**Домашняя работа по предмету   
«Теория массового обслуживания»**

**Студент: Королев Дмитрий**

**Группа: БПИ-173**

**Ссылка:** <https://github.com/DeagleGross/Queueing-Theory-Simulation>

*Программный код:* [*https://github.com/DeagleGross/Queueing-Theory-Simulation/blob/master/Simulation.ipynb*](https://github.com/DeagleGross/Queueing-Theory-Simulation/blob/master/Simulation.ipynb)

Домашняя работа выложена на ресурс github.com. Вы можете просмотреть всю реализацию с небольшими комментариями там *(файл Simulation.ipynb)*.

**Задание**

Для домашнего задания я придумал следующую задачу:

*В цехе находятся 2 станка.*

*В цех поступают детали в среднем через каждые 35 минут (здесь и далее подразумевается, что все времена распределены экспоненциально).*

*Обработка детали производится на одном из двух станков.*

*Первый обрабатывает деталь в среднем за 30 минут, при этом отвергает 10 % деталей, считая их бракованными.*

*Второй обрабатывает деталь в среднем за 50 минут, при этом отвергает 20 % деталей, считая их бракованными.*

*Если деталь была забракована в первый раз, то ее попробуют обработать еще раз после всех деталей, стоящих в очереди (будь то в очереди на первичную или вторичную обработку)*

*Детали, которые были отвергнуты 2 раза, считаются отходами и не обслуживаются цехом.*

*Детали, которые были отвергнуты на любом станке передаются на повторную обработку на второй станок.*

*Моделирование проводится на 500 деталей.*

*Каждый из станков имеет цену обработки (или попытки обработки) детали.*

*Для первого станка цена обработки составляет 5$. Для второго - 2$.*

*Цена готового изделия составляет 12 долларов.*

*Предпринимателю, который владеет цехом, поступило предложение об обмене второго станка. Новый станок имеет цену обработки в 3 $. Время обработки - 40 минут в среднем. Вероятность браковки - 10%.*

*Задача: выяснить какой из станков (старый 2-й или новый) будет более выгодный?*

*а) прибыль*

*б) прибыль \ сек*

Я принял решение реализовать симуляцию на языке Python.

**Тонкости реализации**

В стандартной библиотеке языка Python я нашел функцию, которая возвращает экспоненциально-распределенные величины в зависимости от переданной «лямбды» (функция random.expovariate(lambda)). Я задал такие начальные значения исходя из условия задачи:

arrival\_rate = 35 *# min*

first\_machine\_speed = 30 *# min*

second\_machine\_speed = 50 *# min*

third\_machine\_speed = 40 *# min*

И запускал вышеописанную функцию, передавая в качестве значения

В данном случае *rate* – частота прибытия деталей в цех (в мин) или частота обработки детали машиной (в мин). То есть при генерации времени я вызывал функцию

Генерация времени осуществлялась таким алгоритмом:

1. Пусть *ti* – время, когда в цех поступил *i-ая* деталь.
2. Пока в цех не поступит 500 деталей (из условия задачи) или, иначе говоря, пока *i* не достигнет 500, будем генерировать *t­i* следующим образом:

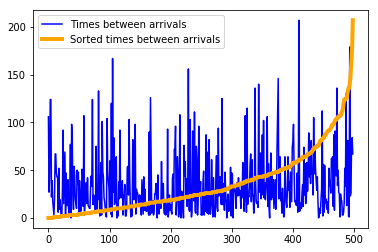
Очевидно,

Затем я сформировал массив значений между временами прихода для того, чтобы проверить, что я верно сгенерировал времена:

Посчитал среднее время для *y* и получил:

E(x) of times between arrivals

34.904



После такой проверки понятно, что условие «детали приходят в среднем через каждые 35 минут» выполняется!

**Сама симуляция**

Наконец, я запустил симуляцию.

Выбор работающей машины производился с помощью обычной функции, возвращающей числа от 0 до 1. Если она возвращала значение больше 0.5, то выбиралась не первая машина.

Для каждой детали я хранил значение ее статуса (например, «arrive», «finished», «repeat» и «garbage»). Сначала обрабатывались все прибытия 500-ста деталей, затем обрабатывались все повторения (которые забраковала первая или вторая машина) только второй машиной.

На каждом шаге производился подсчет количества деталей разных типов.

**Результаты симуляции**

Далее я приведу самый типичный и интересный результат. Конечно, все симуляции будут давать разные значения.

*Симуляция с машиной № 1 и № 2:*

Rejects: first [ 27 ], second [ 50 ]

Successes: first [ 232 ], second [ 254 ]

Garbage items: 14

Finishes: 486

Repeats: 77

Time spent: 20741 minutes

Money earned: 5832 $

Money spent: 1903 $

Profit money: 3929 $

Profit($)\min: 0.1894315606769201

*Симуляция с машиной № 1 и № 3:*

Rejects: first [ 26 ], second [ 28 ]

Successes: first [ 238 ], second [ 254 ]

Garbage items: 8

Finishes: 492

Repeats: 54

Time spent: 19007 minutes

Money earned: 5904 $

Money spent: 2166 $

Profit money: 3738 $

Profit($)\min: 0.19666438680486137

Из-за того, что третья машина бракует всего 10% деталей, а вторая – 20%, то понятно, почему в второй симуляции произошло намного меньше повторений и, соответственно, получилось меньшее число отходов.

Однако, из-за того, что работа с третьей машиной на 1 доллар дороже, чем с второй, расходы в второй симуляции получились намного выше (на 263 доллара)

Прибыль при работе 1й и 2й машины получается больше, чем при работе 1й и 3й. Но, из-за того, что время работы 3-ей машины меньше, да и повторных обработок потребовалось меньше, то прибыль в секунду будет больше у 2-ого эксперимента!

Результат: если требуется достичь бОльшего дохода, то лучше использовать вторую машину. Если же время ограничено, то лучше воспользоваться третьей машиной.