

**Цюцюра Світлана Володимирівна**

Доктор технічних наук, професор, професор кафедри інженерії програмного забезпечення та кібербезпеки,

<https://orcid.org/0000-0002-4270-7405>

Державний торговельно-економічний університет, Київ

**Єрукаєв Андрій Віталійович**

Кандидат технічних наук, доцент кафедри інженерії програмного забезпечення та кібербезпеки,

<https://orcid.org/0000-0002-9956-3713>

Державний торговельно-економічний університет, Київ

**Андрусенко Валерій Павлович**

Luxoft, розробник програмного забезпечення,

<https://orcid.org/0009-0006-0443-0878>

Державний торговельно-економічний університет, Київ

## ОПТИМІЗАЦІЇ ПРОЦЕСІВ ВІДБОРУ ПЕРСОНАЛУ ЗА АЛГОРИТМОМ ПОШУКУ ІДЕАЛЬНИХ КАНДИДАТІВ

**Анотація.** У статті розглянуто алгоритмічний підхід до відбору ідеальних кандидатів, використовуючи аналіз критеріїв та автоматизований процес відбору, для оптимізації процесів відбору персоналу та підвищення ефективності кадрового менеджменту. Досліджено ефективність цього методу, а також його практичні застосування в рекрутингу та внутрішньому переведенні працівників. У сучасному світі, де конкуренція на ринку праці надзвичайно висока, пошук ідеальних кандидатів стає ключовим завданням для багатьох компаній. Вибір правильного претендента на вакантне робоче місце може значно вплинути на успішність бізнесу, а автоматизація процесу відбору кандидатів допомагає роботодавцям значно економити час і ресурси на ручному відборі. Отже, постає необхідність дослідження різних методів та алгоритмів відбору персоналу для досягнення максимальної ефективності автоматизації процесу. Наукові дослідження і публікації, що стосуються відбору кандидатів та застосування алгоритмів у цій сфері, відображають багатоаспектність і складність процесу відбору персоналу. Попередні дослідження розглядали широкий спектр питань, включаючи розробку критеріїв відбору, психологічні тести, аналіз кар'єрних траєкторій, а також ефективність різних методів і підходів до відбору кандидатів. Деякі дослідження зосереджуються на розвитку алгоритмів машинного навчання і штучного інтелекту для автоматизації процесу відбору кандидатів, зокрема за допомогою аналізу резюме, професійних навичок та відгуків. Інші дослідження розглядають роль соціальних мереж у підборі персоналу та розробку алгоритмів для аналізу профілів кандидатів у соціальних мережах. Крім того, було проведено дослідження, що спрямовані на вивчення впливу різних факторів, таких як культурні та соціальні відмінності, на процес відбору кандидатів. Ці дослідження надають цінний внесок у розуміння найефективніших стратегій і методів відбору персоналу в різних умовах. Загальною тенденцією в попередніх дослідженнях є спроба забезпечити об'єктивність, ефективність та інноваційність процесу відбору кандидатів за допомогою розроблення нових алгоритмів і методів, які враховують широкий спектр факторів та потреб бізнесу.

**Ключові слова:** алгоритмічний підхід пошуку ідеальних кандидатів; автоматизація процесу відбору кандидатів

### Мета роботи

Метою пропонованого дослідження є розгляд алгоритмічного підходу до відбору ідеальних кандидатів у IT-індустрії, зосередженого на знаходженні перетину між масивами результатів за критеріями оцінки. Для досягнення цієї мети

визначено такі завдання: охарактеризувати алгоритм для автоматизованого відбору персоналу на основі визначених критеріїв оцінки, пояснити критерії відбору, визначити можливості використання алгоритму в різних сферах IT-індустрії та розглянути можливості подальшого розвитку і вдосконалення алгоритму для оптимізації процесу відбору персоналу.

Дослідження в цьому напрямі стосувалися широкого спектру питань, так алгоритмічний підхід до відбору ідеальних кандидатів включає низку етапів [1 – 6]:

1. *Визначення критеріїв.* Спочатку потрібно визначити критерії, за якими будуть оцінюватися кандидати. Це може включати освіту, досвід роботи, навички, культурну відповідність та інші фактори, які вам важливі.

2. *Створення профілю ідеального кандидата.* На основі визначених критеріїв створюється профіль ідеального кандидата. Це дає змогу зрозуміти, які характеристики потрібні для успішного виконання роботи.

3. *Збирання інформації.* Збирання даних про потенційних кандидатів, включаючи їх резюме, інтерв'ю, рекомендації та будь-яку іншу доступну інформацію.

4. *Автоматизована фільтрація кандидатів.* Використовуючи алгоритми, можна автоматично фільтрувати кандидатів на основі їх відповідності визначеним критеріям. Наприклад, програмне забезпечення може автоматично відсіювати кандидатів, які не мають необхідного рівня освіти або досвіду роботи.

5. *Оцінка.* Оцінка кандидатів на основі даних, зібраних на попередніх етапах. Це може включати використання алгоритмів для аналізу резюме, проведення онлайн-тестування навичок, інтерв'ю та оцінювання рекомендацій.

6. *Прийняття рішення.* На основі результатів оцінки вирішується, які кандидати найбільш придатні для ролі.

7. *Застосування навчання з вчителем (за потреби).* Використання алгоритмів машинного навчання для вдосконалення процесу відбору кандидатів шляхом аналізу результатів попередньо найнятих працівників та їхніх характеристик.

## Виклад основного матеріалу

Важливо пам'ятати, що алгоритмічний підхід може бути корисним інструментом для відбору кандидатів, але він не повинен замінювати людську експертизу і оцінку [5–10]. Людський фактор все ще залишається важливим для прийняття остаточного рішення щодо найняття кандидата.

Важливість оптимізації процесу найму співробітників складно переоцінити, що демонструє опитування CareerBuilder. У ньому стверджується, що існує проблема з традиційним процесом найму. Так, 74% організацій признаються, що найняли на посаду не ту людину. Більше того, 66% співробітників заявили, що погодилися на роботу, яка їм не підходить, причому половина з них звільнилася протягом перших шести місяців [1]. Ця статистика наведена на рис. 1.

Основним принципом алгоритму є отримання підходящих результатів за кожним критерієм окремо та подальший пошук перетину між масивами результатів.



Рисунок 1 – Статистика традиційних методів рекрутингу. Джерело: вебсайт Vervoe[1]

Зважаючи на важливість правильного відбору кандидатів, необхідно визначити чіткі критерії оцінки, за якими будуть оцінюватися потенційні претенденти на позиції. Перелік критеріїв за порядком пріоритетності такий:

1. Технології та інструменти – це один з ключових критеріїв, що визначає наявність потрібних навичок і знань, необхідних для виконання обов'язків певної позиції. Це можуть бути технічні навички, мови програмування, спеціалізовані програми тощо.

2. Область роботи визначає конкретну сферу діяльності або напрям роботи, в якому працівник буде працювати. Він включає такі області, як: front-end, backend, full stack, дизайн, тестування.

3. Посада вказує на конкретну посаду або позицію, на яку кандидат подає заявку. Наприклад, інтерн, Junior-, Middle-, Senior-розробник.

4. Рівень англійської визначає рівень володіння іноземною мовою, який потрібен для виконання обов'язків на посаді. Цей критерій є дуже важливим, оскільки зазвичай робота включає комунікацію з міжнародними клієнтами або колегами, письмову або усну взаємодію англійською мовою.

5. Досвід роботи є важливим показником успішності кандидата. Оцінюються якість, тривалість та рівень відповідності досвіду роботи вимогам певної позиції.

6. Освіта може включати наявність вищої освіти у відповідній галузі, сертифікатів, курсів або тренінгів.

7. Очікувана заробітна плата визначає рівень заробітної плати, який кандидат очікує отримати за відповідну позицію. Цей критерій може бути важливим фактором, особливо якщо роботодавець пропонує конкурентоспроможну зарплату порівняно з іншими компаніями на ринку праці.

8. Область/Місто визначає конкретне місто, де розташована вакансія. Цей критерій може бути важливим для кандидатів, які враховують географічні обмеження або шукають можливості для життя в певному місці.

9. Місце роботи визначає конкретне місце розташування робочого місця, наприклад, віддалена робота, офіс в певному місті або регіоні. Це може впливати на доступність кандидата до роботи та його готовність до переміщень.

Користувачський інтерфейс для заповнення всіх критеріїв наведено на рис. 2. У цій формі роботодавець має можливість вказати всі необхідні характеристики, які відповідають ідеальному кандидату, та розпочати пошук. Кожен критерій у формі є опціональним, що допомагає роботодавцю визначити свої пріоритети та врахувати унікальні вимоги для кожної конкретної вакансії. У випадку, якщо жодне поле не було заповнене, система автоматично виведе список профілів усіх кандидатів, які зареєстровані в базі даних. Це дає змогу роботодавцю отримати повний перелік доступних кандидатів і вибрати тих, які найбільше відповідають його потребам та вимогам.

Рисунок 2 – Користувачський інтерфейс для заповнення критеріїв для пошуку кандидатів (побудовано у Figma)

Зіставлення кандидатів з критеріями є ключовим етапом у процесі відбору ідеальних кандидатів. Це реалізується через поєднання їхніх вказаних атрибутів з критеріями, визначеними користувачем під час пошуку. Кандидати реєструються в системі шляхом заповнення анкети, де вони вказують усю необхідну інформацію, таку як технічні навички, досвід роботи, освіту, рівень англійської тощо. Після цього їхні дані зберігаються у базі даних [5; 6].

Коли користувач-роботодавець встановлює критерії вакансії і починає пошук кандидатів, система використовує ці критерії для порівняння з атрибутами кандидатів. Наприклад, якщо користувач шукає кандидатів з певним рівнем англійської мови, система використовує цей критерій для порівняння з рівнем англійської мови, вказаним кандидатами у своїй анкеті. Для прикладу, розглянемо такий

сценарій: кандидат А та кандидат Б мають рівень англійської B2, а кандидат С – рівень A2, і коли роботодавець шукає людей з B2, у масиві для критерію рівня англійської будуть кандидати А та Б. Після цього для кожного критерію формується масив кандидатів, які відповідають цьому критерію [9; 10]. Результатом роботи цього етапу алгоритму є, у нашому випадку, 10 окремих масивів з підходящими кандидатами для кожного критерію окремо.

Програмна реалізація формування масивів на мові програмування JavaScript наведена на рис. 3. Спочатку він перевіряє, чи є збіг за технологіями. Якщо так, то він виконує запит до бази даних, щоб отримати ID кандидатів, які мають відповідні технології. Далі виконується пошук збігів за рівнем англійської мови.

```
//сбігання по технологіям
let techAndToolsMatches = [];

if (Array.isArray(techAndToolsIds) && techAndToolsIds.length) {
  techAndToolsMatches = await this.getMatchesByFilter(
    techAndToolsIds,
    db.Helper.getSqlCandidateIdsByTechAndTools(techAndToolsIds.toString())
  );
}

//сбігання по англійській
let englishMatches = await this.getMatchesByFilter(
  params.candidateEnglish,
  db.Helper.getSqlCandidateIdsByEnglish(params.candidateEnglish)
);
```

Рисунок 3 – Знаходження збігів на прикладі технологій та рівня англійської (код з Visual Studio Code, рисунок побудовано у Carbon App)

Асинхронний метод `getMatchesByFilter` (рис. 4) використовується для виконання запитів до бази даних і отримання списку збігів. Він приймає фільтр та SQL-запит як параметри. Потім перевіряє, чи фільтр порожній або невизначений. Якщо так, то повертається пустий масив. У іншому випадку виконується SQL-запит за допомогою функції `connectionPromise`, а результат записується у змінну `matches`. Потім залежно від того, чи SQL-запит зв'язаний з таблицею `candidate_technology_tool`, метод повертає масив ID кандидатів або масив ID записів з результатами запиту.

```
async getMatchesByFilter(filter, sql) {
  let matches = [];
  if (filter === "" || filter === null || filter === undefined) {
    return matches;
  } else {
    matches = await connectionPromise(sql, "");
    return sql.includes("candidate_technology_tool") ? matches.map(a => a.candidate_id) : matches.map(a => a.id);
  }
}
```

Рисунок 4 – Асинхронний метод для отримання збігів з бази даних (код з Visual Studio Code, рисунок побудовано у Carbon App)

Аналогічні масиви створюються для інших критеріїв. Отримавши масиви по кожному з критеріїв, система шукає перетин цих масивів, тобто кандидатів, які відповідають всім вказаним

критеріям. Це реалізується шляхом послідовного злиття масивів і відбору кандидатів, які є спільними для всіх критеріїв.

На рис. 5 наведено фрагмент коду, який реалізує цей крок алгоритму. У ньому створюється об'єкт resultsObj, який містить результати пошуку кандидатів за різними критеріями. Потім виконується ітерація за ключами цього об'єкта, починаючи з критерію "techAndTools". Кожен наступний критерій порівнюється з попереднім за допомогою методу getMatchesIntersection (рис. 6), і результат записується у змінну resultSet. На виході отримуємо результат пошуку ідеальних кандидатів, які задовольняють усі критерії одночасно.

```

let resultsObj = {
  techAndTools: techAndToolsMatches,
  workArea: workAreaMatches,
  position: positionMatches,
  english: englishMatches,
  workExp: workExpMatches,
  education: educationMatches,
  salary: salaryMatches,
  region: regionMatches,
  workplaces: workplacesMatches,
  city: cityMatches,
};

let resultSet = resultsObj.techAndTools;
let keys = Object.keys(resultSet);

//поиск идеальных кандидатов
for (const [key, value] of Object.entries(resultSet)) {
  let nextIndex = keys.indexOf(key) + 1;
  let nextItem = keys[nextIndex];
  resultSet = this.getMatchesIntersection(resultSet, resultSet[nextItem]);
}
    
```

Рисунок 5 – Пошук ідеальних кандидатів (код з Visual Studio Code, рисунок побудовано у Carbon App)

У методі getMatchesIntersection (а, b) обчислюється перетин двох масивів а і b. Якщо жоден з масивів не є порожнім, то відбувається фільтрація елементів, які присутні в обох масивах. Якщо один з масивів порожній, то повертається інший масив. Якщо обидва масиви порожні, то повертається порожній масив.

```

getMatchesIntersection(a, b) {
  let intersection;

  if (a !== null && a !== null && a !== undefined && a.length !== 0) {
    if (b !== null && b !== null && b !== undefined && b.length !== 0) {
      intersection = a.filter(el => b.includes(el));
    } else {
      intersection = a;
    }
  } else {
    if (b !== null && b !== null && b !== undefined && b.length !== 0) {
      intersection = b;
    } else {
      intersection = [];
    }
  }
  return intersection;
}
    
```

Рисунок 6 – Метод для отримання перетину між двома масивами (код з Visual Studio Code, рисунок побудовано у Carbon App)

Іноді може виникнути ситуація, коли кандидатів за обраними критеріями було знайдено мало, тому алгоритм передбачає пошук додаткових спеціалістів за найбільш важливими характеристиками у першу чергу (рис. 7). Ідеальні кандидати, які будуть завжди

знаходитися першими в списку, будуть об'єднані з додатковими, які не є ідеальними, але так чи інакше підходять роботодавцю і варті уваги. За мінімальну кількість можливих результатів встановлено значення 20.

У методі getAdditionalResults можна розширити поточний результат, додавши до нього додаткових кандидатів з різних категорій. Він приймає currentResult, тобто поточний результат, та separateMatchesObj – збіг за окремим критерієм. Для цього використовується об'єднання масивів за допомогою методу getMatchesUnion (рис. 8). Цикл for проходить по кожному елементу separateMatchesObj, додаючи його кандидатів до розширеного результату, доки кількість кандидатів не досягне 20 або доки не будуть пройдені всі категорії.

```

getAdditionalResults(currentResult, separateMatchesObj) {
  let expandedResults = this.getMatchesUnion(currentResult, separateMatchesObj.techAndTools);

  let keys = Object.keys(separateMatchesObj);

  //поиск дополнительных кандидатов
  for (const [key, value] of Object.entries(separateMatchesObj)) {
    if (expandedResults.length < 20) {
      let nextIndex = keys.indexOf(key) + 1;
      let nextItem = keys[nextIndex];
      expandedResults = this.getMatchesUnion(expandedResults, separateMatchesObj[nextItem]);
    } else break;
  }

  return expandedResults;
}
    
```

Рисунок 7 – Метод для отримання додаткових кандидатів (код з Visual Studio Code, рисунок побудовано у Carbon App)

У методі getMatchesUnion (а, b) об'єднуються два масиви а та b в один масив union, у якому кожен елемент зустрічається тільки один раз. Перевіряється, чи є масиви непорожніми, після чого вони об'єднуються, але без повторень, за допомогою властивостей масивів та об'єкта Set. Якщо один з масивів порожній або невизначений, результатом буде інший масив або порожній масив.

```

getMatchesUnion(a, b) {
  let union;

  if (a !== null && a !== null && a !== undefined && a.length !== 0) {
    if (b !== null && b !== null && b !== undefined && b.length !== 0) {
      union = [...new Set([...a, ...b])];
    } else {
      union = a;
    }
  } else {
    if (b !== null && b !== null && b !== undefined && b.length !== 0) {
      union = b;
    } else {
      union = [];
    }
  }
  return union;
}
    
```

Рисунок 8 – Метод для об'єднання двох масивів (код з Visual Studio Code, рисунок побудовано у Carbon App)

Спрощена загальна схема описаного алгоритму пошуку кандидатів зображена на рис. 9.

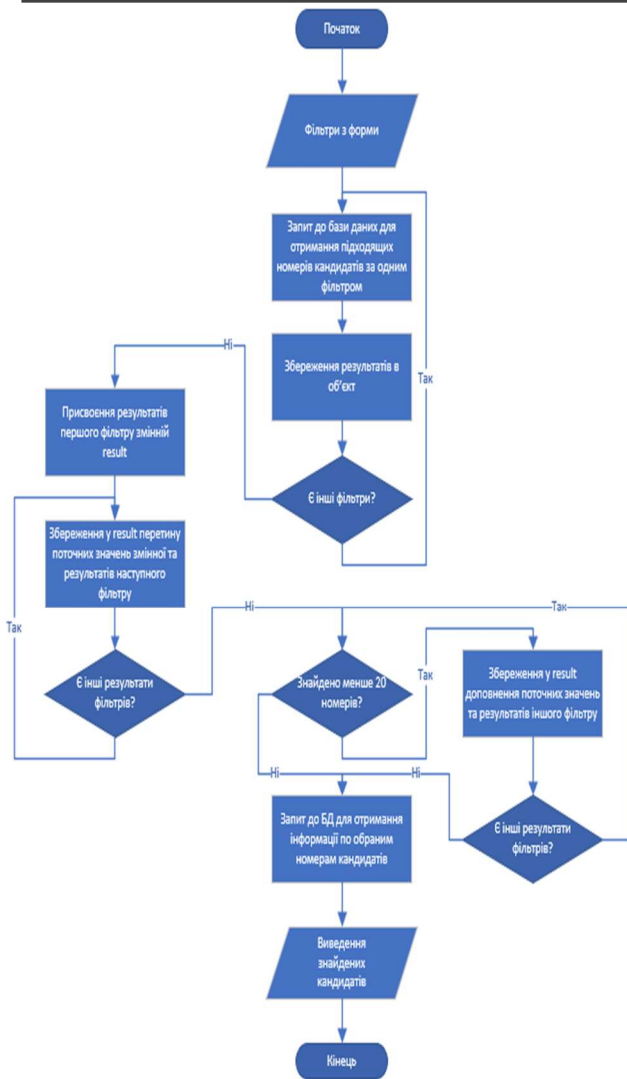


Рисунок 9 – Спрощена схема алгоритму пошуку ідеальних кандидатів

Розглянемо декілька прикладів роботи алгоритму. Заповнимо форму: посада – Мідл-розробник, область роботи – Full Stack, технології – JavaScript, React. (рис. 10).

Рисунок 10 – Заповнення форми для першого прикладу (скріншот програми)

Як видно з рис. 11, ідеальні кандидати розміщені на початку сторінки. Перші три саме мають JavaScript, React у переліку своїх технологій, вони є мідл-Full-Stack-розробниками. Після них можна побачити інші результати, в яких були збіги за критеріями.

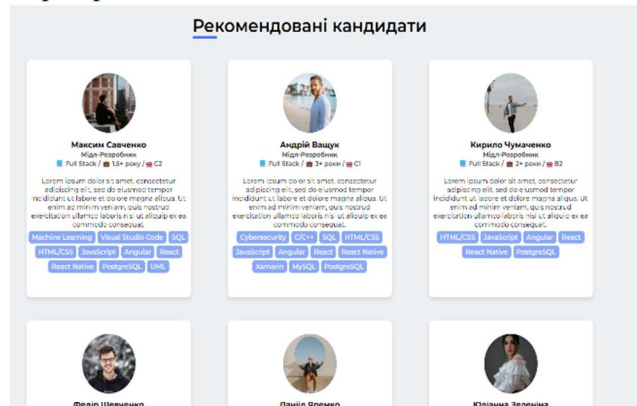


Рисунок 11 – Рекомендовані кандидати для першого прикладу (скріншот програми)

Наступний приклад можна побачити на рис. 12: посада – Сеньйор-розробник, технології – C#, C/C++, .NET, рівень англійської – C2, рівень освіти – вища.

Рисунок 12 – Заповнення форми для другого прикладу (скріншот програми)

Перший кандидат підходить майже за всіма параметрами, оскільки в переліку технологій немає лише .NET, а другий кандидат з технологій має лише C#. Інші – додаткові кандидати.

Цей алгоритм може бути успішно застосований в реальних умовах підбору персоналу для різних компаній та галузей. Він уможливує автоматизувати й оптимізувати процес відбору кандидатів, що веде до скорочення часу та зусиль, витрачених на цей процес, та покращення якості вибору.

Він може застосовуватися ІТ-компаніями для пошуку відповідних кандидатів на різні вакансії, кадровими агенціями, державними установами на позиції, пов'язані з інформаційними технологіями. Також він може бути корисним для внутрішнього переведення працівників на вакантні позиції, щоб максимально використати внутрішній потенціал

компанії, або в освітніх закладах, щоб допомогти студентам і випускникам знайти підходяще місце для застосування набутих навичок.

### Висновки

1. Алгоритмічний підхід до відбору ідеальних кандидатів є потужним інструментом, що допомагає компаніям ефективно впоратися із завданням відбору найкращих претендентів. Його застосування дає змогу знизити витрати часу і ресурсів на відбір персоналу, а також підвищити якість та результативність цього процесу.

2. Алгоритм має доволі багато перспектив розвитку. Можливі напрями його розвитку включають: оптимізацію алгоритму для зменшення часу виконання, що може включати в себе покращення алгоритмів об'єднання та перетину масивів, а також

використання більш ефективних методів опрацювання даних. Вдосконалення критеріїв оцінки: розширення списку критеріїв оцінки кандидатів, враховуючи нові вимоги та потреби роботодавців. Застосування методів машинного навчання для автоматичного визначення ваги кожного критерію та покращення точності відбору ідеальних кандидатів. Розширення областей застосування, тобто використання алгоритму в різних галузях та сферах діяльності, де необхідний відбір ідеальних кандидатів. Розширення функціональності алгоритму для врахування різноманітності кандидатів, таких як студенти, фрілансери, досвідчені фахівці тощо. Додавання нових можливостей, таких як автоматична генерація звітів, редагування даних, інтеграція з іншими системами управління персоналом.

### Список літератури

1. Eaton B. Why traditional recruitment processes no longer cut it. Vervoe. URL: <https://vervoe.com/why-traditional-recruitment-processes-no-longer-cut-it/> (дата звернення: 25.03.2024).
2. Effective Strategies For Talent Acquisition – Innovature BPO. *Innovature BPO*. URL: <https://innovatureinc.com/effective-strategies-for-talent-acquisition/> (дата звернення: 02.04.2024).
3. Koch-Bayram I. F., Kaibel C. Algorithms in personnel selection, applicants' attributions about organizations' intents and organizational attractiveness: An experimental study. *Human Resource Management Journal*. 2023. URL: <https://onlinelibrary.wiley.com/doi/full/10.1111/1748-8583.12528> (дата звернення: 02.04.2024).
4. Williams T. Learning from projects. *Journal of operational research society* 54, 2003. – P.443-451. <https://link.springer.com/article/10.1057/palgrave.jors.2601549>.
5. HR Automation: Best Practices & Trends. *Teambuilding.com*. URL: <https://teambuilding.com/blog/hr-automation> (дата звернення: 02.04.2024).
6. Controlled by the algorithm, coached by the crowd – how HRM activities take shape on digital work platforms in the gig economy / M. Waldkirch et al. *The International Journal of Human Resource Management*. 2021. URL: <https://www.tandfonline.com/doi/full/10.1080/09585192.2021.1914129> (дата звернення: 03.04.2024).
7. Цюцюра М. І. Дослідження функціональних задач аналізу собівартості ІТ-продукту. *Управління розвитком складних систем: зб. наук. праць*. Київ : КНУБА, 2017. Вип. 29 № 29 С. 177 – 183. Видання індексовано в МНБД: Index Copernicus, BASE.
8. Kryvoruchko Olena, Kostiuk Mykhailo, Tsiutsiura Mykola. Architectural solution of time management system in test driven development approach. *International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064 Impact Factor (2012): 3.358; Volume 7 Issue 07, 2017*. Рр. 1098–1100. Видання індексовано в МНБД: Index Copernicus, BASE.
9. James Gareth, Witten Daniela, Hastie Trevor, Tibshirani Robert. An introduction to statistical learning. – Springer Science+Business Media New York, 2013. 426 p.
10. Kriesel D. A Brief Introduction to Neural Networks, 2007, [http://www.dkriesel.com/en/science/neural\\_networks](http://www.dkriesel.com/en/science/neural_networks).

Стаття надійшла до редколегії 11.05.2024

#### Tsiutsiura Svitlana

DSc (Eng.), Professor, Professor of the Department of Software Engineering and Cybersecurity,

<https://orcid.org/0000-0002-4270-7405>

State University of Trade and Economics, Kyiv

#### Yerukaiev Andrii

PhD (Eng.), Associate Professor, Associate Professor of the Department of Software Engineering and Cybersecurity,

<https://orcid.org/0000-0002-9956-3713>

Kyiv National University of Trade and Economics, Kyiv

#### Andrusenko Valery

Luxoft, Software Developer,

<https://orcid.org/0009-0006-0443-0878>

State University of Trade and Economics, Kyiv

**OPTIMIZATION OF PERSONNEL SELECTION PROCESSES  
THE PERFECT CANDIDATES SEARCH ALGORITHM**

**Abstract.** The article discusses an algorithmic approach to the selection of ideal candidates, using criteria analysis and an automated selection process, to optimize personnel selection processes and increase the efficiency of personnel management. The effectiveness of this method, as well as its practical applications in recruiting and internal transfer of employees, were studied. In today's world, where the competition in the labor market is extremely high, finding the ideal candidates becomes a key task for many companies. Choosing the right candidate for a job vacancy can have a significant impact on the success of a business, and automating the candidate selection process allows employers to significantly save time and resources on manual selection. Therefore, there is a need to research various methods and algorithms of personnel selection to achieve the maximum efficiency of process automation. Scientific studies and publications related to the selection of candidates and the application of algorithms in this area reflect the multifaceted and complex nature of the recruitment process. Previous research has addressed a wide range of issues, including the development of selection criteria, psychological tests, career trajectory analysis, and the effectiveness of various methods and approaches to candidate selection. Some research focuses on the development of machine learning and artificial intelligence algorithms to automate the candidate selection process, in particular by analyzing CVs, professional skills and testimonials. Other research examines the role of social media in recruitment and the development of algorithms to analyze candidate social media profiles. In addition, research has been conducted to examine the impact of various factors, such as cultural and social differences, on the candidate selection process. These studies provide a valuable contribution to the understanding of the most effective recruitment strategies and methods in different settings. A general trend in previous research is an attempt to ensure the objectivity, efficiency and innovation of the candidate selection process by developing new algorithms and methods that take into account a wide range of factors and business needs.

**Keywords:** algorithmic approach to finding ideal candidates; automation of the candidate selection process

**References**

1. Eaton, B. (2024). Why traditional recruitment processes no longer cut it. Vervoe. URL: <https://vervoe.com/why-traditional-recruitment-processes-no-longer-cut-it/> (дата звернення: 25.03.2024).
2. Effective Strategies For Talent Acquisition – Innovature BPO. *Innovature BPO*. URL: <https://innovatureinc.com/effective-strategies-for-talent-acquisition/> (date of application: 02.04.2024).
3. Koch-Bayram, I. F., Kaibel, C. (2023). Algorithms in personnel selection, applicants' attributions about organizations' intents and organizational attractiveness: An experimental study. *Human Resource Management Journal*. URL: <https://onlinelibrary.wiley.com/doi/full/10.1111/1748-8583.12528> (date of application: 02.04.2024).
4. Williams, T. (2003). Learning from projects. *Journal of operational research society*, 54, 443–451. <https://link.springer.com/article/10.1057/palgrave.jors.2601549>.
5. HR Automation: Best Practices & Trends. (2024). *teambuilding.com*. URL: <https://teambuilding.com/blog/hr-automation> (date of application: 02.04.2024).
6. Waldkirch, M. et al. (2021). Controlled by the algorithm, coached by the crowd – how HRM activities take shape on digital work platforms in the gig economy. *The International Journal of Human Resource Management*. URL: <https://www.tandfonline.com/doi/full/10.1080/09585192.2021.1914129> (date of application: 03.04.2024).
7. Tsiutsiura, M. I., Tsiutsiura, S. V. and Kryvoruchko, O. V. (2019). Information technologies for the development of the content of education: monograph Kyiv: CP «Comprint», 118. ISBN-978-966-929-967-9.
8. Kryvoruchko, Olena, Kostiuk, Mykhailo, Tsiutsiura, Mykola. (2012). Architectural solution of time management system in test driven development approach. *International Journal of Science and Research*, 7, 07, 1098–1100.
9. James, Gareth, Witten, Daniela, Hastie, Trevor, Tibshirani, Robert. (2013). An introduction to statistical learning. Springer Science+Business Media New York, 426.
10. Kriesel, D. (2007). A Brief Introduction to Neural Networks, [http://www.dkriesel.com/en/science/neural\\_networks](http://www.dkriesel.com/en/science/neural_networks).

**Посилання на публікацію**

APA Tsiutsiura, S., Yerukaiev, A. & Andrusenko, V. (2024). Optimization of personnel selection processes the perfect candidates search algorithm. *Management of Development of Complex Systems*, 58, 86–92, dx.doi.org\10.32347/2412-9933.2024.58.86-92.

ДСТУ Цюцюра С. В., Срукаєв А. В., Андрусенко В. П. Оптимізації процесів відбору персоналу за алгоритмом пошуку ідеальних кандидатів. *Управління розвитком складних систем*. Київ, 2024. № 58. С. 86 – 92, dx.doi.org\10.32347/2412-9933.2024.58.86-92.