podlodka

TechLead Crew

30 мая — 3 июня

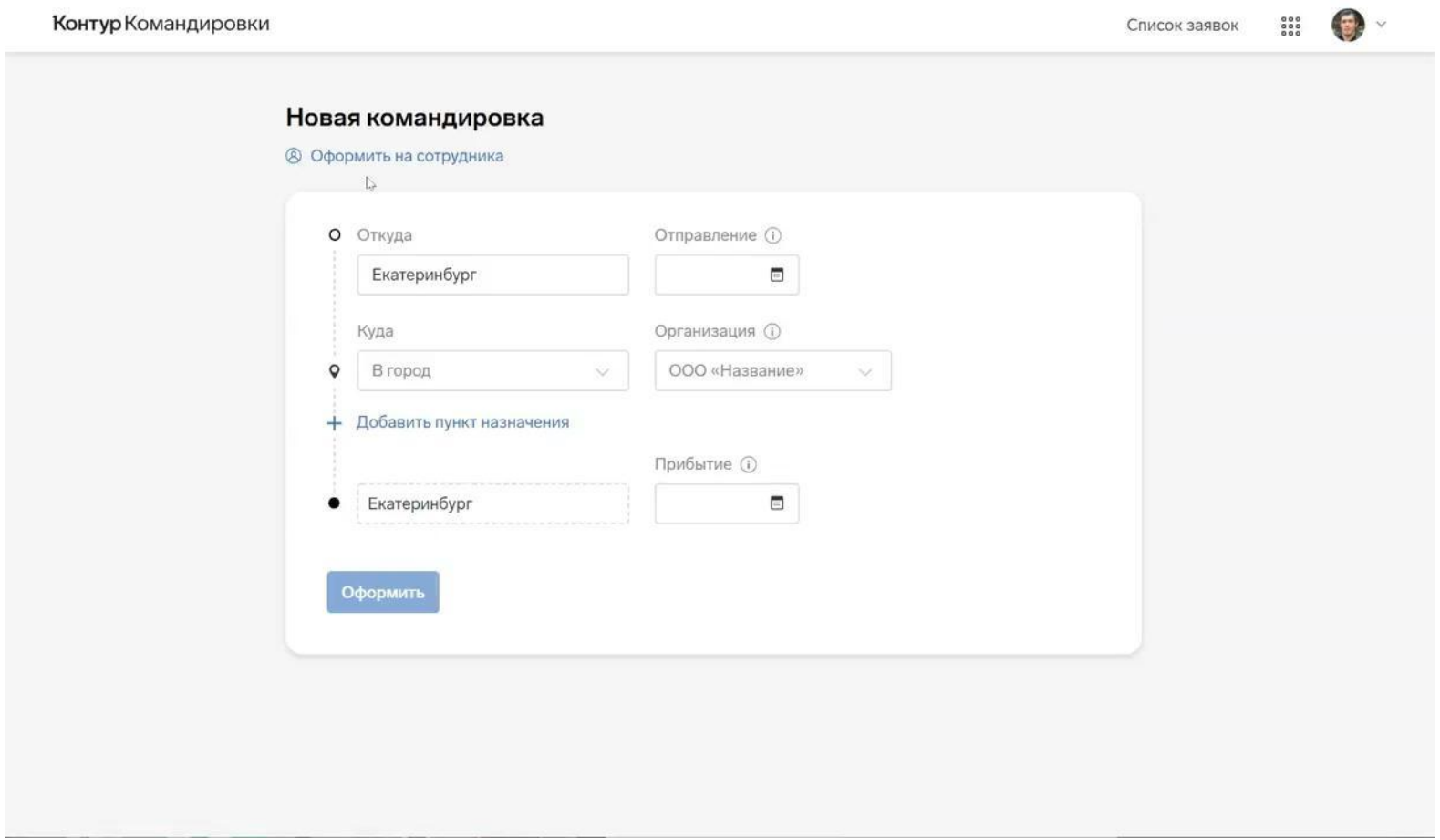
Continuous Delivery

Алексей Шестаков
Контур

Доклад: Саморегулируемый сервис на базе Многоруких бандитов

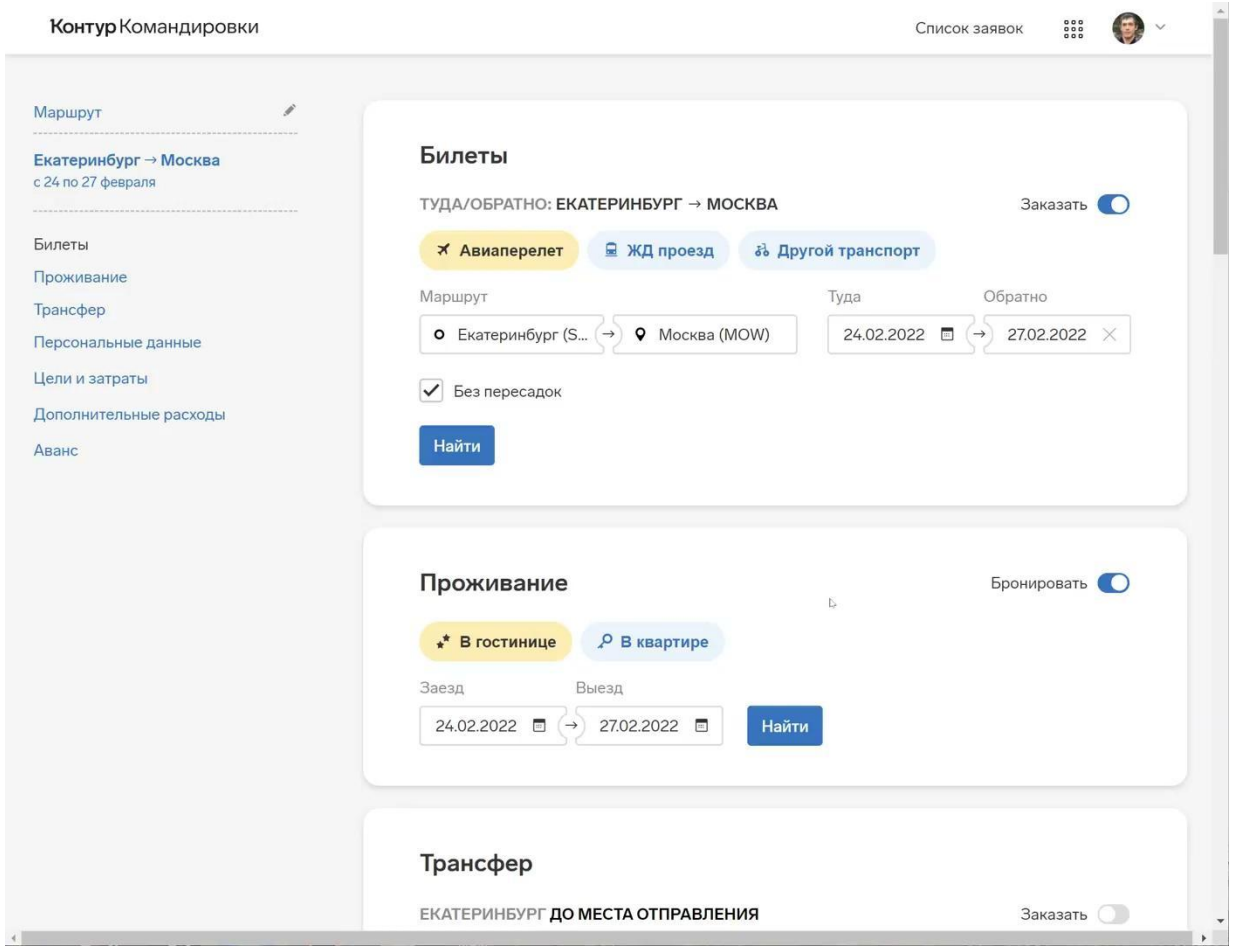
Контур.Командировки

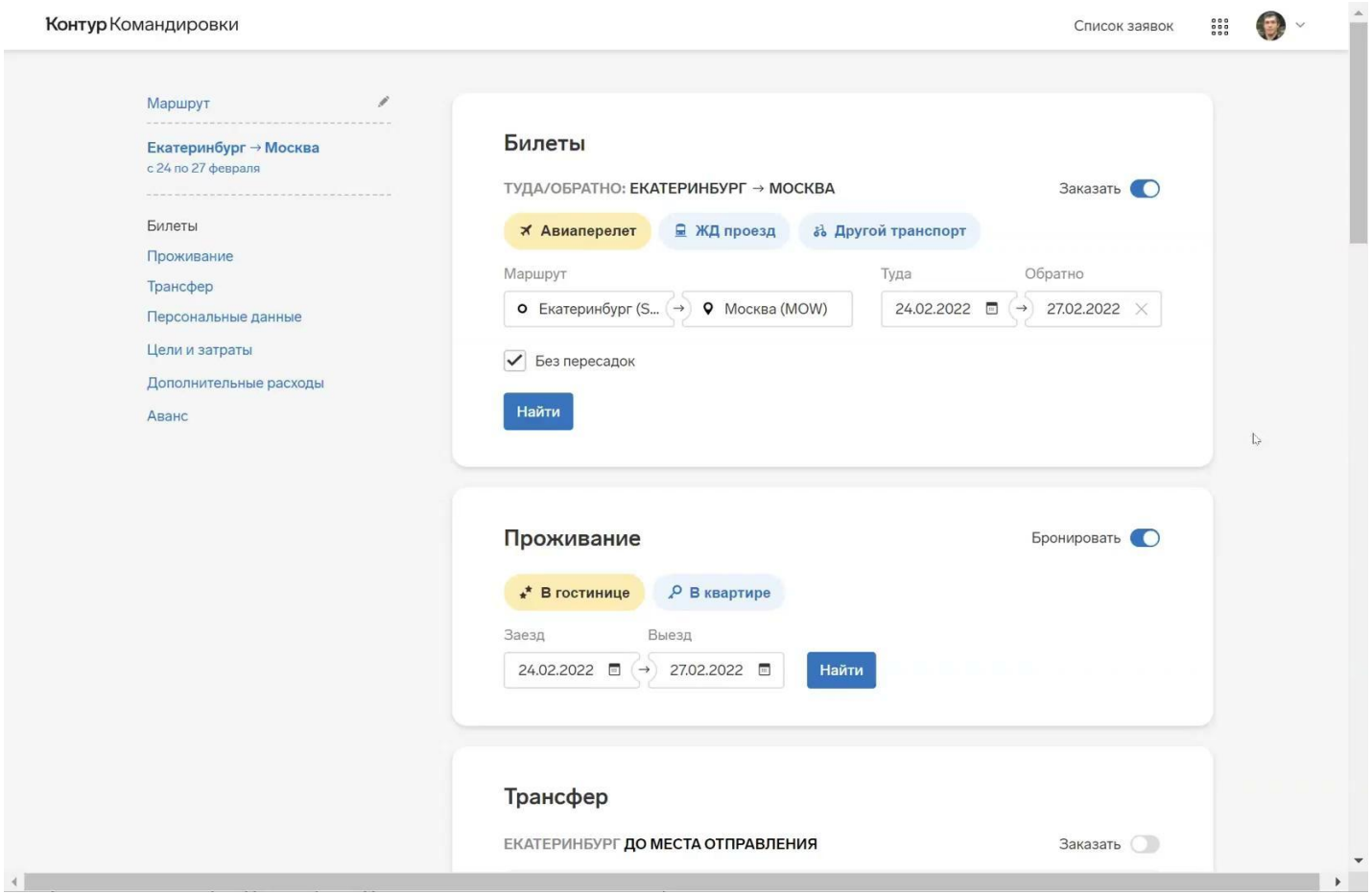
- Сервис для сотрудников
- Входная точка для оформления командировки в компании.
- Оформление приказов + Бронирование номеров + Выкуп авиабилетов

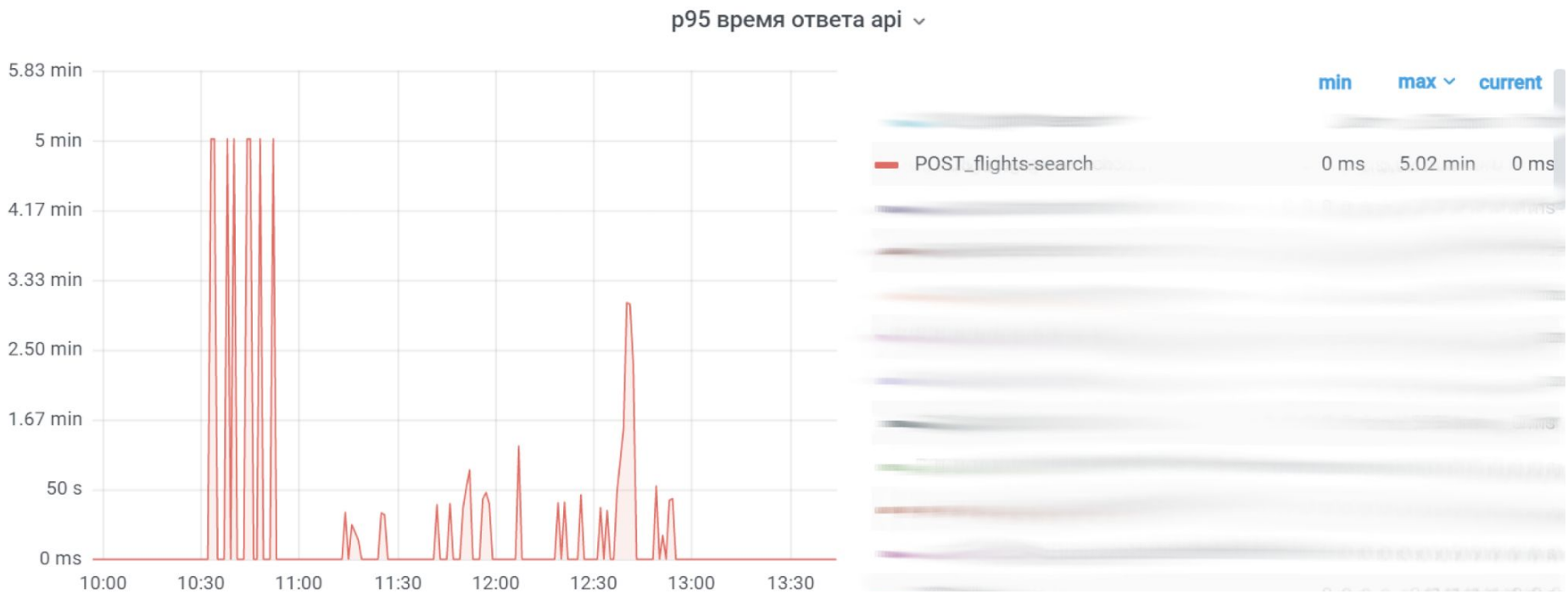

Контур Командировки
Список заявок
Новая командировка
Оформить на сотрудника
Откуда
Екатеринбург
Отправление
Куда
В город
Организация
ООО «Название»
Добавить пункт назначения
Прибытие
Екатеринбург
Оформить

Агрегаторы (Поставщики)

- Поисковая выдача + Бронь + Покупка
- Подобие Booking и Avia-Sales
- Их много)
- Платные API Сервисы
- Вместо договоров с авиакомпаниями/отелями только **один** договор


Контур Командировки
Список заявок
Маршрут
Екатеринбург → Москва
с 24 по 27 февраля
Билеты
Проживание
Трансфер
Персональные данные
Цели и затраты
Дополнительные расходы
Аванс
Билеты
ТУДА/ОБРАТНО: ЕКАТЕРИНБУРГ → МОСКВА
Заказать
Авиаперелет
ЖД проезд
Другой транспорт
Маршрут
Екатеринбург (S...
Москва (MOW)
Туда
Обратно
24.02.2022
27.02.2022
Без пересадок
Найти
Проживание
Бронировать
В гостинице
В квартире
Заезд
Выезд
24.02.2022
27.02.2022
Найти
Трансфер
ЕКАТЕРИНБУРГ ДО МЕСТА ОТПРАВЛЕНИЯ
Заказать


Контур Командировки
Список заявок
Маршрут
Екатеринбург → Москва
с 24 по 27 февраля
Билеты
Проживание
Трансфер
Персональные данные
Цели и затраты
Дополнительные расходы
Аванс
Билеты
ТУДА/ОБРАТНО: ЕКАТЕРИНБУРГ → МОСКВА
Заказать
Авиаперелет
ЖД проезд
Другой транспорт
Маршрут
Екатеринбург (S...
Москва (MOW)
Туда
Обратно
24.02.2022
27.02.2022
Без пересадок
Найти
Проживание
Бронировать
В гостинице
В квартире
Заезд
Выезд
24.02.2022
27.02.2022
Найти
Трансфер
ЕКАТЕРИНБУРГ ДО МЕСТА ОТПРАВЛЕНИЯ
Заказать



Грустная графана нашего бека

Агрегаторы, Боли

- Могут еще и:
  - Лежать
  - Отвечать 500-кой
  - Отдавать не все варианты

- Разные **предложения** и разные **цены**
- Свойства **меняются** со временем

Агрегаторы, Боли

- Могут еще и:
  - Лежать
  - Отвечать 500-кой
  - Отдавать не все варианты

- Разные **предложения** и разные **цены**
- Свойства **меняются** со временем

Наши требования

- Сломанный поставщик ≠ сломанный сервис
- Пользователь хочет, выдачу быстро, уже **вчера**.
- Бизнес хочет **дешевле**
- Пользователь ожидает **больше** вариантов

Простые решения

- Один поставщик

  Мы так **жили**

- Много поставщиков + бизнес-лапша из if-ов

  Мы так **не захотели** жить

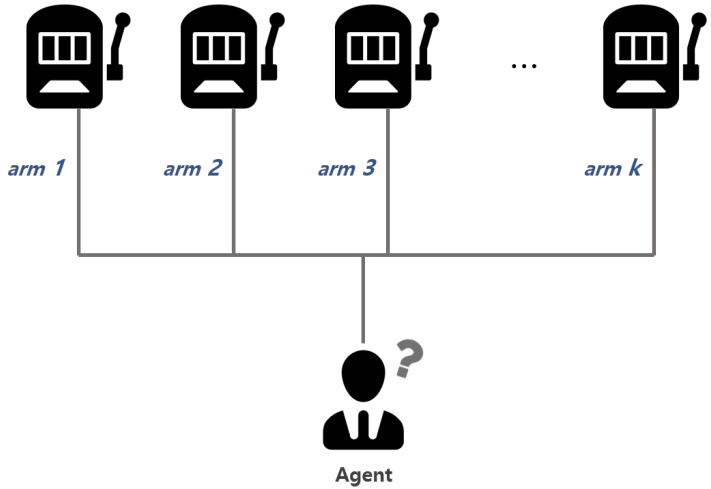
Мы так жить не захотели

- А как поддерживать?
- Как добавлять поставщика?
- Как оценивать поставщиков?

Желаемая картина

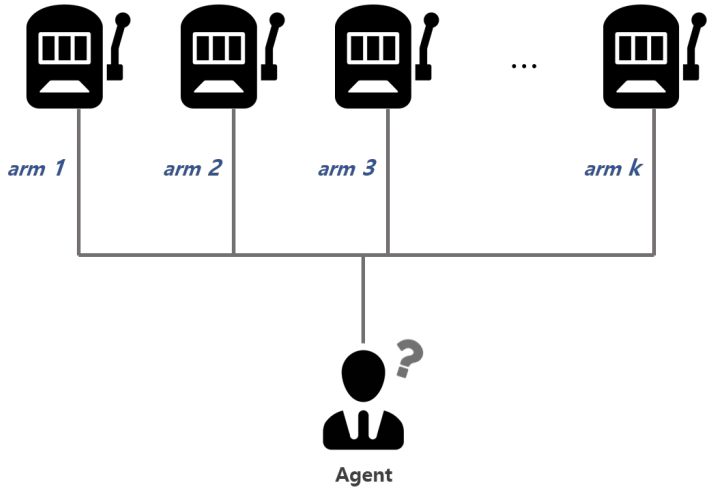
- Много поставщиков
- Сервис сам принимает решение к кому и когда идти
- Хотим Circuit Breaker
- Хотим **найти баланс**

Многорукий бандит
Multi Armed-bandit



"Многорукий бандит" — это математическая модель игрального автомата, с множеством ручек

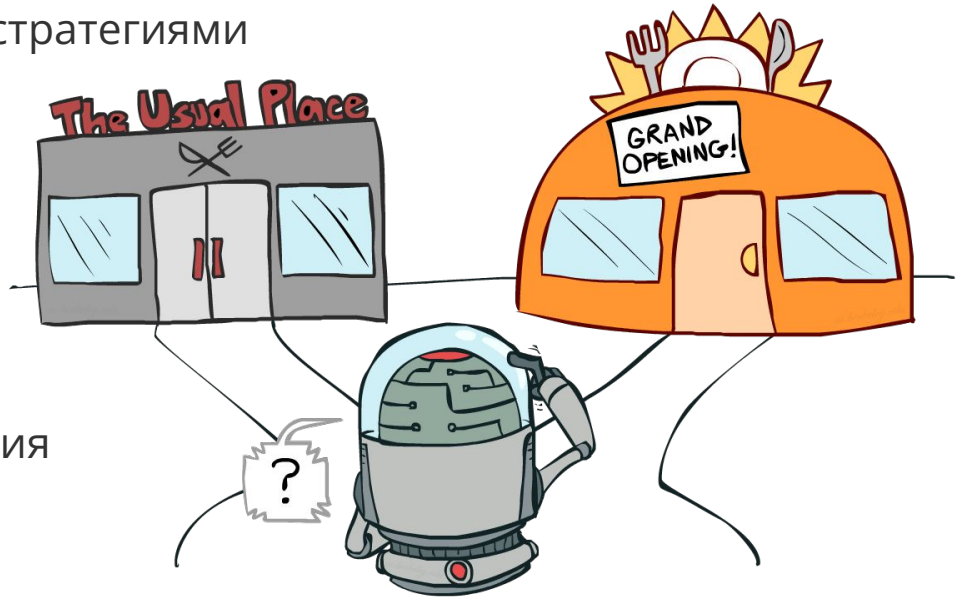
Многорукий бандит
Multi Armed-bandit



Все ручки разные: Не попробуешь - не узнаешь

Многорукий бандит
Multi Armed-bandit

Балансируем между двумя стратегиями



- **Exploit** - Используем знания
- ***Explore*** - Собираем знания

Картинки с роботами взяты из http://ai.berkeley.edu/lecture_slides.html

Когда бандит не подойдет

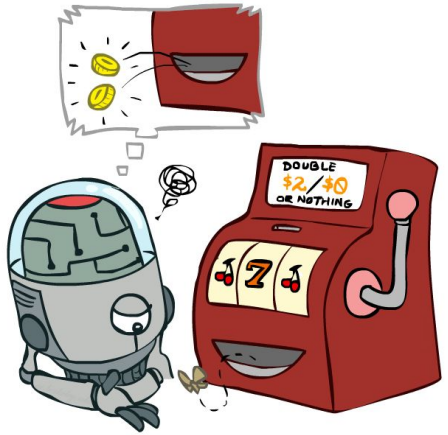
- От **нашего** выбора меняется поведение агрегатора
- Если мы априори знаем все про его поведение (**Полные знания**)

Почему бандит нам зашел?

- Не нужно администрировать.
- Легко добавить еще одного.

Что нужно бандиту?

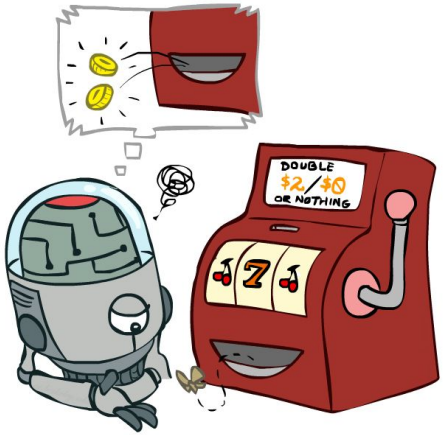
- Для бандита нужна оценка выдачи
- Насколько хорошо отработал поставщик
- Это **одно** число



Роботы взяты из http://ai.berkeley.edu/lecture_slides.html

Как мы оценивали поставщика:

- Скорость ответа агрегатора ***S***,
- Полнота выдачи ***C***



Алгоритм: смешать метрики и получить одно число,

перед этим нужно привести все к одному диапазону

Роботы взяты из http://ai.berkeley.edu/lecture_slides.html

Посчитаем награду

Награда - **нормированная** смесь параметров

$$R(\text{награда}) = \text{norm}(S) + \text{norm}(C) + \text{norm}(....)$$

Как нормировать

- Мы задаем максимальные и минимальные значения параметров
- Приводим в диапазон от 0 до 1, где 1 - все идеально , 0 - все плохо
- **min-max normalization**

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Посчитаем оценку выдачи

Оценка (награда, reward) – **нормированная** смесь параметров

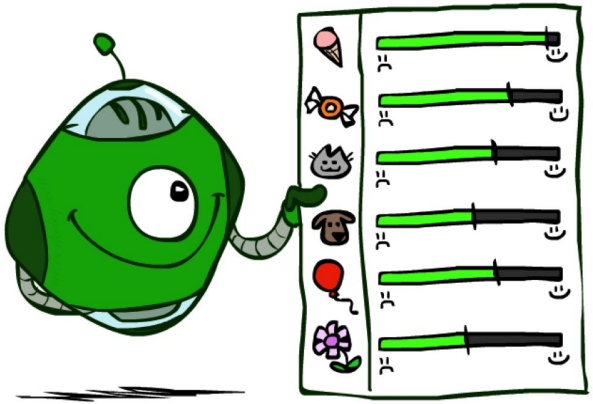
$$R(\text{награда}) = \text{norm}(S) + \text{norm}(C) + \text{norm}(....)$$

Если параметры важны не в равной степени:

$$R(\text{награда}) = \text{вес1}*\text{norm}(S) + \text{вес2}*\text{norm}(C) + \text{вес3}*\text{norm}(....)$$

Какая целевая у вашего сервиса

- Click Rate
  - Всегда ли пользователь переходит?
- Retention Time
  - Время пользователя на странице
- Response Time
  - Время ответа от сервера
- ...



Картинки с роботами взяты из http://ai.berkeley.edu/lecture_slides.html

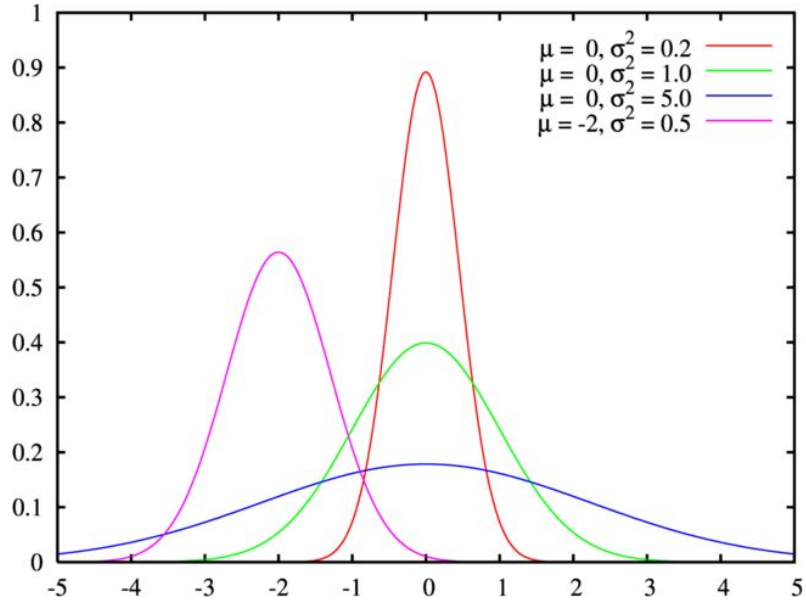
Попробуем промоделировать

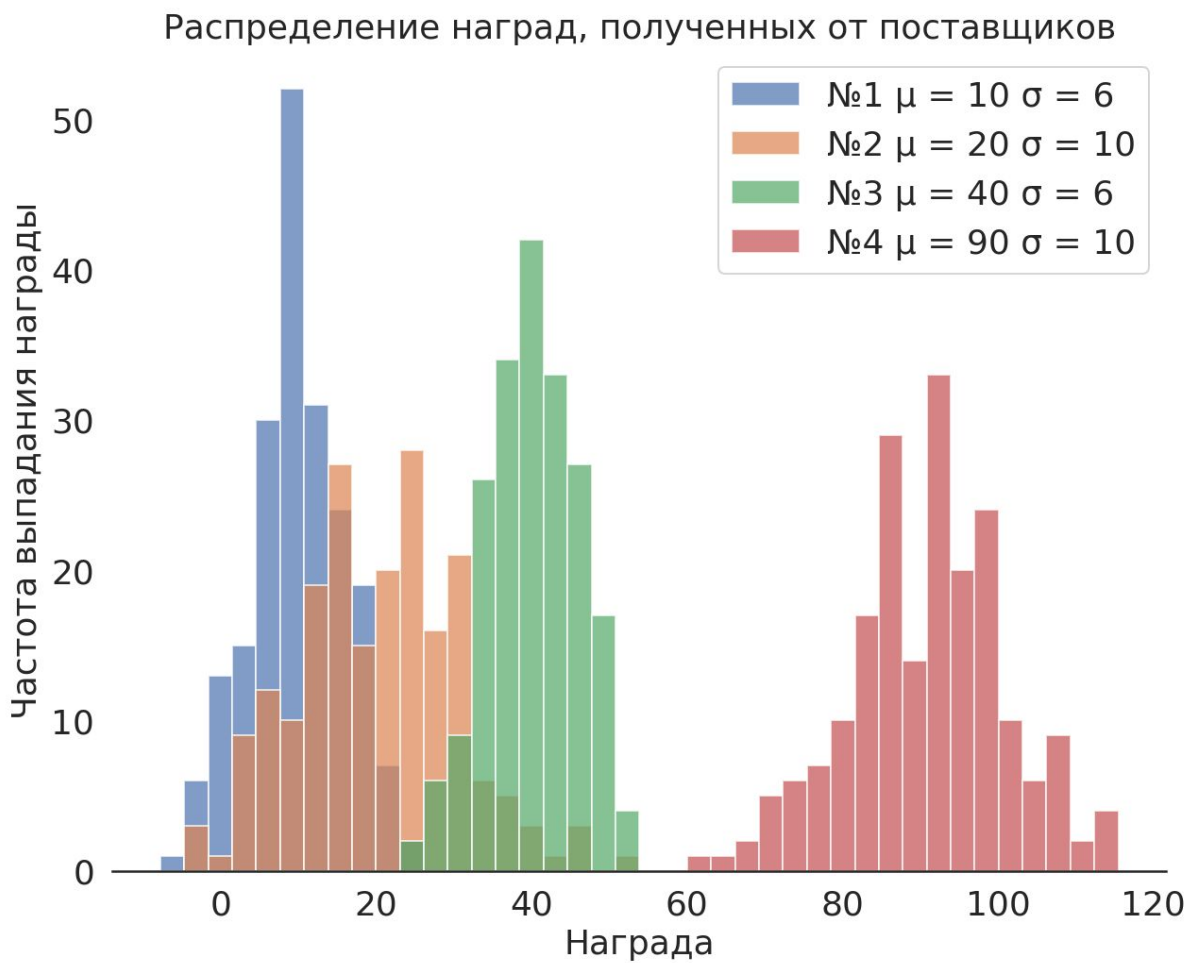
- **Нормальное** распределение
- Параметры:

  Мат. ожидание(μ)

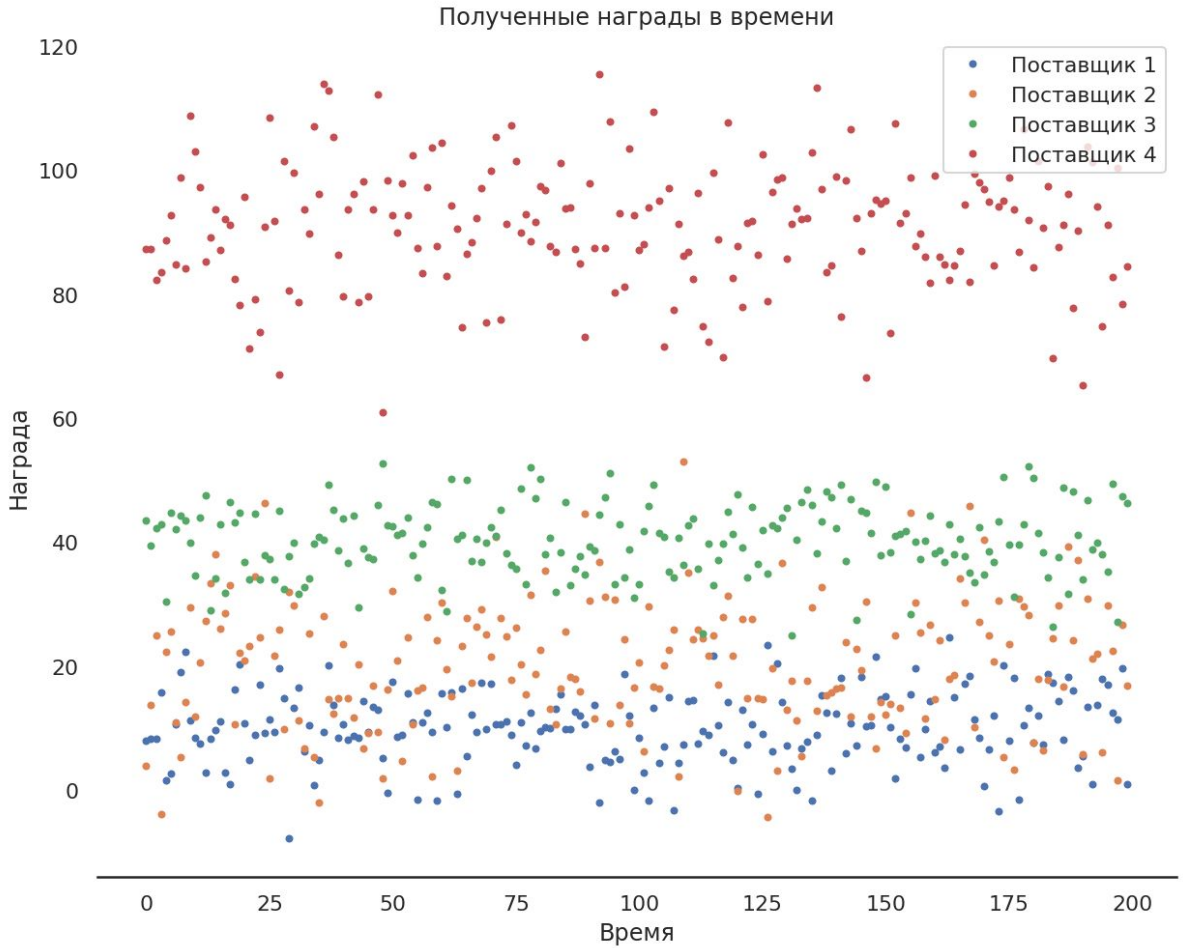
  Дисперсия(σ)

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$





Модели 4-х поставщиков



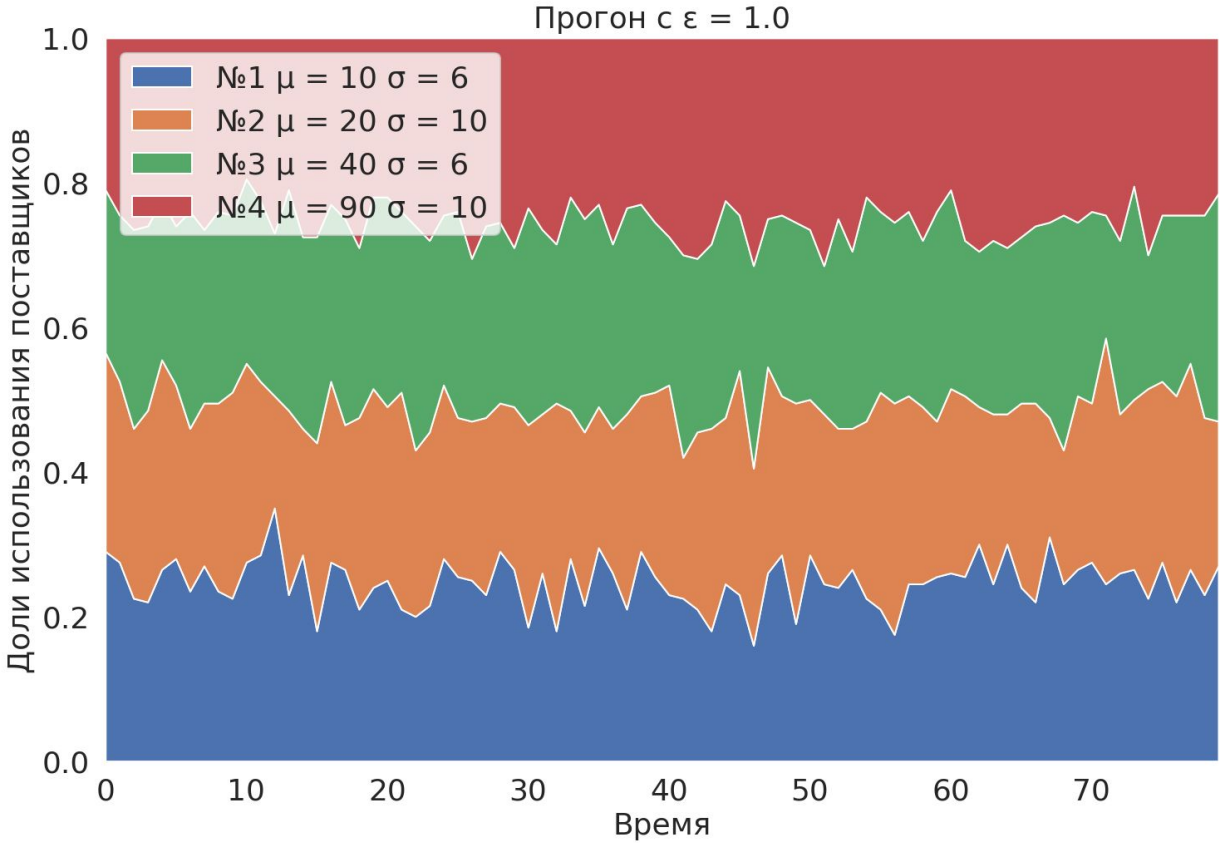
То же самое только во времени

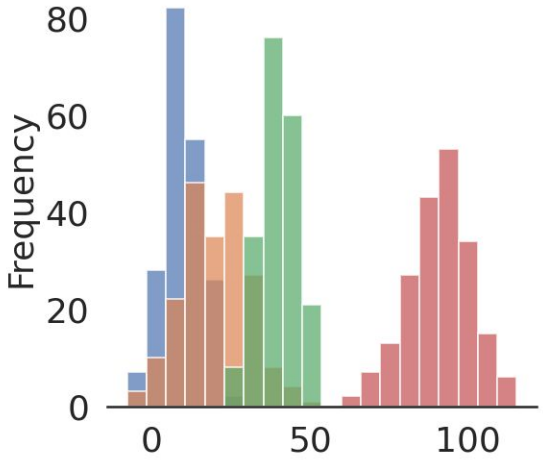
Самые простые бандиты

Крайние случаи

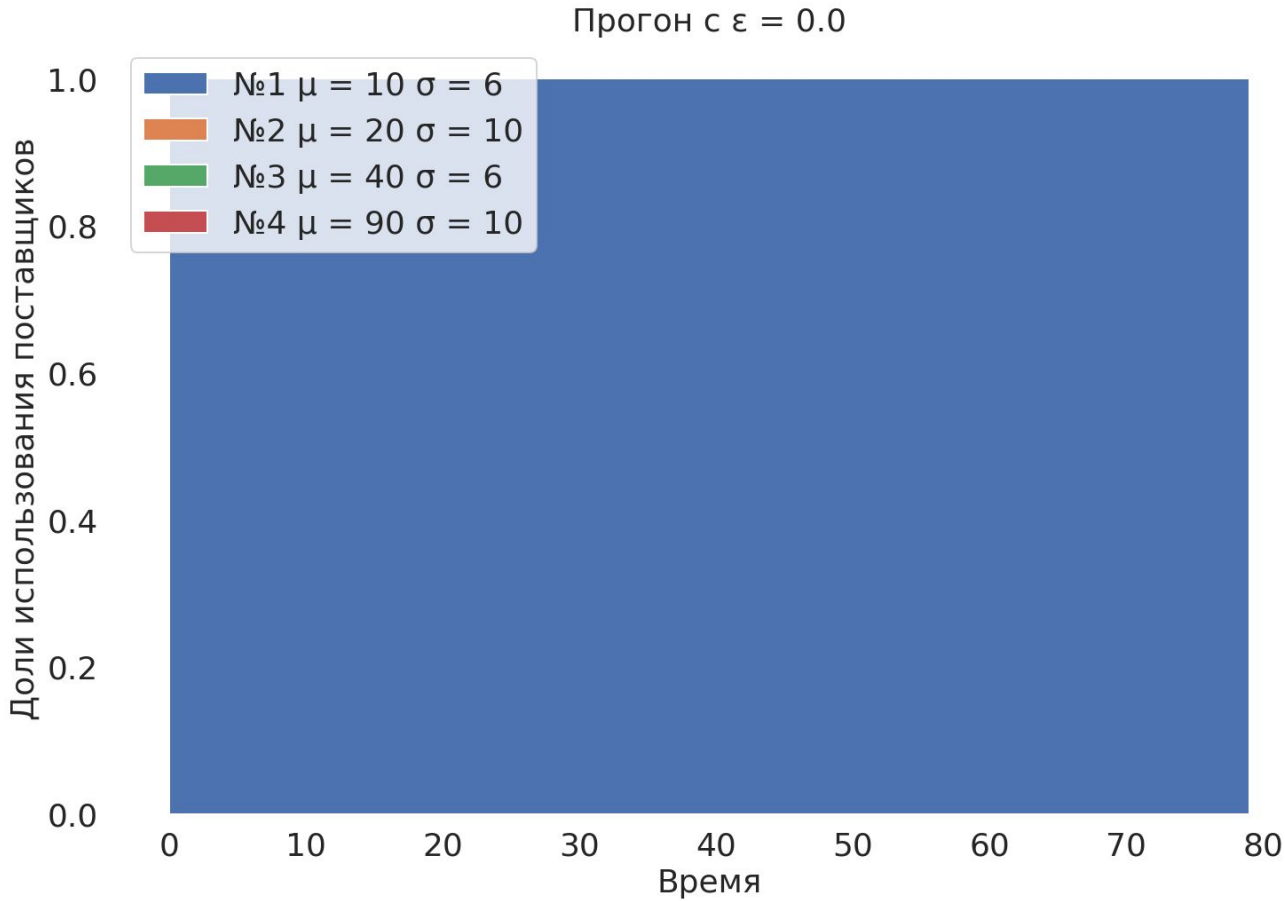
- 100% Greedy ("Жадный")
- 100% Исследователь

- Не очень рабочие





100% Процентный исследователь



100% жадности - берем первый рабочии вариант и используем всегда

Выводы по простым

- Это крайние случаи
- Они не работают

ε-greedy (эпсилон "жадный")

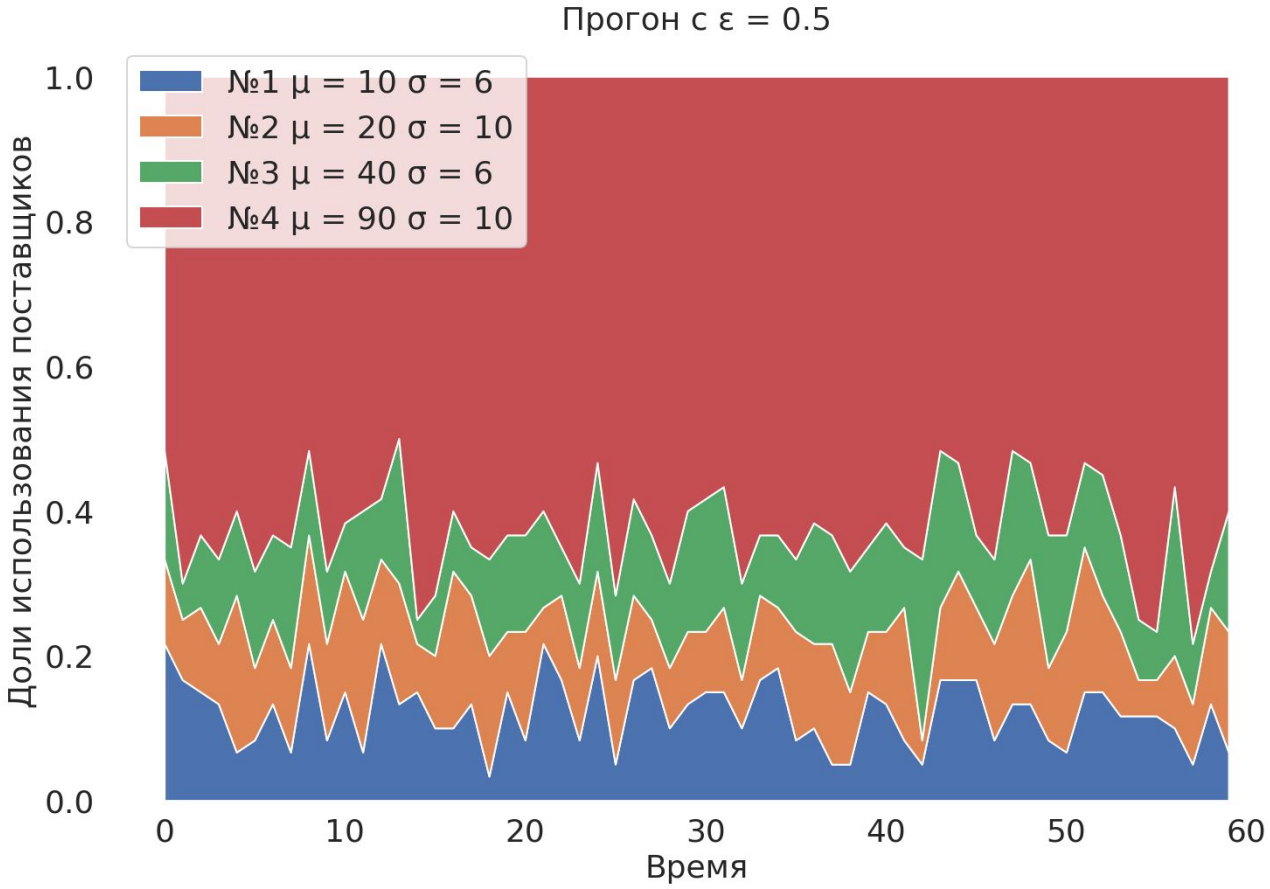
Explore с вероятностью **ε**:

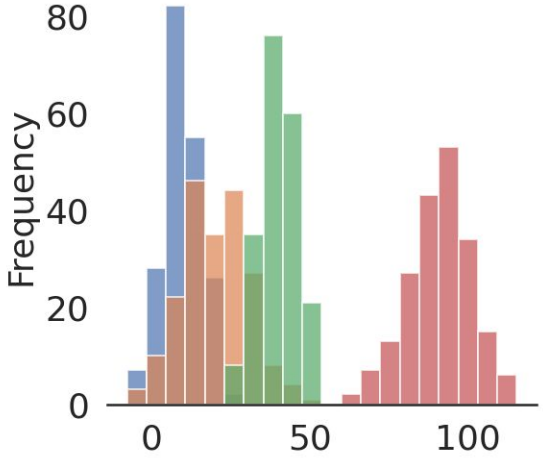
- Дергаем случайного поставщика

Exploit с вероятностью **1 - ε**:

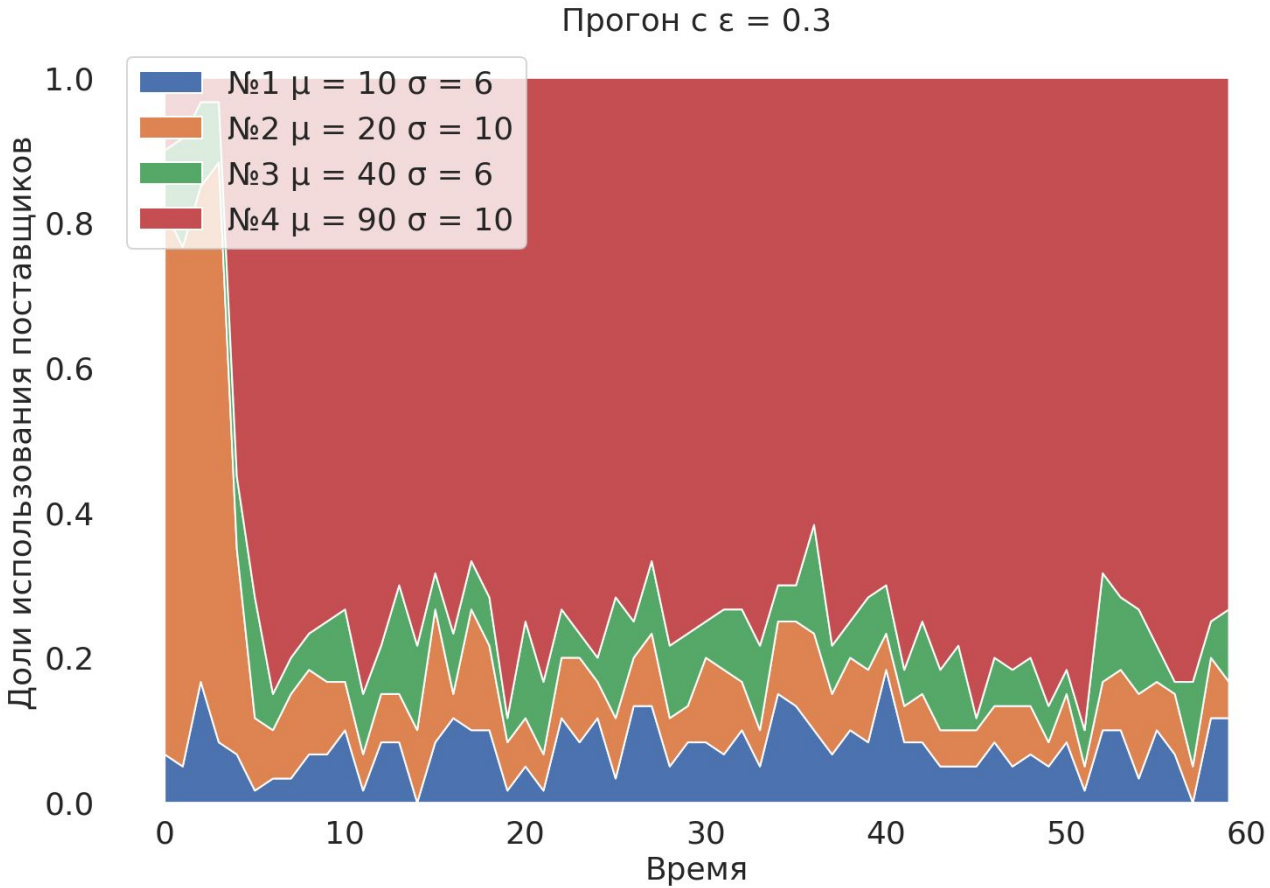
- Дергаем лучшего поставщика

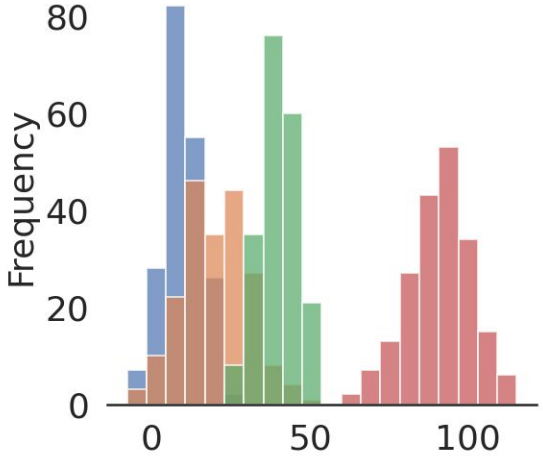
Рабочее решение



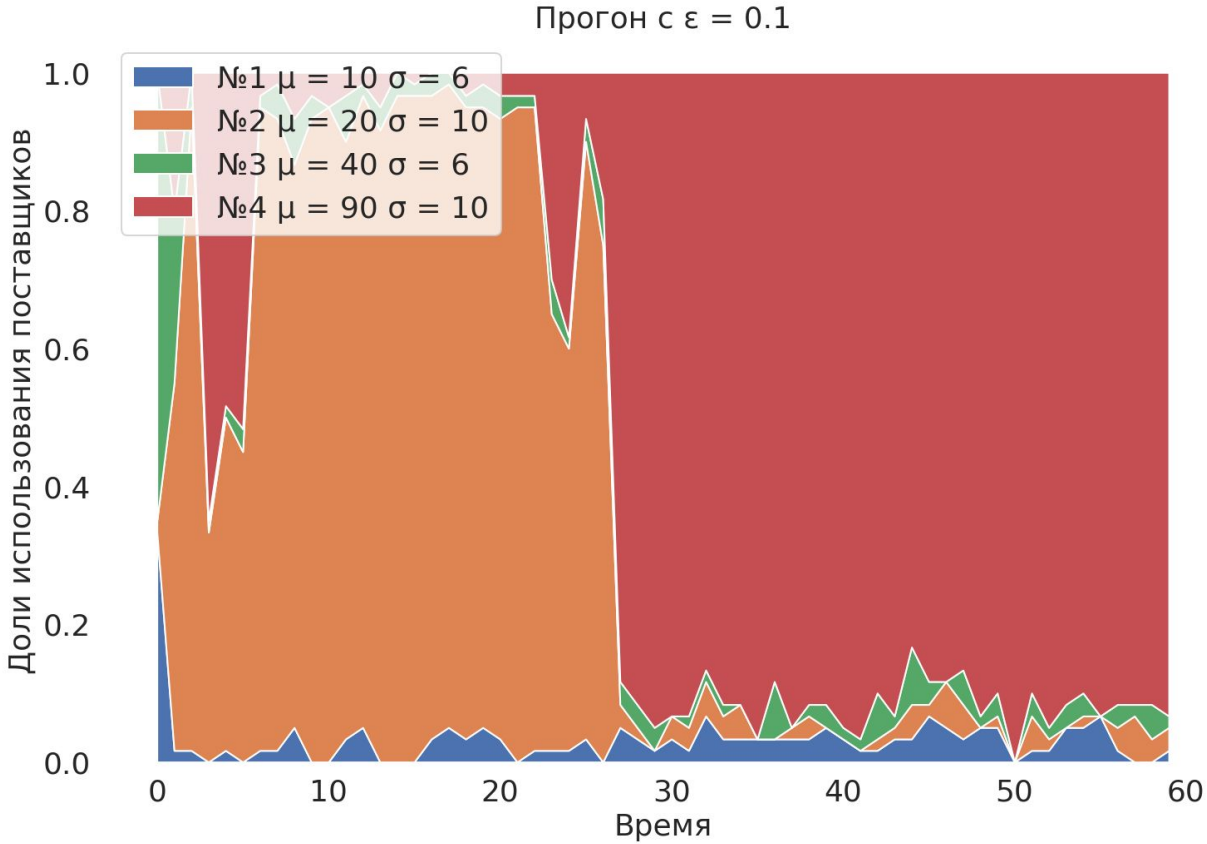


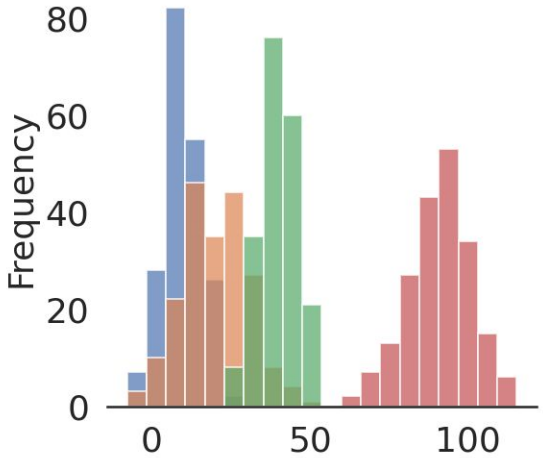
ε-greedy с ε = 0.5 (50% Ресурсов на исследование)



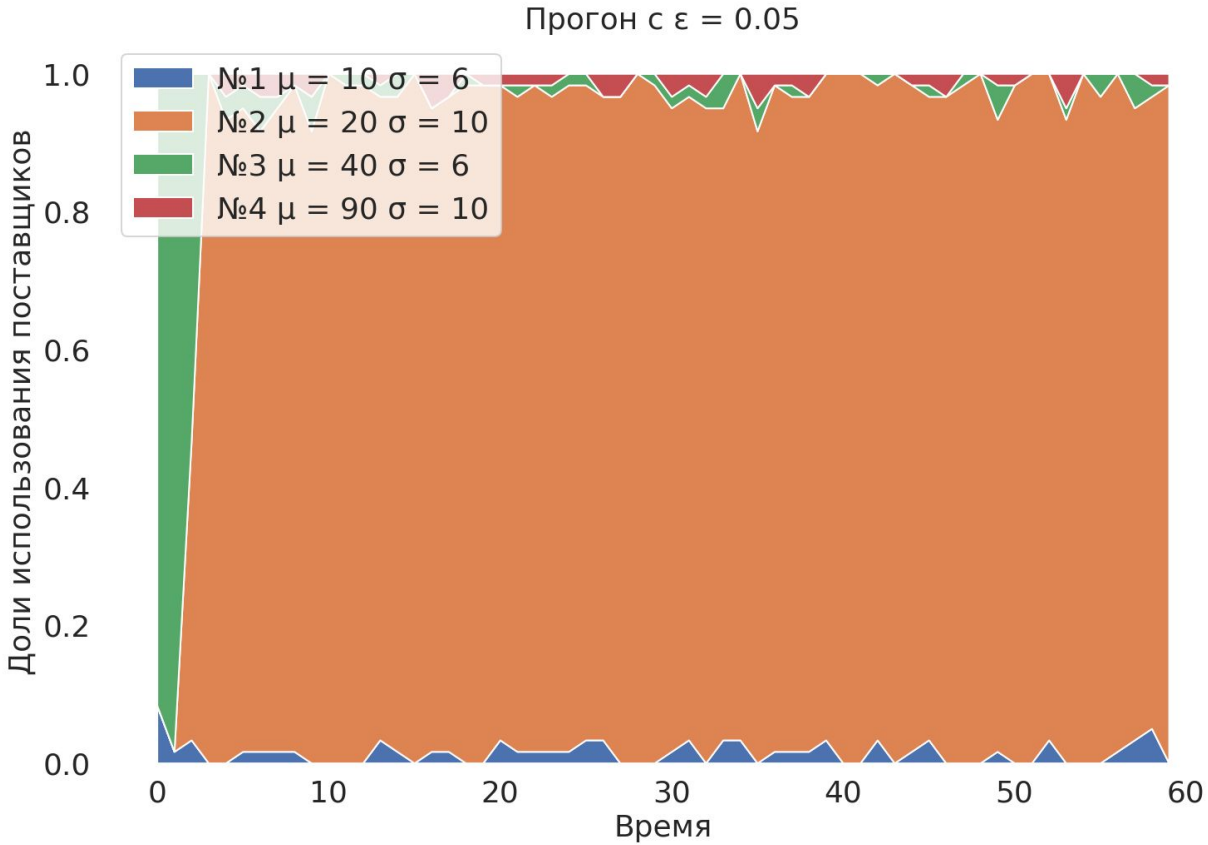


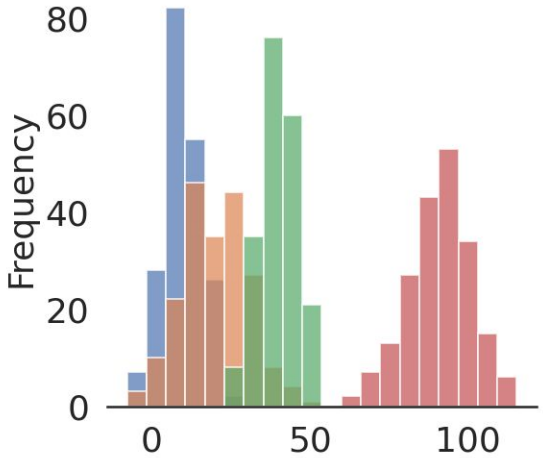
ε-greedy с четырьмя поставщиками и ε = 0.3





ε-greedy с четырьмя поставщиками и ε = 0.1 (10% Ресурсов на исследование)

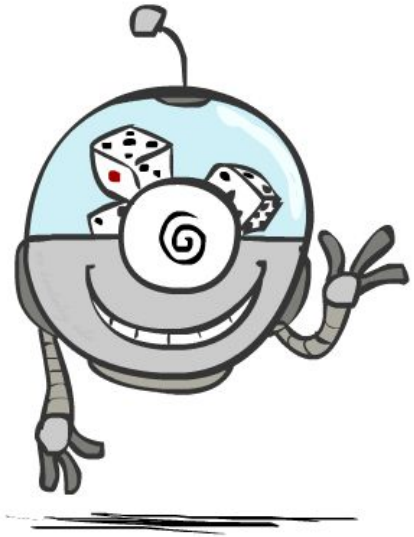




ε-greedy с ε = 0.05 (5% Ресурсов на исследование)

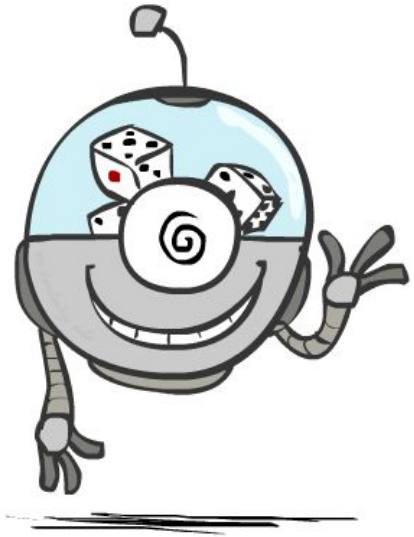
Хранение истории наград

- Нужно хранить значения всех прошлых наград.
- Если поведение агрегаторов меняется со временем, алгоритм не сможет адекватно среагировать — среднее довольно нечувствительно.
- Можно полечить скользящим средним.
- Так-же можно хранить n результатов и вытеснять старые



Выводы по ε-greedy

- Мы показываем пользователям не самую хорошую выдачу чаще, чем могли бы
- Но мы надеемся, что поставщик может исправится



ε можем менять *внешним* алгоритмом или админом

UCB1 - Оптимизм перед неизвестностью

- Более умный алгоритм
- Фактически это эвристика, исследуем в начале, в конце мало исследуем
- Не рассчитываем сильно на то что что то станет в дальнейшем лучше.

The algorithm UCB1 [Auer et al.(2002)Auer, Cesa-Bianchi, and Fischer]

UCB1

Инициализация:

Попробовать каждого из агрегаторов хотя бы один раз

Цикл:

Отправляем запрос к агрегатору ***j***, для которого максимально

$$\tilde{x}_j + c\sqrt{\frac{2\ \ln(n)}{n_j}}$$

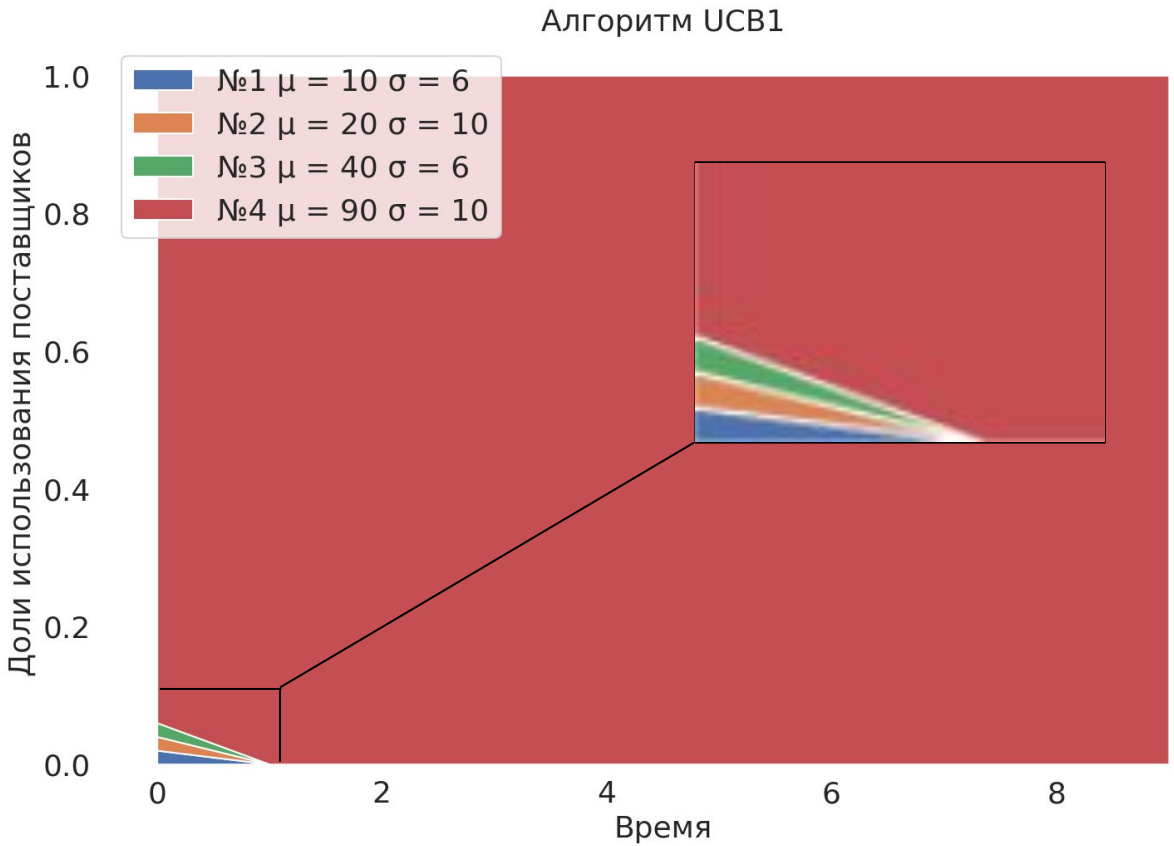
Где x_j - среднее качество работы агрегатора ***j***

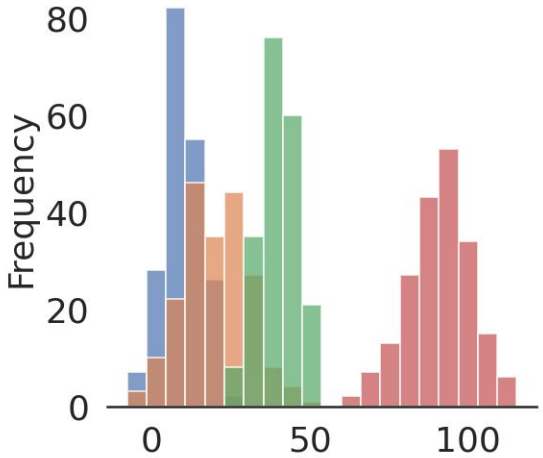
n_j - сколько раз раз мы делали запрос к агрегатору ***j***

n - сколько раз мы обращались ко всем агрегаторам в сумме

c - константа оптимистичности: чем больше она, тем больше мы оптимистичны

The algorithm UCB1 [Auer et al.(2002)Auer, Cesa-Bianchi, and Fischer]

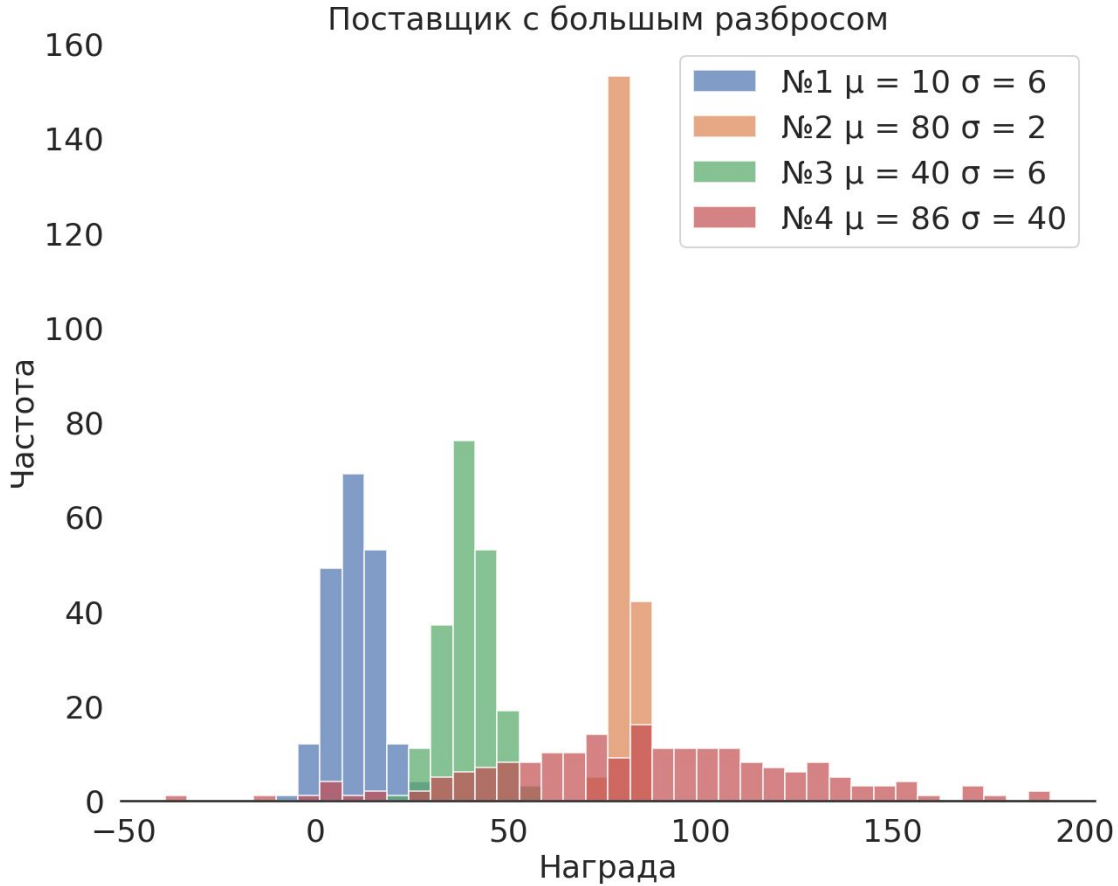




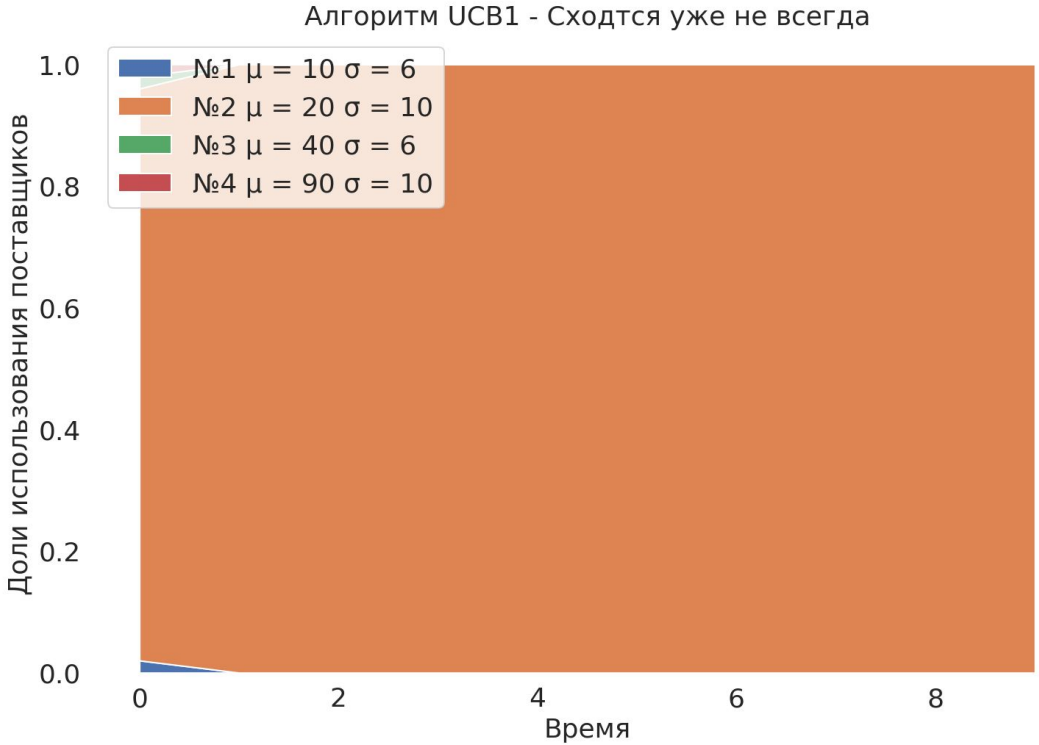
Результат работы алгоритма UCB1

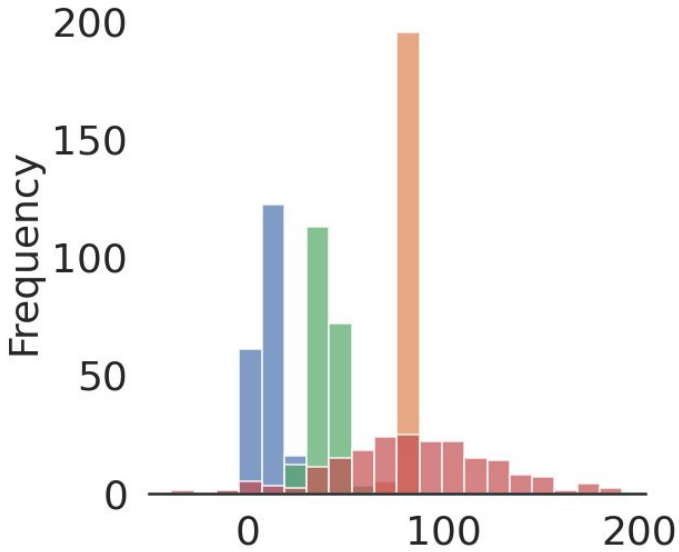
Сходимость к нужному поставщику

- Если будет поставщик с большим разбросом результатов то бандит не всегда выбирает его, даже если он лучше
- Особенно если константа c выбрана слишком маленькой



Один из поставщиков (4-ый) работает лучше всех но с разбросом





Упс, мы не сошлись к нужному поставщику

Выводы по UCB1

- Если бандит не настроен, то может сойтись **сначала** не к самой лучшей руке и лишь потом, очень медленно, к лучшей.
- Сочетает быструю скорость сходимости и не пытается **исследовать** неперспективные ручки.
- Не подходит в чистом виде, если ручка меняет свои свойства со временем.
- Но можно регулярно **сбрасывать статистику**, что бы прошли перевыборы

А еще бандитами можно заменить А/Б тесты

- Автоматом вводить фичу в строй при успехе фичи у пользователей.
- Автоматом реагировать на изменения предпочтений пользователей,
- Автоматом переключиться со "сломанного" варианта фичи при падении части инфраструктуры.

Отличие от А/Б

- Полная автоматизация
- Можно закидывать новые и подчищать неудачные варианты в фоновом режиме (не удачные и так не показываются почти, а удачные будут вводиться плавно)

Какие еще бандиты бывают?

- Есть куча **алгоритмов решения** задачи многорукого бандита
- А еще есть куча различных **постановок (сеттингов)**
- Регулярно появляются новые идеи на arxiv

Дальше: различные варианты постановки

Restless multi-armed bandit

- Параметры ручек **меняются со временем**
- Снимается ограничение стационарности

Whittle, Peter (1988), "Restless bandits: Activity allocation in a changing world"

Journal of Applied Probability, **25A**: 287–298, doi:10.2307/3214163

Dueling Bandits

- В качестве награды **не число**, а результат сравнения **двух ручек**, дернутых одновременно
- **A** > **B** или **B** < **A**
- Хороши там, где получить численную метрику сложно

Simple Algorithms for Dueling Bandits, Tyler Lekang, Andrew Lamperski
https://arxiv.org/abs/1906.07611

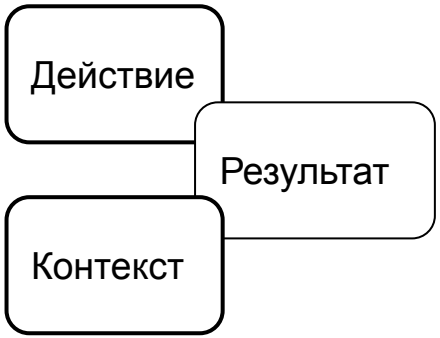
Sleepy multi-armed bandit

- Ручки иногда недоступны
- Нужно выбирать из тех что есть.

https://research.duolingo.com/papers/yancey.kdd20.pdf

Contextual Multi-Armed Bandits

- Действия бандита уже зависят от **контекста**
- Моделирует **рекомендательные системы**



https://research.google.com/pubs/archive/37042.pdf

Combinatorial Multiarmed Bandit (CMAB)

- Бандит выбирает не одну ручку а несколько

Wei Chen, Yajun Wang, Yang Yuan *Proceedings of the 30th International Conference on Machine Learning*, PMLR 28(1):151-159, 2013.

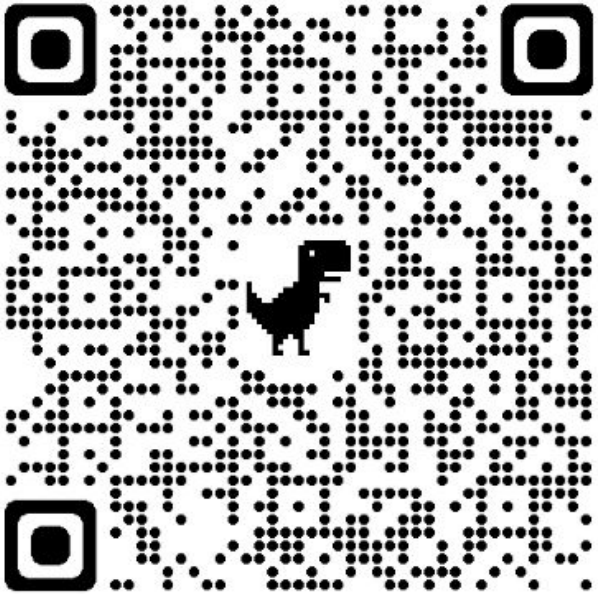
Где еще хороши бандиты

- Рекомендательные системы (Contextual Multi-Armed Bandits)
- Адаптивная маршрутизация

  Сеть как “казино”
- Формирование инвестиционного портфеля

  Asset allocation

Модели доступны в Colab



https://colab.research.google.com/drive/1RpzrMZW2hrjzLaC41DGCBY7Ze4nwRWwC?usp=sharing

Спасибо за внимание

Остались вопросы?