ДИПЛОМНА РОБОТА
БАКАЛАВРА

ДРБ. ІП - 48.00.00.000 ПЗ

Група ІП-18-2

Чучук Дана

2022

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

УДК _____004.4_____

ДИПЛОМНА РОБОТА БАКАЛАВРА

Тема: "Лінійні моделі побудови команд розробників програмного забезпечення"

(назва згідно з наказом ректора)

Спеціальність – 121 "Інженерія програмного забезпечення"

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДРБ.ІП - 48.00.00.000 ПЗ

(позначення)

Консультанти:

Студент

ІП-18-2 _____ /Д.В.Чучук/
(шифр групи) (підпис) (дата) (розшифровка підпису)

Керівник роботи

доц., к.т.н. _____ /Л.М. Гобир/
(посада) (підпис) (дата) (розшифровка підпису)

з питань нормоконтролю

доц., к.т.н _____ /В.Я. Піх/
(посада) (підпис) (дата) (розшифровка підпису)

Допускається до захисту
Завідувач кафедри

д.т.н.,проф. _____ /В.І.Шекета/
(посада) (підпис) (дата) (розшифровка підпису)

2022
(рік)

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**

проф. **В.І. Шекета**

“ ____ ” _____2022 р.

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ БАКАЛАВРА СТУДЕНТОВІ

Чучук Дані Віталіївні

(прізвище, ім’я, по-батькові)

1. Тема проекту (роботи) “Лінійні моделі побудови команд розробників програмного забезпечення ”

керівник проекту (роботи) Гобир Лідія Мирославівна, асистент

затверджені наказом закладу вищої освіти від **“** 25 ” травня 2022 р. №

2. Строк подання студентом проекту (роботи) 12 червня

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки (перелік питань, які потрібно розробити)

1. Огляд існуючих концепцій, рішень та сервісів в даній області

2. Алгоритмічні та математичні рішення для задач даного класу

3. Власне оригінальне алгоритмічне та програмне рішення задачі

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Інтелектуальний аналіз даних

2. Оренда кластера

3. ILP з використанням розгалужувача та вирізання

4. Соціальна мережа експертів

5.Евристичний алгоритм для задачі найму кластерів

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання **25 травня 2022 р.**

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Визначення та обґрунтування теми роботи	26.05.2022	виконано
2	Огляд існуючих концепцій, рішень та сервісів в даній області	29.05.2022	виконано
3	Побудова моделі або алгоритму власного рішення	30.05.2022	виконано
4	Документування реалізації власного оригінального рішення вибраними засобами	10.06.2022	виконано
5	Оформлення пояснювальної записки кваліфікаційної роботи	17.06.2022	виконано

Студент _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

В дипломній роботі основний акцент було зроблено на ролі цілочисельного лінійного програмування у вирішенні проблеми найму кластерів. Як свідчать експериментальні результати, розв'язання задачі наймання кластерів із застосуванням підходу математичного програмування має значний вплив на пошук оптимальної/близько до оптимальної групи експертів і є значно ефективнішим порівняно із запропонованими раніше евристичними рішеннями. Для порівняння з іншими рішеннями ми також реалізували евристичні рішення, які допомагають розрізняти якість результатів, отриманих із застосуванням обох підходів. Основні цілі роботи — максимізувати прибуток, отриманий від доходу від охоплених проектів, і мінімізувати витрати на комунікацію між експертами для формування спільної команди. Наші результати з моделей дають висновок, що він приносить максимальний прибуток і об'єднує експертів, які мають вищий рівень співпраці в порівнянні з евристичним рішенням.

КЛЮЧОВІ СЛОВА: ЦІЛОЧИСЕЛЬНЕ ЛІНІЙНЕ ПРОГРАМУВАННЯ, ЛІНІЙНА МОДЕЛЬ, МАКСИМІЗАЦІЯ ПРИБУТКУ, МІНІМІЗАЦІЯ ВИТРАТ, ЕВРИСТИЧНІ РІШЕННЯ

ANNOTATION

In the thesis, the main emphasis was placed on the role of integer linear programming in solving the problem of hiring clusters. As experimental results show, solving the problem of hiring clusters using a mathematical programming approach has a significant impact on the search for an optimal/near-optimal group of experts and is significantly more effective compared to previously proposed heuristic solutions. For comparison with other solutions, we have also implemented heuristic solutions that help distinguish between the quality of the results obtained using both approaches. The main objectives of the work are to maximize the profit received from the income of the covered projects and to minimize the costs of communication between experts to form a joint team. Our model results conclude that it maximizes profits and brings together experts who have a higher level of cooperation compared to the heuristic solution.

KEY WORDS: INTEGER LINEAR PROGRAMMING, LINEAR MODEL, PROFIT MAXIMIZATION, COST MINIMIZATION, HEURISTIC SOLUTIONS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 7
ВСТУП 8
1 ОГЛЯД ІСНУЮЧИХ КОНЦЕПЦІЙ, РІШЕНЬ ТА СЕРВІСІВ В ОБЛАСТІ СТВОРЕННЯ КОМАНД РОЗРОБНИКІВ 13
1.1 Огляд дослідження 13
1.2 Основні поняття та визначення 17
1.3 Застосування лінійного програмування 19
1.4 Опис алгоритмів для застосування 23
1.5 Огляд літератури по темі дослідження 28
1.6 Постановка проблеми 33
2. АЛГОРИТМІЧНІ ТА МАТЕМАТИЧНІ РІШЕННЯ ДЛЯ ЗАДАЧ ДАНОГО КЛАСУ 37
2.1 Основні позначення, що використовуються в ILP 37
2.2 Модель ILP для максимізації прибутку 38
2.3 Модель ILP з максимальним прибутком і мінімальними витратами на зв’язок 41
2.4 Евристичний алгоритм для задачі найму кластерів 44
3. ВЛАСНЕ ОРИГІНАЛЬНЕ АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ РІШЕННЯ ЗАДАЧІ ФОРМУВАННЯ КОМАНД РОЗРОБНИКІВ 51
3.1 Формалізація моделі прибутку від кластерного найму команд розробників 51
3.2 Евристична модель прибутку від кластерного найму на 10 000 вузлів 55
3.3 Евристична модель прибутку від найму в кластері та моделі вартості зв’язку на 50 вузлів 58
3.3 Опис результатів тестування та ефективність 60
ВИСНОВКИ 65
СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА 66
БІБЛІОГРАФІЧНА ДОВІДКА 70

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ТЗ – Технічне завдання

БД – База даних

СУБД - Система управління базами даних

ПЗ - Програмне забезпечення

ВСТУП

Однією з найпоширеніших проблем, що вивчаються в сучасному світі, є проблема ефективного розподілу ресурсів і планування. Знайшовши команду експертів у процесі найму, рекрутер шукає команду, яка зможе виконати завдання в установлені терміни, а команда, найнята таким чином, є економічно ефективною.

Проект – це добре розроблений і спланований підхід до співпраці з окремими особами для досягнення організаційних цілей і цілей компанії. Кожен проект складається з різноманітних завдань і заходів, які потребують експертів, які мають глибокі знання та навички, необхідні для виконання цього завдання. Організаціям потрібна така команда експертів, які володіють необхідними навичками для безперебійної реалізації проекту. Ця проблема, де ми прагнемо знайти команду експертів, щоб охопити всі навички, необхідні для проекту, називається проблемою формування команди .

Проблема найму в кластері передбачає найм людей для певного набору проектів таким чином, щоб найняті експерти володіли всіма навичками, необхідними для цього набору проектів. Крім того, оскільки кожен експерт має власні витрати (тобто заробітну плату), ми гарантуємо, що наймаємо людей у межах бюджету, виділеного на цей набір проектів.

Формування команди передбачає найм груп експертів у великих установах та організаціях, де кожен експерт має додаткові навички, які допомагають ефективному вирішенню проблем та успішній реалізації проекту. Іншими словами, враховуючи набір експертів, кожен з яких має різноманітний набір навичок, мета полягає в тому, щоб найняти команду експертів, які зможуть спільно охопити необхідні навички експертів, переконавшись, що найнята група експертів створить найбільш спільну команду. . Після завершення проекту кожна організація отримує прибуток. Кожен проект має власний набір навичок і кількість експертів, необхідних для кожного навику. Отже, набір найнятих експертів повинен володіти всіма навичками, необхідними для набору проектів.

Для найму кожного експерта організація має сплатити експерту економічні витрати (тобто зарплату експерта). Організації також призначається бюджет, який не може бути перевищений у процесі найму групи експертів. Інакше кажучи, сума заробітної плати осіб, найнятих для виконання комплексу проекту, не повинна перевищувати визначеного компанією бюджету. Така настройка допомагає максимізувати прибуток організації.

Об’єднання експертів, які разом працювали в попередніх проектах, допомагає швидше завершити проект, а також отримати якісні результати. Припускаючи, що між експертами існує канал зв’язку, вартість зв’язку між двома експертами може бути розрахована на основі попереднього досвіду роботи. Якщо будь-які два експерти раніше працювали разом у минулому, то вважається, що вони мають кращу частоту спілкування в порівнянні з експертами, які представлені в команді вперше. Наявність хорошої частоти дає багато переваг, як-от якість роботи та ефективне дотримання термінів проектів. Крім того, групування осіб з мінімальною або відсутністю зв’язку між ними може збільшити витрати на проект, тим самим зменшуючи дохід, отриманий від проекту. Визначення вартості зв'язку залежить від потреби проекту. Зрештою, організації шукають команди, які приносять більше прибутку від проектів.

Працездатність експерта – ще один важливий параметр при найму експерта для кількох проектів. Менеджери з підбору персоналу також стежать за тим, щоб можливості співробітника не були вичерпані в процесі найму експертів, які принесли найбільший дохід для організації. Кожен експерт має максимальну потужність, яку він/вона може запропонувати. Комбінація витрат на комунікацію між двома експертами та можливостей кожного експерта допомагає створити декілька можливих команд, з яких можна отримати команду, яка приносить максимальний прибуток.

Для вирішення проблеми найму кластерів ми вводимо такі функції, як потужність експерта, вартість співпраці між двома експертами, кількість експертів, необхідних для кожного навику на проект, бюджет, призначений для

найму експертів для певного проекту та прибуток, отриманий кожним проектом для успішне завершення проекту.

У світі технологій багато компаній розвиваються по всьому світу, де власникам потрібно вибрати команду експертів з колективним досвідом, необхідним для використання різноманітних можливостей, які були виявлені на ринку, на який націлена компанія . У цьому контексті важливо зробити ефективний розподіл ресурсів, щоб можна було досягти певних цілей організації, наприклад, щоб можна було досягти законних і соціальних зобов'язань організації щодо системи її робочої сили. Інша відповідна мотивація походить від веб-сайтів ринку праці в Інтернеті, таких як Freelancer і Guru, де вони отримують кілька проектів з кількох віддалених місць. Спочатку фрілансери працюють самостійно, але пізніше зі збільшенням розміру та складності проекту; ці експерти співпрацюють, щоб створити консалтингову компанію, яка отримує проекти, навички та кількість людей, необхідних для проектів. Для кожного навику, необхідного для проекту, консалтингова компанія наймає експертів для виконання комплексу необхідних завдань. Існує бюджет, визначений фінансовим відділом для найму експертів для проекту, і, отже, загальна заробітна плата всіх експертів повинна бути меншою або дорівнювати бюджету. Мета компанії - отримати якнайбільший прибуток від виконання проектів.

Щоб подолати вищеописану проблему, люди запропонували багато способів знайти найбільш спільну команду для окремого проекту та команду, яка максимізує прибуток для ряду проектів, яка складається з найбільш спільних експертів.

Проблема кластерного найму досліджувалася багатьма, щоб сформувати групу експертів, які мають навички, необхідні для виконання підгрупи проектів, які приносять максимальний прибуток за визначеного бюджету. Різновидом цієї проблеми є кластерний найм у соціальній мережі, де поряд з охопленням підгрупи проектів експерти відбираються на основі їх попередньої співпраці з

іншими експертами в групі, тому відібрані експерти мають хорошу сумісність і кращу комунікацію, що допомагає у виконанні проекту вчасно та якісно.

Ми вивчаємо проблему найму в кластері з наступними припущеннями:

1. Кожен експерт володіє набором навичок.

2. Експерти мають ємність робочого навантаження, яка вказує, скільки максимальної кількості навичок можна призначити.

3. Кожен експерт має відповідні витрати на найму. Вартість найму розраховується лише один раз. Тобто експерт може бути призначений для виконання ряду проектів і може виконувати ряд навичок залежно від його/її працездатності.

4. Існує набір проектів, кожен з яких має певні навички та кількість експертів, які мають ці навички, необхідні для завершення проекту.

Експерти об’єднані в мережу, яка моделюється у вигляді графа. Кожен експерт пов’язаний з вузлом графа. Ребро з’єднує двох експертів у графі з вагою краю, що представляє силу співпраці між двома експертами, отриманими в результаті попередньої співпраці в минулому. Якщо експерти не з’єднані, вага призначається сумою ваги на найкоротшому шляху між двома експертами. Найменше комунікація коштувала найбільшої співпраці між експертами. Наприклад, якщо є дві пари експертів, де перша пара експертів працювала в десяти проектах у минулому, а інша пара експертів працювала в дев’яти проектах у минулому, тоді буде обрано колишніх експертів, оскільки вони, як правило, мати кращу частоту, тобто вартість зв’язку, порівняно з останньою парою експертів. Ми моделюємо вартість комунікації між членами команди, використовуючи суму відстаней між кожною парою експертів у групі.

Для виконання певного набору проектів нам надається заздалегідь визначений бюджет, який буде витрачено на найм експертів. Спочатку ми вирішуємо проблему найму групи експертів для комплексу проектів, вважаючи команду найбільш спільною та приносить максимальний прибуток.

Пропонується нова формула для розрахунку вартості зв’язку між двома експертами, яка називається функцією суми відстані. У своєму дослідженні всі вони надають детальний аналіз обчислювальної складності, наближення рішення задачі до оптимального, а також евристичні рішення. Вони продемонстрували ефективність свого підходу, проводячи експерименти на реальних наборах даних експертів і продемонстрували свої переваги та вихідні положення. Суть кластерної проблеми найму в мережі, де вони мають на меті наймати групу експертів для набору проектів, які дають максимальний прибуток, у якому кожен проект має пов’язаний дохід, пов’язану суму, яка отримується після завершення проекту. Вивчають різні атрибути, пов’язані з командою та її членами, такими як співпраця між двома експертами, здатність окремого експерта, заробітна плата експертів. Вони також стежать за тим, щоб вартість найму експертів була в межах бюджету, призначеного фінансовим відділом компанії. Далі вони пропонують два жадібних алгоритми, і їхні результати були отримані в результаті експериментів, проведених на великому графіку експертів.

У наших умовах ми маємо набір проектів, кожен з яких має навички та кількість експертів, необхідних для виконання проекту. Наша мета — максимізувати прибуток, отриманий шляхом підсумовування доходу, пов’язаного з кожним проектом. Крім того, ми модифікуємо наше запропоноване рішення, щоб воно приносило максимальний прибуток, а також вибирало експертів, які найбільше співпрацюють один з одним.

Ми застосовуємо цілочисельне лінійне програмування, яке використовує математичне програмування для вирішення складних завдань прийняття рішень. Ми показуємо моделі , створені для вирішення проблеми найму кластерів із соціальною мережею та без неї, а також порівнюємо нашу роботу з евристичним рішенням.

1 ОГЛЯД ІСНУЮЧИХ КОНЦЕПЦІЙ, РІШЕНЬ ТА СЕРВІСІВ В ОБЛАСТІ СТВОРЕННЯ КОМАНД РОЗРОБНИКІВ

1.1 Огляд дослідження

Процес виявлення прихованих закономірностей у величезних наборах даних за допомогою таких методів, як машинне навчання, статистика та системи баз даних, називається Data Mining. Це одна з підгалузей інформатики та статистики із загальною метою вилучення інформації з набору даних і перетворення інформації в добре описану структуру для подальшого використання. Інтелектуальний аналіз даних допомагає отримати важливу інформацію з даних; це широко відоме як виявлення даних знань. Щоб не відставати від темпів генерації даних, необхідно мати гарну техніку аналізу даних, щоб виявити цікаві закономірності для аналізу тенденції. Залежно від цих тенденцій організація може покращити процес прийняття рішень, щоб отримати більше доходу або залучити більше клієнтів для свого бізнесу. Застосування таких методів є дорогим і може бути доступним не кожному типу бізнесу. Люди вибирають різні методи, виходячи з характеру даних, і це визначило долю цих компаній.

Відкриття знань у базі даних здійснюється покроковою процедурою:

• Очищення даних: на цьому кроці видаляються шуми та суперечливі дані.

• Інтеграція даних: на цьому кроці об'єднуються кілька джерел даних.

• Вибір даних: на цьому кроці дані, що стосуються завдання аналізу, витягуються з бази даних.

• Трансформація даних: на цьому кроці дані перетворюються або консолідуються у форми, придатні для видобутку шляхом виконання операцій зведення або агрегації.

• Data Mining: на цьому кроці застосовуються інтелектуальні методи для вилучення шаблонів даних.

• Оцінка шаблону: на цьому кроці оцінюються шаблони даних.

• Презентація знань: на цьому кроці представлено знання.

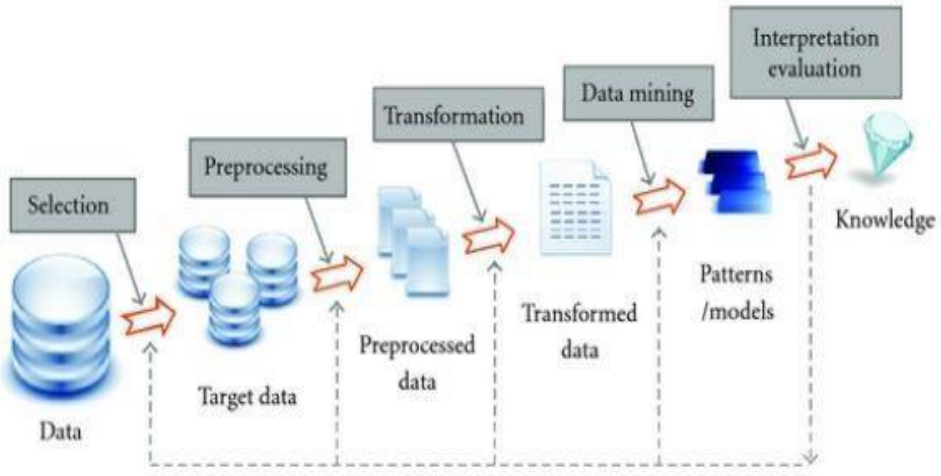


Рисунок 1.1 - Інтелектуальний аналіз даних

Формування команди

Концепція формування команди полягає в тому, щоб найняти набір експертів, які володіють знаннями в одній або кількох навичках, необхідних для завершення проекту, що принесе користь установі після його завершення. Це допомагає організаціям завершити реалізовані проекти, а також допомагає експертам зміцнити свої навички та позицію в компанії, працюючи добре. Основна ідея цієї концепції полягає в тому, щоб зібрати людей, які мають досвід у навичках, необхідних для успішного завершення проекту. Відбір людей для роботи в команді може здійснюватися різними способами з використанням різних обмежень і факторів. Наприклад, команда отримує вигоду від подібності між членами команди. Це може допомогти зменшити будь-які конфлікти або непорозуміння. Наявність кількох відмінностей також може допомогти зменшити кількість часу, необхідного для того, щоб стати ефективною робочою групою, оскільки менше потрібно коригувати індивідуальні стилі роботи. Крім того, залучення більш різноманітних кваліфікованих людей допомагає розширити мислення та перспективи роботи, додаючи приміщення до гнучкості та генерації ідей.

Оренда кластера

Наймаючи групу осіб, які відповідають вимогам набору проектів у рамках бюджету організації, називають кластерним наймом. Наприклад, якщо компанія хоче згрупувати експертів для п’яти проектів, кожен з яких має заздалегідь визначені навички, необхідні для його завершення, завданням тут буде знайти експертів, які володіють цими навичками. Це називається кластерним наймом.

На процес відбору впливають різні фактори, такі як бюджет, призначений для набору проектів, прибуток, отриманий від завершення кожного проекту, вартість комунікації між експертами в команді, працездатність кожного тощо. Ця концепція набуває широкого визнання та важливості в ринку, враховуючи, що сьогодні все більше організацій працюють у середовищі, керованому проектами. Крім того, наймати набір осіб, які охоплюють різні проекти, краще, ніж наймати фіксований набір осіб для кожного проекту. Такий підхід зрештою допомагає знизити фактор бюджету

Фактори, які слід враховувати при формуванні команди

1) Бюджет

Бюджет — це верхня межа грошової величини, що застосовується компанією до вартості найму осіб для певного ряду проектів. Мета організації — підібрати команду експертів таким чином, щоб бюджет не був перевитрачений.

2) Експерти та їх атрибути

Навички експерта: кожен експерт або особа на ринку праці чи в компанії відомі своїми знаннями в певному наборі навичок. Ми прагнемо відбирати людей для проектів залежно від навичок, які експерти можуть принести разом із собою, що допоможе досягти швидшого та кращого результату проекту.

Вартість експерта: кожен експерт вимагає певної грошової винагороди за свої навички, які він/вона надає. Необхідно пам’ятати, що сума витрат на найм відібраних експертів не повинна перевищувати бюджет, призначений для найму команди для певної групи проектів.

Здатність експерта: експерт може володіти кількома навичками. У ситуаціях, коли існує кілька навичок, якими володіє експерт, які є корисними для множини проектів, ми призначаємо експертам лише ті навички, які є в його працездатності. Наприклад, експерт має шість навичок, корисних для даного набору проектів, але експерт може працювати лише з трьома навичками. Ми подбаємо про те, щоб фахівець не був переобтяжений і отримав навички відповідно до його робочих можливостей.

Вартість співпраці експертів: у цьому спеціальному налаштуванні ми також дбаємо про те, щоб вибрані експерти мали найкращу співпрацю, тобто частоту спілкування між ними. Ми використовуємо їх попередню співпрацю та випадки, коли обидві особи працювали разом, щоб підрахувати вартість зв'язку між ними. Формування такої команди експертів допомагає скоротити накладні витрати часу, витраченого на розуміння походження один одного тощо.

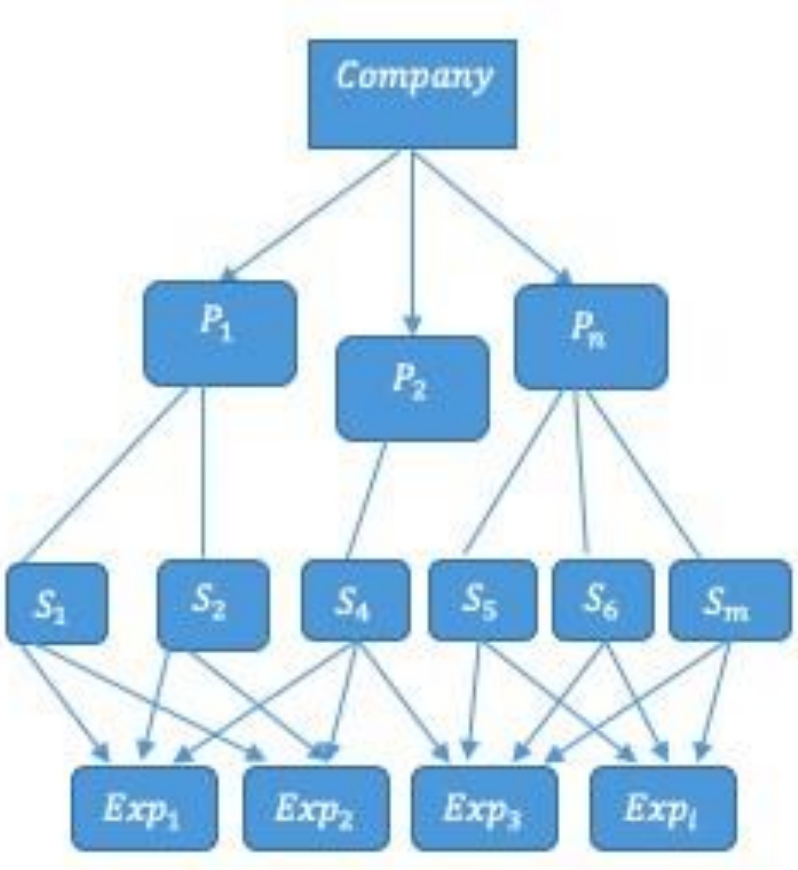


Рисунок 1.2 - Оренда кластера

3) Проект та його атрибути

Навички проекту: кожен проект складається із заздалегідь визначених навичок, необхідних для отримання бажаного результату. Ми шукаємо людей, які мають хороший рівень знань у навичках, необхідних для проекту.

Попит на експертів: ми також повинні переконатися, що знайдемо не менше, ніж кількість експертів, необхідних для виконання певних навичок, необхідних для проекту. Наприклад, для проекту потрібні дві особи, які є експертами зі штучного інтелекту, і три експерти з управління базами даних. Отже, наше завдання полягає в тому, щоб знайти ту кількість експертних пар навичок, які потрібні проекту для його успішного завершення

Дохід від проекту: враховуючи всі вищезазначені фактори, група експертів, відібрана для кожного проекту, допомагає отримувати дохід для проекту. З кожним проектом пов’язаний заздалегідь визначений дохід (грошовий прибуток), який компанія отримує після успішного завершення проекту.

Мета нашого підходу – залучити достатню кількість експертів для набору проектів для отримання максимального прибутку. Роблячи це, ми дбаємо про те, щоб усі вищезгадані обмеження не були порушені, а сформована команда була найбільш спільною.

1.2 Основні поняття та визначення

Лінійне програмування — це математична модель, яка використовується для розв’язування задач прийняття рішень, які мають багато змінних рішень, які обмежені набором обмежень. Для лінійної програми обмеження та цільові функції повинні бути лінійно пов’язані зі змінними задачі.

Термінологія лінійного програмування

1. ЗМІННА ПРИЙНЯТТЯ РІШЕННЯ: Ці змінні визначають кількість, яку хотіли б визначити особи, які приймають рішення. Зазвичай їх оптимальні значення визначають за допомогою методу оптимізації. Загалом, змінні рішення представлені за допомогою алгебраїчних позначень, наприклад

2. ОБМЕЖЕННЯ: Обмеження – це нерівність або рівність, що визначає обмеження для рішень. Загалом кажуть, що LP має m лінійних обмежень, які можна сформулювати як

3. ОБ'ЄКТНА ФУНКЦІЯ: Цільова функція мінімізує або максимізує кількісний критерій, такий як вартість, прибуток, корисність або прибутковість. Загальну лінійну цільову функцію можна записати так

4. ПРОСТА ВЕРХНЯ МЕЖА: з кожним значенням тут може бути пов'язана певна величина, _ яка обмежує її значення зверху;

Якщо для змінної не вказано просту верхню межу, кажуть, що змінна необмежена зверху

5. ОБМЕЖЕННЯ ТРИВАЛОСТІ: У більшості практичних завдань змінні повинні бути невід’ємними;

$\geq 0, =1\ldots.$

Цей особливий тип обмеження називається обмеженням відсутності негативу.

6. ПОВНА МОДЕЛЬ ЛІНІЙНОГО ПРОГРАМУВАННЯ: Об'єднання вищезгаданих компонентів в один оператор дає:

Звести до мінімуму

Обмеження, включаючи невід’ємність і прості верхні межі, визначають можливу область проблеми.

Цілочисельне лінійне програмування (ILP)

Лінійне програмування (LP) — це підхід до математичного програмування, коли задача моделюється, щоб знайти рішення для мінімізації або максимізації функції за певних обмежень. Цілочисельне лінійне програмування (ILP) — це галузь лінійного програмування, яка обмежує всі змінні моделі лише цілими числами. Загальна цілочисельна модель лінійного програмування може бути представлена у вигляді

$$Max\ cx$$

$$Ax = b\ldots\ldots\ldots (2)$$

$$x \geq 0 \; integer$$

Усі записи , і вважаються цілими. Рівняння 2 надає одну можливу формулювання проблеми ILP. Крім того, ми маємо проблему мінімізації або проблеми з обмеженнями нерівності.

Проблеми мінімізації можна перевести на задачу максимізації, просто використовуючи наступне позначення

$$-min\,(-f(x)) \;=\; max(f(x))$$

Обмеження нерівності можна перетворити на рівність шляхом додавання допоміжних змінних. Наприклад

$$ax \;<=\; b \; can \; also \; be \; denoted \; as \; ax \;+\; s = b,$$

$$and \quad ax >= b \; can \; also \; be \; denoted \; as \; ax - t = b$$

Змінні s і t відомі як змінні слабкості або надлишку.An important special case of the ILP problem is called binary ILP problem described by

$$Max \; cx$$

$$Ax \;=\; b \ldots\ldots\ldots (2)$$

$$x \geq 0 \; binary$$

Тут двійковий означає =0 або =1 для всіх

1.3 Застосування лінійного програмування

Цілочисельне лінійне програмування знайшло свій шлях як один з провідних методів пошуку оптимальних рішень у сфері дослідження операцій. Добре відомі проблеми, такі як розподіл ресурсів, призначення завдань, планування ресурсів, можна вирішити за допомогою методів цілочисельного

лінійного програмування. Нижче наведено деякі класичні програми, в яких ILP створив еталон як засіб вирішення проблем.

1) Інформатика: одна з найбільших проблем, з якою стикаються організації в сучасному світі, — це працювати з великим набором даних і знаходити способи отримати важливе уявлення про дані, які можуть допомогти у кращому прогнозуванні або аналізі майбутньої тенденції продажів компанії або формування прибутку. Процес визначення закономірностей і значущих ідей з великого набору даних називається інтелектом даних. Методи аналізу даних допомагають організаціям отримувати дохід і приймати конкретні рішення на основі відкритів і прогнозів, зроблених за допомогою даних. Лінійне програмування надає методи для визначення такої моделі та надання даних більш розумним і корисним для прогнозування та аналізу. Процес цілочисельного лінійного програмування включає формулювання моделі ILP для задачі, використовуючи змінні рішення та обмеження, які надаються цільовій функції.

2) Виробництво: ILP є одним із найбільш широко використовуваних математичних підходів, що використовуються у обробній промисловості для розподілу ресурсів та максимізації прибутку. Планування ресурсів важливе для обробної промисловості, яка вимагає належного планування оптимального набору продуктів, які необхідні для виробництва. Одним з найпоширеніших методів ILP, які допомагають у плануванні та виробництві ресурсів, є симплексний метод, який використовує концепцію слабких змінних, таблиць і зведених змінних для пошуку оптимального рішення задачі оптимізації.

3) Оптимізація транспортування: транспортні системи залежать від лінійного програмування для отримання рішень, які дають ефективні за витратами та часом результати. Майже всі транспортні системи вважають планування перевезень, пасажирів і своєчасну доставку своїм ключовим пріоритетом. Авіакомпанії застосовують методи лінійного програмування, встановлюючи вартість і попит на подорожі як фактори для максимізації свого

прибутку. Так само службовці автобусного транспорту використовують LP для планування маршрутів і розподілу водіїв. Оптимізація з використанням ILP сприяє ефективному прийняттю рішень і зменшує витрати.

4) Енергетична промисловість: у сучасній системі управління мережею ми залучаємо не лише традиційне джерело електроенергії, а й джерела невідновлюваних джерел, таких як вітер та фотоелектричні пристрої. Таким чином, стає важливим оптимізувати навантаження між вимогами, передачею, генераторами та розподільними лініями. Лінійне програмування використовувалося в електротехнічній промисловості як метод перепроектування конструкції енергосистеми, що допомагає відповідати електричному навантаженню з огляду на попит на електроенергію та час її виробництва.

Проблема капітального бюджету

Фірма має n проектів, але через бюджетні обмеження не всі можуть бути обрані.

Проект j має поточну вартість cj і потребує інвестицій у aij за період часу i, де $i = 1 \ldots \ldots, m$. Доступний капітал за період часу i становить bi . Проблему максимізації загальної теперішньої вартості з урахуванням бюджетних обмежень можна записати у вигляді

$$0 \leq xj \leq uj \ for \ j = 1 \ldots n$$

де xj=1, якщо проект j вибрано, і xj=0, якщо проект j не вибрано

Проблема фіксованого заряду

Загалом, вартість діяльності є спеціальною нелінійною функцією рівня активності x, заданою як

Якщо $d > 0$ і f потрібно мінімізувати, ми маємо проблему

хв $cx + dy$

$x \geq 0$ предмет

$x - vy \leq 0$ $y = 0,1$

де y є показником того, виконується чи ні, а u — відома кінцева верхня межа для x. Друге обмеження гарантує, що x > 0 означає y=1

Проблема розташування заводу

Розглянемо n клієнтів, $j - th$ той, якому потрібно bj одиниць товару. Є m місць, у яких заводи можуть працювати, щоб задовольнити потреби.

Існує фіксована плата в розмірі di за відкриття заводу i, а вартість одиниці постачання клієнту j з заводу становить cij. Потужність заводу i становить hi.

Сформульована модель ILP є

$$\min \sum_{i=1}^{m} \left(\sum_{j=1}^{n} c_{ij} x_{ij} + d_i y_i \right)$$
$$\sum_{j=1}^{m} x_{ij} = b_i$$
$$\sum_{j=1}^{n} x_{ij} - h_i y_i \leq 0$$
$$x_{ij} \geq 0, \qquad y_i = 0, 1.$$

Проблема з ранцем

Припустимо, n різних типів наукового обладнання береться до уваги для включення до космічного апарату. Нехай cj — наукова цінність одиниці aj ваги одиниці j-го типу. Якщо загальне обмеження ваги b, то проблема максимізації загальної вартості узятого обладнання є

$$\sum_{j=1}^{n} a_j x_j \leq b$$
$$xj \geq 0, integer$$

де xj — кількість включених одиниць типу $j - th$.

1.4 Опис алгоритмів для застосування

Алгоритми, що використовуються для задач цілочисельного лінійного програмування, спираються на дві основні концепції:

1. LP Релаксації

Видалення обмежень цілісності з цілочисельної лінійної програми називається релаксацією. Це дозволяє змінним моделі ILP набувати неінтегральних значень. Наприклад,

$xi \epsilon \{0,1\}$ можна замінити двома неперервними змінними для $xi \geq 0$ і $xi \leq 0$ відповідно. Отже, результуюча релаксація для задачі цілочисельного лінійного програмування є лінійною програмою. Цей метод допоможе перетворити задачу оптимізації NP-Hard у пов'язану проблему, яка розв'язується за поліноміальний час, забезпечуючи майже оптимальне рішення вихідної задачі. LP релаксацію можна використовувати для отримання інформації про рішення вихідної цілочисельної програми.

2. Межі LP релаксації вирішує проблему ILP і забезпечує рішення LP, яке дає верхню межу цілочисельної лінійної задачі. Задача ILP вирішується, щоб знайти найкраще рішення у можливій області, що дає результат поблизу верхньої межі, отриманої релаксаціями LP.

Для цілочисельного лінійного програмування використовуються три основні типи алгоритмів

а) Точний алгоритм:

Класичні підходи до точного вирішення комбінаторної задачі оптимізації називаються точними алгоритмами. Вони гарантують знаходження оптимального рішення, але можуть потребувати експоненційної кількості ітерацій. Деякі з прикладів точних алгоритмів, широко відомих, - це січуючі площини, розгалуження та межі та динамічне програмування.

б) Евристичні алгоритми

Евристичні алгоритми - це алгоритми, які забезпечують неоптимальне рішення, але не гарантують якість отриманого рішення. Хоча час обчислення не гарантується поліноміальним, з емпіричних даних відомо, що деякі з цих алгоритмів знаходять хороші рішення.

в) Алгоритми апроксимації

Це алгоритми, які забезпечують неоптимальне рішення за поліноміальний час разом із обмеженням ступеня субоптимальності.

Точний алгоритм

Алгоритм розгалужень і обмежень використовує підхід «розділяй і володарюй» для дослідження набору можливих рішень. У ньому використовується концепція меж оптимальної вартості, щоб уникнути дослідження всіх можливих цілочисельних рішень. Алгоритм складається з систематичного перерахування кількох рішень, що подібне до кореневого дерева. Досліджується кожна гілка цього дерева, яка представляє підмножини набору рішень. Перед вибором будь-якої гілки він перевіряє значення з верхньою та нижньою межами для оптимального рішення і відкидається, якщо не може створити краще рішення, ніж найкраще, знайдене на даний момент за допомогою алгоритму.

Кроки, які використовує алгоритм для визначення оптимального цілого рішення для моделі ILP, можна перерахувати наступним чином:

1. Знайти оптимальне рішення для моделі лінійного програмування зі послабленими цілочисельними обмеженнями.

2. У вузлі 1 нехай розслаблене рішення є верхньою межею, а округлене цілочисельне рішення — нижньою.

3. Виберіть змінну з найбільшою дробовою частиною для розгалуження. Створіть два нових обмеження для цієї змінної, що відображають розділені цілі значення. Результатом буде нове обмеження ≤ і нове обмеження ≥.

4. Створіть два нових вузли, один для обмеження ≤ і один для обмеження ≥.

5. Розв’яжіть розслаблену модель лінійного програмування з новим обмеженням, доданим у кожному з цих вузлів.

6. Розслаблене рішення є верхньою межею для кожного вузла, а існуюче максимальне ціле рішення (у будь-якому вузлі) є нижньою межею.

7. Якщо процес дає можливе ціле рішення з найбільшим значенням верхньої межі будь-якого кінцевого вузла, оптимальне цілочисельне рішення було досягнуто. Якщо можливе ціле рішення не виходить, відгалужуйте від вузла з найбільшою верхньою межею.

8. Поверніться до кроку 3.

Деякі з недоліків алгоритму Branch і Bound

1. Розгалуження на підзадачі, а потім їх розв’язування призводило до великого часу обчислень

2. Розгалуження на вузлі з найменшою межею збільшило простір для зберігання

3. Час виконання зростає в геометричній прогресії з розміром проблеми, але невеликі помірні проблеми можна вирішити за розумний час

Для подолання вищезгаданих обмежень і недоліків було введено алгоритм Branch and Cut, який детально пояснюється нижче.

Алгоритм розгалуження та розрізання

Метод Branch and Cut є одним із найпотужніших та найуспішніших алгоритмів розв’язування задач цілочисельного лінійного програмування. Це точний алгоритм, що складається з методу січної площини та алгоритму розгалужень і обмежень. Алгоритм розгалуження та розрізання використовує два прийоми

1.Сікучі площини: для вирішення ILP спочатку розглядається послаблення задачі, щоб відрізати частини багатогранника шляхом додавання нових обмежень, щоб отримати найкраще цілочисельне рішення. Для того щоб

розрізати площини, вивчається і потім відокремлюється дійсна нерівність ILP, яка спричиняє порушення.

2. Розгалуження та обмеження: усі змінні впорядковані подібно до дерева перерахування, потім це дерево частково обходить для обчислення глобальних нижніх і локальних верхніх меж, які використовуються, щоб уникнути частин дерев, які не дають оптимального рішення.

На малюнку нижче показано загальну стратегію, яку використовує алгоритм Branch and Cut

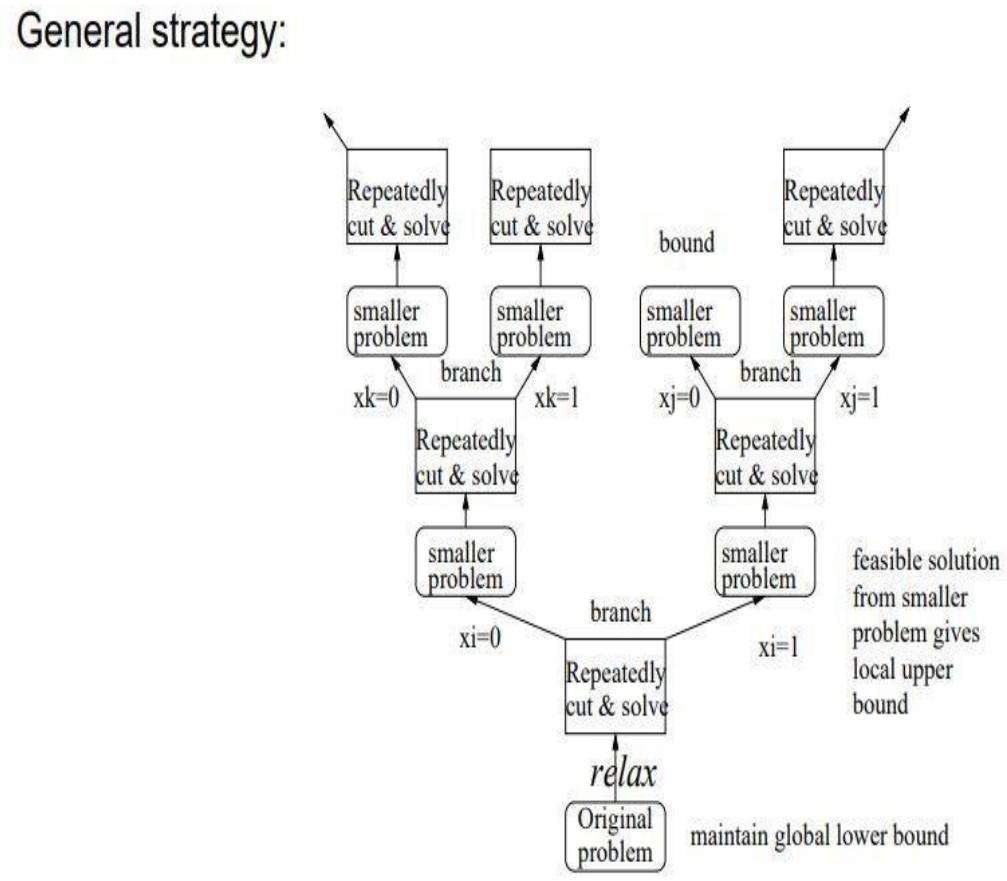


Рисунок 1.3 - ILP з використанням розгалужувача та вирізання

Евристичні алгоритми

Термін евристичний використовується для алгоритмів, які знаходять найкраще рішення серед усіх можливих рішень, але вони не гарантують, що найкраще рішення знайдено. Підхід намагається знайти близьке до

оптимального або оптимальне рішення в деяких випадках. Залежно від використовуваної евристичної логіки, алгоритм вирішує, якої гілки слід слідувати, щоб досягти найкращого можливого рішення для даної проблеми. Рішення приймається залежно від наявної інформації на кожному кроці.

Єдиним недоліком цього підходу є те, що він суперечить оптимальності, точності результату, який повертає алгоритм, і повноті. Перевага використання цього підходу полягає в тому, що він забезпечує швидкі результати, хоча оптимальність результату, отриманого за допомогою алгоритму, не може бути гарантовано.

Для розв'язання задачі комівояжера використовується добре відомий приклад евристичного алгоритму. Проблема полягає в наступному: наведено список міст і відстань між містами, який є найкоротшим маршрутом, за допомогою якого можна відвідати якомога більше міст рівно раз.

Евристичний алгоритм, який використовується для швидкого вирішення проблеми, — це використання алгоритму найближчого сусіда (NN), у якому ми починаємо з випадкового вибору міста, а потім знаходимо найближче місто. Потім відвідують інші міста, поки не буде знайдено найближче місто.

Нижче наведено кроки алгоритму NN.

1. Почніть з випадкової вершини з заданого набору вершин.

2. Визначте найкоротшу відстань, яка з'єднала поточну вершину з невідвіданою вершиною V

3. Зробіть поточну вершину, невідвідану вершину

4. Зробіть V як відвідану вершину

5. Зверніть увагу на пройдену відстань

6. Завершіть роботу, якщо не залишилося інших невідвідуваних вершин

7. Повторіть кроки з 2 по 5

Цей алгоритм є евристичним, тому він не враховує можливість виключення кращих кроків через процес відбору.

1.5 Огляд літератури по темі дослідження

Формування команди Проблемні та вартісні функції для найкращого формування команди в

Соціальні мережі автор Lappas та інші були першими, хто представив проблему формування команд у спільноті інтелекту та управління даними. Вони дали дві функції для оцінки вартості комунікації команди. Перша функція визначає найбільший найкоротший шлях між будь-якими двома вузлами утвореного підграфа. Друга функція враховує вартість мінімального остовного дерева (MST) на підграфі. Ці функції мали певні недоліки і потім були вдосконалені Kargar et. al, де вони ввели дві нові функції вартості. Перша функція, відома як функція sumofdistance, виявляє вартість комунікації команди на основі підсумовування найкоротшої відстані між експертами, які володіють навичками, необхідними для проекту. Друга функція включає вартість комунікації лідера з усіма іншими експертами команди і відома як функція відстані лідера, яка обчислює суму найкоротшого шляху між лідером і кожним експертом у проекті. Ми використовуємо наведену вище функцію сумовідстані та перекладаємо проблему формування команди в математичну модель, використовуючи підхід цілочисельного лінійного програмування.

Соціальна мережа – важливий фактор ефективного формування команди

Ефективна організація команди тісно пов’язана зі структурою, яка передбачає формування команди для виконання складних завдань. Автор Matthew et. ін. доводять, як такі соціальні структури відіграють життєво важливу роль у виконанні завдань та успішному завершенні проектної роботи. Кожен проект і команда моделюються підграфом, де набір вершин представляє експерта, який володіє колективними навичками, необхідними для виконання поставлених завдань. Потім вони проводять експерименти для вивчення впливу мережі на реальні дані та показують, як соціальна мережа серед членів команди впливає на продуктивність команди.

Динаміку формування груп досліджують автори Lars et. ін., де вони розглядають три параметри: членство, зростання та еволюцію, для аналізу розвитку великих груп в організації/спільноті. Вони вивчають, як різні фактори впливають на бажання людини приєднатися до спільноти і як компанії проводять дослідження цих факторів для їх масштабованості. У цій статті сказано, що бажання людини приєднатися до спільноти залежить від кількості людей, з якими вона дружить в цій організації. Методи дерева рішень використовуються для визначення найбільш помітних детермінант, що впливають на таку соціальну групу. Крім того, вони створили нову методологію, яка вимірює вхід і вихід людей з однієї спільноти в іншу залежно від рівня їхнього інтересу до певної теми. Експеримент та результати отримані шляхом виконання запропонованого методу на наборі даних DBLP.

Генетичний алгоритм формування команди

Цілочисельне лінійне програмування використовується для моделювання моделі формування команди разом із обмеженнями на основі знань і співпраці як факторів, які впливають на цільову функцію. Оцінка знайомства розраховується як середній час знайомства двох осіб. У контексті набору даних DBLP автори використали раніше опубліковані статті та інтервал часу між ними, щоб сформулювати оцінку знайомства. Аналогічно, оцінка знань вираховується на основі ключових слів, витягнутих із статей, опублікованих цими авторами. Автори Chien et. ін. зосереджуються на призначенні хорошого менеджера команди на основі аналізу знань, які він/вона має, та вибору інших членів команди, щоб існувала хороша синергія, необхідна для успішного та своєчасного завершення проекту. Оскільки це NP-складна нелінійна задача, для розв’язання задач пропонується генетичний алгоритм

Проблема формування команди розв'язується різними прийомами

Автор Adil et. ін. у цій роботі пропонується оптимізаційна модель, що складається з цільових функцій, обмежених обмеженнями, заснованими на

критеріях, які керують вибором найкращої команди для проекту. Вони вивчають різні людські та нелюдські фактори, які впливають на здоров'я проекту, і формулюють теми, які керують рішеннями щодо формування команди. Математична модель спрямована на максимізацію сумісності між членами команди на основі таких факторів, як вартість експерта, розмір команди, бюджет, виділений для команди, кількість заходів, задіяних у проекті тощо. Нечіткі цілі та обмеження потім перекладаються на змодельований алгоритм відпалу, який дає бажаний результат. У роботі автор Xinyu et. ін. вирішує проблему формування команди з урахуванням різних підходів. Кожен підхід використовує різні програми для формування команди експертів.

1. Алгоритми R-TF. Цей підхід використовувався для найму групи відомих експертів для рецензування статті.

2. Алгоритм Штайнера-ТФ. Цей алгоритм припустив, що команда рецензентів, найнята на роботу, може давати різноманітні результати, оскільки рецензенти мають широкий спектр досвіду.

3. Алгоритми SD-TF: автор припустив, що цей підхід можна використовувати для найму експертів, які прагнуть піднятися на своїх посадах і є новими в галузі.

4. Алгоритми LD-TF: гарантує, що під час призначення експертів за навичками працездатність експертів не перевантажується

Усі наведені вище підходи використовують один атрибут або функцію для формування команди, що робить це трохи нереалістичним, оскільки для кожного визначення нам потрібно буде використовувати різний підхід, щоб отримати рішення.

Техніка кластеризації для задачі формування команди

Проблема формування команди (TFP) у соціальних мережах широко досліджується багатьма авторами та дослідниками. Для вирішення TFP було запропоновано багато методів, щоб знайти експертів, які могли б охопити всі завдання, необхідні для проекту, і що сформована команда є найбільш спільною.

З цією метою Kalyani у своїй роботі пропонує метод кластеризації для групування набору експертів у формі кластера, щоб точки кластера складалися з експертів, які володіють навичками, необхідними для проекту. Вони прийняли зважений алгоритм SCAN, який є розширеною версією SCAN і: е Structural Clustering Algorithm для вирішення TFP з мінімальними витратами на зв’язок. Комунікаційні витрати – це витрати, пов’язані між двома експертами на основі їхньої минулої співпраці або попереднього досвіду роботи в одному або кількох проектах. Основна ідея їхньої техніки полягає в тому, щоб ідентифікувати кластери, концентратори та викиди в мережі експертів, де кожен вузол представляє експерта, а ваги краю визначають вартість зв'язку. Спочатку вони підходять до проблеми, знайшовши пул експертів, які можуть охопити навички, необхідні для проекту. Потім серед усіх експертів шукають експерта з високою зв’язком і оголошують його основним. Потім кластер розширюється від ядра до сусідніх вузлів і: е від вузлів щільного зв’язку до вузлів із слабким зв’язком. Кластер розширюється до досягнення порогового діапазону вартості зв'язку. Порівняння зваженого сканування виконується з іншим жадібним, генетичним, випадковим і точним алгоритмом і представляє аналіз досягнутих результатів.

Формування команди онлайн у соціальних мережах

Anagnostopoulos та ін. у своїй роботі досліджують онлайн-версію проблеми формування команди. Вони використовують соціальну мережу експертів, яка володіє набором навичок і сумісності один з одним, отриманими з попереднього досвіду роботи. Вони розглядають послідовність завдань, які надходять у режимі онлайн, кожне завдання вимагає певних навичок. Ідея цієї статті полягає в тому, щоб сформувати кілька команд для кількох проектів, кожен проект і набір завдань для проекту з’являються в онлайн-моді. Вони спрямовані на формування команди з урахуванням трьох атрибутів. По-перше, охоплюються навички, необхідні для завершення проекту. По-друге. сформований колектив є спільним і по-третє, працездатність особистості в колективі не перевантажена. Різні евристичні та наближені алгоритми використовуються для вирішення проблеми

TFP із зазначеними вище цілями з подальшим аналізом результатів та висновками.

Проблема найму в кластері

Першими, хто досліджував проблему найму в кластері, були Гольшан та ін. для найму групи експертів, кожен з яких володіє однією або кількома навичками, для набору проектів, де кожен проект складається з набору завдань, що вимагає певних навичок. Це називається проблемою CLUSTER HIRE. Вони вивчають варіанти цієї проблеми, роблячи реалістичні припущення, які впливають на процес найму в кластері. Метою роботи є найняти групу осіб для виконання ряду проектів, де сформована команда максимізує загальний прибуток і знаходиться в межах бюджету, виділеного на найм групи експертів для завершення проекту. Крім того, враховують працездатність кожного індивіда та стежать за тим, щоб робоче навантаження не було перевищено для особи, яка працює в кількох проектах. Вони пропонують два варіанти жадібного алгоритму та використовують реальні набори даних для проведення експериментів та аналізу результатів. Тоді як Гольшан та ін. вивчають проблему кластерного найму з метою максимізації прибутку, Meet et al. першими досліджували проблему CLUSTER HIRE у соцмережі. Метою статті є двох об'єктна, де пропонують найняти групу людей, які найбільше співпрацюють і дають максимальний прибуток від успішного завершення проекту. Вони припускають, що серед експертів існує соціальна мережа, де кожен експерт пов'язаний з іншим через попередній досвід роботи в одному або кількох загальних проектах. Кожен експерт у цьому контексті пов'язаний із зарплатою, тобто вартістю найму та працездатністю. Найм здійснюється з урахуванням того, що сума заробітної плати відібраних експертів не перевищує бюджет, виділений на комплекс проектів, і що жоден спеціаліст у команді не перевантажений. Щоб досягти всіх вищевказаних атрибутів відбору команди, проблема біоб'єктивності потім вирішується шляхом пропонування двох жадібних алгоритмів, тобто стратегії вибору експертів і стратегії вибору проекту. Обидва ці алгоритми

використовують функцію оцінки. Стратегія вибору експерта оцінює кожного експерта в мережі, який може охопити одну або кілька навичок, необхідних для проектів, таким чином, щоб вона охоплювала багато навичок, щоб отримати найбільш прибутковий проект, вартість найму є дешевою, а вартість спілкування експерта найменшою. Використовуючи цей підхід, потім обирають експертів з найвищим балом. Оскільки менш можливо, що дешевий експерт, який охоплює багато навичок, може мати найбільшу співпрацю, вони вводять λ параметр компромісу між прибутком і вартістю зв'язку, який має значення від 0 до 1. Цей компроміс допомагає визначити, чи варто приділяти більше значення прибутку або витратам на комунікацію. Друга стратегія, яка називається стратегією вибору проекту, використовує ту саму стратегію для оцінки проектів, які мають високий прибуток, набір експертів, необхідних для завершення проекту, є дешевими, а набір відібраних експертів здатний добре спілкуватися. Далі вони проводять експерименти з реальним набором даних і представляють ефективність свого підходу в порівнянні з випадковим алгоритмом.

1.6 Постановка проблеми

Нехай $E = \{e1, e2, e3, \ldots . . en\}$ позначаємо множину з n експертів і S =$\{s1, s2, s3, \ldots . . sm\}$ позначають m навичок. Усі символи, використані в цій дипломній роботі, зведені в Таблицю 1. Кожен експерт і володіє набором навичок, і ми використовуємо $CSiu$, чи володіє експерт і навичкою u за $CSiu$. Кожен експерт i має грошові витрати (зарплату) для виконання завдання, пов'язаного з проектом, який йому доручено. Це вимірюється вартістю в доларах, що позначається Ci. У цьому налаштування ми також використовуємо набір проектів $P =$

$\{p, p2, p3, \ldots . . pk\}$. Кожен проект також містить необхідні навички та кількість експертів, необхідних для успішного завершення проекту. Цей набір представлено за допомогою $PSRup$ для проекту p . Крім того, кожен експерт i

може запропонувати свої знання щонайбільше $Capi$ разів (тобто здатність експерта i). Це пов'язано з тим, що наші цілі полягають у тому, щоб надати експертам навички таким чином, щоб вони не були перевантажені, коли призначені для кількох проектів. Таким чином, кожен експерт має максимальну можливість для участі. Ми додаємо це як обмеження до нашої моделі ILP.

Визначення 1: Група експертів

Враховуючи набір з n експертів E, набір з m навичок S і набір з k проектів P, група експертів E може завершити підмножину проектів P, якщо виконується наступне.

1. Експерт e призначається для виконання необхідних навичок s для p.

2. Кількість експертів, призначених для кожної навички, має бути такою ж, як попит r на навичок s для кожного проекту p, коли один і той самий експерт e не може виконувати однакові навички s в одному проекті p двічі

3. Кожному експерту не слід призначати більше ніж $Capi$ навичок

4. Сума Ci серед відібраних має бути меншою за бюджет B

5. Загальна вартість комунікації є найнижчою, тобто вони мають найвищу співпрацю.

Експерти пов'язані один з одним у мережу, яка моделюється у неорієнтованому графі G. Кожен експерт i позначається як вузол графа G. Два експерта в графі з'єднані один з одним за допомогою ваги краю, причому вага вказує на вартість зв'язку між цими двома особами. Вартість комунікацій витягується з попередньої співпраці в минулому (тобто участі в тому самому проекті). У цьому випадку вага краю вказує на міцність співпраці між експертами. Чим менша вартість зв'язку, тобто вага краю, тим міцніша попередня співпраця між експертами. Наприклад, якщо два експерти брали участь у чотирьох попередніх проектах, їхня комунікаційна вартість буде меншою, ніж у експертів, які співпрацювали в попередніх двох проектах. Якщо між експертами не представлено співробітництво, вага найкоротшого шляху між ними в G береться для визначення вартості зв'язку між ними. Подібно до

попереднього розуміння для більшості співробітництв, щонайменша відстань між людьми, максимальна кількість двох експертів може працювати. Тут ми прагнемо вибрати групу експертів, серед яких мінімізовані витрати на спілкування. Приклад графіка соціальної мережі експертів

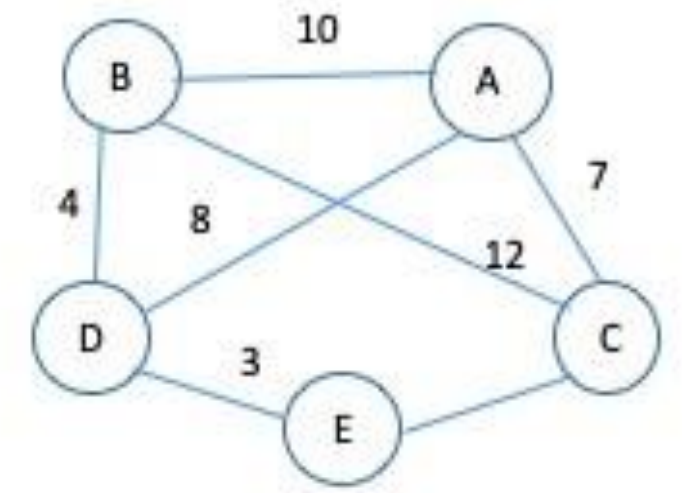


Рисунок 1.4 - Соціальна мережа експертів

Визначення 2: Вартість зв'язку

Вартість зв’язку між двома експертами, i і j, розраховується аналогічним чином, як зазначено Kargar відома як функція суми відстаней, представлена нижче,

Завершення кожного проекту приносить прибуток у доларах, що показано як $Revenue(p)$. У нашій дослідницькій роботі мета полягає в тому, щоб вибрати набір проектів, які максимізують наш прибуток, який отримується шляхом підсумовування доходу від вибраних/охоплених проектів.

Визначення 3: Прибуток від проектів:

Враховуючи набір проектів P, прибуток, отриманий при виконанні цих проектів, визначається так:

де функція доходу $Revenue(p)$ відображає кожен проект цілим числом, що позначає суму в доларах, отриману в результаті завершення проекту.

Для виконання набору проектів нам також надається попередньо визначений бюджет B. Вибираючи набір проектів, які забезпечують максимальний прибуток, ми також стежимо за тим, щоб загальна вартість найму, яка є зарплатою кожного експерта, була в межах бюджету, виділеного на набір проектів.

У нашому дослідженні ми пропонуємо дві моделі ILP, одна з яких повертає групу експертів з максимальним прибутком, а інша вибирає підмножину проектів, які приносять максимальний прибуток і мають мінімальні витрати на комунікацію серед експертів, відібраних для виконання проекту.

Задача 1: Для набору з n експертів E, набору з m навичок S, набору з k проекту P, ми зацікавлені у виборі групи експертів E’ та підмножини проектів у P’, щоб наступна мета була максимальною:

Проблема 2: Подібно до постановки задачі 1, ми пропонуємо вирішити кластерний найм у соціальній мережі, де ми вибираємо групу експертів E’ та підмножину проектів у P’ в P, так що наступна мета була максимальною:

Вищезазначена мета має на меті дві речі: (i) охоплення підгрупи проектів, які дають максимальний прибуток, і (ii) група експертів, відібраних для підготовки над проектами, є спільною та має мінімальні витрати на комунікацію між ними. Тут відстань розраховується між експертами одного проекту, а не між проектами

Теорема 1 Задача 1 є NP-складною

Доказ 1 Знайти групу експертів для охоплення підмножини проектів P та максимізувати

Прибуток(P) за даного бюджету B доведено до NP-складної задачі в Kargar et al. Оскільки мета даної задачі 1 лінійно пов’язана з $Profit(p)$, сучасна задача оптимізації 1 також є NP-складною.

2. АЛГОРИТМІЧНІ ТА МАТЕМАТИЧНІ РІШЕННЯ ДЛЯ ЗАДАЧ ДАНОГО КЛАСУ

2.1Основні позначення, що використовуються в ILP

У цій главі ми пропонуємо дві різні стратегії для пошуку групи експертів, яка приносить максимальний прибуток і має найбільш спільну команду експертів, тобто з мінімальними витратами на комунікацію. Перша стратегія використовує евристичний підхід, який є жадібним і охоплює найбільш прибутковий проект на кожній ітерації.

Таблиця 2.1 Позначення в ILP

E	Множина з n експертів {e1, e2, e3, … . . en}
S	Набір m Навичок {s1, s2, s3, … . . см}
G	Введіть граф G, який моделює соціальну мережу
P	Набір з k проектів {p1, p2, p3, … . . pk}
Distij	Вартість зв’язку між експертом i та експертом j
CSiu	навички експерта
Capi	Здатність експерта запропонувати свій досвід
PSRup	Кількість експертів, необхідних для кваліфікації u в проекті
Wp	двійкова змінна, яка представляє проект p,
Ci	вибрано чи ні
Xi	Вартість найму кожного експерта i
Бюджет	Загальний бюджет найму групи експертів
Виручка	Прибуток кожного проекту c
Viup	$Binary$ Змінна рішення щодо того, чи є експерт i та його навички
Ziujvp	було обрано для проекту c
	Бінарна змінна рішення для вибору двох експертів i та

Тоді як у другому підході ми використовуємо розв'язувачі ILP, які дають точне рішення поставленої задачі та знаходять найкращу можливу групу експертів із доступного вхідного графа вузлів. Далі ми створюємо дві моделі ILP, одна з яких дає максимальний прибуток, тобто охоплює проекти з найвищим прибутком, а інша, яка охоплює прибуткові проекти, а також вибирає експертів, які мають найвищу співпрацю з іншими експертами в проекті.Як було зазначено раніше, наша головна мета цього дослідження полягає в тому, щоб знайти групу експертів, яка охоплює максимальну кількість проектів у рамках даного бюджету, щоб група відібраних експертів мала максимальну співпрацю при розгортанні для роботи з вибраною підгрупою проектів.

У нашому дослідженні ми пропонуємо дві моделі

a) Модель спрямована на вибір групи експертів з метою охоплення проектів, які приносять максимальний прибуток.

b) У цій шкалі вищенаведена модель також обирає експертів на основі їхньої співпраці з іншими учасниками проекту, одночасно охоплюючи проект з високою віддачею прибутку.

У цьому розділі ми окреслимо позначення, використані для формулювання проблеми з метою, обговорюваної в попередньому розділі цієї глави.

2.2 Модель ILP для максимізації прибутку

Спочатку ми представляємо модель ILP, яка повертає групу експертів, які охоплюють підмножину проектів, які приносять максимальний прибуток і не гарантують між експертами хорошу співпрацю. У наведеному нижче описі ми показуємо цільову функцію та набір обмежень. Номер обмеження з підпозначеннями використовується для опису комбінації навичок експерта та проекту.

Цільова функція:

$$maximize \sum_{p=1}^{k}(W_p \times Revenue_p) \quad (2.1)$$

Піддають:

1. Значення двійкової змінної Viup дорівнює 1, якщо воно збігається зі значенням, встановленим у вхідних даних CSiu, в іншому випадку встановлюється значення 0

$$\forall u \in S, \forall i \in E \ \sum_{p=1}^{k} V_{iup} \leq \ CS_{iu} \tag{2.2}$$

2. Для всіх охоплених проектів кількість відібраних експертів не повинна бути меншою за кількість експертів, яких вимагає кожна навичка цього проекту

$$\forall p \in P, \forall u \in S \quad \sum_{i=1}^{n} V_{iup} \geq PSR_{up} \times W_p \tag{2.3}$$

2. Обраному експерту не слід приписувати кількість навичок, що перевищує його працездатність. Ємність надається як вхідні дані для моделі ILP

$$\forall i \in E \quad \sum_{u=1}^{m} \sum_{p=1}^{k} V_{iup} \leq Cap_i \tag{2.4}$$

4. Експерт і призначається до якогось проекту для певних навичок, коли існує проект p і навичка u, до яких він/вона призначений,

$$\forall p \in P, \forall u \in S, \forall i \in E, \quad Xi \geq Viup \tag{2.5}$$

5. Сума вартості Viup для кожного проекту та навичок має бути більшою або дорівнювати значенню

$$Xi^{\forall i \in E} \qquad X_i \leq \sum_{u=1}^{m} \sum_{p=1}^{k} V_{iup} \tag{2.6}$$

6.Сума витрат на найм експертів має бути меншою за вказаний бюджет

$$\sum_{i=1}^{n} C_i \times X_i \leq Budget \tag{2.7}$$

Обґрунтування

Як пояснювалося раніше, метою нашого алгоритму є охоплення найбільш прибуткових проектів при заданому бюджетному обмеженні. Таким чином, цільовою функцією є максимізація прибутку, розрахованого шляхом підсумовування доходу кожного вибраного проекту Wp, значення якого дорівнює 1. Обмеження 2.2 відповідає значенню двійкової змінної Viup, що дорівнює 1, якщо експерту i призначено навик u в проект k, лише якщо експерт володіє навичками u. У обмеженні 2.3 ми впевнені, що для кожного проекту та навичок, пов'язаних із цим проектом, ми наймаємо точно таку ж кількість експертів, яка потрібна для кожної навички, зазначеної у вхідному PSRup. Це також встановлює значення змінної Wp як 1, що показує, чи був охоплений проект, і призначає експертам, які володіють навичками та потребами, таку кількість необхідних експертів. Обмеження 2.4 надається, щоб переконатися, що сума змінної Viup для кожного проекту та навичок для проекту повинна бути меншою або дорівнює працездатності Capifor expert i, яка надається як вхідні дані для нашої моделі. У наших налаштуваннях ми також хочемо переконатися, що вартість найму експерта вираховується з бюджету лише один раз, незалежно від того, скільки навичок і в скількох проектах призначено одному експерту. Щоб це сталося, ми включаємо обмеження 2.5 і 2.6. Обмеження 2.5 гарантує, що сума експертів у Xi повинна бути більшою або дорівнює значенню Viup для кожного проекту, навику та експерта. По суті, ми переконаємося, що значення Xi встановлено на 0 для кожного експерта у Viup, чиє значення дорівнює 0. З іншого боку, значення Xi має бути меншим або дорівнювати сумі кожного проекту та навику. Це гарантує, що значення Xi встановлено на 1, якщо експерт i призначений для будь-якого проекту з будь-яким навиком. У 2.7 ми підсумовуємо зарплату всіх відібраних експертів, щоб переконатися, що вона менша або дорівнює заданому бюджету, який надається як вхідні дані для моделі.

Усі наведені вище обмеження забезпечують врахування наведених нижче атрибутів для вирішення питань найму в кластері

1) Один і той самий експерт не призначається більше одного разу до однієї і тієї ж навики в одному проекті

2) Вартість найму експерта віднімається лише один раз із заданого бюджету, враховуючи, що експерт може виконати до n навичок у m кількості проектів 3) Жодному експерту не призначено навичок більше, ніж його призначені можливості.

2.3 Модель ILP з максимальним прибутком і мінімальними витратами на зв'язок

Наведений нижче ILP пропонується для вирішення проблеми Cluster Hire у соціальній мережі. Ми обираємо роботу за допомогою техніки ILP, оскільки вона проходить через усі можливі вузли розв'язання, щоб повернути найкраще рішення для заданої цільової функції. Тут ми також доводимо масштабованість існуючої моделі, показуючи, як модель ILP для найму кластеру з максимізованим прибутком може бути модифікована, щоб включати функцію витрат на комунікацію, яка забезпечує максимальний прибуток вибраної групи експертів і гарну комунікацію між кожним членом компанії. проект, до якого він/вона призначений

Цільова функція:

$$\sum_{p=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u=1}^{m}\sum_{u\neq v}^{m} \quad (W_p \times Profit_p) \quad -\left(Dist_{ij} \times Y_{iujvp}\right) \tag{2.8}$$

Піддають:

1. Значення двійкової змінної Viup дорівнює 1 лише тоді, коли експерт і володіє навичкою u

$$\forall u \in S, \forall i \in E \ \sum_{p=1}^{k} V_{iup} \leq \ CS_{iu} \tag{2.9}$$

2. Для будь-якого охопленого проекту кількість експертів, відібраних для цього проекту, повинна дорівнювати кількості експертів, необхідних для кожної навики цього проекту

$$\forall p \in P, \forall u \in S \quad \sum_{i=1}^{n} V_{iup} \geq PSR_{up} \times W_p \tag{2.10}$$

2. Відібраному експерту не слід приписувати навички більше, ніж його працездатність. Ємність надається як вхідні дані для моделі ILP

$$\forall i \in E \quad \sum_{u=1}^{m} \sum_{p=1}^{k} V_{iup} \leq Cap_i \tag{2.11}$$

4. Значення Viup має бути менше або дорівнювати значенню Xi

$$\forall p \in P, \forall u \in S, \forall i \in E, \quad Xi \geq Viup \tag{2.12}$$

5. Сума вартості Viup для кожного проекту та навичок має бути більшою або дорівнювати значенню Xi

$$\forall i \in E \qquad X_i \leq \sum_{u=1}^{m} \sum_{p=1}^{k} V_{iup} \tag{2.13}$$

6. Сума витрат на найм експертів має бути меншою за вказаний бюджет

7. Щоб розрахувати вартість зв'язку між двома експертами, ми пишемо наведені нижче обмеження. Оскільки нам потрібно помножити дві змінні Viup і Vjup, що зробить модель нелінійною, ми лінеаризуємо її за допомогою загальної техніки, вводячи третю змінну Ziujvp, як пояснюється нижче:

$\forall p \in P, \forall u \in S, \forall i \in E$

$$Ziujvp \leq Viup \tag{2.15}$$

$$Ziujvp \leq Vjup \tag{2.16}$$

$$Ziujvp \geq Viup + Vjup - 1 \tag{2.17}$$

8. Ми вводимо змінну Yiujvp, щоб переконатися, що вартість зв'язку розраховується лише серед експертів у межах обраного проекту.

$\forall p \in P, \forall u \in S, \forall i \in E$

$$Yiujvp \leq Ziujvp \quad (2.18)$$

$$Yiujvp \leq Wp \quad (2.19)$$

$$Yiujvp \geq Ziujvp + Wp - 1 \quad (2.20)$$

Для цієї моделі ми модифікуємо цільову функцію попередньої моделі ILP таким чином, що поряд із прибутковими проектами вирішувач вибирає експертів, які мають вартість зв’язку оренди. Оскільки це задача максимізації, ми використовуємо стандартний метод, щоб перетворити задачу мінімізації на максимізацію, вводячи обернену до неї. Отже, цільова функція включала прибуток, доповнений мінусом вартості зв’язку між експертами, обраними для цього проекту. Обмеження від 2.9 до 2.14 нагадують ті, що використовувались у попередній моделі ILP. Новий набір обмежень, доданий з 2.15 до 2.20, в основному призначений для розрахунку відстані/вартості зв’язку між експертами. Обмеження від 2.15 до 2.16 є лінеаризаційною формою для Viup* Vjup. Оскільки це рівняння зробило б модель нелінійною, ми використали стандартний спосіб розкладання множення змінних на наведений вище набір рівнянь. Таким чином, значення Ziujvp встановлюється рівним 1, коли існує вартість зв’язку між експертами i та j з навичками u та v відповідно. Крім того, далі ми розраховуємо вартість комунікації між учасниками кожного обраного проекту. Знову ж таки, вихідне рівняння Ziujvp * Wpp було розкладено на лінійні рівняння за допомогою стандартної техніки з введенням змінної Yiujvp

Ця модель для ILP забезпечує додаткову функцію, а саме вартість зв’язку, яка допомагає приймати кращі рішення щодо найму на роботу в кластері в соціальній мережі. Він забезпечує всі вищезгадані атрибути найму в кластері з ще одним атрибутом, щоб вибрати осіб, які раніше співпрацювали з іншими

експертами в проекті, і, отже, виконує завдання проекту у співпраці та співпраці з підвищеною синергією.

2.4 Евристичний алгоритм для задачі найму кластерів

У цьому розділі ми представляємо наш евристичний алгоритм для вирішення проблеми найму кластерів із попитом на потрібні навички проекту, тобто конкретну кількість експертів, необхідних для кожного навику на проект, а також намір найняти групу експертів, які співпрацюють і приносять максимальний прибуток. Алгоритм розроблений таким чином, щоб охопити якомога більше прибуткових проектів у заданому бюджеті. Ми призначаємо оцінку кожному неохопленому проекту на кожній ітерації, щоб вибрати проект, який отримує найвищий бал з надією, що наш алгоритм повертає найбільший прибуток.

Оцінка для пошуку найбільш прибуткового проекту складається з урахуванням таких інтуїцій:

1. Ми хочемо вибрати менш дорогих експертів, щоб вони охоплювали всі навички, необхідні для проекту, і щоб бюджет не був перевитрат на один проект.2. Набір експертів, відібраних для проекту, повинен мати можливість ефективно спілкуватися один з одним

3. Виберіть проект, який приносить найбільший прибуток.

Ми структуруємо функцію оцінки, яка враховує поєднання всіх перерахованих вище інтуїцій. Отже, на кожній ітерації для кожного неохопленого проекту p ми знаходимо набір експертів, щоб охопити необхідні навички для p. Наша мета полягає в тому, щоб вибрати ті підгрупи експертів, які охоплюють усі навички та експертів, необхідних для навичок для кожного проекту, а також, щоб жоден експерт не охоплював одні й ті самі навички більше одного разу для одного проекту, який вимагає від кількох осіб для виконання однієї і тієї ж навички. Таким чином, алгоритм прагне знайти підмножину експертів, які мінімізують витрати на найм і комунікацію. На кожній ітерації ми

знаходимо набір експертів, які можуть охопити проект. На початку ми маємо порожній набір; отже, для навику, який з'являється на першій ітерації проекту, ми знаходимо та призначаємо експерта, який покриває навички з найменшими витратами. Після вибору першого експерта ми вибираємо експертів, які мають як мінімальні витрати, так і мінімальні витрати на спілкування з кожним членом команди, обраної на даний момент в проекті. Потім ми використовуємо нашу функцію підрахунку балів, як зазначено нижче, щоб ранжувати проекти, щоб вибрати проект з найвищим прибутком

$$Profit(p) = Revenue(p) - min\{Ce\{Ce + Revenue(p) - minCe$$

Тут e, e' представляють сусідніх експертів у тому самому проекті. Таким чином, наше рівняння має на меті вибрати підмножину проектів і команду експертів, які приносять максимальний прибуток.

Евристичний алгоритм — це наше рішення проблеми найму кластерів, де ми надаємо певні вхідні дані для нашого алгоритму, як-от графік G з вузлами, що представляють експертів, а також атрибути проекту, як-от прибуток на проект, навички, необхідні для кожного проекту, кількість людей, необхідних для кожного навику. проект.

Алгоритм найму кластерів для максимального прибутку та спільної команди

Вхідні дані: набір з n експертів $E = \{e1, e2, e3 \dots . en\}$, набір s навичок $S = \{s1, s2, s3 \dots . sm\}$ і набір з k проектів $P = \{p1, p2, p3 \dots . pk\}$, графік G, що моделює мережу експертів, потужність експертів $Capi$, вартість експертів Ci. Навички проекту та кількість експертів, які потрібні для роботи над цією навичкою $PSR(up)$

Результат: команда експертів для підгрупи проектів разом із їхніми навичками, максимізований прибуток за заданого бюджету B

У першому рядку ми ініціалізуємо необхідні змінні, такі як мінімальна вартість, бюджет, temp_team, high_profit, які відповідають за зберігання

тимчасової команди експертів, проект з найбільшою прибутковістю з усіх доступних проектів і залишок бюджету. Цикл while у рядку 2 повторюється до тих пір, поки не буде вичерпано заданий бюджет і не залишиться проекти, які потрібно призначити команді. Рядок 3, 4, 5 містить ітерацію кожного проекту, їхні навички та кількість експертів, необхідних для кожного навику для даного проекту. Таким чином, ми прагнемо призначити експерта за навичками і повторити процес, якщо не буде обрано необхідну кількість експертів для навичок. Рядок 6 виконує ітерацію всіх експертів і перевіряє, чи експерт у поточному циклі має потенціал і володіє навичками, необхідними для проекту, у рядку 7. Якщо експерт володіє навичками, необхідними для проекту, вартість найму цього експерта зберігається у змінній exp_cost у рядку 8. Ми перевіряємо, чи немає експертів, вибраних у temp_team для цього проекту. Якщо ні, ми знаходимо та призначаємо експерта з найменшою зарплатою, який пропонує навички, який з'являється першим у проекті в рядку 9. Рядки з 10 по 12 виконується, якщо в команді temp_team є хоча б один експерт. Співпраця між експертом у поточному циклі та кожним експертом у команді додається до вартості найму поточного експерта в циклі. У рядках з 12 по 17 ми знову знаходимо та призначаємо експерта, який має мінімальну вартість найму плюс вартість зв'язку, тобто має найвищу співпрацю з іншими учасниками проекту. Для кожного експерта, призначеного до temp_team для конкретного проекту, ми потім використовуємо нашу функцію оцінки в рядку 19, яка допомагає обчислити temp_profit після того, як усім проектам призначено необхідних експертів, які володіють навичками. У рядку 20 вибирається проект, який повертає максимальний прибуток. temp_team для цього проекту вибирається як остаточна команда для проекту. Усі експерти в остаточній команді потім читаються один за одним, де їхня спроможність та вартість найму коригуються таким чином, щоб експерти, відібрані в остаточну команду, використовувалися першими для інших проектів, доки їх працездатність не буде вичерпана. Це пояснюється з рядків 22 по 24. У рядку 25 видаляється проект, якому були

призначені експерти. Потім ітерації виконуються до тих пір, поки умови рядка 2 не будуть порушені. Остаточна команда разом із залишком бюджету повертається в рядку 26. Алгоритм кластерного найму для максимального прибутку та спільної команди

```
min_cost = MAX_VALUE, budget = 0, temp_team(p) = Ø, min_cost_expert =          1.
0, exp_cost = 0 , highest_profit = 0
While (budget < B and P ≈ Ø) do                                                2.
  While (p € P) do                                                             3.
    While (s € PSR(p)) do                                                      4.
      While (r € PSR(p)) do                                                    5.
    While(e € E)do                                                             6.
      If e holds the skill s belonging to p && Cap(e) > 0 then                 7.
        exp_cost = C(e)                                                        8.
        if ( exp_cost < min_cost and temp_team(p) = Ø) then                    9.
          min_cost_expert = e                                                  10.
          min_cost = C(e)                                                      11.
        else                                                                   12.
          for all e' € temp_team do                                            13.
            exp_cost = exp_cost + Dist(e, e')                                  14.
      if (exp_cost < min_cost)                                                 15.
      min_cost = exp_cost                                                      16.
      min_cost_expert = e                                                      17.
    temp_team(p). add(e, s)                                                    18.
    temp_profit(p) = Revenue(p) – min_cost                                     19.
  if (temp_profit(p) > highest_profit(p))                                      20.
  final_team(p) = temp_team(p)                                                 21.
 for all e in final_team(p) do                                                 22.
   budget = budget − C(e)                                                      23.
   update Cap(e), C(e)                                                         24.
   remove p from P
return final_team, budget
```

Рисунок 2.1 - Евристичний алгоритм для задачі найму кластерів

Зразок спостережень із набору даних

Тепер ми проілюструємо результати, отримані всіма трьома підходами, щоб побачити різницю між процесом прийняття рішень алгоритмами з урахуванням того самого набору зразків вхідних даних.

Навички: {AI, DB, ML, IR}

Експерти: {Expert0, Expert1, Expert2, Expert3}

Соціальна мережа серед експертів:

Експерт 0<-> Експерт 1, Експерт 2

Експерт 1 <-> Експерт 0, Експерт 2

Експерт 2<->Експерт 1, Експерт 3

Експерт 3<->Експерт 2

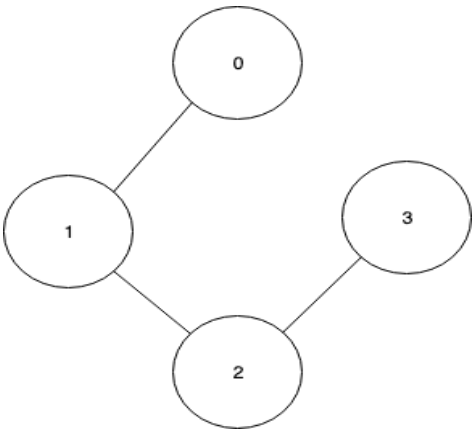


Рисунок 2.2. - Зразок мережі експертів

Зазначений бюджет:

Охоплені проекти: Проект 1, Проект 2, Проект 3

Вибрані експерти: експерт 0, експерт 1, експерт 2, експерт 3

Вихід із моделі ILP з максимальним прибутком

Проект 1: {AI: експерт 0, DB: експерт 0, експерт 2, ML: експерт 1}

Проект 2: {ML: Expert1, Expert 2, IR: Expert 3}

Проект 3: {AI: експерт 0, IR: експерт 3}

Бюджет вичерпано: 0

Вихід із моделі ILP з максимальним прибутком і мінімальними витратами на зв’язок

Проект 1: {AI: експерт 0, DB: експерт 0, експерт 2, ML: експерт 1}

Проект 2: {ML: Expert2, Expert 3, IR: Expert 3}

Проект 3: {AI: експерт 0, IR: експерт 2}

Бюджет вичерпано: 0

Таблиця 2.2 - Експертні атрибути

Ідентифікатор експерта	Ідентифікатор експерта	Ідентифікатор експерта	Ідентифікатор експерта
Експерт 0	AI, DB	150	3
Експерт 1	AI, ML	120	2
Експерт 2	DB, AI, ML, IR	120	3
Експерт 3	ML, IR	110	4

Вихід евристичного алгоритму

Проект 2: {ML: Expert1, Expert 2, IR: Expert 2}

Проект 3: {AI: експерт 1, IR: експерт 2}

Бюджет вичерпано: 260

Таблиця 2.3 - Атрибути проекту

Ідентифікатор проекту	Попит на навики проекту	Прибуток проекту
Проект 1	AI: 1 DB: 2 ML:1	150
Проект 2	ML: 2 IR: 1	120
Проект 3	IR:1AI:1	150

Усі три алгоритми дають результати відповідно до логіки їхнього проектування. ILP з максимальним прибутком гарантує, що спочатку вибирають експертів з меншою зарплатою та більшими можливостями, коли їхні навички збігаються з навичками, необхідними для проекту.

Найцікавіше - вивчити, як ILP з максимізованим прибутком і вартістю зв'язку та евристичний алгоритм поводяться при спробі розв'язати поставлену проблему. Евристичний алгоритм спроектований, він вибирає експерта з мінімальною вартістю найму та мінімальною вартістю співпраці, і намагається спочатку вичерпати потенціал експерта, коли знайдеться відповідність між навичками проекту та експертними знаннями експерта. У той час як ILP переглядає всі доступні вузли рішень, щоб знайти експертів, які мають найменшу зарплату, більше можливостей і мають найбільший набір навичок, необхідних для груп проектів, забезпечуючи найбільш спільну команду та охоплюючи проекти з високим рівнем доходу. Ми спостерігаємо, що експерт 2 вичерпує свої навички за допомогою евристичного алгоритму. Отже, хоча він має БД навичок, він не може брати участь, щоб покрити навички та попит проекту 1, покриваючи решту проекту з більшим доходом, тобто проект 2 і проект 3

3. ВЛАСНЕ ОРИГІНАЛЬНЕ АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ РІШЕННЯ ЗАДАЧІ ФОРМУВАННЯ КОМАНД РОЗРОБНИКІВ

3.1 Формалізація моделі прибутку від кластерного найму команд розробників

У цьому розділі ми оцінюємо продуктивність запропонованих нами алгоритмів, щоб знайти найкращу команду експертів для вирішення проблеми найму кластерів у мережі експертів. Формулювання ILP, оцінене в цьому дослідженні, дає кращі результати, які є відносно близькими до оптимальних, порівняно з евристичними рішеннями. Основна мета цієї дипломної роботи полягає в тому, щоб вивчити та порівняти вплив формулювання ILP при застосуванні на проблему найму в кластері, яка є NP-складною за своєю природою.

Ми сформулювали дві моделі ILP, як описано, щоб створити рішення для пошуку групи експертів, які задовольняють усі надані обмеження. Перша модель створює групу експертів, які охоплюють проекти, які приносять максимальний прибуток. Друга модель ILP створює найбільш спільну команду разом із максимальним прибутком, який формується шляхом додавання додаткових обмежень і цільової функції, що використовуються для розрахунку вартості комунікації. Це дає моделі ILP простір для гнучкості та масштабованості проблемної області.

Це означає, що для кожної групи експертів оцінюються моделі ILP1. Максимальний прибуток, який повертає група експертів

2. Відібрана група експертів потрапляє в можливий регіон. Тобто задовольнити всі обмеження, необхідні для вирішення проблеми.

Мета моделі полягає в тому, щоб загальний прибуток був максимальним, а сформована команда – максимально спільною.

Ми використали Gurobi Optimizer, математичний вирішувач, написаний на Python, щоб сформувати модель для відбору групи людей, які можуть допомогти завершити підмножину проектів із заданим бюджетом. Gurobi Optimizer — це комерційний розв'язувач оптимізації, який використовується для вирішення завдань лінійної оптимізації, відомих як лінійне програмування (LP), включаючи цілочисельне лінійне програмування (ILP). Guorbi Optimizer надає гнучкий, високопродуктивний розв'язувач математичного програмування для задач лінійного програмування (LP), задач квадратичного програмування (QP), програмування з квадратичними обмеженнями (QCP) та задач змішаного цілочисельного програмування (MIP).

Для аналізу результатів ми також розробили евристичний алгоритм, описаний у розділі 3, для порівняння з рішенням ILP. Нижче наведено вхідні дані як для моделі ILP, так і для нашого евристичного алгоритму.

1. Файл, що містить мережу експертів із вагами краю, що представляють співавторство двох експертів

2. Файл, що містить інформацію про вартість та потужність експерта, а також перелік навичок, у яких він має досвід

3. Файл, що складається з набору проектів. Кожен проект складається з

а) дохід від проекту

б) Навички, необхідні для проекту

в) Кількість експертів, необхідних для виконання проекту

Ми формуємо наші моделі ILP для задачі найму в кластері з цільовою функцією та обмеженнями – щоб відповідати всім критеріям, необхідним для формування групи експертів. Виконання вирішувача ILP для нашої моделі виконується наступними кроками.

1. Читає проект, навички, необхідні для проекту, і кількість експертів, необхідних для кожної навички

2. Формат обмежень для нашої моделі ILP, щоб вибрані експерти відповідали меті формування команди. Ці обмеження враховують експертний

потенціал, вартість експерта, навички експерта, вартість зв’язку між експертами з вхідних файлів і зберігаються у файлі типу lp.

3. Викликає вирішувач Gurobi. Розв’язувач читає файл .Lp і намагається знайти, якщо можливо, оптимальне рішення проблеми.

4. Розв’язувач ILP спочатку знаходить найкраще рішення для даної задачі, не враховуючи обмежень. Це допомагає визначити верхню межу для даної задачі максимізації. Наприклад, розв’язувач ILP спочатку розглядає найменшу вартість комунікацій та максимальну величину прибутку, яку можна отримати, вибравши групу експертів, які найбільше співпрацюють, яка охоплює весь набір проектів. /навички.

5. Потім розв’язувач знаходить найкраще рішення, яке дає значення, близьке до верхньої межі, і виконує всі умови, сформульовані для вирішення проблеми. У цьому сценарії розв’язувач ILP на кожному кроці формує групи експертів, які задовольняють весь набір необхідних обмежень. для нашої задачі з наймом кластерів і перевіряє, чи повернене значення менше або дорівнює значенню, отриманому на кроці 4, щоб повернути групу експертів і проектів.

Модель ILP, сформульована для вирішення проблеми найму в кластері, перевірена на основі різноманітного розміру наборів даних, що генерує п’ять двійкових змінних, кожна з яких складається з різної комбінації кількості експертів k, списку навичок s та проектів p.

Розмір змінної рішення, що позначає, які експерти відбираються для якого навику якого проекту (Viup), дорівнює k*m*n.

Аналогічно, кількість змінних рішення для розрахунку співпраці між експертами одного проекту (Ziujvp) дорівнює $k*m2 * n2$

Наприклад, припустимо, що m=8 експертів, n=3 навичок і k – 2 проекти. Тоді нижче буде розмір вищеописаних змінних

Експертна навичка, обрана для проекту (Viup), буде 8 * 3 * 2 = 48

Expert-Skill-Expert-Skill для проекту, щоб отримати вартість зв'язку (Ziujvp) буде 8 * 3* 8 * 3 * 2 = 4608

Отже, розмір змінних збільшується зі збільшенням розміру набору даних, створюючи обмеження пам’яті для завантаження моделі в пам’ять і вирішення проблеми. Таким чином, для нашого експерименту ми запускаємо модель ILP оренди кластера з максимальним прибутком на 10 тис. вузлів, тоді як ми обмежуємо модель ILP з максимальною прибутком і мінімальними витратами на зв’язок до 50 вузлів експертів.

Нижче наведено моделі ILP з їх цілями, скільки вузлів експертів використовуються в цьому дослідженні для експериментів, а також кількість змінних, необхідних для моделі для різної колекції проектів.

Таблиця 3.1 - Розмір змінної

Модель ILP	Кількість вузлів	№	Кількість змінних	Розмір файлу lp
Модель ILP для кластерного найму з максимальним	10 000 вузлів	Навички	10 проектів-	10 Проектів-0,69
прибуток	50 вузлів	26	5 900 000 08 Проекти- 27 050 400	ГБ 05 Проекти - 4,31 Гб 08 Проекти- 7,38 Гб

Щоб отримати набір даних, ми спочатку створюємо вхідний графік (тобто мережу експертів) з DBLP. Набір даних складається з інформації про набір робіт та їх авторів. Він складається з інформації про 10 000 експертів. Краї представляють інформацію про співпрацю або співавторство між дослідниками. Другий набір даних являє собою випадкову мережу з 50 вузлів із вагами краю, застосованими для представлення співпраці, подібної до DBLP, але з випадковими числами. Зауважте, що для наших експериментів ваги країв (тут співпраця), витрати на експерта, потенціал експерта, прибуток від проекту, потреба в навичках для експертів нормалізуються, щоб мати однакову шкалу. Ми випадковим чином створюємо колекцію проектів, кожна з яких складається з 5, 8, 10 і 20 проектів, використовуючи наведені нижче шкали для двох наших

мереж. У наших експериментах ми використовуємо різні значення для бюджету, щоб побачити загальний прибуток, отриманий від кожної методики.

Генерація синтетичних даних і використовувані шкали.

Таблиця 3.2 - Використана шкала значень

Вартість експерта з наборів даних	Вартість експерта з наборів даних	Вартість експерта з наборів даних	Вартість експерта з наборів даних	Вартість експерта з наборів даних
10 000 Вузли	100 to 700	5 to 8	100 to 200	4 to 9
50 вузлів	100 to 200	1 to 3	50 to 100	1 to 3

Ми вивчаємо результати, досягнуті з використанням як ILP, так і евристичного підходів. Обидва методи мають власний підхід до відбору групи експертів, які прагнуть завершити заданий набір проектів у межах необхідного бюджету. Ми порівнюємо результати ILP, реалізовані на python та евристичний алгоритм, реалізований на Java на комп'ютері Intel Core i7 3,3 ГГц з 32 ГБ оперативної пам'яті.

3.2 Евристична модель прибутку від кластерного найму на 10 000 вузлів

Загальний прибуток проти бюджету.

У цьому розділі ми представляємо результати експерименту. Ми досліджували, наскільки ефективно модель ILP приносить прибуток порівняно з евристичним алгоритмом. Оцінюємо вплив бюджету на загальний прибуток проектів. На малюнку нижче наведені значення загального прибутку для збільшення значень бюджету для різної кількості проектів. Тут для порівняння з евристичним алгоритмом ми розглянули k проектів, де k= {10,20}. Потім розраховуємо загальний прибуток для різних значень бюджету. Результати показують, що наша модель ILP досягла більшого прибутку в порівнянні з евристичним алгоритмом, коли у нас обмежений бюджет.

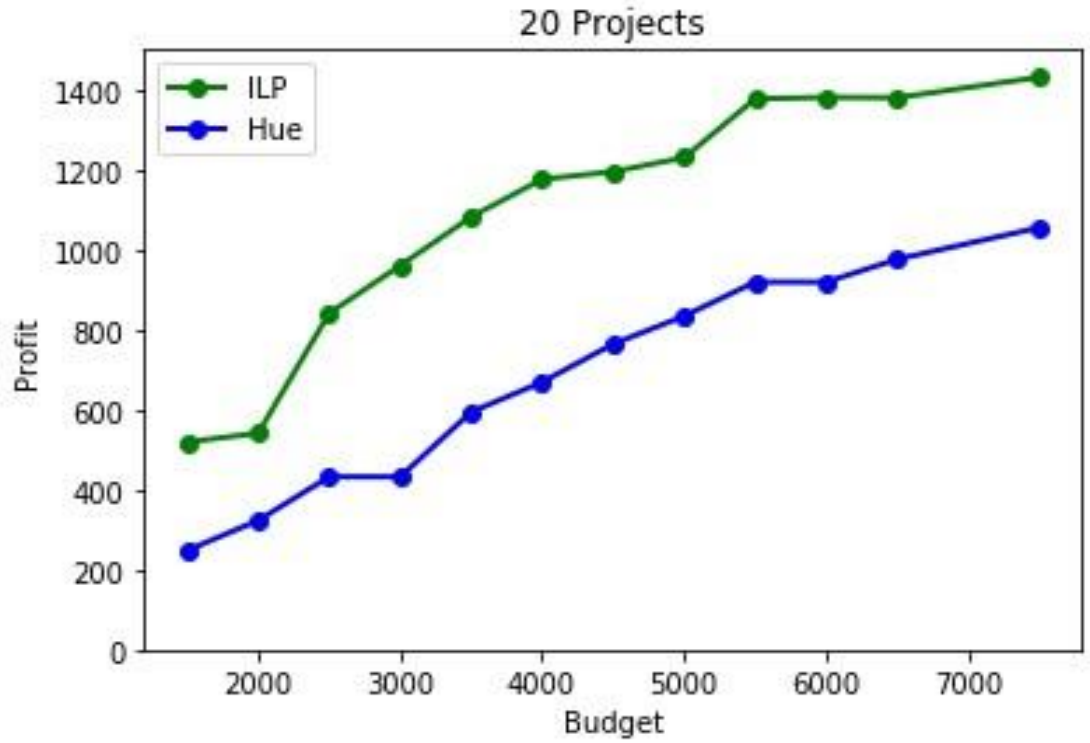


Рисунок 3.1 - Порівняння бюджету та прибутку з розміром проекту 20

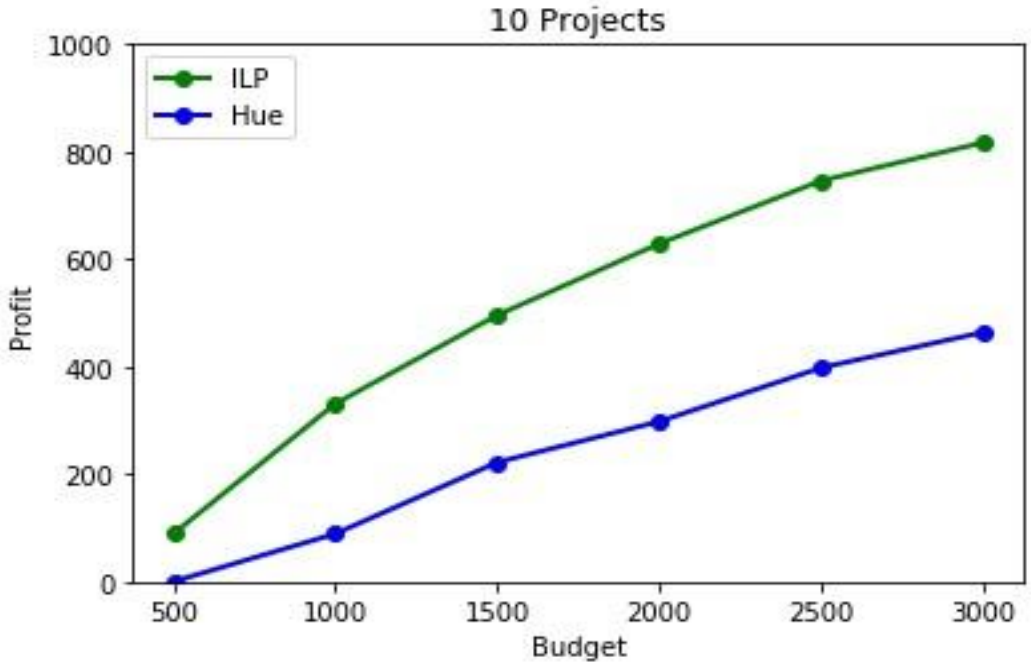


Рисунок 3.2 - Порівняння бюджету та прибутку з розміром проекту 10

Кількість завершених проектів у порівнянні з бюджетом

На малюнку нижче показано кількість завершених проектів при збільшенні вартості бюджету для іншої кількості проектів. Результати показують, що ефективність ILP перевершує ефективність евристичного рішення, навіть якщо у нас обмежений бюджет.

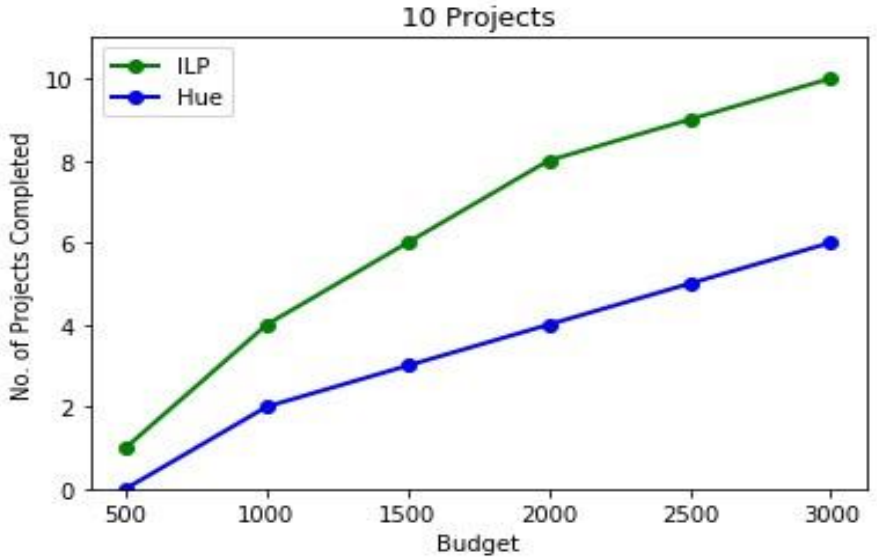


Рисунок 3.3 - Кількість завершених проектів у порівнянні з бюджетом (10 проектів)

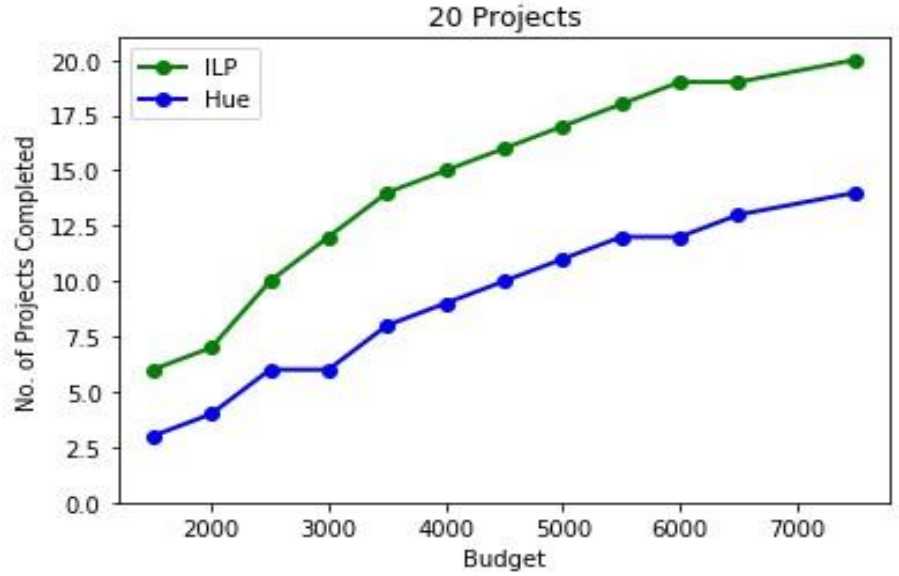


Рисунок 3.4 - Кількість завершених проектів у порівнянні з бюджетом (20 проектів)

3.3 Евристична модель прибутку від найму в кластері та моделі вартості зв’язку на 50 вузлів

Загальна вартість зв'язку проти бюджету

Тепер ми вивчаємо вплив бюджету на вартість комунікації з порівнянням між моделлю ILP, яка дає максимальний прибуток з найбільш спільною командою, та евристичним алгоритмом, що має ту саму мету. Ми експериментували на 50 вузлах експертів і двох наборах проектів с. Кожен набір проекту, як обговорювалося раніше, містить інформацію про те, які навички необхідні для розробки проекту та кількість персоналу, необхідного для успішного завершення проекту. З наведеного нижче графіка ми спостерігаємо, що спочатку і евристична, і ILP-модель призводять до майже однакової суми витрат на комунікацію, але в міру збільшення бюджету ILP, як правило, створює найбільш спільну команду, тоді як евристична модель постійно збільшує суму витрат на комунікацію між експертів, відібраних для кожного завершеного проекту. ILP забезпечує мінімальні витрати на зв’язок і, отже, створює найбільш спільну команду.

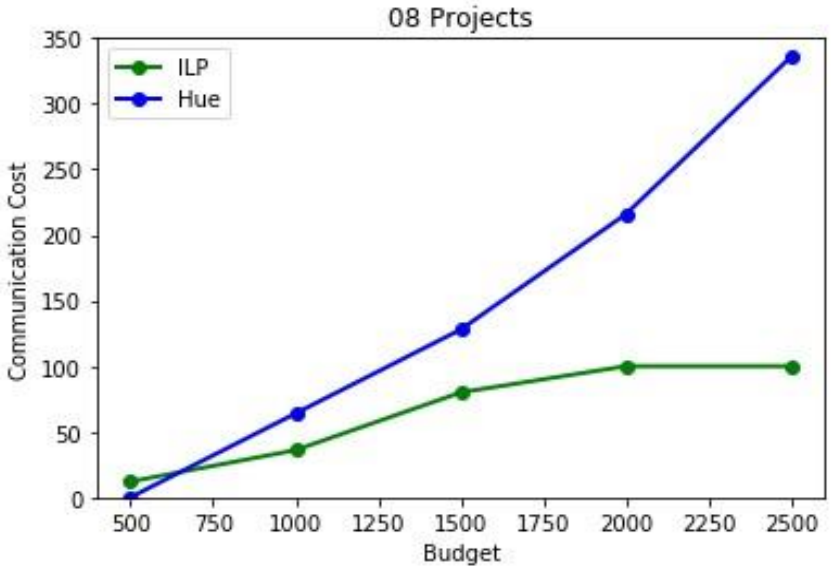


Рисунок 3.5 - Вартість зв'язку проти бюджету (08 проектів)

Загальний прибуток проти бюджету

Як і вище, тепер ми порівнюємо максимальний прибуток, отриманий ILP, з евристичним алгоритмом, щоб побачити вплив збільшення бюджету на загальний отриманий прибуток. Ми помічаємо, що евристичний алгоритм дає результати, близькі до оптимальних, отриманих ILP.

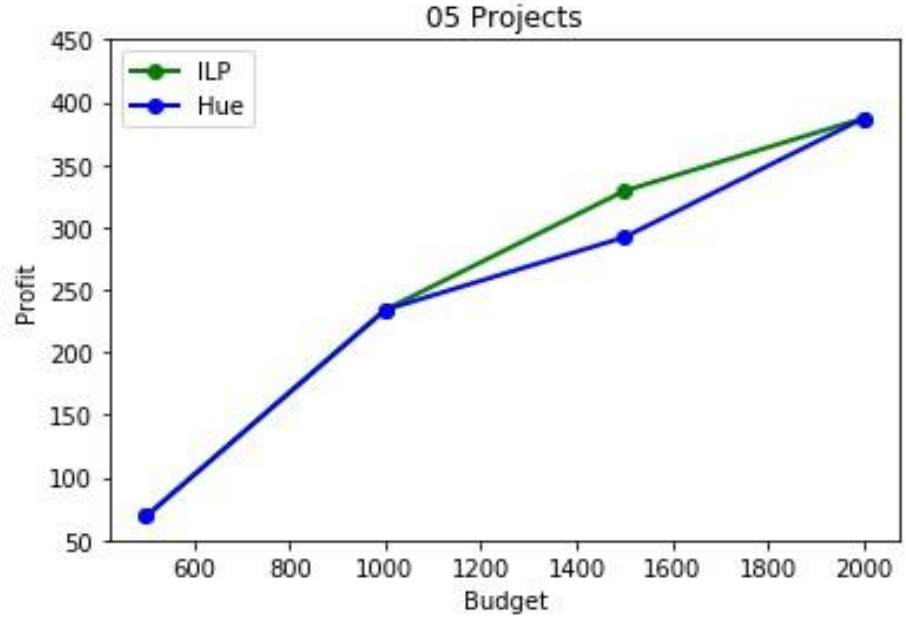


Рисунок 3.6 - Прибуток проти бюджету (05 проектів)

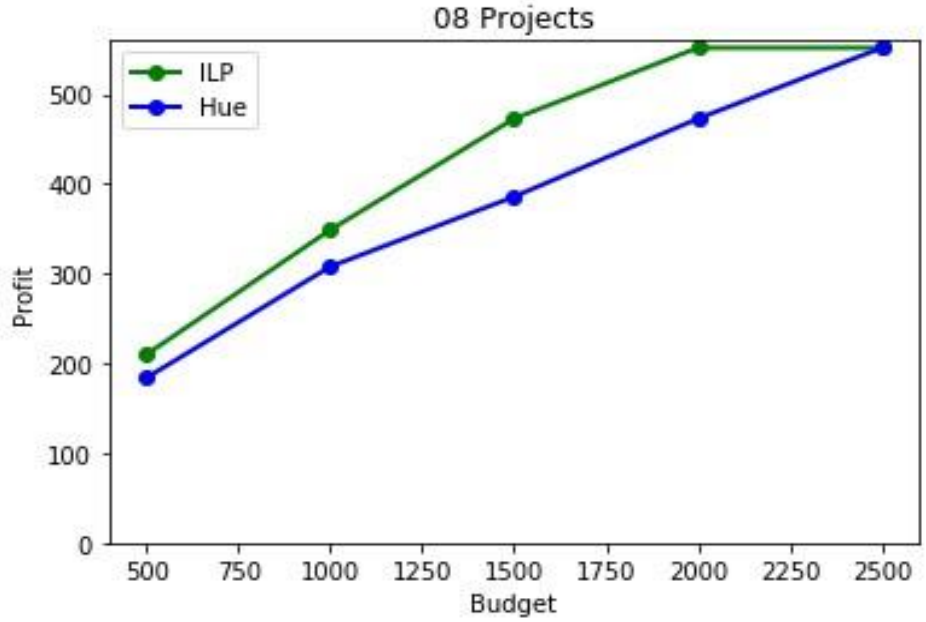


Рисунок 3.7 - Прибуток проти бюджету (08 проектів)

3.3 Опис результатів тестування та ефективність

У цьому розділі ми обговоримо ефективність запропонованих нами алгоритмів для вирішення проблеми найму кластерів з метою вибрати групу експертів, які принесуть мінімальний прибуток і мінімальні витрати на комунікацію, якщо є співпраця між учасниками проекту. Ми представляємо дві моделі ILP для вирішення проблеми наймання кластерів, кожна з яких має свою мету. Ми провели різні експерименти та порівняли результати з евристичним алгоритмом, щоб оцінити ефективність нашого підходу.

Використовуючи набір даних DBLP, ми порівнюємо модель ILP з метою максимізації прибутку з евристичним алгоритмом. Ми проводимо експерименти, які порівнюють загальний прибуток з бюджетом і кількість завершених проектів з бюджетом. Ми виконуємо порівняння загального прибутку та бюджету з двома наборами проектів розміром 10 та 20 відповідно. Результати показують, що вирішення проблеми найму кластерів шляхом формулювання її у вигляді моделі ILP дає кращі результати порівняно з евристичною технікою. Однією з основних причин є те, як працює вирішувач ILP. У той час як евристичне рішення веде себе так, як воно розроблено, тобто підбирає експерта з найменшою зарплатою та витратами на комунікацію для проекту, розв’язувач ILP проходить через різні комбінації групи експертів, що повертає найбільш прибуткову підмножину проектів. Оскільки вирішувач ILP охоплює всі найкращі можливі комбінації з доступних експертів, навичок і проектів, він займає більше часу на виконання в порівнянні з евристичним алгоритмом. Наприклад, евристичний рішення займає час виконання 11,45 секунди, тоді як розв’язувач ILP займає приблизно 6 хвилин, щоб повернути групу експертів для вибраного проекту, обмеженого заданим бюджетом. Подібна тенденція спостерігається, коли однакові експерименти проводять різною кількістю проектів. Розглядаючи експеримент, який порівнює кількість завершених проектів з бюджетом, ми помічаємо, що техніка ILP охоплює більше проектів, ніж евристичний алгоритм.

Після обговорення експериментів з набором даних DBLP ми тепер обговорюємо експерименти, проведені з використанням Synthetic Dataset, де ми порівнюємо модель ILP, яка має на меті максимізувати як прибуток, так і співпрацю між учасниками проекту за допомогою того ж евристичного рішення, яке ми використовували вище. Спочатку ми порівнюємо загальну вартість зв'язку з бюджетом. Зауважте, чим вище вартість комунікації, тим менше співпраця між учасниками проекту. Знову ж таки, тут ми помічаємо, що і ILP, і евристичні рішення створюють хороші спільні команди, коли є невелика кількість проектів, але коли ми збільшуємо розмір проекту, ми можемо помітити помітну різницю. Підхід ILP створює більш спільну команду на відміну від евристичного підходу. Ми помічаємо, що вартість зв'язку постійно зростає зі збільшенням бюджету. Після цього експерименту ми порівнюємо загальний прибуток із бюджетом. Ми помічаємо, що для невеликої кількості проектів значення загального прибутку майже таке ж, яке повертає розв'язувач ILP. Коли кількість проекту збільшується, розв'язувач ILP перевершує евристичний алгоритм

З усіх проведених вище експериментів ми виявили, що моделі ILP для найму в кластері, як правило, дають кращі результати. Однак він має два обмеження. По-перше, розмір змінних збільшується зі збільшенням розміру набору даних. Загалом, якщо ми збільшимо кількість навичок, необхідних для проектів, розмір змінних, які складаються з експертних навичок для кожного проекту, різко збільшується, оскільки враховуються всі можливі комбінації кожного експерта з цією навичкою. Оскільки змінні створеної нами моделі ILP спочатку завантажуються в пам'ять, із збільшенням розміру даних нам потрібно більше пам'яті. Таким чином, ми пропонуємо різні розміри задач, щоб перевірити максимальний розмір введення, який можна надати, враховуючи обмеження пам'яті, як показано нижче на рисунку.

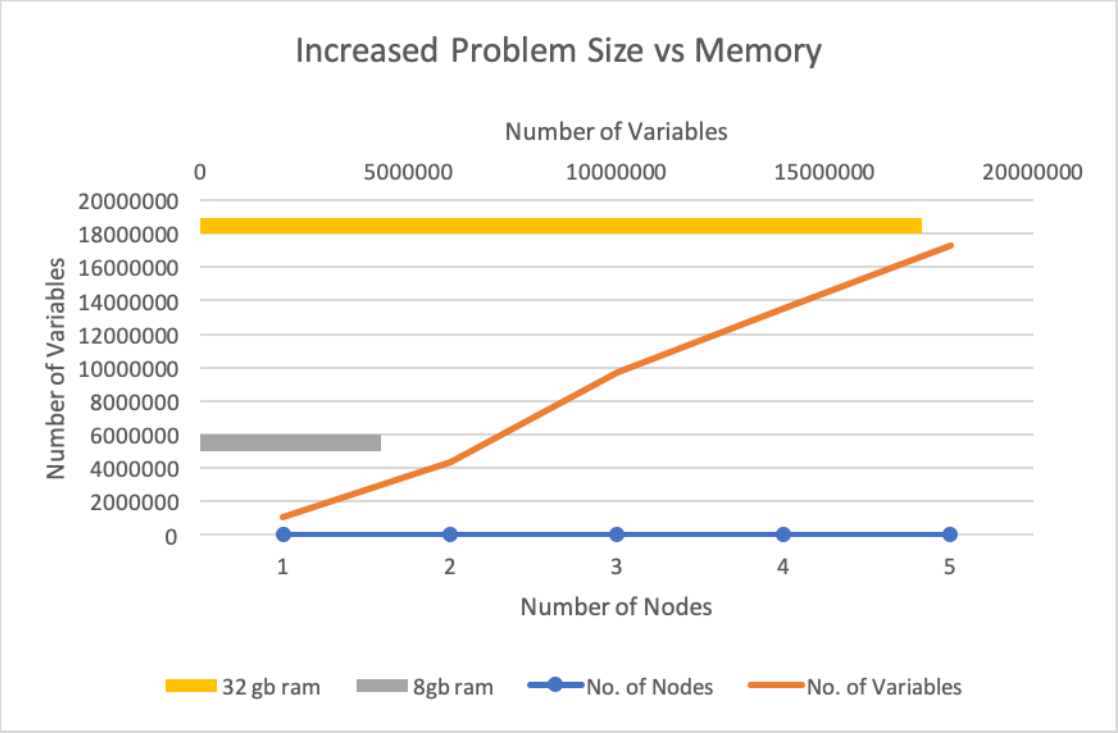


Рисунок 3.8 - Розмір проблеми проти розміру пам'яті

Ми помічаємо, що машина пам’яті 8 Гб може вирішити проблему, наприклад, зі змінним розміром. Аналогічно, з 32 ГБ оперативної пам’яті максимальний вхідний розмір може становити 50 вузлів з 26 навичками та 8 проектами відповідно. Перевищення вхідного розміру, зазначеного вище, викликає помилку «Недостатньо пам’яті».

Другим обмеженням є час роботи підходу ILP. Проблема найму кластерів є проблемою NP-Hard, яка передбачає, що будь-який оптимальний алгоритм, включаючи алгоритми ILP, ймовірно, займе багато часу, оскільки розмір проблеми стає великим. Щоб оцінити масштабованість підходу ILP для сформульованої задачі найму в кластері, ми експериментували з проблемами різних розмірів. Результати виконання показані на малюнку.

Як ми бачимо з малюнка, час обчислення досить короткий, коли розмір задачі невеликий. Перші експерименти закінчувалися від кількох секунд до хвилини. Однак, коли розмір проблеми збільшується, час виконання швидко збільшується. Останній експеримент проводився на моделі ILP з максимізацією та мінімізацією прибутку вартість комунікації для вхідного розміру 50 експертів,

26 навичок і 8 проектів. Щоб повернути групу експертів, знадобилося приблизно 360 секунд. Загалом, з тим, як ми налаштовуємо наші моделі для найму в кластері, підхід ILP виглядає сприятливим для невеликих розмірів проблем, які вимагають меншої кількості змінних.

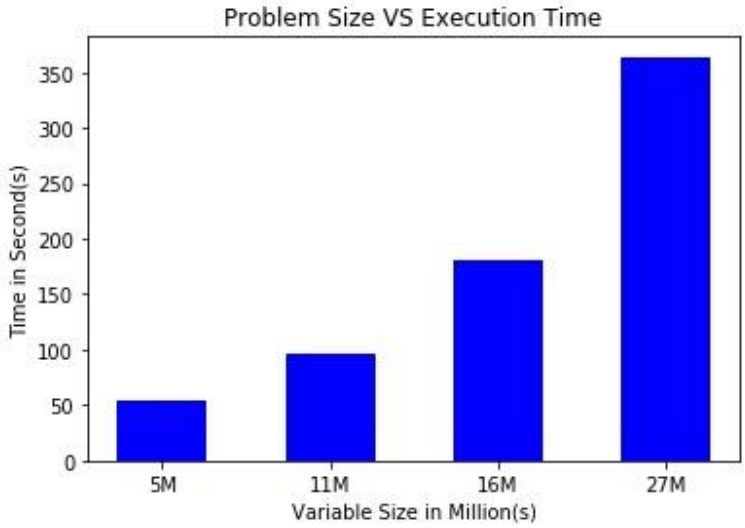


Рисунок 3.9 - Розмір проблеми проти часу виконання

Метою цього дослідження було вивчити придатність використання формулювань та алгоритмів ILP для вирішення проблеми найму кластерів із різними цілями. Як пояснюється, такі проблеми можуть бути дуже складними, оскільки вони пов’язані з багатьма складними обмеженнями. Розробити хороші евристичні алгоритми для вирішення таких задач непросто. Відомі евристичні алгоритми, як правило, працюють над простішими версіями задач і навіть у цих випадках часто не знаходять можливих рішень та/або не досягають з точки зору оптимізації цілі.

З цього дослідження ми можемо зробити висновок, що для проблем до певного розміру підхід ILP цілком застосовний. За допомогою ILP можна точно описати проблему, а потім вдатися до алгоритмів ILP, щоб знайти не тільки здійсненні, але й оптимальні рішення. Для випадків використання, коли проблеми вирішуються за допомогою ILP, немає потреби шукати евристичні алгоритми. Евристичні алгоритми стають корисними лише тоді, коли варіанти використання породжують великі проблеми.

Розмір проблеми проти часу виконання

Метою цього дослідження було вивчити придатність використання формулювань та алгоритмів ILP для вирішення проблеми найму кластерів із різними цілями. Як пояснюється, такі проблеми можуть бути дуже складними, оскільки вони пов’язані з багатьма складними обмеженнями. Розробити хороші евристичні алгоритми для вирішення таких задач непросто. Відомі евристичні алгоритми, як правило, працюють над простішими версіями задач і навіть у цих випадках часто не знаходять можливих рішень та/або не досягають з точки зору оптимізації цілі.

З цього дослідження ми можемо зробити висновок, що для проблем до певного розміру підхід ILP цілком застосовний. За допомогою ILP можна точно описати проблему, а потім вдатися до алгоритмів ILP, щоб знайти не тільки здійсненні, але й оптимальні рішення. Для випадків використання, коли проблеми вирішуються за допомогою ILP, немає потреби шукати евристичні алгоритми.

ВИСНОВКИ

В дипломній роботі основна увага приділена ролі цілочисельного лінійного програмування у вирішенні проблеми найму кластерів. Останнім часом було проведено багато досліджень для вирішення проблеми найму кластерів, коли ми хочемо найняти групу експертів для виконання певної групи проектів у межах даного бюджету. Як свідчать експериментальні результати, розв'язання задачі наймання кластерів із застосуванням підходу математичного програмування має значний вплив на пошук оптимальної/близько до оптимальної групи експертів і є значно ефективнішим порівняно із запропонованими раніше евристичними рішеннями.

Крім того, ми використали більш практичну постановку проблеми найму в кластері, отриману на основі більш реалістичного сценарію найму що може бути застосована майже до всіх задач, де потрібен найм людей, які мають досвід у різних доменах. Ми розглядаємо групу експертів, які мають бути найняті для певної групи проектів, кожен з яких має свої *навички*, і кожна навичка має свою вимогу, яка запитує, скільки експертів потрібно найняти для кожного проекту.

Для порівняння з іншими рішеннями ми також реалізували *евристичні* рішення, які допомагають розрізняти якість результатів, отриманих із застосуванням обох підходів. Наші головні цілі тут — *максимізувати* прибуток, отриманий від доходу від охоплених проектів, і *мінімізувати* витрати на комунікацію між *експертами* для формування спільної команди. Наші результати з моделей дають висновок, що він приносить максимальний прибуток і об'єднує експертів, які мають вищий рівень співпраці в порівнянні з *евристичним рішенням*. Рішення займає значний обсяг пам'яті для створення *змінних і обмежень*. Крім того, оскільки він проходить через усі можливі цільові функції, для повернення рішення потрібно більше часу, ніж при евристичному підході.

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Bazerman, M. (2006). Climate change as a predictable surprise. Climate Change, 77, 170-193.
2. Beersma, B., Hollenbeck, J. R., Humphrey, S. E., Moon, H., Conlon, D. E., & Ilgen, D. R.(2003). Cooperation, competition, and team performance: Toward a contingency approach.Academy of Management Journal, 46, 572-590.Woolley et al. / Bringing in the Experts 369
3. Bottger, P. C., & Yetton, P. W. (1988). An integration of process and decision scheme explanations of group problem solving performance. Organizational Behavior and Human Decision Processes, 42, 234-249.
4. Bunderson, J. S. (2003). Recognizing and utilizing expertise in work groups: A status characteristics perspective. Administrative Science Quarterly, 48, 557-591.
5. Bunderson, J. S., & Sutcliffe, K. M. (2002). Comparing alternative conceptualizations of functional diversity in management teams: Process and performance effects. Academy of Management Journal, 45, 875-893.
6. Cabeza, R., & Nyberg, L. (2000). Imaging cognition II: An empirical review of 275 PET and FMRI studies. Journal of Cognitive Neuroscience, 12, 1-47.
7. Caruso, H. M., & Woolley, A. W. (in press). Manifesting the value of cognitive diversity in teams: The critical role of emergent interdependence. In M. A. Neale, E. Mannix, & K. Phillips (Eds.), Research on managing groups and teams: Groups & diversity (Vol. 9) Oxford, UK:Elsevier Science Press.
8. Chugh, D., & Bazerman, M. H. (2007). Bounded awareness: What you fail to see can hurt you.Mind and Society, 6, 1-18.
9. Cronin, M. A., & Weingart, L. R. (2007). Representational gaps, information processing, and
10. conflict in functionally diverse teams. Academy of Management Review, 32, 761-773.

11. Dahlin, K. B., Weingart, L. R., & Hinds, P. J. (2005). Team diversity and information use. Academy of Management Journal, 48, 1107-1123.
12. Davis, J. H. (1996). Group decision making and quantitative judgments: A consensus model.In E. Witte & J. H. Davis (Eds.), Understanding group behavior: Consensual action by small groups (Vol. 1, pp. 35-59). Mahwah, NJ: Lawrence Erlbaum.
13. Devine, D. J. (1999). Effects of cognitive ability, task knowledge, information sharing, and conflict on group decision-making effectiveness. Small Group Research, 30, 608-634.
14. Duchaine, B., & Nakayama, K. (2006). The Cambridge Face Memory Test: Results for neurologically intact individuals and an investigation of its validity using inverted face stimuli and prosopagnosic participants. Neuropsychologia, 44, 576-585.
15. Ellis, A. P. J., Hollenbeck, J. R., Ilgen, D. R., Porter, C. O., West, B. J., & Moon, H. (2003). Team learning: Collectively connecting the dots. Journal of Applied Psychology, 88, 821-835.
16. Ericsson, K. A. (2005). Recent advances in expertise research: A commentary on the contributions to the special issue. Applied Cognitive Psychology, 19, 233-241.
17. Ericsson, K. A., & Lehmann, A. C. (1996). Expert and exceptional performance: Evidence of maximal adaptation to task constraints. Annual Review of Psychology, 47, 273-305.
18. Faraj, S., & Sproull, L. (2000). Coordinating expertise in software development teams. Management Science, 46, 1554-1568.
19. Fiore, S. M., & Schooler, J. W. (2004). Process mapping and shared cognition: Teamwork and the development of shared problem models. In E. Salas & S. M. Fiore (Eds.), Team cognition: Understanding the factors that drive process and performance (pp. 133-152). Washington, DC: American Psychological Association.

20. Gurtner, A., Tschan, F., Semmer, N. K., & Nagele, C. (2007). Getting groups to develop good
21. strategies: Effects of reflexivity interventions on team process, team performance, and shared mental models. Organizational Behavior and Human Decision Processes, 102, 127-142.
22. Hackman, J. R., Brousseau, K. R., & Weiss, J. A. (1976). The interaction of task design and
23. group performance strategies in determining group effectiveness. Organizational Behavior and Human Decision Processes, 16, 350-365. 370 Small Group Research
24. Hackman, J. R., & Morris, C. G. (1983). Group tasks, group interaction process, and group performance effectiveness. In H. H. Blumberg, A. P. Hare, V. Kent, & M. Davies (Eds.), Small groups and social interaction (Vol. 1, pp. 331-345. Chichester, UK: Wiley.
25. Hackman, J. R., & O'Connor, M. (2004). What makes for a great analytic team? Individual vs. Team approaches to intelligence analysis. Washington, DC: Intelligence Science Board,Office of the Director of Central Intelligence.
26. Hackman, J. R., & Wageman, R. (2005). A theory of team coaching. Academy of Management Review, 30, 269-287.
27. Henry, R. A. (1995). Improving group judgment accuracy: Information sharing and determining the best member. Organizational Behavior and Human Decision Processes, 62, 190-197.
28. Heuer, R. J., Jr. (1999). Psychology of intelligence analysis. Washington, DC: Government Printing Office.
29. Howard, M. W., & Kahana, M. J. (2002). When does semantic similarity help episodic retrieval? Journal of Memory and Language, 46, 85-98.
30. Jehn, K. A., Northcraft, G. B., & Neale, M. A. (1999). Why differences make a difference: A

31. field study of diversity, conflict, and performance in workgroups. Administrative Science Quarterly, 44, 741-763.
32. Kerr, N. L., & Tindale, S. R. (2004). Group performance and decision making. Annual Review of Psychology, 55, 623-655.
33. Kosslyn, S. M. (1994). Image and brain: The resolution of the imagery debate. Cambridge, MA: MIT Press.
34. Kozhevnikov, M., Kosslyn, S. M., & Shephard, J. (2005). Spatial versus object visualizers: A new characterization of visual cognitive style. Memory & Cognition, 33, 710-726.
35. LePine, J. A. (2005). Adaptation of teams in response to unforeseen change: Effects of goal difficulty and team composition in terms of cognitive ability and goal orientation. Journal of Applied Psychology, 90, 1153-1167.
36. Littlepage, G., Robison, W., & Reddington, K. (1997). Effects of task experience and group experience on group performance, member ability, and recognition of expertise. Organizational Behavior and Human Decision Processes, 69, 133-147.

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи бакалавра: "Лінійні моделі побудови команд розробників програмного забезпечення"

Обсяг пояснювальної записки: 69 аркуші

Дата закінчення дипломної роботи 13 червня 2021р.

Підпис студента _____