай існує N кандидатів на участь у стартап-проекті, де $i = \overline{1, N}$ індекс кандидата на участь в стартапі, x_i – кількість проектів, в яких і-ий кандидат брав участь відповідно до зазначених в даній вакансії компетенціях; w_i – бажана заробітна платня кандидата, W – максимальний фонд заробітної плати.

Ціллю є визначити кандидатів, що мають найбільший досвід роботи але вимагають найменшу зарплатню.

Сформульована задача є типовим представником задачі про рюкзак, яка є задачею комбінаторної оптимізації і полягає у наповнені рюкзака, що здатен витримати деяку максимальну масу, предметами, кожен з яких має певну вартість або корисність та масу. Необхідно обрати об’єкти в такий спосіб, аби максимізувати сумарну вартість (або користь), що не перевищить максимально припустиму масу [1].

В термінах означеної предметної галузі предметами, що пакують в рюкзак, є кандидати; рюкзаком є стартап-проект, для виконання якого слід сформувати команду; маса предмету – це заробітна платня кандидата на участь в стартап-проекті; вартість або корисність – це досвід кандидата (кількість проектів, в яких кандидат брав участь).

Для побудови цільової функції математичної задачі, введемо бульовий вектор H, елементами якого є h_i = 1, якщо і-ий кандидат працюватиме в проекті, та h_i = 0, якщо кандидат не буде відібраний:

$$z(H) = \sum_{i=1}^{N} h_i x_i \to \max, \qquad (1)$$

В якості обмежень означені обмеження на фонд заробітної плати по всім учасникам проекту:

$$\begin{aligned} &\sum_{i=1}^{N} w_i x_i \le W, \\ &w_i > 0, \\ &x_i \in \{0,1\} \end{aligned} \quad , \qquad (2)$$

Опис методу розв'язання задачі. Задача про рюкзак є NP-повною задачею [2], для розв'язання якої існує багато методів. Для вирішення задачі, що розглядається, вибраний метод гілок та меж [3]. В порівнянні з іншими відомими алгоритмами, цей алгоритм потребує менше пам'яті, що робить його більш прийнятним для вирішення великих задач, ніж, наприклад, розв'язання методом динамічного програмування, та є більш точним ніж, наприклад, жадібний алгоритм.

Алгоритм полягає в побудові дерева рішень і нагадує алгоритм повного перебору, але на відміну від нього, гілки, що не відповідають умові відхиляються одразу ж після того, як визначено невідповідність, і алгоритм пошуку рішення переходить до наступної гілки.

Для вирішення задачі визначимо функцію [3]:

$$\phi_k(y) = \max \sum_{i=1}^{k} h_i x_i, \quad 1 \le k \le N \ , \qquad (3)$$

$$\sum_{i=1}^{k} h_i x_i \le y, \quad 0 \le y \le b \qquad (4)$$

Рівняння (3) визначає максимальне значення цільової функції, коли кандидати обираються з перших k варіантів, а загальна заробітна плата обраних кандидатів обмежена величиною y. Вважаємо, $\phi_0(y) = 0, \ \forall y$ у випадку, коли не вибрано жодного кандидата, тобто обмеження загальної заробітної плати дорівнює 0. Для функції $\phi_k(y)$ при $k \ge 1$ справедливо наступне рекурентне співвідношення (5):

$$\phi_k(y) = \max[\,\phi_k(y), \phi_k(y-\alpha_k) + c_k\,], \qquad (5)$$

Для обґрунтування співвідношення (5) вважатиме, що коли при отриманні $\phi_k(y)$ кандидати обираються з перших k варіантів, k-ий кандидат може виявитися або обраним, або ні. Якщо він не обраний, то $\phi_k(y) = \phi_{k-1}(y)$, і загальна вага інших кандидатів обмежується величиною $y - \alpha_k$. Максимальне значення цільової функції, яке можна отримати, використовуючи перших k кандидатів, при обмеженні $y - a_k$ на загальну заробітну платню, позначено $\phi_k(y-\alpha_k)$.

За допомогою рекурентного співвідношення (5) величини $\phi_k(y)$ можна підрахувати в наступному порядку: $\phi_1(1), \dots, \phi_1(b), \dots, \phi_n(1), \dots, \phi_n(b)$, де величини $\phi_k(y), k = \overline{1,N}; y = \overline{1,b}$ є елементами матриці Φ з N рядків та b стовпців.

Побудуємо матрицю I вимірністю n рядків та b стовпців, елементи якої $i(k,y)$ мають таку семантику: якщо $i(k,y) = j$, то це означає, що j-ий кандидат використовується в $\phi_k(y)$.

Елементи 1-го рядка i(1,y) матриці I визначають за правилом (6):

$$i(1,y) = \begin{cases} 0, \text{ якщо } \phi_1(y) = 0 \\ 1, \text{ якщо } \phi_1(y) \neq 0 \end{cases}, \qquad (6)$$

Інші елементи матриці I визначаються за правилом (7):

$$i(1,y) = \begin{cases} i(k,y), \text{ якщо } \phi_{k-1}(y) > \phi_k(y-\alpha_k) + c_k \\ k, \text{ якщо } \phi_{k-1}(y) \leq \phi_k(y-\alpha_k) + c_k \end{cases}, \qquad (7)$$

Зауважимо, що якщо $i(1,y) = 1$, це означає, що 1-ий кандидат взятий в команду. Припустимо, що $i(k-1,y) = j$. Це означає, що $x_j \geq 1$ у $\phi_{k-1}(y)$. Для визначення значення x_α, яке відповідає $\phi_{k-1}(y-a_j)$ знаходимо елемент $i(k-1, y-\alpha_j)$. Якщо $i(k-1, y-\alpha_j) = j$, це означає,

що $x_\alpha \geq 1$ у $\phi_{k-1}(y-\alpha_j)$. Якщо ж $i(k-1, y-\alpha_j)=q$, це означає, що $x_q \geq 1$ у $\phi_{k-1}(y-\alpha_j)$.

Для того щоб визначити, чи вигідно брати в команду k-го кандидата, порівняємо значення $\phi_{k-1}(y)$ та $\phi_{k-1}(y-\alpha_k)+c_k$.

Якщо $\phi_{k-1}(y) \leq \phi_k(y-\alpha_k)+c_k$, беремо в команду k-го кандидата і вважатиме $i(k,y)=k$. Якщо ж $\phi_{k-1}(y) > \phi_k(y-\alpha_k)+c_k$, то невигідно брати цього кандидата і тому вважаємо $i(k,y)=i(k-1,y)$.

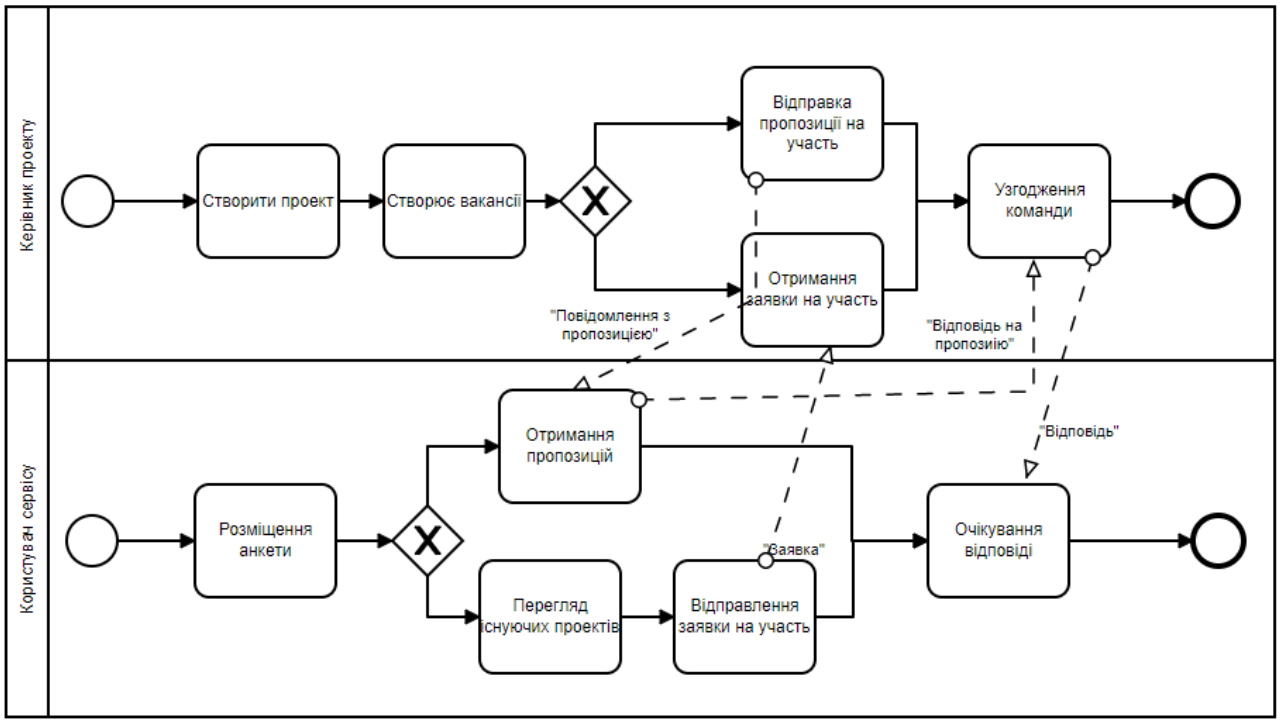

Рисунок Ошибка! Текст указанного стиля в документе отсутствует.. **Схема бізнес-процесів модулю обробки заявок**

Опис алгоритму.

Крок 0. Визначити рейтинг кожного кандидата, його заробітну платню та обмеження загальної заробітної плати.

Крок 1. Знайти значення $\phi_k(y), k=\overline{1,N}; y=\overline{1,b}$ матриці Φ з N рядків та b стовпців, де N – кількість кандидатів, b – обмеження максимальної заробітної плати.

Крок 2. Побудувати матрицю I значень $i(k, y)$.

Крок 3. За допомогою матриці I значень $i(k, y)$ знайти найкращий варіант команди для стартап-проекту.

Крок 4. Запропонувати керівнику проекту знайдену команду на затвердження.

Робота алгоритму формування команди для стартап-проекту закінчена.

Реалізація алгоритму. Розглянутий алгоритм реалізований у веб-застосуванні в модулі обробки заявок кандидатів у команду проекту.

BPMN діаграма бізнес-процесу подана на рисунку 1.

Висновки. В роботі розглянуто застосування методу гілок та меж для задачі формування команди стартап-проекту та пошуку інвестора для його реалізації. Описана змістовна постановка задачі. Побудована математична модель. Задача зведена до задачі про рюкзак, для якої визначений алгоритм на основі методу гілок та меж. Отриманий алгоритм покладений в основу модулю обробки заявок кандидатів у команду проекту веб-застосування.

Література:

1. Задача пакування рюкзака [Електронний ресурс] // Режим доступу: https://www.wikiplanet.click/enciclopedia/uk/Задача_пакування_рюкзака

2. NP-повна задача [Електронний ресурс] // Режим доступу: http://znaimo.com.ua/NP-повна_задача

3. Метод гілок і меж [Електронний ресурс] // Режим доступу: https://studfiles.net/preview/5163079/page:5/

References:

1. The task of packing a backpack. [Electronic resource] // Access mode: https://www.wikiplanet.click/enciclopedia/uk/Задача_пакування_рюкзака

2. NP-complete problem. [Electronic resource] // Access mode: http://znaimo.com.ua/NP-повна_задача

3. The method of branches and boundaries. [Electronic resource] // Access mode: https://studfiles.net/preview/5163079/page:5/