

УДК 004.94

ПОСТАНОВКА ИМИТАЦИОННЫХ ЭКСПЕРИМЕНТОВ ПО СРАВНЕНИЮ СТРАТЕГИЙ УПРАВЛЕНИЯ ЛИФТАМИ В МНОГОЭТАЖНОМ ЗДАНИИ

П.Л. Чечет¹, А.В. Воруев¹, Е.А. Левчук²

¹Гомельский государственный университет им. Ф. Скорины

²Белорусский торгово-экономический университет потребительской кооперации, Гомель

DESIGN OF SIMULATION EXPERIMENTS FOR COMPARING ELEVATORS CONTROL STRATEGIES IN A HIGH-RISE BUILDING

P.L. Chechat¹, A.V. Varuyeu¹, A.A. Liauchuk²

¹F. Scorina Gomel State University

²Belarusian Trade and Economics University of Consumer Cooperatives, Gomel

Рассматривается постановка имитационных экспериментов для сравнения стратегий управления лифтами в многоэтажном здании. Параметрами модели являются алгоритмы управления, а не значения отдельных параметров. Предлагается шаблон проектирования симулятора, позволяющий как тривиально добавлять стратегии управления, так и создавать новые стратегии управления на основе уже разработанных.

Ключевые слова: стратегия управления лифтами, симулятор, имитационный эксперимент, сравнение альтернатив.

The design of simulation experiments for comparing elevators control strategies in a high-rise building is considered. Model parameters are control algorithms, not the values of individual parameters. A design pattern is proposed for the simulator that allows both trivially to add control strategies and create new control strategies based on already developed ones.

Keywords: elevators control strategy, simulator, simulation experiment, comparison of alternatives.

Введение

Системы управления лифтами являются программно-аппаратными комплексами. Они различаются по ряду признаков, например, типу командоаппарата и его расположению; количеству лифтов, управляемых по единой схеме; очередности обработки сигналов вызова и старта свободной кабины и т. п. При разработке системы управления лифтами обычно на первом этапе инженеры создают аппаратную базу, а затем занимаются отладкой алгоритмов [1]. В данной статье предлагается обратный подход. На первом этапе создается имитационная модель системы управления лифтами, позволяющая варьировать алгоритмическую часть системы. На следующем этапе инженеры-электронщики должны разработать аппаратную базу под один или несколько эффективных алгоритмов.

Несмотря на относительную доступность использования систем математического моделирования, в данном случае оправдана разработка программы имитационной модели, т. е. симулятора, на универсальном языке программирования. При всей своей сложности, такой подход обладает неоспоримыми преимуществами: более высокая эффективность работы симулятора, возможность использования современных библиотек для вычислений и конвертирования данных, использование возможностей многопоточного выполнения.

В данной статье рассматривается разработка и использование имитационной модели движения лифтов в многоэтажном здании на языке программирования Java. Одно из преимуществ приложения, работающего под управлением виртуальной машины Java, – это возможность работы с несколькими потоками. Виртуальная машина Java оптимизирована для многопроцессорных систем и может управлять большим количеством потоков.

1 Концептуальная модель

Объектом моделирования является многоэтажное здание с системой лифтов, предназначенных для перевозки людей. Здание имеет классическую структуру с входом на первом этаже, перемещение пассажиров между этажами осуществляется с использованием фиксированного числа взаимосвязанных лифтов. Алгоритм управления каждым лифтом имеет доступ к информации о состоянии всех лифтов в здании.

Поток посетителей для каждого этажа здания может быть задан с учётом того, что интервал времени между посетителями полагаем распределённым по закону Пуассона с параметром λ_{ij} , где i – номер исходного этажа, j – номер целевого этажа, а t – номер временного интервала. Это позволяет моделировать офисные здания, где в начале рабочего дня посетители приходят на первый этаж, в течение дня перемещаются

между этажами, а в конце дня покидают здание. Также существует возможность моделировать жилые здания, где посетители генерируются на всех этажах здания, моделируя таким образом движения жильцов, выходящих из своих квартир.

Количество этажей в здании, количество лифтов, скорость их перемещения и вместимость задаются параметрами программы имитационной модели. В зависимости от решаемой задачи это могут быть как параметры, так и переменные имитационной модели (варьируемые исследователем значения или значения, зависящие от свойств реального объекта моделирования).

2 Проектирование симулятора

Симулятор позволяет ставить имитационные эксперименты с моделью системы управления лифтами и обладает следующими функциональными возможностями обмена данными с корпоративной информационной системой [2]:

- загрузка настроек конфигурации для генерации начального состояния имитационной модели;
- загрузка входной информации для реализации имитационного эксперимента;
- предварительный просмотр конфигурации до запуска имитационного эксперимента;
- мониторинг откликов имитационной модели во время выполнения;
- вывод результатов в файл в необходимом формате;
- оперативный вывод результатов имитационного эксперимента, что позволяет исследователю принять решение о продолжении или об окончании имитационного эксперимента.

На рисунке 2.1 изображена развернутая диаграмма последовательностей событий во время работы с симулятором.

Настройка начальной конфигурации. Основной поток событий заключается в задании начальной конфигурации. Альтернативными событиями являются отмена ввода конфигурации

или ввод ошибочной конфигурации. Симулятор в этом случае предлагает исправить неправильную конфигурацию или завершает работу приложения с выдачей соответствующего сообщения об ошибке.

Последовательность мониторинга состояния имитационной модели. Основной поток событий заключается в том, что после подтверждения конфигурации симулятор допускает клиента к мониторингу откликов модели и пользователь может анализировать данные. Альтернативным событием является возможность отмены мониторинга откликов имитационной модели. В этом случае работа приложения завершается.

Последовательность получения результатов имитационного эксперимента. Основной поток событий заключается в том, что после мониторинга результатов клиенту необходимо получить, а после получения – сохранить результаты имитационного эксперимента. Альтернативным событием является ошибка обработки, что обычно бывает при ошибках прогона модели. Приложение в этом случае предложит перезапустить имитационный эксперимент.

Диаграмма потоков данных разработанного симулятора управления лифтами многоэтажного здания представлена на рисунке 2.2.

3 Компоненты симулятора

Симулятор включает в себя набор взаимодействующих компонентов.

1. Информационные классы – специальные классы, которые берут на себя роль хранения информации в зависимости от их объявления. В качестве информационных классов определены следующие классы: Passenger, Floor, Elevator, Building. Для организации взаимодействия динамических потоков имитационной модели был реализован шаблон синхронизации транзактов [3].

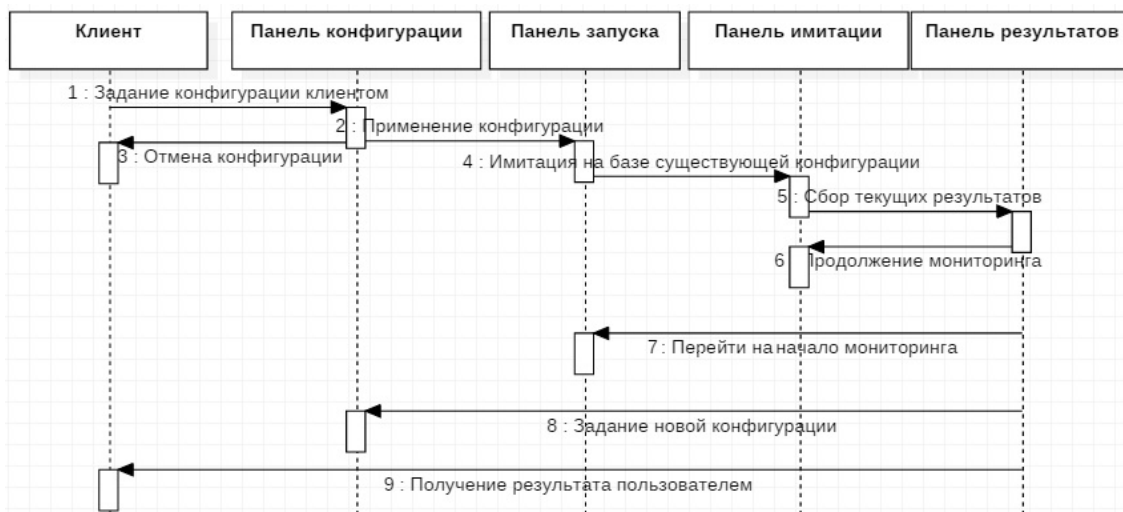


Рисунок 2.1 – Диаграмма последовательностей имитационной модели

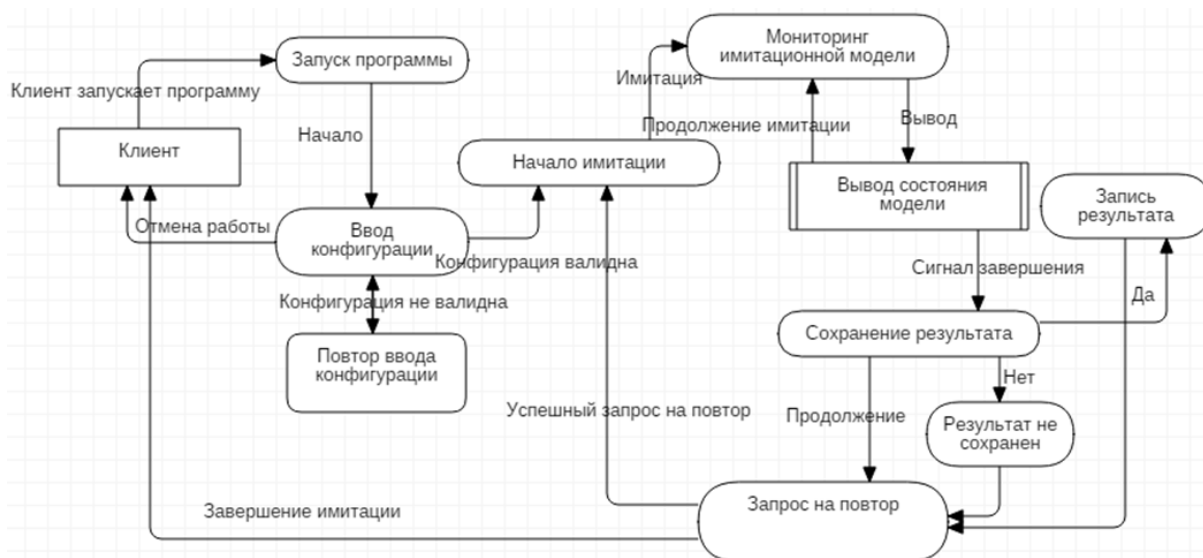


Рисунок 2.2 – Диаграмма потоков данных

2. Компоненты загрузки информации – компоненты, которые берут на себя роль загрузки начальной конфигурации из файла и последующей ее передачи в компоновщик имитационного эксперимента. В качестве компонентов загрузки определен семантический интерфейс Loader и его реализации – PropertyLoader и XMLConfigLoader.

3. Компонент для сборки имитационной модели. Данный компонент выполняет роль порождения выполняемых компонентов в имитационной модели. В качестве входной информации компонент принимает информацию из загрузчика и на выходе поставляет созданные компоненты, которые будут принимать участие в моделировании. В качестве этого компонента определен класс ElevatorTaskBuilder.

4. Компоненты исполнения модели – компоненты, которые являются управляющими в данной модели. Они несут в себе роль исполнения – выполняют определенный алгоритм работы модели. В качестве таких компонентов определены классы ElevatorController, ElevatorTask, TransportationTask и Runner.

5. Компонент валидации системы. Этот компонент выполняет роль проверки компонентов имитационной модели на корректность выполнения (верификация модели). С этой ролью справляется класс Validator.

6. Компонент регистрации событий – элемент в имитационной модели, отвечающий за вывод информации о состоянии модели. Данный компонент регистрирует события на панель выполнения и в файл. В программе имитационной модели представлен класс AppLogger.

На рисунке 3.1 показано взаимодействие компонентов в имитационной модели.

Особенностью разработанного симулятора является возможность варьирования не только параметров, но и алгоритмов управления лифтами. Для этого алгоритмы вынесены в отдельные подклассы суперкласса AbstractLiftStrategy (рисунок 3.2).

Подобная организация структуры имитационной модели позволяет варьировать стратегии обслуживания посетителей здания не только варьированием значений переменных имитационной модели, а гораздо шире: путём выбора различных стратегий функционирования лифтов, задаваемых на уровне алгоритмов управления, что, безусловно, является более гибким, чем традиционное изменение значений параметров имитационной модели.

4 Описание имитационного эксперимента

Для проведения имитационного эксперимента по сравнению эффективности нескольких стратегий обслуживания посетителей многоэтажного здания были реализованы три различных алгоритма функционирования лифтов. Каждый алгоритм реализован в программе имитационной модели в одном из подклассов суперкласса AbstractLiftStrategy.

Первая стратегия представлена классом SpeedLiftStrategy. Идея данной стратегии заключается в максимально быстром обслуживании посетителей: для перемещения пассажира на этаж вызова отправляется ближайший по времени доступа лифт со свободным местом из тех лифтов, которые потенциально могут выполнить перевозку посетителя в нужном направлении (рисунок 4.1). В этой стратегии удобство посетителей поставлено выше загрузки лифта и оптимизации потребления энергии.

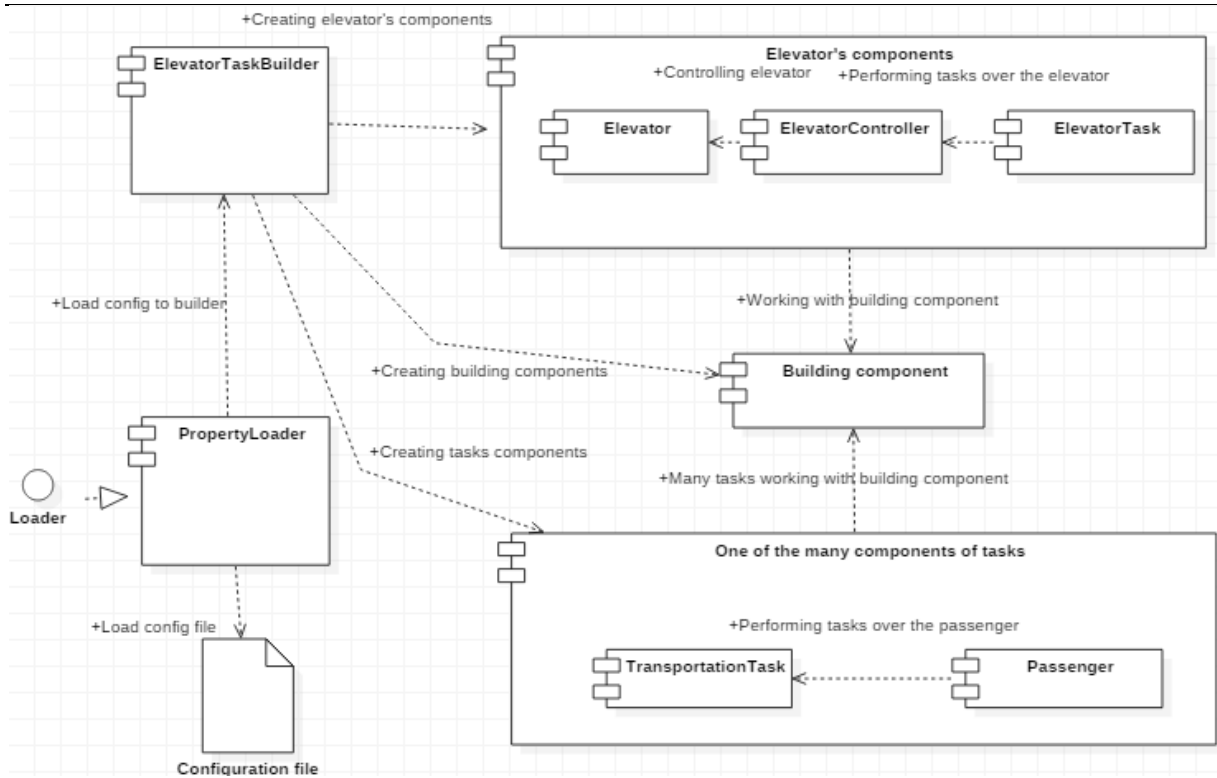


Рисунок 3.1 – Взаимодействие компонентов имитационной модели

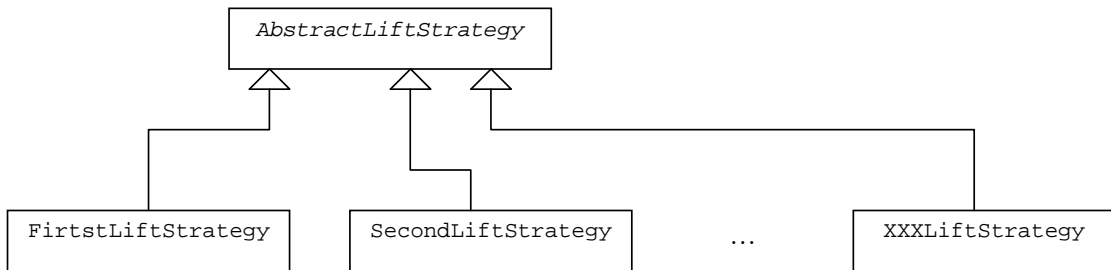


Рисунок 3.2 – Обобщенная диаграмма классов стратегий

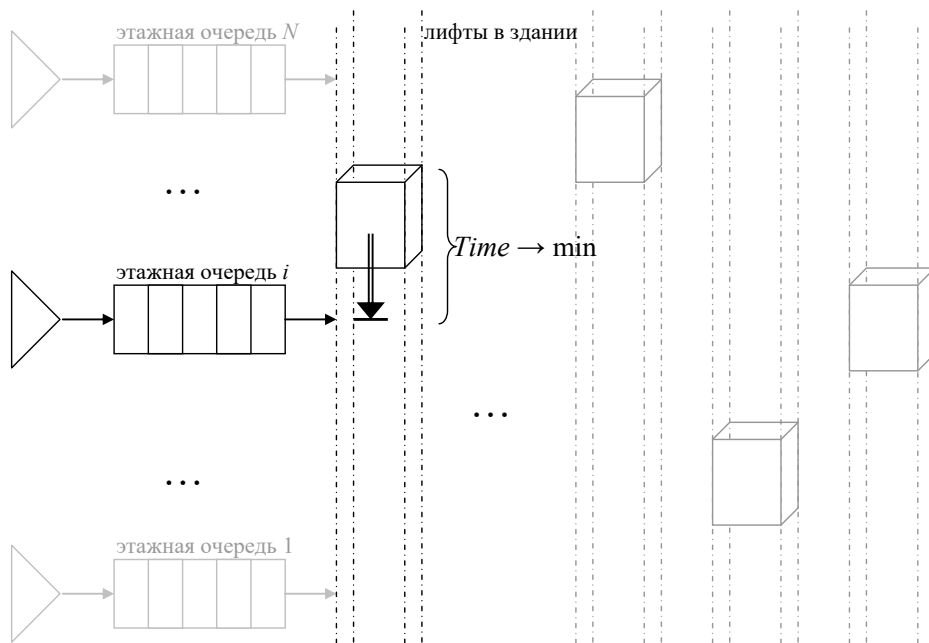


Рисунок 4.1 – Скоростная стратегия перевозки посетителей

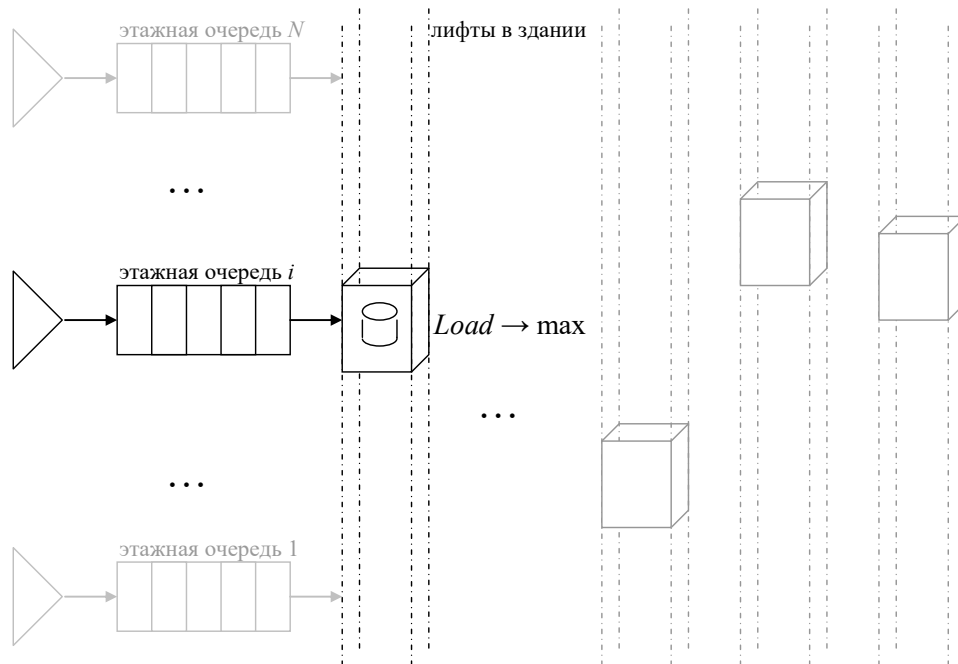


Рисунок 4.2 – Загрузочная стратегия перевозки посетителей

Вторая стратегия представлена классом `LoadLiftStrategy`. Эта стратегия ориентирована на максимальную загрузку лифтов для снижения расхода потреблённой лифтами энергии. В этом случае выбор лифта осуществляется так, чтобы обеспечить максимальную его заполняемость посетителями (рисунок 4.2), т. е. для обслуживания отправляется лифт со свободным местом, уже задействованный для перевозки посетителей. В целях недопущения возрастания времени ожидания лифта посетителем, введён лимит τ_{\max} , по истечении которого для перевозки пассажира может быть задействован даже пустой лифт.

Третья стратегия перевозки посетителей представлена классом `FloatLiftStrategy`. Она является разновидностью стратегии максимальной загрузки лифта. Её отличие от стратегии, представленной классом `LoadLiftStrategy`, заключается в том, что лимит времени ожидания посетителем лифта не является постоянным, а изменяется по закону (4.1):

$$\tau = \tau_{\max} \frac{l_{\max} - l}{l_{\max}}, \quad (4.1)$$

где τ_{\max} – задаваемый исследователем максимальный лимит времени ожидания лифта посетителем, l_{\max} – максимально возможная длина очереди на этаже, l – среднее значение длины очереди к лифту по всем этажам.

Как видно из выражения (4.1), с ростом очередей на этажах время, через которое для посетителя будет отправлен даже свободный лифт, снижается вплоть до нулевого при максимальной длине очереди. Эта стратегия обслуживания

ожидаемо должна обеспечить более качественное (более быстрое, с меньшим временем ожидания) обслуживание посетителей здания, чем загрузочная стратегия, но при этом обеспечить более высокий коэффициент загрузки лифтов, чем скоростная стратегия обслуживания посетителей здания.

Иерархия классов стратегий в программе имитационной модели представлена на рисунке 4.3.

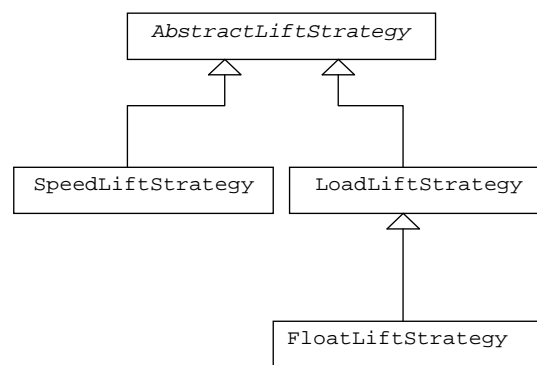


Рисунок 4.3 – Диаграмма классов стратегий обслуживания

Для проверки выдвинутой гипотезы об эффективности третьей стратегии, т. е. с изменяемым лимитом времени ожидания, были проведены имитационные эксперименты для каждой из трёх описанных выше стратегий перевозки посетителей здания имеющимися в нём лифтами.

Результаты имитационного эксперимента полученные в процессе моделирования представлены в таблице 4.1.

Таблица 4.1 – Результаты имитационного эксперимента

Показатель	Стратегии обслуживания		
	Speed	Load	Float
Средняя длина очереди к лифту на этаже (по всем этажам здания)	1,3	7,0	4,2
Среднее время ожидания лифта посетителем, с	32,5	97,2	45,5
Изменение загрузки (коэффициента, характеризующего заполнение лифта посетителями)	0,38	0,7	0,53

Как видно из таблицы, различные стратегии обслуживания приводят к изменению значения загрузки (наполняемости) лифтов.

Для удобства сравнения рассмотренных стратегий обслуживания нормируем значения коэффициента загрузки лифтов их максимальным значением.

Два оставшихся отклика: среднюю длину очереди и среднее время ожидания, преобразуем выражением (4.2):

$$x_i^* = \frac{\max x_i - x_i}{\max x_i - \min x_i}, \quad (4.2)$$

где i – номер стратегии от 1 до 3.

После преобразования значений максимальное значение каждого отклика характеризует большую эффективность. То есть предпочтительным для функционирования системы лифтов в здании является максимальная загрузка и минимальные очередь и время ожидания. Гистограмма преобразованных значений откликов с разбивкой по стратегиям обслуживания выглядит следующим образом (рисунок 4.4).

Из гистограммы на рисунке 4.4 видно, что стратегии, нацеленные исключительно на скорость обслуживания и хорошую загрузку (использование) лифтов, обеспечивают хорошие показатели лишь по тем характеристикам системы, на которые они изначально нацелены. Между тем, остальные характеристики при этом неизбежно «проседают».

Поэтому целесообразным выглядит применение промежуточной, третьей, стратегии, которая варьирует время максимального ожидания t

в зависимости от текущего размера l средней очереди к лифтам.

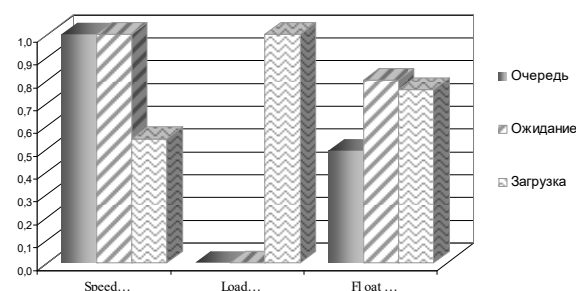


Рисунок 4.4 – Сравнительная характеристика различных стратегий

Заключение

Несмотря на условный характер принятых в имитационной модели управления лифтами в многоэтажном здании допущений по рассмотренному в статье шаблону проектирования можно провести сравнение любых произвольных стратегий управления, которые могут быть представлены в виде некоторого разумного алгоритма. Использование возможностей наследования и полиморфизма в коде программы имитационной модели делает возможным как тривиальное добавление совершенно новых стратегий управления, так и создание новых стратегий управления на основе уже разработанных.

ЛИТЕРАТУРА

1. *Elevator* – *Wikipedia* [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/Elevator>. – Дата доступа: 27.12.2018.
2. Левчук, В.Д. Реализация обмена данных между имитационной моделью и корпоративной информационной системой / В.Д. Левчук, Е.А. Левчук // Известия Гомельского государственного университета им. Ф. Скорины. – 2010. – № 5 (62). – С. 115–118.
3. Левчук, В.Д. Реализация шаблона синхронизации транзактов в системном модуле симулятора MICIC4 / В.Д. Левчук, Е.А. Левчук // Известия Гомельского государственного университета им. Ф. Скорины. – 2013. – № 6 (81). – С. 130–133.

Поступила в редакцию 18.01.19.