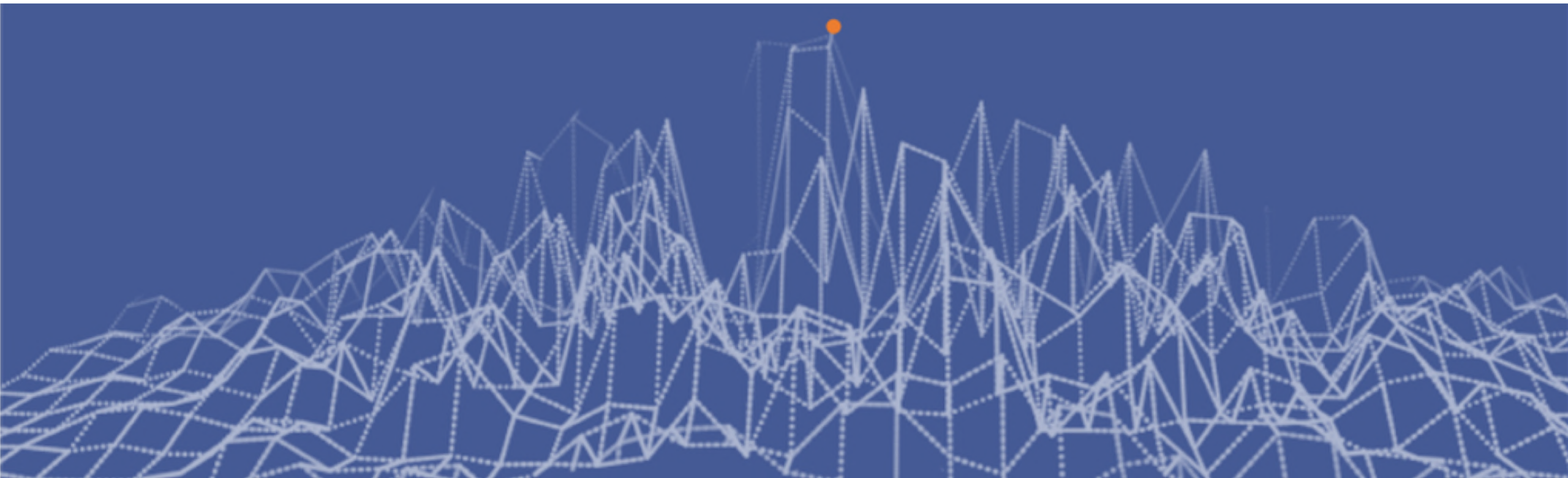


Интерпретируемый ИИ:

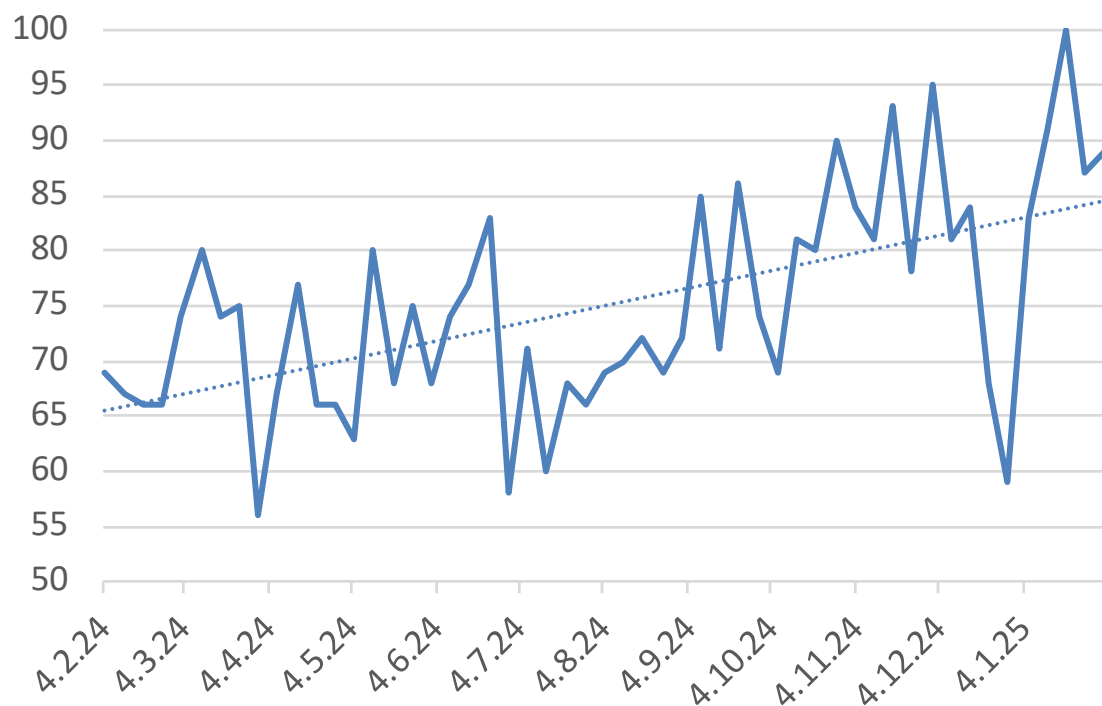
рынок, теория и практическая реализация
с применением MaxSAT-солвера OptJet



Интерпретируемый ИИ: есть ли запрос?

Интерпретируемый (объяснимый) ИИ – это набор методов и процессов, которые позволяют пользователям-людям понимать принципы получения результата алгоритмом машинного обучения и доверять решениям, принимаемым на его основе, формируя обоснованные ожидания.

Индекс активности: Explainable AI: Весь мир*



*<https://trends.google.com>

Кейсы от Amazon

- ✓ Рекомендательная система Personalize
- ✓ CV-сервис Recognition
- ✓ Прогнозирование временных рядов - Forecast
- ✓ Alexa
- ✓ Рекламная платформа
- ✓ Оптимизация цепочек поставок
- ✓ HR (после скандала внедрили XAI и поправили алгоритмы)
- ✓ SaaS XAI-сервис SageMaker Clarify

Два основных подхода к интерпретации ML-моделей: эвристический и строгий

- ML-Модели: нативно интерпретируемые (деревья решений, LR, ...) **vs.** black-box-модели (deep learning). Вторые требуют post-hoc методов интерпретации.
- Эвристические методы, как правило, универсальны, строгие – специфичны для конкретной модели
- Второй разрез для классификации – глобальность (объяснение модели, в целом) **vs.** локальность (объяснение конкретного инстанса)
- Распространенные эвристические методы - LIME (Local interpretable model-agnostic explanation), SHAP, Anchor – дают высокий процент ошибок на некоторых датасетах:

		LIME	Anchor	SHAP
adult	(5579)	61.3%	80.5%	70.7%
lending	(4414)	24.0%	3.0%	17.0%
rcdv	(3696)	94.1%	99.4%	85.9%
compas	(778)	71.9%	84.4%	60.4%
german	(1000)	85.3%	99.7%	63.0%

Строгий подход для ансамблевых моделей: релевантен ли пример?

- При переходе от дерева к ансамблю повышается точность модели, но теряется возможность нативной интерпретации
- Ансамбль случайных деревьев (RF) и градиентный бустинг (GBT) – основные модели класса

Популярные продукты на основе GBT

- ✓ Alibaba MaxCompute (предиктивная аналитика для ecom, финансов, логистики)
- ✓ Google Ads
- ✓ Stripe Risk (выявление мошенничества в реальном времени)
- ✓ Salesforce Einstein (скоринг лидов, сегментация клиентов)
- ✓ Palantir Foundry (предиктивная аналитика)

Немного вводных (1/2)

Ансамблевая модель решает задачу классификации:

Набор признаков: $\mathcal{F} = \{1, \dots, m\}$

Набор классов: $\mathcal{K} = \{c_1, \dots, c_K\}$

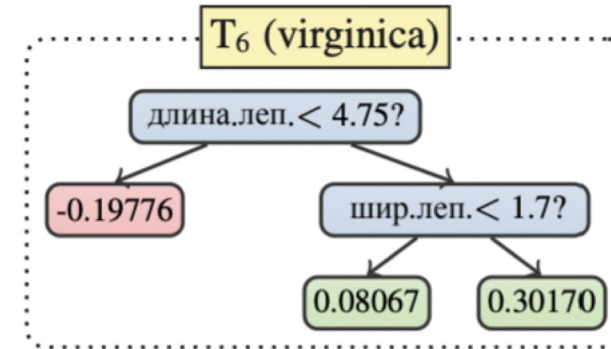
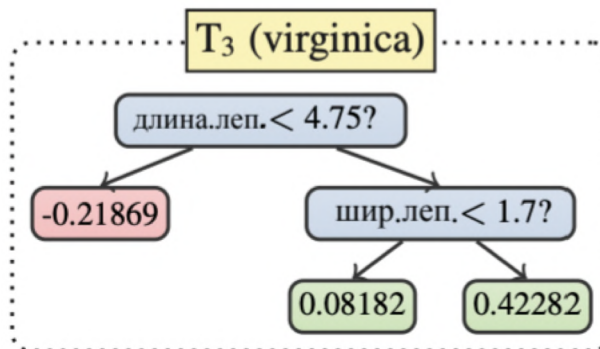
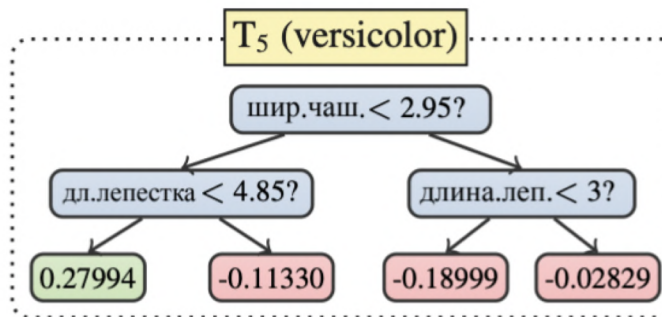
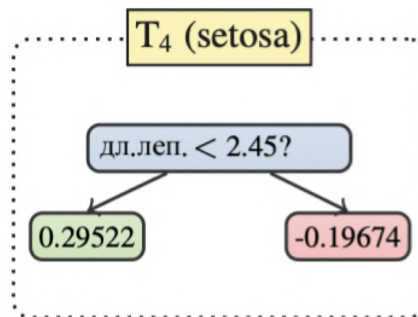
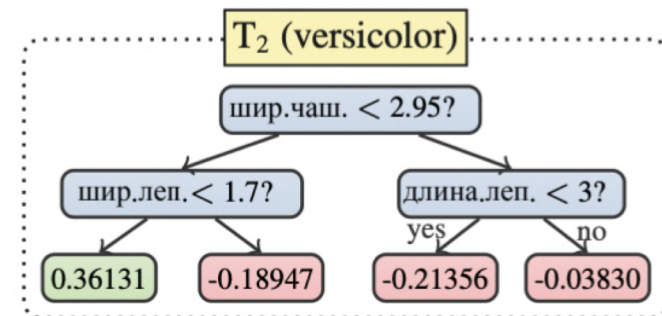
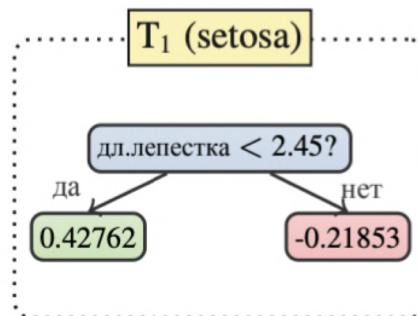
Любой признак $j \in \mathcal{F}$ имеет область значений D_j

Пространство признаков: $\mathfrak{F} = D_1 \times D_2 \times \dots \times D_m$

Инстанс (точка в пространстве признаков):

$\mathbf{v} = (v_1, \dots, v_m)$

Найти $\tau : \mathfrak{F} \rightarrow \mathcal{K}$



Классификация Ирисов по лепесткам и чашелистикам

Немного вводных (2/2)

- Подход к строгой интерпретации ансамблевых моделей основан на представлении ансамбля в виде набора (линейных) ограничений
- Цель: найти абдуктивные объяснения (интерпретации) \equiv простые импликанты*
- Применяемые локальные методы: MI(L)P, SMT, MaxSAT
- MaxSAT не только реализован в солвере OptJet, но и считает быстрее

*Сильно упрощая: Если [**импликанта**] то [**принадлежность к конкретному классу**].

Простая импликанта – та, которая использует минимальный набор признаков для однозначной классификации

SAT: маленький набор кубиков для большой башни

$x \in \{0,1\}$

$\wedge, \vee, \neg, \rightarrow, \leftrightarrow$

SAT: маленький набор кубиков для большой башни

$x \in \{0,1\}$

$\wedge, \vee, \neg, \rightarrow, \leftrightarrow$



SAT: маленький набор кубиков для большой башни

$x \in \{0,1\}$

$\wedge, \vee, \neg, \rightarrow, \leftrightarrow$



MaxSAT-представление ансамбля: основные идеи (1/2)

Проблема 1: представить непрерывную область значений признаков с помощью булевых переменных

Решение:

- рассматривая **весь ансамбль**, для каждого признака j сформировать упорядоченный набор пороговых значений $s_{j,k}$ и связанный с ним набор интервалов $\{I\}$
- Задать булевы переменные* $l_{j,k}: l_{j,k} = 1 \text{ iff } v_j \in I_k$ и $o_{j,k}: o_{j,k} = 1 \text{ iff } v_j < s_{j,k}$

*можно использовать одну переменную, но кодировка получится менее экономной

MaxSAT-представление ансамбля: основные идеи (2/2)

Проблема 2: соотнести целевую функцию и простую импликанту

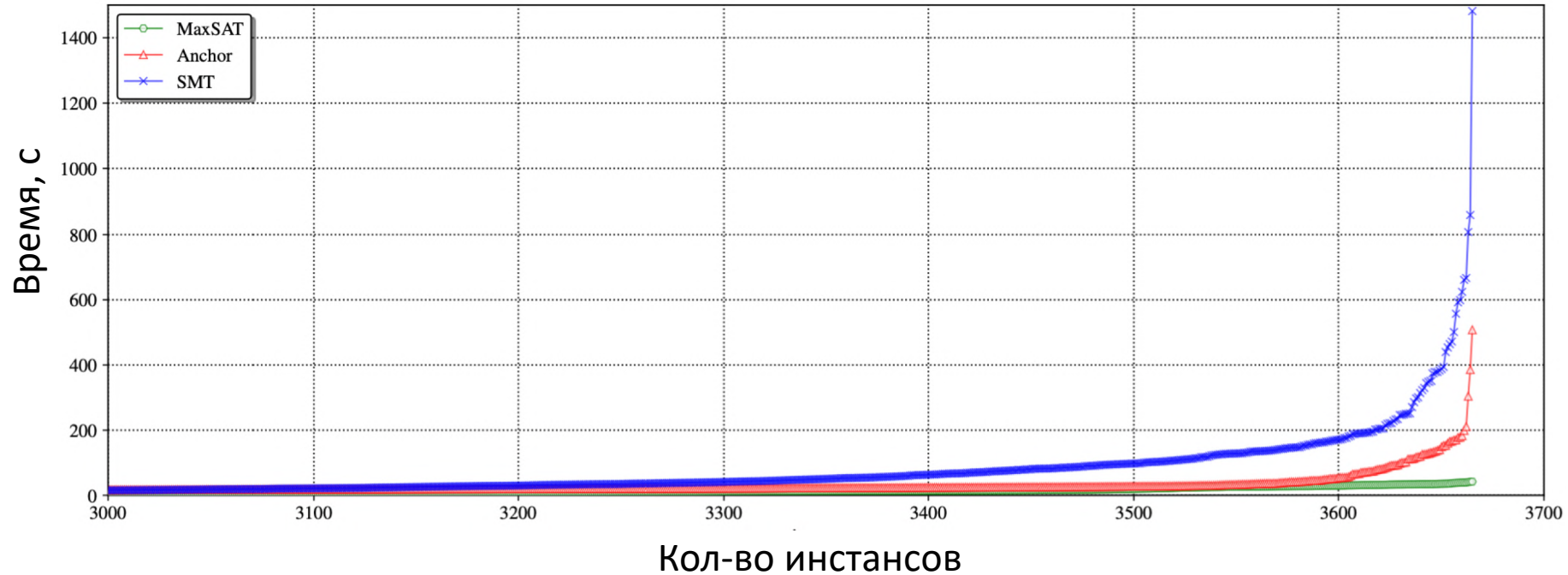
Ход мыслей: проверить, возможна ли ошибочная классификация для каждого возможного объяснения χ (для каждого возможного набора использованных для этого объяснения признаков), т.е. существует ли инстанс, который соответствует χ , но имеет больший суммарный вес признаков для некоторого класса $l \neq i$

Решение: задать целевую функцию в виде $\Sigma_l - \Sigma_i \rightarrow \max$

Эффект:

- Декомпозиция задачи (для каждого i рассматриваем все $l \neq i$)
- Не нужно искать глобальный оптимум (останавливаем солвер, как только найдено отрицательное значение ЦФ, используем особенности алгоритма для оценки нижнего предела ЦФ и останавливаем солвер, если $LB > 0$)

Результаты: MaxSAT рулит



MaxSAT на порядок быстрее, чем Anchor, который на порядок быстрее, чем SMT, который быстрее, чем CP

```
--- [stop_running]  
[2025-02-05_01:34:22.544](00:00:00.211){NOTIC} |[core] Control thread : finished  
successfully [Thread id = 34944]
```

Пример проекта, использующего MaxSAT: оптимизация заказа и погрузки вагонов в конвертерный цех

Контекст

- Заказ вагонов для подачи в конвертерный цех и определение комбинации слябов для погрузки в каждый вагон осуществлялась вручную
- Показатель стат.нагрузки находился на уровне 65 тн/вагон при средней грузоподъемности 69 тн/вагон
- Объем отгрузки составлял 300-450 вагонов в сутки в 33 типа подвижного состава (полувагоны разной грузоподъемности, габаритов, кодов годности) по 300 схемам погрузки

Периметр

- Оптимизация и автоматизация расчёта заявок на вагоны по 4 тупикам цеха с учётом складского остатка и характеристик слябов (вес, геометрия, марка стали), сбытовых заказов и направлений отгрузки, наличия и характеристик вагонов
- Оптимизация выбора схемы погрузки и слябов под фактически поданные вагоны с учётом расположения слябов
- Оцифровка ограничений по рисункам схем погрузки (ТУ, МТУ)

Результат

- Увеличение средней загрузки полувагонов (тн/вагон) на 4%, оптимизация транспортных затрат на привлечение вагонов и платежи РЖД
- Оптимизация и расчёт модели за 5 мин.
- Автоматизация взаимодействия участка отгрузки цеха и ж/д станции по заявкам через интерфейс системы, ведения модели данных и НСИ по ограничениям и схемам погрузки

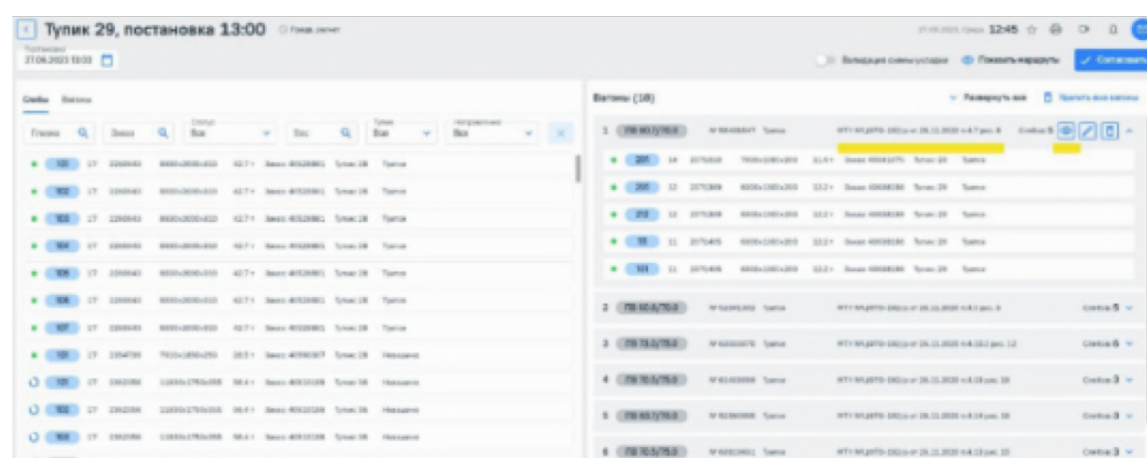
Ведение ограничений и схем погрузки



Рисунок 8

1 – упорный брусок сечением не менее 100х60 мм
2 – распорный брусок сечением не менее 120х100 мм

Экран заказа вагонов и оптимизации погрузки



QuSolve - разработчик промышленного солвера OptJet

- ✓ Российский программный продукт (в реестре)
- ✓ Позволяет решать оптимизационные задачи с более 10 млн переменными
- ✓ Поддерживает LP, MIP, MILP, QP, CP, MaxSAT
- ✓ Разработан на C++ для высокой скорости решения задач on-premise и on-cloud
- ✓ Имеет лучшую производительность, чем Cplex и Gurobi в целочисленных задачах
- ✓ Оперативная поддержка на русском языке
- ✓ Создан и совершенствуется математиками – разработчиками прикладных моделей
- ✓ Удобный API и библиотека на Python + инструменты логирования и отладки



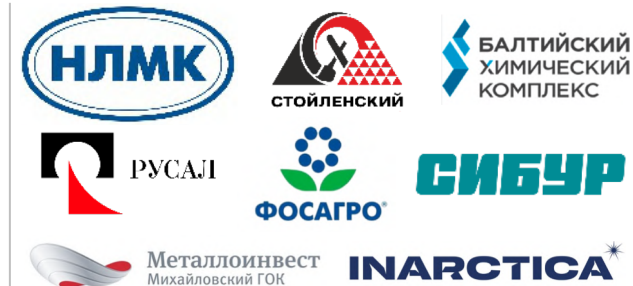
QuSolve имеет большой опыт проектов по математической оптимизации и интегрированному планированию

Почему QuSolve?

- ✓ >15 оптимизационных моделей планирования производства и цепочек поставок с применением солверов OptJet, IBM CPLEX + всей линейки open-source солверов (в т.ч. OR-TOOLS, HiGHS, CBC)
- ✓ Клиенты получают реальные экономические результаты, которые многократно окупают стоимость наших услуг (напр., увеличение объёма производства на 7%, снижение простоев)
- ✓ Комплексная экспертиза команды в математике, бизнесе и разработке ПО + свой солвер OptJet
- ✓ Мат.оптимизация – это наш профиль, мы берём модели на долгосрочную поддержку и развитие

Опыт внедрения оптимизационных моделей и IBP

Разработка оптимизационных моделей, внедрение систем **планирования производства и цепей поставок** (S&OP, MPS, APS, SCP, IBP)



Оптимизация управления парком **транспорта, диспетчеризации и маршрутизации**



Оптимизация **режимов работы оборудования**



Оптимизация **расписаний** и назначения исполнителей



Список использованных источников

1. <https://www.ibm.com/think/topics/explainable-ai>
2. <https://chat.qwenlm.ai/>
3. <https://qusolve.ru/documentation/>
4. Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System.
5. Ignatiev, A., Izza, Y., Stuckey, P., Marques-Silva, J. 2022. Using MaxSAT for Efficient Explanations of Tree Ensembles.
6. Ignatiev, A. 2020. Towards Trustable Explainable AI.
7. Rai, A. 2019. Explainable AI: from black box to glass box

QuSolve

ООО «Квантовые системы»

г. Москва, Пресненская наб., 12

+7 (495) 142-51-61

info@qusolve.ru

