

УДК 004.9: 004.4

DOI 10.36910/10.36910/6775-2313-5352-2023-22-13

Левончук С.А., Гуменюк Л.О., Сацик О.В.

Луцький національний технічний університет, м. Луцьк, Україна

АВТОМАТИЗАЦІЯ ПРОЦЕСУ ВИБОРУ НАПРЯМКУ РУХУ КАБІНИ ЛІФТА

Проведено детальне дослідження автоматизації процесу вибору напрямку руху кабіни ліфта. Цей процес є важливою складовою ефективною роботи будь-якого будинку з великою кількістю поверхів, оскільки забезпечує швидке та безпечне транспортування пасажирів до призначених місць. Робота присвячена розробці імітаційної моделі для дослідження алгоритмів автоматизації процесу вибору напрямку руху кабіни ліфта з метою вибору системи, що має найбільшу продуктивність і найменший час очікування. Модель включає в себе можливість змінювати різні параметри, такі як кількість ліфтів, потужності ліфтів, навантаженість, час моделювання та ін.

Ключові слова: модель, ліфт, моделювання, вибір напрямку, кабіна ліфта, моделювання дискретної події, Python.

Постановка проблеми. Один з основних підходів до розробки алгоритмів вибору напрямку руху кабіни ліфта базується на оптимізації траєкторії руху ліфта. Даний підхід має на меті мінімізувати сумарну відстань, пройдену ліфтом, і таким чином зменшити енергоспоживання та час очікування пасажирів. Прикладом такого підходу є алгоритм "best-first search", який обирає наступний поверх для відвідування на основі мінімальної відстані подорожі від поточного поверху [1].

Інший підхід базується на прогнозуванні пасажирського попиту. Цей підхід має на меті передбачити майбутній пасажиропотік і відповідно до нього скоригувати траєкторію руху ліфта. Прикладом такого підходу є алгоритм "навчання з підкріпленням", який використовує історичні дані для прогнозування пасажиропотоку та коригування траєкторії руху ліфта в режимі реального часу [2].

Імітаційні моделі також широко використовуються для вивчення поведінки систем управління ліфтами. Ці моделі використовуються для аналізу продуктивності різних алгоритмів вибору напрямку руху кабіни ліфта. Наприклад, метод "дискретно-подієвого моделювання" може бути використаний для моделювання роботи системи керування ліфтом з урахуванням прибуття пасажирів, траєкторії руху ліфта та напрямків руху пасажирів [3].

Актуальність розробки нового алгоритму вибору напрямку руху кабіни ліфта полягає в тому, що зі зростанням кількості поверхів у будівлях і кількості ліфтів, постає потреба в ефективному управлінні рухом кабіни. Не ефективна робота алгоритму вибору напрямку руху може призвести до затримок у підйомі та спуску, збільшення часу очікування ліфта та збільшення витрат на його утримання.

Крім того, існуючі алгоритми вибору напрямку руху кабіни ліфта були розроблені десятиліття тому, і не враховують сучасні технології та нові вимоги до ефективності та безпеки пасажирів. Розробка нового алгоритму може допомогти вдосконалити роботу ліфтів, зменшити час очікування, збільшити швидкість руху та забезпечити безпеку пасажирів.

Метою роботи є вибір системи, що найкраще підходить для автоматизації процесу вибору напрямку руху кабіни ліфта на основі аналізу існуючих алгоритмів.

Основне завдання та одержані результати роботи. Одним із найпоширеніших методів автоматизованого вибору напрямку є використання систем керування призначенням (DCS) [4]. Ці системи використовують алгоритми для розрахунку найефективнішого маршруту руху ліфта, виходячи з бажаного місця призначення пасажирів. Система DCS також дозволяє оптимізувати рух ліфтів, зменшуючи кількість зупинок кожного ліфта. Спрямовуючи пасажирів до конкретних ліфтів, система може зменшити кількість непотрібних зупинок, що може значно скоротити час, необхідний пасажирам, щоб дістатися до потрібного поверху.

Іншим методом автоматизованого вибору напрямку є використання предиктивних алгоритмів [5]. Ці алгоритми використовують такі дані, як історичні схеми руху та очікуване використання, щоб передбачити, де і коли пасажирам знадобляться ліфти. Потім система заздалегідь розміщує ліфти, щоб задовольнити очікуваний попит, скорочуючи час очікування і підвищуючи ефективність.

Алгоритми прогнозування використовують різні джерела даних для точного прогнозування, включаючи історичні схеми руху, дані про зайнятість і заплановані події [6]. Алгоритми можуть навчатися на основі даних і коригувати свої прогнози з часом, щоб зробити систему більш точною. Система безперервно відстежує та аналізує дані в режимі реального часу, щоб спрогнозувати попит на ліфти в будівлі. На основі цієї інформації система може скеровувати ліфти на поверхи, де вони найімовірніше потрібні, скорочуючи час очікування пасажирів і підвищуючи загальну ефективність. Алгоритм прогнозування також може визначати пріоритетність ліфтів на основі кількості пасажирів, які очікують на кожному поверсі, кількості доступних ліфтів та очікуваного часу прибуття ліфта на кожен поверх. Це дозволяє системі надавати персоналізоване та ефективне обслуговування кожному пасажирові, зменшуючи затори та час очікування.

Ліфтові системи є важливим аспектом сучасної інфраструктури, що забезпечує ефективне вертикальне транспортування в багатоповерхових будівлях. Протягом багатьох років для оптимізації цього процесу були розроблені різні методи, починаючи від простих підходів, заснованих на правилах, і закінчуючи більш просунутими методами, заснованими на машинному навчанні. Ці методи можна умовно поділити на п'ять категорій: методи, засновані на аналізі трафіку; методи, засновані на пріоритетах виклику поверху; методи на основі машинного навчання; методи оптимізації руху ліфтів на основі Інтернету речей; гібридні методи оптимізації руху ліфтів. Вибір відповідного методу залежить від різних факторів, таких як тип будівлі, схеми руху та потреби користувачів.

Для проведення аналізу роботи алгоритмів управління ліфтовою кабіною було розроблено програмне забезпечення, яке дозволило порівняти ефективність різних алгоритмів, а саме: «Першим прийшов, першим обслужений» (FCFS); найкоротшого часу пошуку (SSTF); сканування запитів SCAN; пропускання пустих поверхів, які не містять запитів LOOK.

Аналізуючи роботу алгоритму FCFS можна зробити висновок, що він досить простий та легко реалізовується в програмах, однак, при збільшенні кількості запитів загальна кількість операцій пошуку значно збільшується, що може призвести до збільшення часу очікування для пасажирів.

Алгоритм SSTF працює значно ефективніше за FCFS, коли маємо справу зі сценаріями, де послідовність операцій відрізняється за місцем розташування запитів. У конкретному випадку з нашими даними SSTF показав зниження кількості операцій пошуку на 60-70% порівняно з FCFS. Таким чином, можна стверджувати, що алгоритм SSTF є більш ефективним для управління роботою кабіни ліфта, оскільки дозволяє зменшити час очікування пасажирів та збільшити пропускну здатність.

SCAN-алгоритм має вищу ефективність порівняно з алгоритмом FCFS, тому що він зменшує середній час очікування пасажирів. Він забезпечує рівномірний розподіл навантаження на кабіні, оскільки кабіна переміщується в обидві сторони з центральної точки, що зменшує затримки в обробці запитів в окремих секторах і працює досить добре в умовах високого навантаження. Але його недоліком є те, що він проходить весь шлях від першого поверху до останнього (і навпаки) не зважаючи, чи є запити з цих поверхів.

Алгоритм LOOK є ефективним рішенням для оптимізації вибору руху кабіни ліфта в будівлі. Основна ідея алгоритму полягає в тому, щоб пересуватись тільки в той бік, де знаходяться запити на пересування, що дозволяє уникнути зайвих зупинок та прискорити процес пересування.

Виходячи з результатів аналізу алгоритмів, була розроблена імітаційна модель автоматизації процесу вибору напрямку руху кабіни ліфта.

Модель побудована за допомогою мови програмування Python і використовує бібліотеку SimPy для моделювання дискретних подій. Параметри симуляції, такі як кількість поверхів, кількість ліфтів, вантажопідйомність ліфта, тривалість симуляції та інші, визначаються на початку коду.

Програмна реалізація дозволяє обрати, яким методом буде здійснена імітація роботи ліфтової кабіни - FCFS або SSTF.

Розглянемо результати моделювання, отримані при використанні програми «першим прийшов - першим обслужений» (FCFS).

Вхідні дані моделювання:

- кількість ліфтів = 1;

- потужність ліфта (кількість людей, що перевозиться одночасово) = 1;

- швидкість ліфта = 10;

- значення лямбда = 0.5;
- тривалість моделювання = 10000;
- модель управління ліфтом = FCFS;
- кількість поверхів = 7.

Якщо дані середнього часу очікування представити в діаграмі, то отримаємо залежність середнього часу очікування пасажирів від кількості ліфтів (рис. 1).

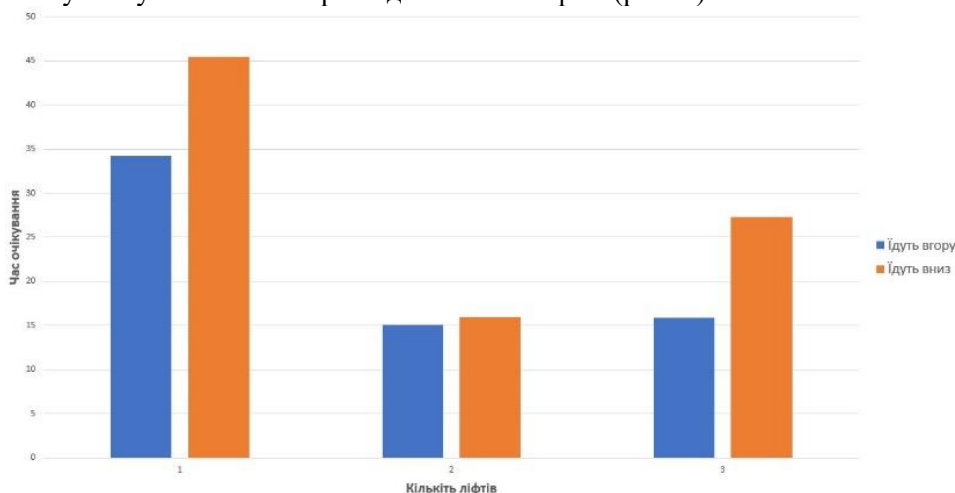


Рисунок 1 – Залежність середнього часу очікування пасажирів від кількості ліфтів при моделюванні методом FCFS.

Загальна тенденція з цих даних полягає в тому, що зі збільшенням кількості ліфтів зменшується час очікування пасажирів. Це пов'язано з тим, що більша кількість ліфтів забезпечує більшу пропускну здатність і меншу кількість людей, які очікують на один ліфт.

Наприклад, при переході від одного до двох ліфтів час очікування пасажирів, що їдуть вгору, зменшився з 34.24 до 15.12 одиниць, а для тих, що їдуть вниз, з 45.48 до 16.0 одиниць. Це є досить значним зменшенням часу очікування.

При додаванні третього ліфту час очікування для тих, що їдуть вгору, зменшився до 15.89 одиниць, а для тих, що їдуть вниз, збільшився до 27.30 одиниць, порівняно з двома ліфтами. Зростання часу очікування може бути пов'язане з більшою кількістю конкуруючих запитів на ліфт. Коли до системи додається третій ліфт, запити розподіляться між трьома ліфтами, що може призвести до більшої конкуренції між ними. Окрім того, якщо система не оптимізована для розподілу пасажирів між ліфтами, то деякі ліфти можуть залишатися непотрібними, тоді як інші можуть бути перевантажені, що призводить до збільшення часу очікування.

Таким чином, збільшення кількості ліфтів не завжди призводить до скорочення часу очікування, якщо система не оптимізована для оптимального розподілу пасажирів між ліфтами (модель FCFS).

Порівняння часу очікування пасажирів, що їдуть вгору, для різної кількості ліфтів, дало наступний результат:

- у випадку з одним ліфтом максимальний час очікування становить 100 одиниць. Деякі пасажирів очікують досить довго (більше 50 одиниць), тоді як інші майже відразу отримують ліфт. Це свідчить про те, що модель FCFS може бути не дуже ефективною, оскільки вона не приділяє уваги пасажирам, які чекають найбільше;

- у випадку з двома ліфтами максимальний час очікування становить 60 секунд - це значно краще, ніж у випадку з одним ліфтом, оскільки обидва ліфти можуть обслуговувати пасажирів одночасно, зменшуючи час очікування;

- у випадку з трьома ліфтами максимальний час очікування становить 60 секунд - це краще порівняно з одним ліфтом, але гірше, ніж у випадку з двома ліфтами. Це може бути пов'язано з тим, що при трьох ліфтах пасажирів розподіляються менш рівномірно, що може призвести до того, що один з ліфтів залишається не використаним у деякий час.

Порівняння часу очікування пасажирів, що їдуть вниз, для різної кількості ліфтів, дає аналогічні результати – час очікування при запуску другого ліфту зменшився від 120 до 60 одиниць, при додаванні третього ліфту час очікування залишається 60 одиниць, але кількість пасажирів, що очікують ліфт зростає.

Розглянемо результати моделювання, отримані при використанні програми «з найменшим часом пошуку обслуговується першим» (SSTF).

Вхідні дані моделювання ті ж, що і у моделі FCFS. Залежність середнього часу очікування пасажирів від кількості ліфтів приведена на рис. 2.

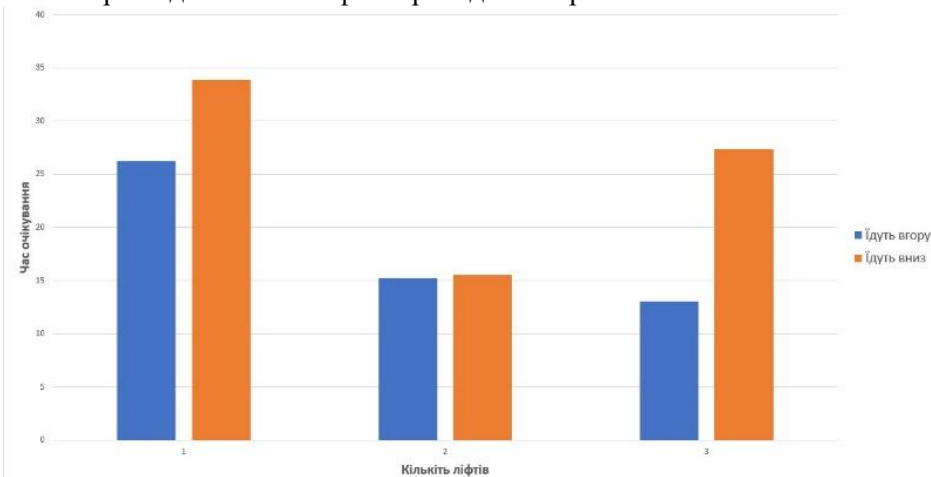


Рисунок 2 – Залежність середнього часу очікування пасажирів від кількості ліфтів при моделюванні методом SSTF.

Застосування SSTF моделі підвищує ефективність управління ліфтом, що призводить до скорочення часу очікування пасажирів. При наявності одного ліфта, середнє значення часу очікування пасажирів, що їдуть вгору, становить 26.24 одиниці, а пасажирів, що їдуть вниз - 33.87 одиниць. Однак, з двома ліфтами середнє значення часу очікування пасажирів, що їдуть вгору, знижується до 15.23 одиниць, а пасажирів, що їдуть вниз - до 15.51 одиниць. Це може бути пов'язано з тим, що пасажирів мають доступ до більшої кількості ліфтів, тому можуть швидше дістатись до своєї пункту призначення. При трьох ліфтах середнє значення часу очікування пасажирів, що їдуть вгору, знижується до 13.03 одиниць, а пасажирів, що їдуть вниз, збільшується до 27.34 одиниць. Це може бути пов'язано з тим, що пасажирів, які їдуть вниз, можуть мати менше доступних ліфтів, оскільки частину ліфтів можуть займати пасажирів, які їдуть вгору. Порівняємо результати часу очікування пасажирів для випадків з одним, двома та трьома ліфтами:

- у випадку з одним ліфтом максимальний час очікування становить 60 одиниць, деякі пасажирів очікують досить довго (більше 20 -50 одиниць).

- у випадку з двома ліфтами та трьома ліфтами максимальний час очікування складає 60 секунд і максимально кількість пасажирів очікувала 5 одиниць. Це значно краще, ніж у випадку з одним ліфтом.

Приведемо також графіки залежності середнього часу очікування пасажирів від вмістимості ліфта для моделей FCFS і SSTF (рис. 3).

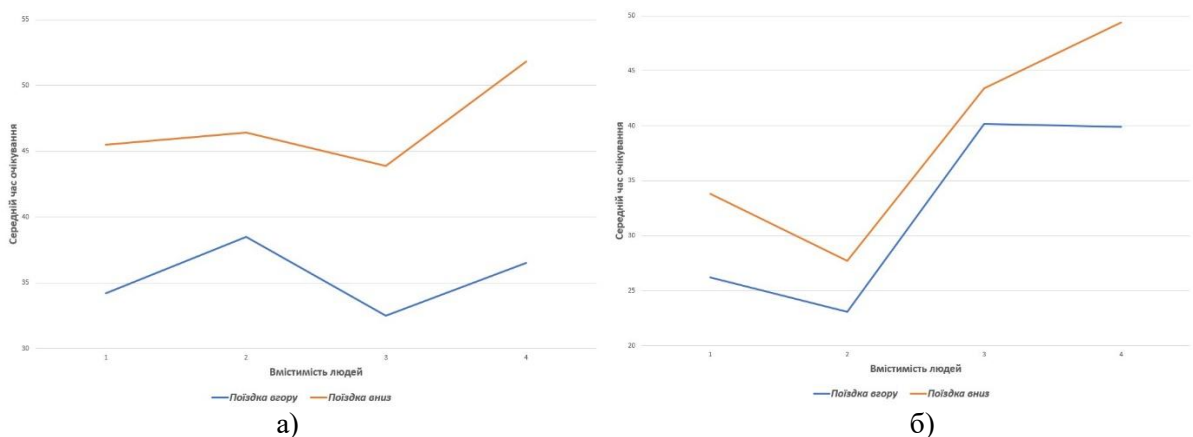


Рисунок 3 – Графіки залежності середнього часу очікування пасажирів від вмістимості ліфта: а) модель FCFS; б) модель SSTF.

Висновок. За результатами моделювання можна зробити висновок, що застосування моделі управління кабіною ліфта SSTF дає кращі результати, ніж модель FCFS у семиповерховому будинку.

По-перше, при використанні моделі FCFS деякі пасажирів можуть очікувати досить довго, тоді як інші можуть отримати ліфт майже відразу. Це свідчить про те, що модель FCFS може бути не дуже ефективною, оскільки вона не приділяє уваги пасажирів, які чекають найбільше.

По-друге, тривалість середнього часу очікування пасажирів при русі вгору і вниз для по моделі SSTF складає 15 одиниць, що є найкращим результатом для обох напрямків одночасно.

Таким чином, застосування моделі управління кабіною ліфта SSTF дає кращі результати для двох ліфтів у семиповерховому будинку, оскільки вона знижує час очікування пасажирів порівняно з моделлю FCFS.

Інформаційні джерела

1. Shang, W., & Zheng, P. (2015). Development and design of an intelligent elevator control system based on PLC. *Journal of Control Science and Engineering*, 2015, 1-8.
2. Shah, R. C., Shah, P., & Shah, M. (2018). An intelligent elevator control system using internet of things. *International Journal of Computer Science and Information Security*, 16(10), 49-56.
3. Hu, X., Wang, X., Yang, Y., & Ma, X. (2019). Design and implementation of elevator group control system based on BIM technology. *Advances in Mechanical Engineering*, 11(1)
4. What is a Distributed Control System? URL : <https://controlstation.com/blog/what-is-a-distributed-control-system/> (дата звернення: 26.05.2023).
5. Колдун М. М., Грудзинський Ю. Є. Побудова алгоритму прогностичної моделі при створенні предиктивного модуля передбачення нештатних ситуацій в роботі автоматизованої системи керування технологічними процесами. *Вісник Національного технічного університету "ХПІ". Сер. : Нові рішення в сучасних технологіях: зб. наук. пр. Харків : НТУ "ХПІ", 2020. № 2. С. 58-63.*
6. Алгоритми прогнозування. URL : <https://www.ibm.com/docs/ru/cognos-analytics/11.2.0?topic=fm-forecasting-algorithms> (дата звернення: 26.05.2023).

Levonchuk S., Gumeniuk L. Satsyk O.

Lutsk National Technical University, Lutsk, Ukraine

AUTOMATION OF THE PROCESS OF CHOOSING THE DIRECTION OF MOVEMENT OF THE ELEVATOR CABIN

A detailed study of the automation of the process of selecting the direction of movement of the elevator car was carried out. This process is an important component of the efficient operation of any building with a large number of floors, as it ensures fast and safe transportation of passengers to their destinations. This work is devoted to the development of a simulation model for studying algorithms for automating the process of selecting the direction of movement of the elevator car in order to select the system that has the highest performance and the shortest waiting time. The model includes the ability to change various parameters, such as the number of elevators, elevator capacities, load, simulation time, etc.

Key words: model, elevator, simulation, direction selection, elevator cabin, modeling a discrete event, Python.