МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ

КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский Технический Университет Связи И Информатики (MTUCI)»**

Кафедра «Математическая кибернетика и информационные технологии»

**Курсовая Работа**

по дисциплине «**Структуры и алгоритмы обработки данных**»

Выполнил:

студент 3 курса гр. БВТ2204

Ибрагимов Т.Э.

Москва 2024 г.

**Содержание**

[Цель 3](#_Toc184743264)

[Задача работы 3](#_Toc184743265)

[Реализация 4](#_Toc184743266)

[Визуальная часть 8](#_Toc184743273)

[Итог 10](#_Toc184743274)

[Список источников 23](#_Toc184743275)

**Цель**

Разработка расписания движения автобусов с использованием метода прямого перебора, то есть в лоб для оптимального расписания работы водителей с учетом всех ограничений и обеспечить равномерное распределение смен и соблюдений правил по отдыху.

**Задача работы**

1. Разработка оптимального расписания работы водителей автобусов

а) Создать расписание для 8 водителей, учитывая специфику их смен

б) Учитывать временные ограничения, такие как продолжительность смен и допустимые часы работы.

1. Учёт ограничений и распределение ресурсов

а) Убедиться, что каждый автобус обслуживается не более чем одним водителем в каждый час.

б) Гарантировать, что водители работают строго свои смены (8 или 12 часов подряд) без перекрытия.

1. Распределение выходных дней и смен

а) Организовать выходные для водителей с короткими сменами в субботу и воскресенье.

б) Разработать график для водителей с длинными сменами, чтобы чередовать их работу через каждые три дня.

**Реализация**

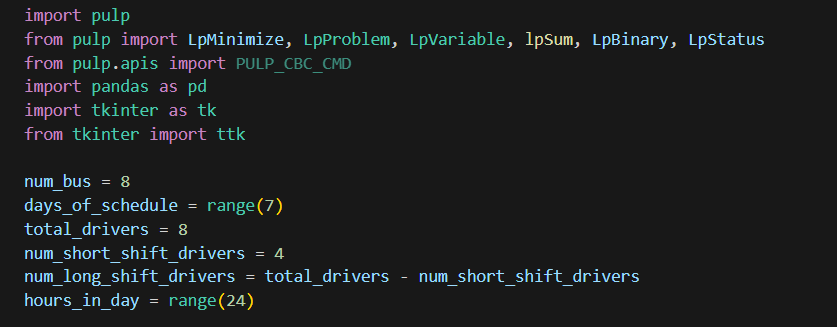
****

Рис 1 – Импорт библиотек определение требования для формирования расписания

Импортируются необходимые библиотеки:

Pulp **-** используется для создания и решения задачи оптимизации

Pandas - для формирования таблицы с расписанием

Tkinter - для создания графического интерфейса (визуализация расписания)

Создание задачи оптимизации:

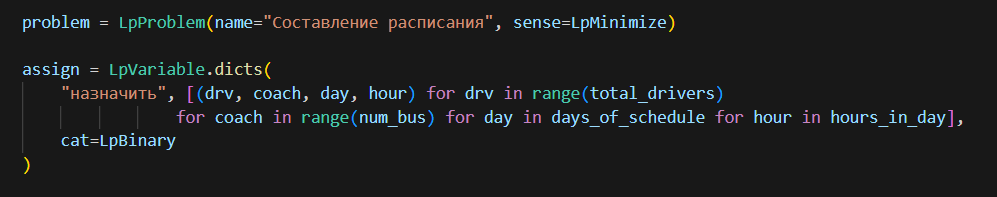


Рис 2 – Задача оптимизации

**assign** — словарь переменных. Каждая переменная имеет значение 0 или 1 (тип **LpBinary**)

Создаётся задача оптимизации, то есть problem , что определяет минимизацию времени, потраченного на решение задачи

Массив переменных assign определяет, будет ли водитель работать на автобусе в конкретный день и час

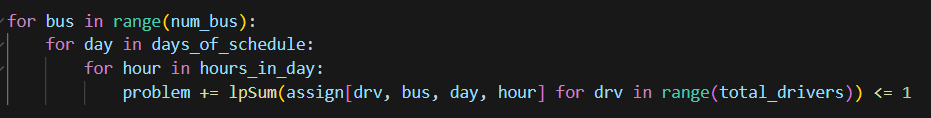


Рис 3 – Ограничение один водитель значит один автобус

Тут оно гарантирует, что на одном автобусе в один час может работать только один водитель

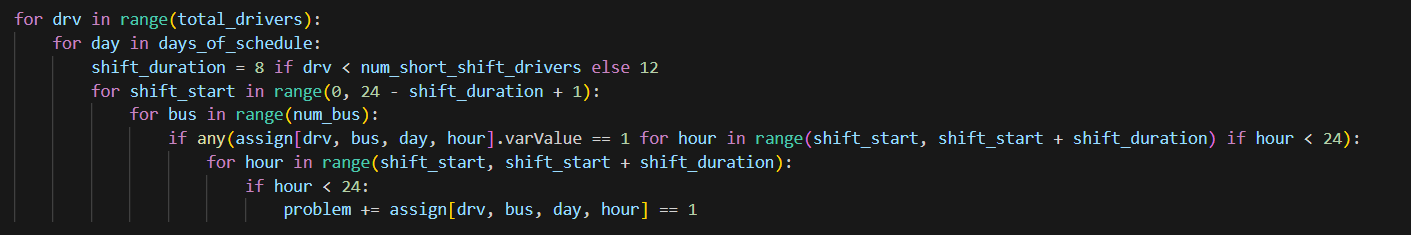


Рис 4 – Соблюдение длительности смен

У нас короткая смена – 8 часов

А для длинной – 12 часов

Здесь оно формирует ограничение на продолжительность смены и её последовательность

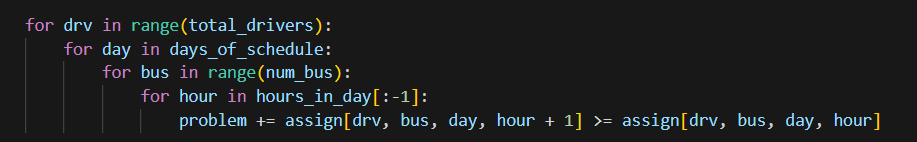


Рис 5 – Последовательность работы

Устанавливается непрерывность смены: если водитель работает в один час, то он также должен работать в следующий час, пока смена не завершится

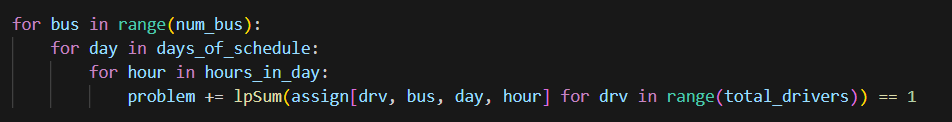
****

Рис 6 - Все автобусы заняты

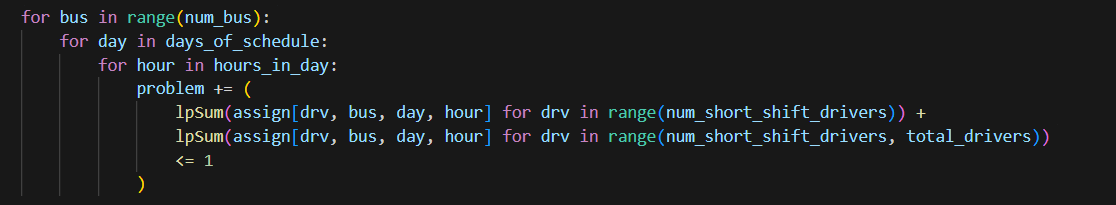
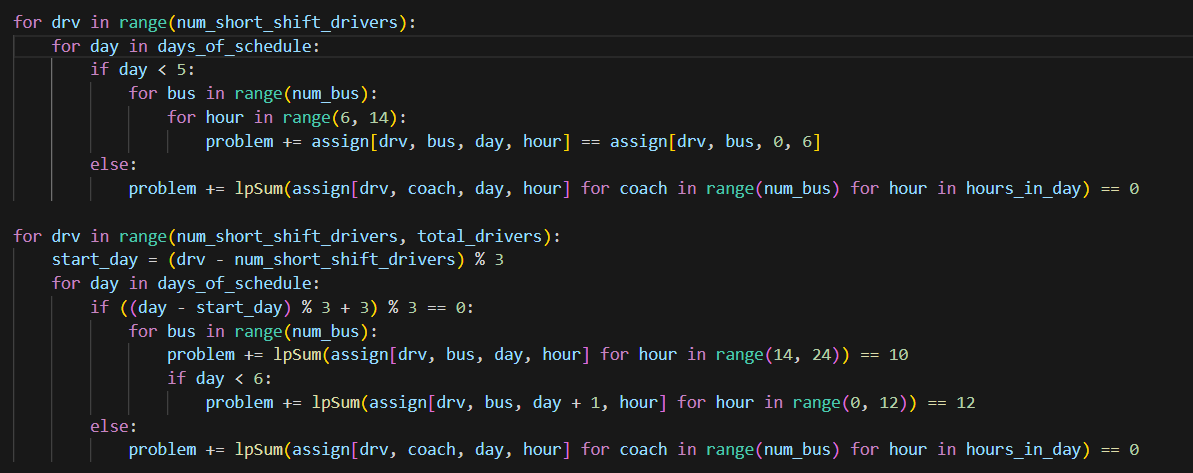


Рис 7 – Еще одно ограничение

Отвечает за **ограничение** на количество водителей, которые одновременно работают на одном автобусе в определённый момент времени



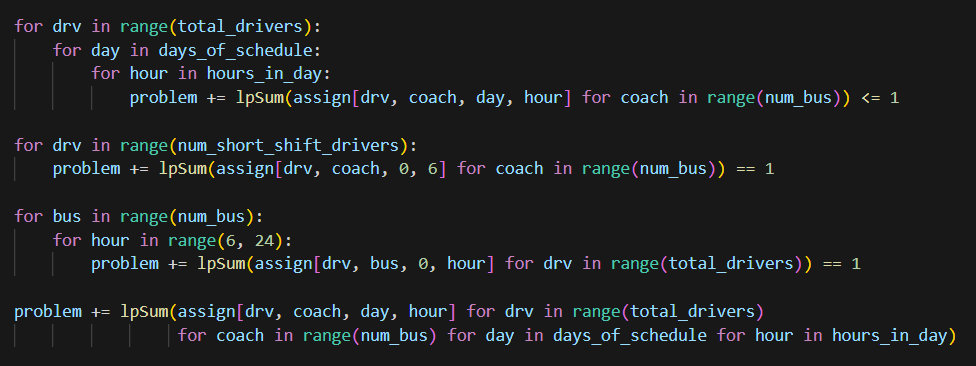


Рис 8,9 – Выходные

Здесь создаются ограничения для каждого автобуса в каждый час каждого дня.

Например, для автобуса 1 в понедельник в 10:00 код гарантирует, что за рулем не будет более одного водителя.

Затем суммируются **все короткосменные водители**, работающие в этот час на этом автобусе

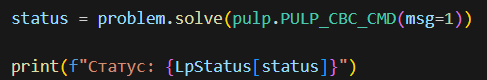


Рис. 10 – Решение задачи оптимизации

**Функция solve** запускает процесс решения задачи линейного программирования. Используется CBC - это встроенный решатель линейных программ, поддерживаемый PuLP.

msg = 1 означает вывод в консоль отладочного сообщения о процессе решения задачи. После этого возвращается статус, который показывает как завершилось решение задачи

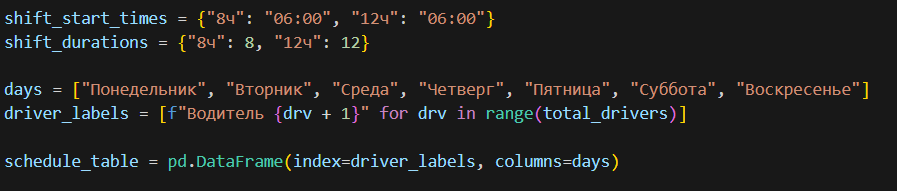
Может быть три случая:

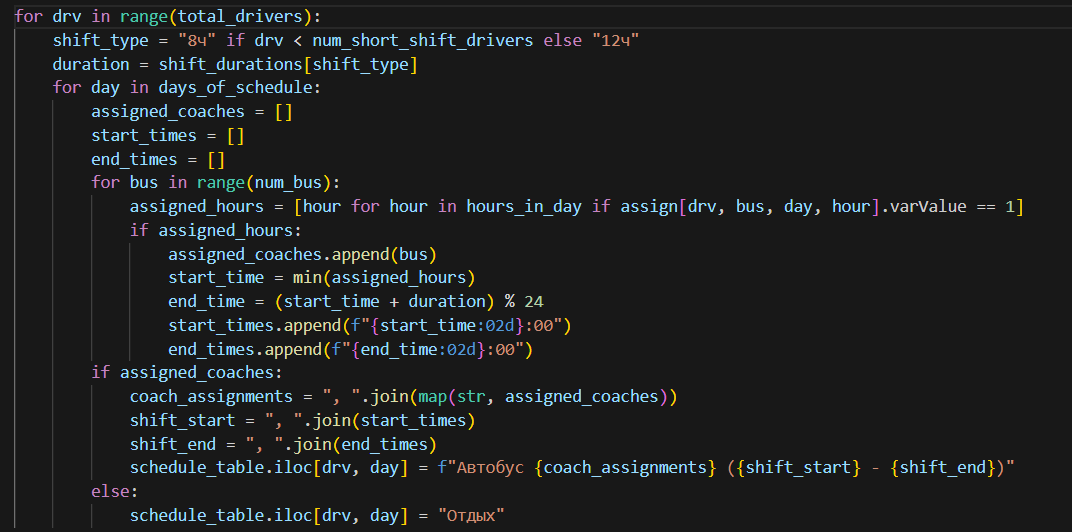
Optimal - Решение успешно найдено, и оно оптимально в рамках поставленной задачи

Infeasible - Решение невозможно из-за слишком жестких ограничений

Underfined - Решение не было найдено

Визуальная часть





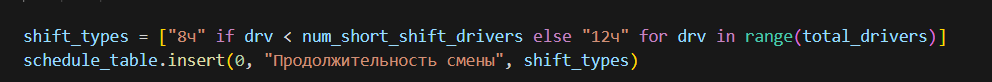


Рис 11,12,13 – Формирование таблицы расписания

То есть наши результаты конвертируются в удобный формат для визуализации



Рис 14 – Создание таблички

Создаем визуальную таблицу с нашими данные

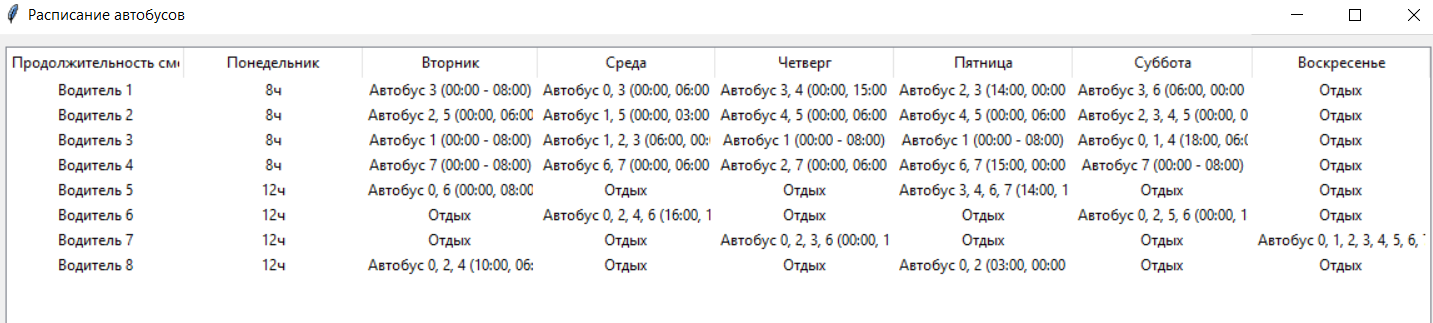


Рис 15 - Вывод

**Итог**

Итогом работы стало создание алгоритма для автоматического составления расписания водителей автобусов с учетом смен, выходных и нагрузки. Полученное решение позволяет оптимально распределить работу водителей и визуализировать расписание в таблице.

Список источников

1. В. Хилльер, Дж. Либерман. "Введение в математическое программирование и оптимизацию".
2. Д. Бертсек и Дж. Царацис. "Линейное и нелинейное программирование" (Linear and Nonlinear Programming).
3. Официальный сайт Tkinter: <https://docs.python.org/3/library/tkinter.html>
4. Python Documentation: <https://docs.python.org/3/>.
5. Видеоролики про оптимизацию на Python в ютуб