



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. Ломоносова



Факультет вычислительной математики и кибернетики

Кафедра автоматизации систем вычислительных комплексов

Лаборатория вычислительных комплексов

**Курс "Имитационное моделирование в исследовании и разработке
информационных систем"**

ЗАДАНИЕ № 2

«Моделирование работы лифтов 2-го учебного корпуса и потока пассажиров»

ОТЧЕТ

о выполненном задании

Выполнил: Аграновский Михаил Леонидович,
студент 321 учебной группы

Москва, 2016 г.

Table of Contents

1	Постановка задачи	- 3 -
1.1	Исходные данные.....	- 3 -
1.2	Цели.....	- 3 -
2	Конфигурация системы программирования.....	- 3 -
3	Описание реализации.....	- 3 -
3.1	Задержки.....	- 4 -
4	Результаты работы имитационной модели.....	- 5 -
4.1	Базовая версия	- 5 -
4.2	Изменим загруженность.....	- 5 -
4.2.1	Слабая нагрузка (время выбора студентом этажа назначения = 0..100)	- 5 -
4.2.2	Высокая нагрузка (время выбора студентом этажа назначения = 0)	- 5 -
4.2.3	Все пассажиры стартуют с 0 этажа.....	- 5 -
4.2.4	Идеальная ситуация: циркуляция пассажиров между 2 этажами.....	- 5 -
5	Выводы.....	- 6 -

1 Постановка задачи

1.1 Исходные данные

Дано описание исследуемой системы на естественном языке (возможно, неполное). Заданы цели исследования системы (которые также могут быть уточнены или дополнены студентом самостоятельно или в ходе диалога с преподавателем).

1.2 Цели

- уточнить исходные данные;
- построить концептуальную модель (указать, какие упрощающие предположения принимаются, описать структуру системы, взаимодействия между компонентами);
- построить имитационную модель в системе моделирования по выбору студента;
- проверить правильность построения модели и адекватность;
- провести эксперименты с моделью в соответствии с целью исследования;
- уточнить модель или цель исследования самостоятельно или по указанию преподавателя;
- сделать выводы и составить отчёт о работе.

В частности, необходимо собрать сведения о функционировании лифтов и построить модель, отражающую:

- движение лифта между этажами;
- реакцию на кнопки вызова;
- открытие и закрытие дверей;
- поведение пассажира (подход к лифту, занятие очереди в лифт, выход на нужном этаже)
- (дополнить по усмотрению студента)

В ходе исследования необходимо воспроизвести различные сценарии использования лифта (лифтов в холле и даже в здании в целом), например: перемещение одиночных пассажиров, массовые перемещения (например, на основе анализа расписания занятий). Определить задержку в очереди, среднее время движения пассажира на этаж, загрузку лифта, и т. д. (по усмотрению студентов и предложению преподавателя)

Следует сравнить различные алгоритмы планирования движения лифта (лифтов в одном холле): «базовый этаж», приоритет движения вверх или вниз — с точки зрения влияния на характеристики обслуживания пассажиров.

2 Конфигурация системы программирования

В качестве системы моделирования выбран фреймворк SimPy для языка программирования Python (SimPy 3.0.10, Python 3.5)

Модель выполнялась на Ubuntu 16.04 x86-64 на Intel Core i5 6300HQ, 24GB RAM.

3 Описание реализации

Ниже рассматривается разработанная имитационная модель системы лифтов. В скобках указаны имена классов или переменных из кода реализации.

В рамках модели рассматриваются следующие процессы:

- лифт (Elevator)
- пассажир (Person)

Лифтами управляет *лифтовая система (ElevatorSystem)*, пассажирами – великий рандом.

Модель основана на следующих упрощениях и абстракциях:

- **2 ГУМ** – 8 этажное здание с одним лифтовым холлом. При старте модели студенты (NUM_PEOPLE=500 человек) распределены по этажам равномерно, востребованность этажей также равномерна. Вместительность этажей бесконечна. Лестницами студенты не пользуются.
 - **Лифт**
 - Вместительность лифтов одинакова и составляет CAPACITY=10 человек на лифт. Грузоподъемность и надежность лифтов бесконечны.
 - Лифт может находиться в одном из 3 состояний: “vacant”, “go-to-the-first-person” и “on-the-go”.
 - Изначально все лифты стоят на 0 этаже в состоянии “vacant”. При достижении этажа назначения лифт принимает состояние “vacant” и остается ждать следующего вызова.
 - **Лифтовый холл** рассматривается как интерфейс к одной **лифтовой системе** с 6 лифтами.
 - Вызов лифта осуществляется по нажатию на идеальную, срабатывающую с первого раза кнопку. При этом в каждом лифтовом холле расположены панели с номерами обслуживаемых этажей. Студенты поочередно выбирают нужные им этажи, а далее встают в очередь ожидать *своих* лифтов.
 - При каждом нажатии на панель лифтовая система принимает решение об отправке лифта к месту вызова. Если все лифты заняты, система отказывает студенту. Он повторит попытку позже. Итак, как лифтовая система выбирает, какой лифт отправить на вызов?
 - Наивысший приоритет имеет лифт, уже движущийся к этажу назначения через этаж дислокации пассажира. Т.е. поддерживается подбор пассажиров по пути следования.
 - Меньший приоритет имеет ближайший к нужному этажу лифт, находящийся в ожидании.
- У автора есть предположение, что подобная неоптимальная для пассажиров система как раз и применяется во многих домах. Возможно, из-за простоты реализации и минимизации износа лифтов.
- **Студент** принимает решение о выборе этажа назначения за 0..DECISION_MAKING_TIME_MAX секунд (здесь и далее «тики» времени имитационной модели будем для упрощения называть секундами).

3.1 Задержки

Время достижения студентом этажа назначения зависит от следующих задержек:

- Время ожидания лифта.
- Время передвижения. Зависит от:
 - OPEN_TIMEOUT – время открытия (или закрытия) дверей лифта
 - COME_IN_TIMEOUT – время продвижения студента в лифт (из лифта)
 - непосредственно время движения в лифте

Все эти задержки зависят от следующих параметров системы:

- CAPACITY – вместительность лифта
- NUM_ELEVATORS – число лифтов
- NUM_FLOORS – число этажей
- NUM_PEOPLE – число студентов
- SPEED – скорость движения лифта (этажей в секунду)
- WAIT_TIMEOUT -- время обновления внутреннего состояния лифта (деталь реализации)

4 Результаты работы имитационной модели

Рассмотрим результаты выполнения модели. Проведем замеры для описанной выше базовой версии, а также посмотрим, как будут менять результаты при внесении изменений в модель. Результаты приведены с усреднением.

4.1 Базовая версия

mean_sum_time	mean_waiting_time	mean_moving_time	mean_people_number
263.35	185.84	77.51	7.81

4.2 Изменим загруженность

4.2.1 Слабая нагрузка (время выбора студентом этажа назначения = 0..100)

mean_sum_time	mean_waiting_time	mean_moving_time	mean_people_number
226.38	162.62	63.76	7.93

4.2.2 Высокая нагрузка (время выбора студентом этажа назначения = 0)

mean_sum_time	mean_waiting_time	mean_moving_time	mean_people_number
295.01	214.44	80.57	8.62

Вывод: повышение нагрузки приводит к росту времени ожидания

4.2.3 Все пассажиры стартуют с 0 этажа

mean_sum_time	mean_waiting_time	mean_moving_time	mean_people_number
130.42	112.76	17.89	9.25

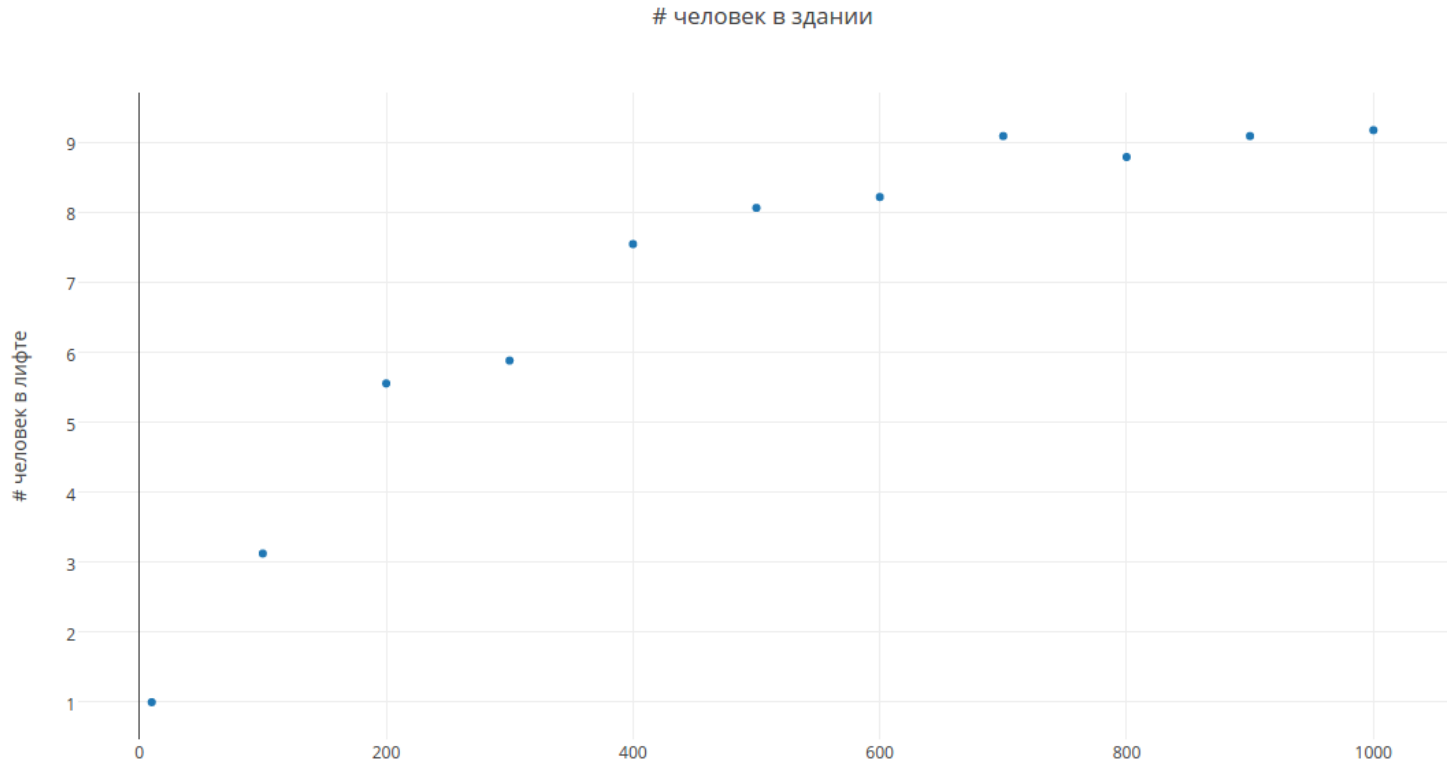
Вывод: лифт едет долго из-за задержек на подбор пассажиров. Если все пассажиры столпились на 1 этаже, система будет работать быстрее, а время в пути (в лифте) будет минимальным!

4.2.4 Идеальная ситуация: циркуляция пассажиров между 2 этажами

mean_sum_time	mean_waiting_time	mean_moving_time	mean_people_number
116.63	101.65	14.98	9.26

Примечание: большие абсолютные значения времени ожидания связаны с большим временем ожидания перед вторичным запросом лифта. Неприятная особенность реализации. На относительное изменение значений она влиять не должна.

Ниже приведена зависимость заполнения лифтов исходя из числа человек в здании. Видно, что начиная с некоторого числа человек, лифты загружены практически полностью (>90%).



5 Выводы

- Уточнены исходные данные, построена концептуальная модель (упрощения и абстракции обозначены).
- По концептуальной модели построена имитационная модель в системе моделирования SimPy.
- Правильность и адекватность проверены методом «пристального взгляда», а также на реакцию к изменению входных данных. «Боевая» проверка затруднительна, так как полноценная имитация системы лифтов 2 ГУМ (и поведения пассажиров!), которую можно было бы сравнивать с действующей системой, по сложности выходит за рамки данного задания.
- Над построенной моделью проведены эксперименты. Были рассмотрены случаи различной нагрузки на системы. Их результаты представлены в форме таблиц и графиков.

Исходный код имитационной модели доступен в репозитории на GitHub:

https://github.com/agrml/ImitationModeling_project2