

РАЗРАБОТКА И АНАЛИЗ МОДЕЛИ СВЕТОФОРНОГО РЕГУЛИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ НЕЧЕТКОЙ ЛОГИКИ

Аннотация. В работе построена математическая модель системы светофорного регулирования. Решается задача оптимизации параметров модели с помощью методов нечеткой логики. Реализация программной части выполнена на базе языка программирования Python. Проанализирован пример последовательной связи перекрестков. Проведена оценка общей эффективности применения методов нечеткой логики в городской системе транспортных потоков.

Ключевые слова: модель светофорного регулирования, методы нечеткой логики, оптимизация движения транспортных потоков, нечеткая логика в системе светофорного регулирования, нечеткая логика, оптимизация, светофорное регулирование

Введение. Развитие городов рука об руку идет с расширением транспортной сети. Но не всегда удается создать сбалансированную дорожную карту, способную стабильно выдерживать возможные перепады загруженности автомобильных потоков. Попадая в Час Пик в очередную пробку, любой ее участник невольно задумается: «Почему нельзя избавиться от пробок?». Ответ на этот вопрос, можно найти в разных направлениях. Будь то перестройка транспортной сети с ее расширением, принудительное изменение скоростного режима или же индивидуальная настройка каждого светофора системы. Но можно ли сделать так, чтобы светофоры научились самостоятельно подстраиваться под дорожную ситуацию?

Проблема исследования. Цель работы заключается в демонстрации эффективности использования методов нечеткой логики для оптимизации системы светофорного регулирования. Для достижения поставленной цели требуется построить математическую модель системы, подобрать метод решения на базе методов нечеткой логики и провести анализ конкретного примера.

В задаче оптимизации системы светофорного регулирования требуется максимизировать число проезжающих исследуемые перекрестки автомобилей:

$$Q = \sum_{i=1}^n \sum_{l=1}^M \sum_{j=1}^u \left[\frac{T_{(i,j)}^l - \hat{t}_{(i,j)}}{k_i} \right] \cdot z_{(i,j)} \rightarrow \max.$$

Здесь использованы следующие обозначения:

Q — количество автомобилей, проехавших перекрестке за все циклы во всех доступных направлениях;

n — количество доступных к изучению перекрестков;

M — количество исследуемых циклов работы перекрестков;

u — количество доступных направлений на i -м перекрестке;

$\hat{t}_{(i,j)}$ — среднее время проезда i -го перекрестка в j -м направлении;

$T_{(i,j)}^l$ — время зеленого сигнала;

$z_{(i,j)}$ — количество полос движения на j -м направлении движения;

k_i — время проезда автомобилем средней дистанции между машинами.

Таким образом, в вышеприведенной формуле вычисляется общее количество проехавших автомобилей по всем исследуемым перекресткам и по всем циклам работы.

В модели управляемым параметром выступает время зеленого сигнала светофора. Распишем время зеленого сигнала, как систему, где время выбирается в зависимости от назначения направления движения:

$$T_{(i,j)}^k = \begin{cases} T_{(i,j)}^{k-1} + \Delta t, & \text{— время на ведущем/сонаправленном направлении;} \\ TM_i - (T_{(i,j)}^{k-1} + \Delta t), & \text{— время на оставшихся направлениях.} \end{cases}$$

Здесь

TM_i — время полного цикла на i перекрестке;

Δt — изменение времени зеленого сигнала цикла.

Ограничение изменений времени зеленого сигнала будет основано на правилах дорожного движения:

$$20 \leq \Delta t \leq TM_i - 20.$$

Материалы и методы. Вычисление изменения времени зеленого сигнала светофора основано на методе центра масс, который требует введения дополнительных параметров, представляемых в виде нечетких чисел [1]. Предполагается, что определение величины изменения происходит на основе системы правил, регулирующих направление изменения времени (табл. 1).

Таблица 1

Таблица правил регулирования движения

<i>Время сигнала, с</i>	<i>Количество машин, шт.</i>	<i>Решение</i>
малое	малое	не изменять
малое	среднее	увеличить
малое	большое	увеличить
среднее	малое	уменьшить
среднее	среднее	не изменять
среднее	большое	увеличить
большое	малое	уменьшить
большое	среднее	уменьшить
большое	большое	не изменять

Метод решения поставленной задачи заключается в корректном изменении времени зеленого сигнала светофора на каждом из доступных светофоров. Для этого определяется наиболее загруженное направление. Далее, процесс преобразования нечеткого вывода основан на алгоритме Мамдани [2]. Вначале входные данные проходят процесс фаззификации и сопоставление системе правил. Следом, выполняется этап активизации подзаключений. Последним действием выступает процесс дефаззификации, результатом которого станет величина изменения длительности зеленого сигнала светофора на следующем цикле работы. Внедрение данного алгоритма и позволит оптимизировать движение транспортных потоков.

Для реализации вычислительной части используется язык программирования Python [3]. На основе методов библиотек skFuzzy, numPy и matplotlib была написана программа, которая способна считывать входные данные перекрестков, выполнять расчеты по статистике проезда автомобилей как в статическом режиме, так и с использованием методов нечеткой логики, а также сохранять результаты исследований в разном виде [4-6]. В процессе разработки были реализованы функции считывания и генерации начальных данных разных масштабов, разработаны классы для хранения сведений об объектах перекрестков и подсчетов статистики работы перекрестков в разных режимах.

Результаты. Демонстрация результатов работы происходит на примере анализа последовательно связанных перекрестков (рис. 1). Данные начальной загруженности представлены в табл. 2.

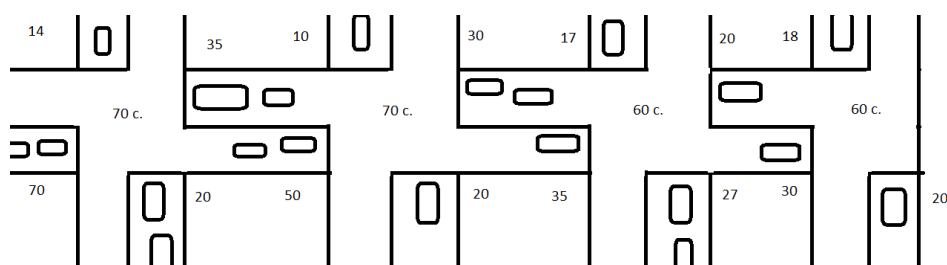


Рис. 1. Схема четырех перекрестков

Таблица 2

Таблица начальных данных загруженности перекрестков

Номер перекрестка	Направление 1, шт.	Направление 2, шт.	Направление 3, шт.	Направление 4, шт.
№ 1	70	20	35	14
№ 2	50	20	30	10
№ 3	35	27	20	17
№ 4	30	20	18	

Анализируя представленные сведения загруженности перекрестков, можно заметить общую тенденцию перегруженности на одном направлении. Данная картина достаточно популярна на улицах современных городов. Обратимся к результатам работы программы (рис. 2).

```

Информация по оптимизированному перекрёстку
Номер цикла: 1 2 3 4 5 6
Машин в начале цикла: 445 341 251 175 127 98
Проехало перекрёсток: 139 125 111 83 64 46
Машин в конце цикла 306 216 140 92 63 52
Информация по статическому перекрёстку
Номер цикла: 1 2 3 4 5 6
Машин в начале цикла: 438 348 260 205 178 165
Проехало перекрёсток: 120 118 85 57 43 40
Машин в конце цикла 318 230 175 148 135 125
Коэффициент эффективности: 1.2267818574514038

```

Рис. 2. Сравнительная характеристика системы перекрестков

Можно заметить, что число проехавших автомобилей на каждом цикле исследования больше у перекрестков, которые работают в оптимизированном режиме. Вычислим общую эффективность как отношение показателей пропускной способности оптимизированного режима к статическому. Показателем пропускной способности выступает суммарное число проехавших автомобилей каждый исследуемый перекресток за фиксированное количество циклов работы светофора. Общая эффективность находится на уровне 22%, что является достаточно хорошим результатом.

Заключение. В данной работе рассмотрена задача определения эффективности использования методов нечеткой логики в системах светофорного регулирования перекрестков. Построена математическая модель рассматриваемой системы, подобран метод решения на базе методов нечеткой логики, выполнена программная реализация расчета эффективности использования указанного метода для оптимизации движения транспортных потоков. Проанализирован пример оптимизации системы последовательно связанных перекрестков. Результатом исследования стал вывод об эффективности использования методов нечеткой логики в работе систем светофорного регулирования.

СПИСОК ЛИТЕРАТУРЫ

1. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л.А. Заде. — М.: Мир, 1976. — 161 с.
2. Mamdani E.H. An Experiment in Linguistic Synthesis with Fuzzy Logic Controller / E.H. Mamdani, S. Assilian // International Journal Man-Machine Studies. — 1975. — Vol. 7, № 1. — Pp. 1-13.
3. Руководство Python: [сайт]. — URL: <https://www.python.org/> (дата обращения: 05.05.2024). — Текст: электронный.
4. Skfuzzy Docs: [сайт]. — URL: <https://pythonhosted.org/scikit-fuzzy> (дата обращения: 05.05.2024). — Текст: электронный.
5. Классы Python: [сайт]. — URL: <https://python-scripts.com/python-class> (дата обращения: 05.05.2024). — Текст: электронный.
6. Руководство NumPy: [сайт]. — URL: <https://numpy.org/learn> (дата обращения: 05.05.2024). — Текст: электронный.