



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Разработка алгоритма управления динамической системой на основе аппроксимированного динамического программирования

*Проект НУГ № 21-04-039*

*«Динамическая оптимизация параметров контура  
управления киберфизической системы проветривания  
подземного горнодобывающего предприятия»*

- И. А. Черницин, ПИ-19-2, НИУ ВШЭ – Пермь

Задача: разработать алгоритм управления динамической системой с использованием ADP

1. Анализ существующих методов ADP
2. Определение требований к алгоритму
3. Определение особенностей для процесса проветривания
4. Выбор аппроксимации
5. Стек технологий
6. Визуализация



# Существующие методы ADP

## Прогностическое управление (стратегия ограниченных перспектив)

Предлагается сократить горизонт оптимизации и использовать в каждый момент времени (на каждом шаге) решения, основанные на краткосрочном прогнозировании изменений на некоторое число шагов вперед.

В случае дискретного момента времени  $k$  и состояния процесса проветривания  $x_k$  будем выбирать управление  $u_k$  в соответствии с функцией  $\bar{\mu}_k(x_k)$

$$\min_{u_k \in U_k(x_k)} E \left\{ g_k(x_k, u_k, w_k) + \bar{J}_{k+1}(f_k(x_k, u_k, w_k)) \right\} \quad \text{где}$$

$\bar{J}_{k+1}$  - аппроксимированное значение истинной функции затрат на энергоресурсы  $J_{k+1}$ , при  $\bar{J}_N = g_N$

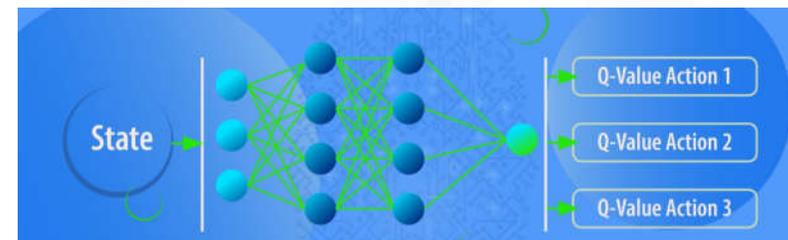


НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Существующие методы ADP

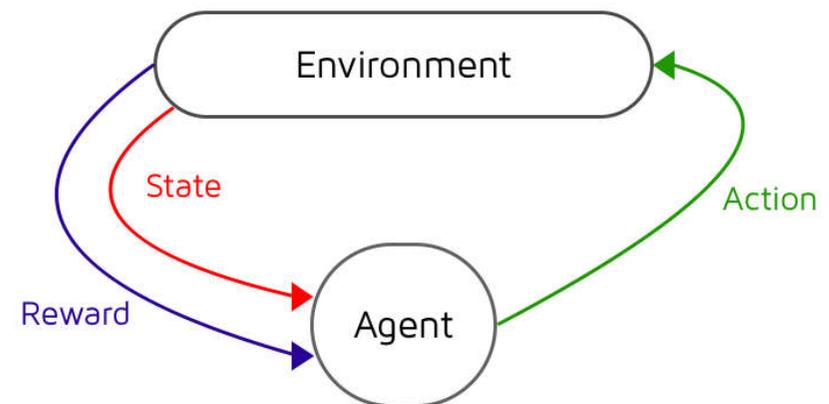
Q - Learning - алгоритм, вычисляющий значение какого либо действия в определенном состоянии.

Для любого процесса принятия решений Q - обучение находит оптимальную политику в смысле максимизации ожидаемого значения общего вознаграждения на всех без исключения последовательных шагах, начиная с текущего состояния.



## Особенности Q-Learning:

1. Q- обучение может определить оптимальную политику выбора действий для любого процесса, учитывая бесконечное время исследования и частично случайную политику.
2. На каждом шаге мы получаем больше информации о процессе.
3. Model-free





# Почему именно Q-Learning?

## Преимущества Q-Learning

1. Выработка собственной политики управления системой в процессе обучения
2. Возможность использовать текущую модель Open Modelica
3. Идеально подходит для работы в динамических системах



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Задачи алгоритма

1. Основная задача: выработка алгоритмом политики управления динамической системой в процессе обучения
2. Субоптимальное управление ГВУ
3. Сравнение базового алгоритма и алгоритма ADP



# Основные требования к алгоритму

1. Соблюдение ограничений по безопасности
2. Стремление к минимальной стоимости электроэнергии
3. Самостоятельная работа после обучения
4. Возможность продолжения обучения в процессе работы

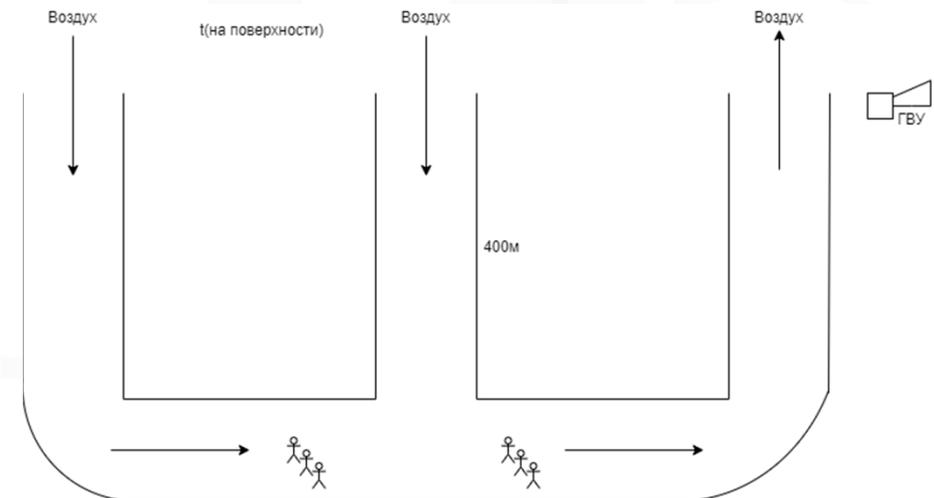


# Входные параметры алгоритма

## Цифровой двойник

На начальных этапах используется упрощенная модель горного рудника.

Собираются данные о состоянии системы: текущее состояние ГВУ, температура





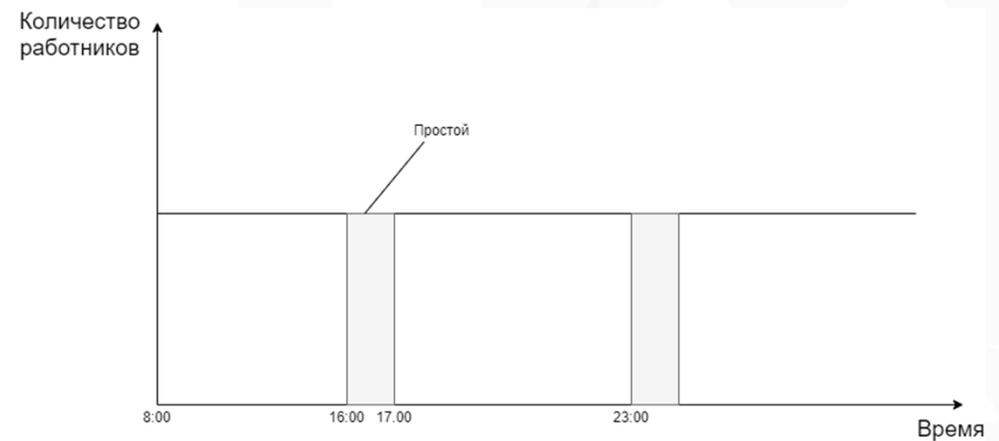
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Входные параметры алгоритма

## План производства

Предполагается, что мы заранее будем знать время пребывания пикового количества народа в руднике.

Имеется почасовой план работы с небольшим количеством окон простоя.

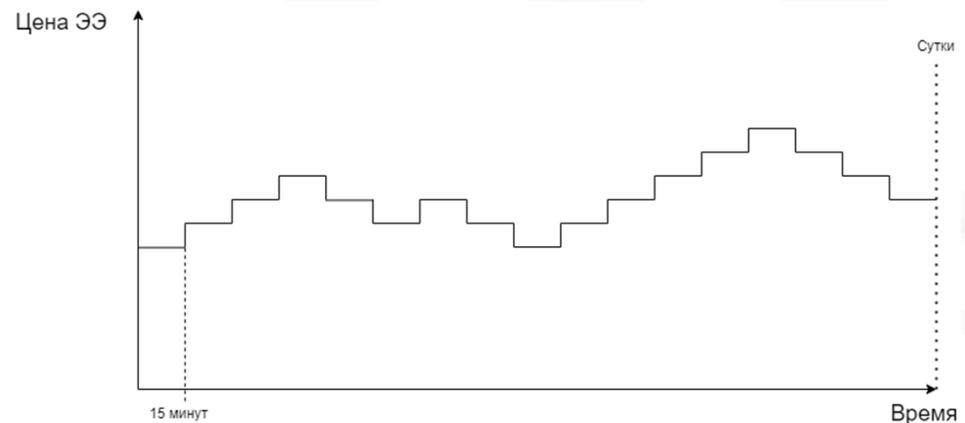




# Входные параметры алгоритма

## Цена электроэнергии

Предполагается, что цена на электроэнергию заранее известна в текущий день утром для следующего дня.



Интервал 15 минут



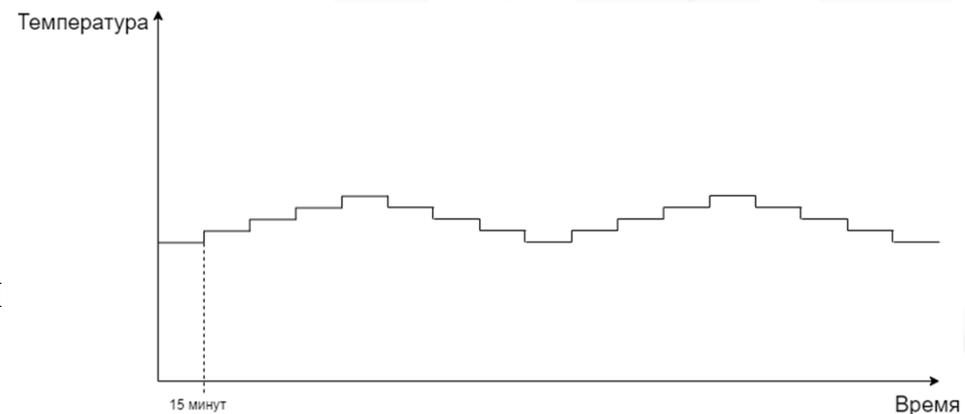
# Входные параметры алгоритма

## Температура воздуха

Используется два массива данных:

1. Прогноз
2. Фактическая температура

Прогноз известен заранее на следующий день, фактическая температура динамически изменяется.



Интервал 15 минут



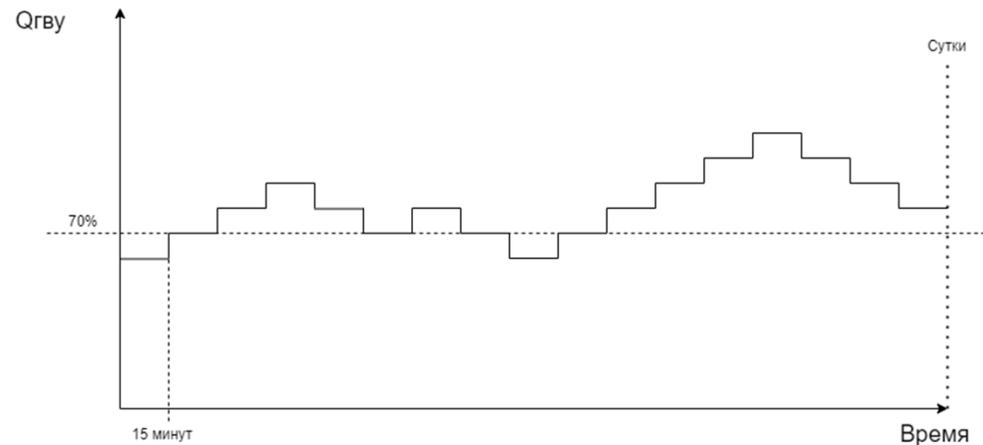
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Выходные результаты алгоритма

## Производительность ГВУ

В результате работы алгоритма будут получен план управления ГВУ на следующие сутки с интервалом в 15 минут.

План динамически изменяется в результате изменений состояния системы.



Интервал 15 минут



# Особенности при проветривании

1. Имеется минимальное состояние ГВУ, ниже которого производительность быть не может
2. Учет температуры окружающей среды
3. Учет плана производства
4. Возможность подстроить максимальное потребление в часы “дешевой” электроэнергии
5. Объем подаваемого воздуха всегда постоянный



# Логика Q-Learning

Метод Q-Learning основан на введении функции Q, отражающей ценность каждого возможного действия системы для текущего состояния, в котором сейчас находится симуляция.

$$x_t^n = \arg \max_{x_t \in X_t^n} (C_t(S_t^n, x_t) + \gamma \bar{V}_{t+1}^{n-1}(S_{t+1}(S_t^n, x_t, W_{t+1}(\omega^n))))$$

Процесс обучения представляет из себя итерационное уточнение значения функции Q на каждом шаге.

$$\bar{Q}_t^n(S_t, x_t) = (1 - \alpha_{n-1}) \bar{Q}_t^{n-1}(S_t^n, x_t^n) + \alpha_{n-1} \hat{q}_{t+1}^n$$

Таким образом, мы оцениваем перспективность перехода в новое состояние посредством совершения действия. В нашем случае Q фактор – это стоимость затрат на энергоресурсы при процессе проветривания.

$$x_t^n = \arg \max_{x_t \in X_t^n} \bar{Q}_t^{n-1}(S_t^n, x_t)$$

## Post-decision

$$\hat{v}_t^n = \max_{x_t \in X_t^n} (C_t(S_t^n, x_t) + \gamma \bar{V}_t^{n-1}(S^{M,x}(S_t^x, x_t))) = \max_{x_t \in X_t^n} \hat{Q}_t^{x,n}$$

, где

$\hat{Q}_t^{x,n} = C_t(S_t^n, x_t) + \gamma \bar{V}_t^{n-1}(S^{M,x}(S_t^x, x_t))$  является формой Q-фактора, вычисленного с использованием значения функции после принятия решения (post decision).



# Логика Q-Learning

## Шаг 0. Инициализация

**Шаг 0а.** Инициализируем аппроксимацию значения функции  $\bar{Q}_t^0(S_t, x_t)$  для всех состояний  $S_t$  и решений  $x_t \in X_t, t \in T$ .

**Шаг 0б.** Установим  $n = 1$

**Шаг 0с.** Инициализируем  $S_0^1$

**Шаг 1.** Выбираем примерный путь  $\omega^n$

**Шаг 2.** Выполняем для  $t = 0, 1, \dots, T$ .

**Шаг 2а.** Найдем решение используя текущий Q-фактор

$$x_t^n = \arg \max_{x_t \in X_t^n} \bar{Q}_t^{n-1}(S_t^n, x_t)$$

**Шаг 2б.** Подсчитываем:

$$\hat{q}_{t+1}^n = C_t(S_t^n, x_t^n) + \gamma \bar{V}_{t+1}^{n-1} S^M(S_t^n, x_t^n, W_{t+1}(\omega^n))$$

**Шаг 2с.** Обновляем  $\bar{Q}_t^{n-1}$  и  $\bar{V}_t^{n-1}$  используя:

$$\bar{Q}_t^n(S_t, x_t) = (1 - \alpha_{n-1}) \bar{Q}_t^{n-1}(S_t^n, x_t^n) + \alpha_{n-1} \hat{q}_{t+1}^n$$

$$\bar{V}_t^n(S_t^n) = \max_{x_t} \bar{Q}_t^n(S_t^n, x_t^n)$$

**Шаг 2д.** Находим новое состояние

$$S_{t+1}^n = S^M(S_t^n, x_t^n, W_{t+1}(\omega^n))$$

**Шаг 3.** Увеличиваем  $n$ . Если  $n \leq N$  перейти к шагу 1.

**Шаг 4.** Возвращаем Q-факторы  $(\bar{Q}_t^n)_{t=1}^T$





НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Этап обучения

1. Моделирование различных ситуаций и отслеживание поведения алгоритма
2. Использование упрощенной модели
3. Последующее обучение при изменении модели
4. Расчет минимальной стоимости электроэнергии



# Этап работы

1. Сравнение с базовым алгоритмом:  $Q_{гву} = 70\%$
2. Жесткое ограничение по безопасности
3. Оптимизация за счет температуры окружающей среды



# Аппроксимация

Возможные варианты аппроксимации:

1. Безусловная эквивалентность
2. Аппроксимация на основе уравнения линейной регрессии
3. Нейронная сеть

Наиболее важно, что Q-обучение обычно представлено в ситуациях, когда состояния и действия дискретны. Мы можем использовать общие регрессионные модели, которые помогают нам избежать вычисления значений для каждой пары: действие-состояние. С точки зрения объема вычислений разница велика.



# Выбор разумной аппроксимации

## Аппроксимация нейронной сетью

Почему именно нейронная сеть:

1. Неизвестно точное решение задачи аппроксимации, но присутствует достаточное количество исходных данных
2. Большие объемы входной информации
3. Использование цифрового двойника



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Стек

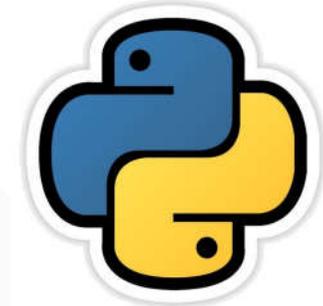
Для реализации алгоритма  
используется:

Язык Программирования Python 3.8

СУБД InfluxDB

Библиотеки для реализации  
машинного обучения TensorFlow,  
Keras, Padasip

Модель OpenModelica



# OpenModelica



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

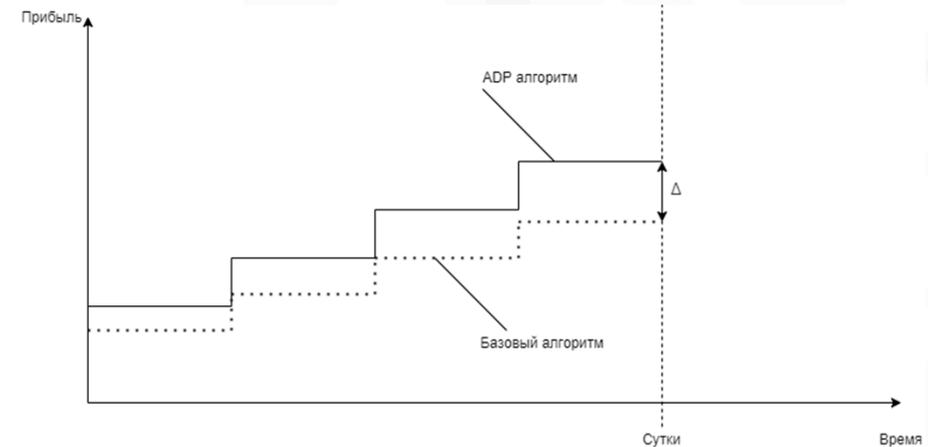
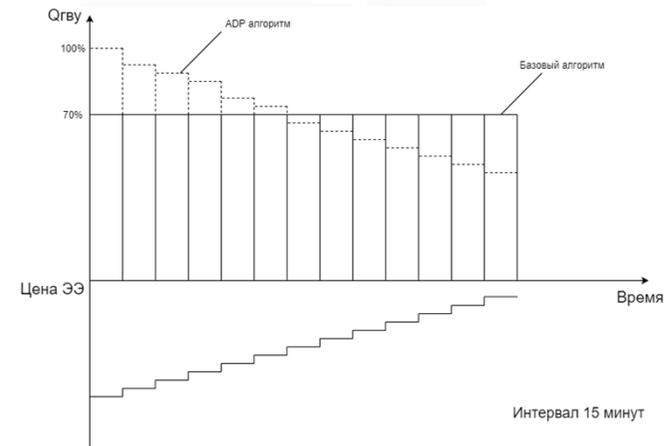
# Визуализация

Визуализация будет представлена в среде **Grafana**

**Grafana** - это платформа с открытым исходным кодом для визуализации, мониторинга и анализа данных.

Необходимые графики:

- 1. График обучения**  
(показывает процесс построения политики проветривания при котором потребление уменьшается в “дорогие” часы)
- 2. График прибыли**  
(Накапливает суммарную прибыль за месяц в сравнении с базовым алгоритмом)





НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

# Спасибо за внимание!

Черницын Игорь, НИУ ВШЭ – Пермь