Портфолио

1. Достижения в школе и университете
2. Основная работа по базам данных пригородных и пассажирских перевозок.
3. Обработка данных по увольнениям сотрудников
4. Обработка данных по загрузке сервера и очередям
5. Оптимизация ценообразования для поездов
6. Упрощение работы с расписанием поездов
7. Оптимизация работ локомотивов
8. Нейросетевое прогнозирование
9. Выявление мошенничества на сахалинской ЖД
10. Собеседование в Газпром на должность программиста-математика
11. Написание книги по астрофизике
12. Достижения в школе и университете

Во время обучения в школе, в ФМШ-30, многократно участвовал в городских олимпиадах по математике, физике, химии и программированию. По программированию дважды имел диплом 3 степени, по химии ничего, по физике 3 диплома 3 степени и в 11 классе 1 степени с рекомендацией поехать на всесоюзную, но пришлось отказаться из за математики.

По математике в школе дважды имел диплом 1 степени по городу, в 10 и 11 классе, после чего ездил на всесоюзные олимпиады в Смоленск и Алма-Ату, оба раза получал диплом 3 степени.

В университете на 1 курсе прошёл неформальную внутри МатМеховскую олимпиаду, занял 3 место среди студентов всех курсов, и официально был признан лучшим студентом-первокурсником, с выдачей премии.

Со 2 семестра 1 курса в составе группы лучших студентов МатМеха был переведён в ЛОМИ-группу (половину лекций слушали непосредственно в бывшем ЛОМИ, ныне ПОМИ РАН имени Стеклова, там же сдавали экзамены), которая существовала до окончания 2 курса.

На 2 и 3 курсах участвовал в городских олимпиадах студентов по математике, выиграл 1 место, после чего на 3 курсе отправился на олимпиаду всесоюзную в Челябинске, где получил 3 место (отличительная особенность этой олимпиады – было единственное 1 место, единственное 2 и единственное 3 на всю Россию, все 3 места заняли наша группа МатМеховцев).

На 3 курсе участвовал в студенческой олимпиаде по механике, и был единственным от МатМеха студентом, отправленным на городскую олимпиаду, проходившую в Политехническом университете. Там занял 1 место, решив все задачи, но так как олимпиада оказалась командной (5 человек от каждого ВУЗа) а я был один, меня засчитали вне конкурса.

Дипломную работу писал на кафедре теории вероятностей и мат статистики, по теме «поведение P-значений 2 типа при альтернативе», работа в итоге была опубликована в итоговом журнале работ ПОМИ РАН 1997г. Закончил университет с отличием, по баллам был 3-4-й диплом на всём Матмехе.

На 1-4 курсах университета помогал вести математический кружок в ФМШ-239. На 5 курсе в 1996г проходил педагогическую практику в Финансово-Экономическом университете, преподавал практику по математике, в результате руководимая иною группа на экзамене показала средний балл свыше 4.5, в трёх других группах максимум был 3.7.

1. Основная работа по базам данных пригородных и пассажирских перевозок.

Первая главная работа на РЖД состояла в том, чтобы из базы данных проданных билетов (пригородных и пассажирских) создать базу агрегированной ежемесячной и ежесуточной отчётности, с целью выдачи справок и отчётов нескольких разных типов для вышестоящего начальства, для дальнейшей оптимизации перевозочного процесса.

Основные параметры входных данных. По пригороду база данных продажи билетов на ОктЖД– это ежемесячно пополняемая база данных, объёмом от 50тыс строк (1998г) до 3млн строк(2023г) в месяц, постепенно увеличивающейся детализации, бессрочного хранения. По пассажирским перевозкам – база данных до 1млн строк в сутки, сроком хранения 2 года, по ней была потребность в самых простейших агрегированных данных, и в множестве реестров разных льготников.

Исходные пригородные данные представляли из себя 3 большие таблицы – одна по всем проданным билетам, а остальные две – по каждому из проданных билетов указан маршрут движения каждого конкретного пассажира, и деление стоимости его проезда по регионам (стоимость билета складывается из стоимостей по регионам, назначаемых по согласованию с этими регионами), и были почти непригодны для выдачи серьёзной отчётности. Дополнительные таблицы имели в 3-5 раз большую длину, нежели основная таблица продажи билетов.

Для ускорения и упрощения работы первым шагом была проведена нормализация данных – осталась одна таблица собственно данных, и несколько справочников уникальных вариантов вида билета, места его продажи, и собственно маршрута (включая сюда и разбиение стоимости по регионам). В результате самая длинная накопительная справочная таблица (маршрутов) за 25 лет работы имела в объёме в 2-3 раза меньше строк данных, нежели в сырых данных в неё записывалось за 1 месяц. Вторым результатом нормализации была экономия памяти, что серьёзно сказывалось на дальнейшей глубокой обработке данных.

По пригороду в результате была создана структура из нескольких видов агрегатов, для принципиально разных типов отчётов, и справочных таблиц для них. Основная потребность СЗППК была в справках типа «кто сколько продал билетов разных типов, или перевёз пассажиров, в разные периоды времени, и сравнение с предыдущими периодами». Для этих целей была разработана структура, состоящая из нескольких таблиц-справочников, и таблицы агрегатов примерно такой структуры:

-отчётный месяц

-станция (отделение, дорога, административный район) совершивший действие.

-вид билета (разовый или абонемент, льготность, способ оплаты, срок действия).

-произведённое действие (наименование = продажа билета, получение денег, начисление суммы потерь, отправка пассажира или пассажирооборот и многое другое)

-сам вычисленный объём (билеты, выручка, потери, сформированные или отправленные пассажиры или пассажирооборот)

-дата и номер расчёта

В результате, таблица агрегированных данных имела объём порядка 100тыс строк в месяц, исполнение любого запроса укладывалось в требуемые системой 0.2 секунды, иногда при сильной загрузке сервера возрастая до 1сек.

Для оптимизации создания SQL запросов для разных справок был написан один универсальный запрос, выдающий все возможные параметры во всех возможных разрезах, какие только требовались для отчётностей (иногда запрос приходилось пополнять, при появлении новых требований от заказчика), после чего специально написанная программа «отсекала всё лишнее» от него, оставляя только те параметры, которые нужны в данной конкретной справке – и именно такой запрос уже и вкладывался в листинг конкретной программы, выдающей конкретную справку. По сути, то же самое делает система SAP HANA, с которой пришлось познакомиться в компании Транс-ИТ в 2015-2017гг, только у нас это работало уже в 2007г.

Благодаря предложенной структуре, содержащей в себе дату и номер вычисления, был сильно упрощён процесс поиска разных ошибок программного обеспечения, которое приходилось регулярно дорабатывать под новые требования. При изменении ПО и пересчёте агрегатов за прошлые месяцы, программа сперва насчитывает по каждому показателю полный объём, затем вычитает объёмы, уже записанные в базу в результате предыдущих расчётов, после чего выкидывает все строки с нулевыми данными, и записывает в агрегаты только результат изменения, с соответствующей датой и номером расчёта. По этой причине после пересчёта данных всегда можно было посмотреть, что и насколько изменилось, и если изменилось не то что надо, или не на столько, насколько ожидалось – сделать откат назад, и ещё раз откорректировать ПО.

Помимо этих основных агрегатов, так же имелись агрегаты для отчётов видов - «сколько пассажиров проехало по маршруту от станции А до станции Б», «какова была ежесуточная отправка пассажиров от каждой станции, и как менялась средняя дальность поездки» (исходные данные приходили раз в месяц, но начиная с 2010г уже с разбивкой по дате начала действия билета) – по ним были сделаны подобные же агрегаты, чуточку отличающейся структуры, с целью минимизации объёмов в каждой таблице агрегатов.

Самая сложная таблица была создана для справок типа «каков был пассажирооборот, и сколько уникальных пассажиров проехало за отчётный период по разным регионам сети ОктЖД» - имея в виду, что справкой пользовались несколько компаний (СЗППК, МТППК, ДОСС), и каждая из них хотела видеть одни и те же данные в разных сверлениях, описание которых укладывалось в отдельный справочник, имевший рекуррентную структуру ссылок. Справка выдавалась в виде {значение по дороге, её деление на нужные районы {значение по району, его деление на подрайоны {…}}} – и так от данных по всей дороге шло постепенное уточнение данных до каждого мельчайшего перегона (=отрезок ЖД между двумя соседними станциями). Математическая сложность этой отчётности заключалась в том, что в числе прочего указывать надо было уникальных проехавших пассажиров, то есть, если пассажир проехал от А до Б через 20 перегонов, то в каждом из них он должен был быть посчитан 1 раз, и в любом регионе хоть как-нибудь задевавшем его маршрут движения – он тоже считался 1 раз, а не по количеству перегонов, входящих в рассматриваемый участок. То есть, программе надо было выдавать по каждому региону значение, вычисляемое не суммированием, и при этом не на основе сырых данных, а по очень небольшим агрегатам и произвольно заданному пользователем справочнику. В результате, ПО для выдачи справок было основано на теории плоских графов, при этом агрегаты содержали данные только по каждому перегону (их на ОктЖД всего около 1500 штук), без сведений о каких либо более крупных сущностях, благодаря чему справка отрабатывала за считанные секунды.

Аналогичные базы агрегированных данных были построены по пассажирскому движению, но они были мало востребованы, в связи с отсутствием частных компаний, осуществляющих на них работу, а руководство централизованной РЖД оптимизацией своих процессов не занималось.

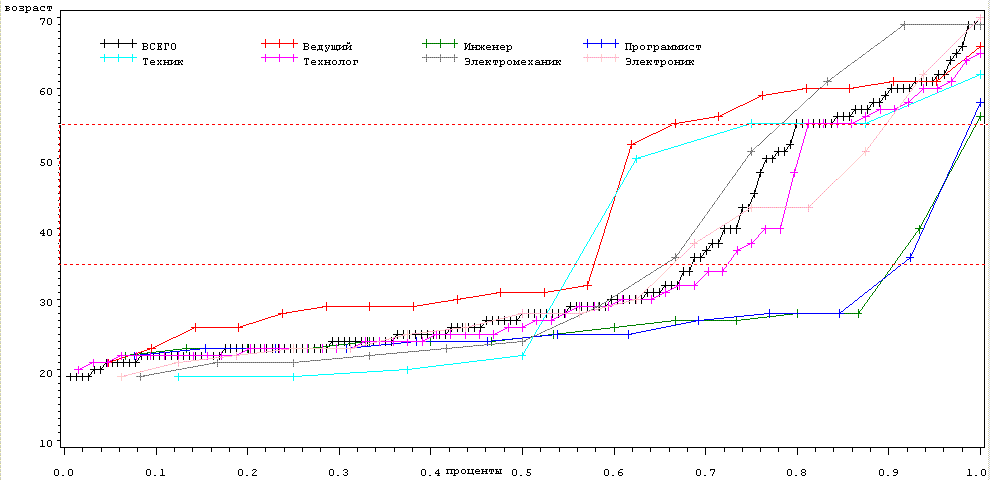
Весь комплекс расчётов данных по пригородным перевозкам для ОктЖД был написан мною без посторонней помощи на языке SAS, с подключением к хранилищу исходных и агрегированных данных на DB2-сервере, и с выдачей справок по SQL запросам в экселевские формы – в этой системе я осуществлял все работы до разрисовки экселевских отчётов, которыми занимались уже другие программисты.

Примерно в таком же виде базы данных планировалось сделать в до сих пор разрабатываемой системе Экспресс-НП, осуществляющей выдачу отчётности уже на всю сеть РЖД.

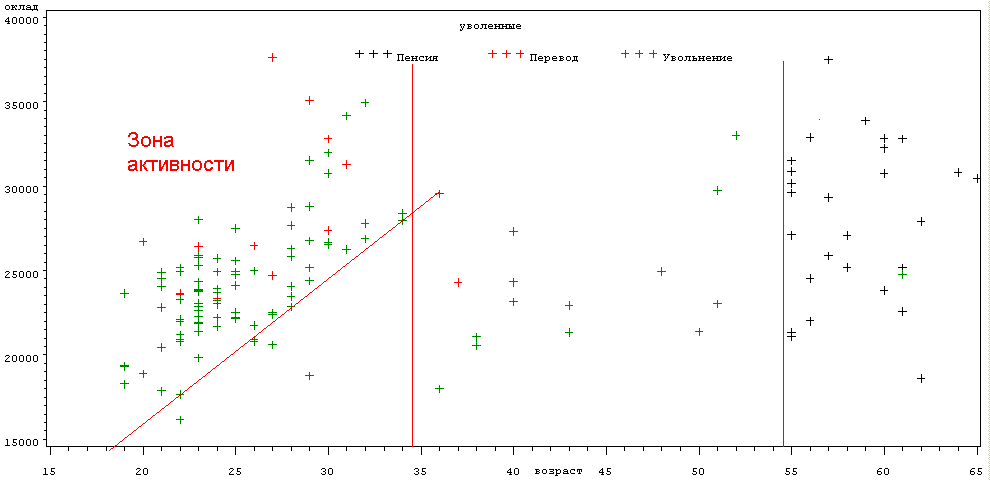
1. Обработка данных по увольнениям сотрудников

В 2012г ещё в бытность работы в ИВЦ ОктЖД, от отдела кадров поступил запрос – выяснить причины и критерии увольняемости сотрудников РЖД, с целью дальнейшей оптимизации работы с персоналом. На вход была дана база данных всех сотрудников РЖД с многочисленными параметрами на тот момент, и отдельно список всех уволившихся за последние 15 лет, с их показателями на момент увольнения.

После проверки всех данных, отсева увольнений по причине перевода на другую должность, смерти или ухода на пенсию, проверка всех данных по увольнениям почти ничего не смогла выявить сразу, по каким бы разрезам не происходило рассмотрение данных. Типичный пример – график вероятности увольнения сотрудника в течение года в зависимости от возраста и слова в названии профессии.



Но затем было произведено приближённое нормирование данных, приведение их к текущему состоянию, а именно – за рассматривавшийся период истории зарплаты возросли более чем в 10 раз, и потому зарплаты уволившихся и оставшихся сотрудников были приведены к единому значению, а именно – домножены на экспоненту с показателем инфляции и временем от увольнения до рассматриваемого момента. В результате этого выявилась чёткая закономерность, выраженная графиком



Из этого графика стало ясно видно, что на пенсию в основном выходят по возрасту (но есть и иные, например по инвалидности), а по переводу и уход насовсем из РЖД подчиняется чёткому критерию – только если сотрудник имеет хорошую специальность и возможность роста (= зарплата выше линейной функции от возраста, и возраст до 35 лет, то есть молодой специалист), тогда он часто увольняется, вне зависимости от всех прочих показателей (разумеется зарплата коррелирует с профессией). А в противном случае, если сотрудник неконкурентоспособен на рынке труда, его увольняемость минимальна.

В этом виде ответ на поставленный вопрос был более чем хорош для отдела кадров, на чём задачу посчитали решённой. Какие из этого были сделаны управленческие выводы – мне неизвестно.

(Иллюстрации взяты из сохранившегося файла презентации итоговой работы начальству)

1. Обработка данных по загрузке сервера и очередям

Примерно в 2007г поступило задание – проанализировать причину проблем с вводом данных в систему ИОММ (Интегрированная Обработка Маршрутов Машинистов), выливавшуюся в возникновении длинных очередей к серверу обработки данных, на обработку запросов ввода данных суточных работ машинистов.

То есть – для учёта и контроля работы машинистов диспетчера вручную вводили на локальных компьютерах данные о работе машиниста, нажимали «ввод», задание отправлялось на сервер, и там оно находилось в очереди до момента его исполнения, иногда в течение десятков минут. При этом все логи работы сервера говорили о том, что каждый конкретный маршрут в норме обрабатывается в течение 1-2 секунд, а в часы пик – до 10 секунд, но очереди не давали нормально работать всей ОктЖД.

Для решения проблемы были собраны логи сервера, показывающие какой процент загрузки у него был в каждый момент времени, когда какая программа запускалась, и конкретно для ИОММ для некоторых запросов – в какое время запрос отсылался на сервер (сам лог записывал только момент начала и окончания обработки одного маршрута, не учитывая время его стояния в очереди).

Для прогнозирования времени ожидания запроса в очереди был применён раздел математики - теория очередей. И в результате была написана программа, которая с большой точностью предсказывала, когда именно поступивший на обработку запрос достигнет своей очереди и будет обработан. Программа использовала всего 2 легко вычисляемых из таблицы логов параметра – сколько времени из последних 10 минут сервер занимался обработкой запросов, и сколько в результате запросов было обработано.

Обработка данных позволила выявить, что сервер вообще никогда не работал более чем на 50% своей мощности, очереди образовывались только на уровне 25 или 50% загрузки сервера, но при этом очень часто при таком уровне загрузки очередей не было. Более детальное рассмотрение периодов возникновения очередей показало, что оно возникает только в момент большой занятости сервера двумя программами из порядка 20 штук, постоянно работавшими на нём, и почти не возникает при других программах.

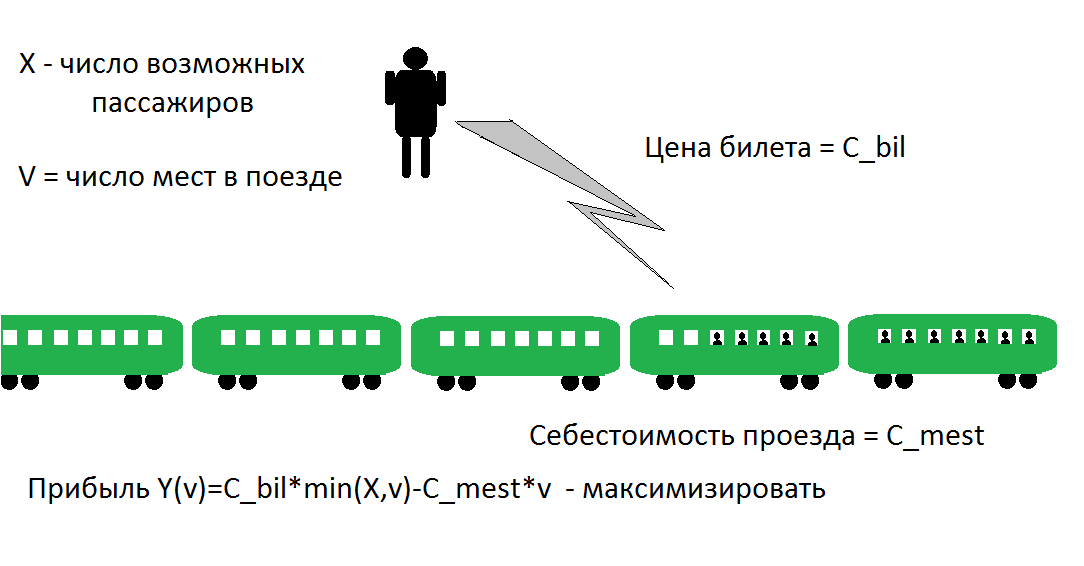
В результате было высказано предположение, что сервер представляет из себя 4-процессорный компьютер, у которого разные программы исполняются на разных процессорах, всегда на одних и тех же. Когда это предположение было передано администраторам, выяснилось, что это был на тот момент новейший однопроцессорный сервер с 4 ядрами, работавший в точности как было предсказано. И так как программам никто не присваивал никакого приоритета или места их работы, то они разбили «зоны ответственности» согласно датам установки программ на сервер, выполняясь каждая строго на одном единственном своём ядре, и никак не загружая все соседние ядра. При этом одно из четырёх ядер вообще никогда не исполняло никакой полезной работы.

После исправления этого недочёта, программа ИОММ работала уже без очередей, исполняя запросы «на лету», работа всех прочих программ тоже улучшилась.

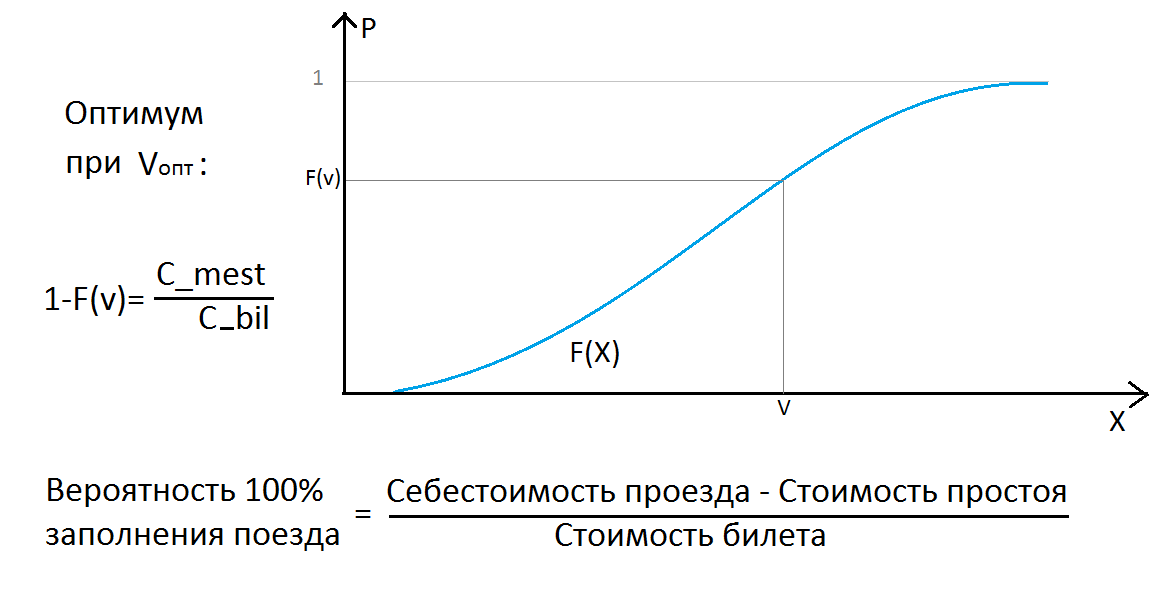
Результатом работы стало предложение мне выступить со своим докладом на слёте молодых специалистов ОктЖД, проходившем осенью 2008г в гостинице Прибалтийской, после чего, по непроверенным сведениям, мой доклад был опубликован в Вестнике РЖД. Другим результатом этой работы стало присвоение мне категории ведущего программиста, с повышением разряда по тарифной сетке РЖД.

1. Оптимизация ценообразования для поездов

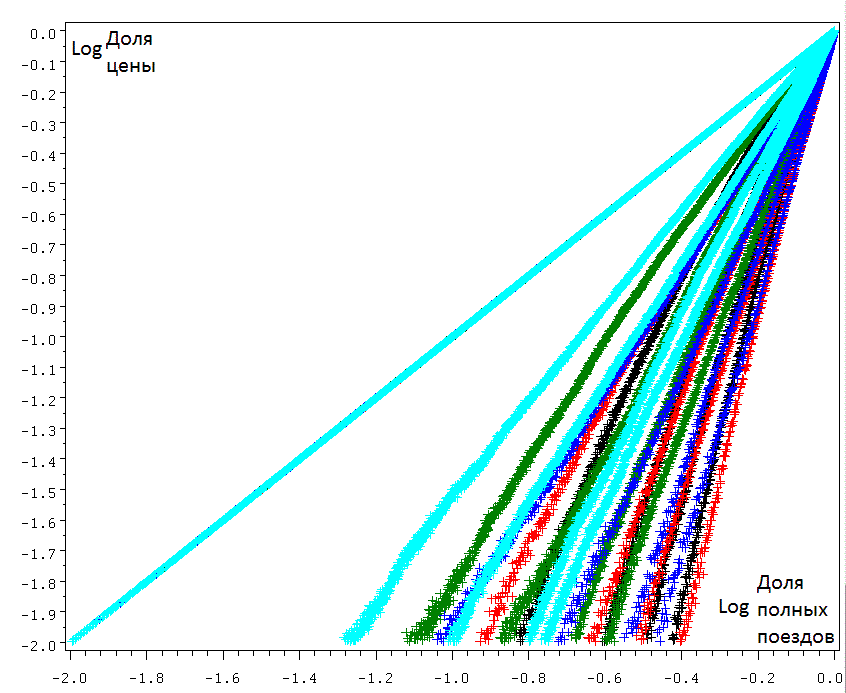
В 2013г поступило от руководства ИВЦ ОктЖД провести семинар по теории вероятностей, и её возможном применении в работе ИВЦ, с вольным выбором примеров. В качестве одного из возможных примеров я выбрал задачу – как максимизировать возможную прибыль перевозчика, пускающего в рейс поезд имеющий V свободных мест, из которых пассажирами будет выкуплено Х штук – являющееся случайной величиной с неизвестным распределением. Перевозчик имеет право изменять число мест, а так же (не всегда) стоимость билетов (очень часто стоимость проезда регулируется государством).



В полном виде решить такую задачу вряд ли возможно, но есть частные случаи – варианты покупки билетов без отложенного спроса и конкуренции, то есть когда пассажирам необходимо поехать «именно сегодня и только на этом поезде», а завтра будет уже поздно. В таком предположении был проделан математический расчёт, и выяснилось, что критерий оптимальности предложения количества мест изумительно прост и легко вычислим. А именно – оптимальным будет именно такое количество предложенных мест, или цены их продажи, когда вероятность, что поезд отправится битком забитым (выкуплено 100% мест) в точности равна отношению себестоимости билета к предложенной стоимости розничной продажи. Так же формула может учесть и стоимость простоя вагона – когда она ненулевая, например по причине потребности обслуживания стоящих в отстойнике вагонов.



Дальнейшее исследование с помощью математического моделирования показало, что при самом широком спектре функций распределения пассажиров, и при разных критериях отложенного спроса и конкуренции, формула оптимальности изменяется весьма незначительно.



Результатом озвучивания этой работы стало то, что пассажирская компания ДОСС заказала во всех своих справках пассажирской отчётности за длительные периоды ещё одну графу – «количество поездов, отправленных заполненными целиком», подразумевая под этим продажу свыше 95% предложенных в продажу билетов, и разумеется, долю полностью заполненных поездов среди всех отправленных. По всей видимости, этот оптимизирующий коэффициент используется и поныне.

(Иллюстрации взяты из сохранившегося файла презентации итоговой работы начальству)

1. Упрощение работы с расписанием поездов

В 2016г, во время непродолжительной работы в компании Транс-ИТ, у коллег из соседнего отдела возникла серьёзная проблема с мощностями серверов, хотя работали тогда на SAP HANA, с наимощнейшим из возможных на тот момент серверов.

Ситуация была такова. Исходно на РЖД было расписание поездов – порядка 3000 регулярных поездов, ходящих по приблизительно постоянным маршрутам с приблизительно постоянным графиком, но с множественными мелкими отличиями. В старой системе за расписания отвечали множество файлов, где каждому поезду устанавливалось, по какому из возможных графиков он пойдёт в каждый конкретный день. В новой системе это всё должно было быть в базе данных, и поэтому, так как база должна была быть единообразной, таблицу создали примерно в таком виде

-Номер поезда (около 3000 штук)

-дата отправления (от -50 до +360 дней от текущей даты)

-станция маршрута со всеми её параметрами (от 20 до 1000 строк записи, в зависимости от длины маршрута).

В итоге результирующая таблица имела свыше 100млн строк записей, что создавало простую структуру с простой адресацией обращений, но однако же из за её размеров, и требований постоянной (тысячи раз в минуту) её склейки с приходящими запросами данных, просто обрушивало всю систему. Результатом стал паралич процесса разработки новой системы Экспресс-5 на SAP HANA.

Когда с вопросом, что с этим можно поделать, обратились ко мне, я предложил и частично самостоятельно реализовал следующий алгоритм. База данных всех расписаний делилась на 2 отдельные таблицы. Первая таблица – уникальные расписания – при вводе в базу данных нового расписания движения конкретного поезда на конкретную дату проверяется, не бывало ли прежде именно такого уникального расписания; если оно встречалось – брался его уникальный номер, если нет – оно вносилось с новым номером, и брался уже он. Вторая таблица – соответствия конкретного номера поезда с конкретной датой отправления уникальному номеру расписания. В итоге, вся таблица уникальных расписаний содержала всего лишь порядка 1млн строк, а таблица дат отправлений поездов – ещё того меньше, и такая связка работала влёт, почти не загружая процессор.

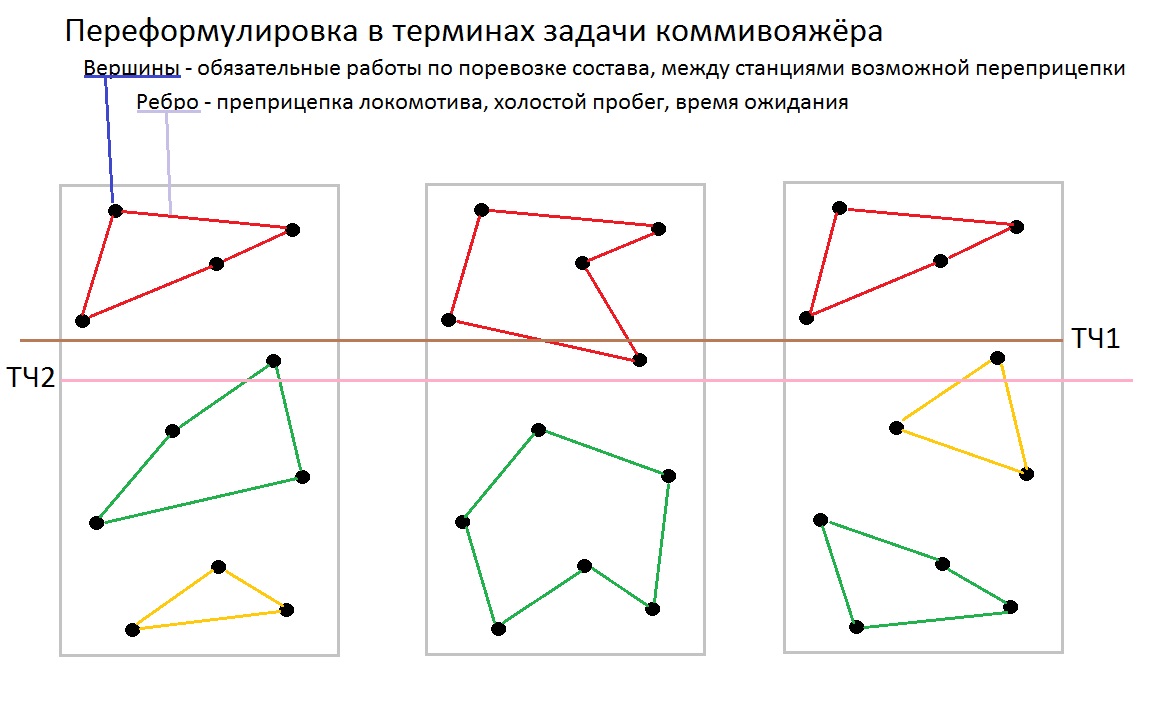
Самой сложной алгоритмической частью стало выяснение – как узнать, имеется ли новое расписание маршрута движения поезда уже известным в таблице всех уникальных расписаний, или его надо туда дописать? Дело в том, что если новое расписание имеет 1000 строк, а таблица уникальных 1млн, то прямая склейка со всеми уникальными расписаниями работает слишком долго – а таких проверок надо делать многие десятки тысяч ежесуточно (если на месте решают, что в данную дату обычно проходящий без остановки поезд надо остановить, и прицепить к нему вагон, либо отцепить, то это уже изменение расписания движения поезда, и таких изменений могут быть десятки у каждого грузового поезда; помимо этого, из за аварии или прочих факторов поезд могут пустить по другому маршруту или в другое время). Для этой цели был придуман алгоритм хеширования – для каждого уникального расписания, по сути являющегося текстом состоящим из множества отсортированных строк единой структуры, вычислялась его хэш-функция, и то же самое вычислялось для новоприбывшего кандидата. Если значение хэш-функции было новым, значит и расписание гарантированно новое, а если нет – то на всякий случай делалась проверка точного соответствия расписаний. Такой алгоритм вычислений работал уже практически мгновенно, позволяя провести идентификацию новоприбывшего расписания среди списка уникальных за микросекунды.

В итоге данный алгоритм в компании Транс-ИТ так и не был введён, потому что именно тогда решением свыше компания была расформирована, а наш отдел перевели во ВНИИЖТ. Но сам алгоритм нормализации маршрутов через процедуру хеширования был мною опробован в старой, всё ещё работающей базе на DB2, а после реализован в системе Экспресс-НП для пригородного и дальнего сообщения – с его помощью осуществляется быстрая нормализация исходных «сырых» данных на SQL-запросах, без привлечения сложных вычислений на других языках программирования. Фактически, этот простой алгоритм осуществляет ту же нормализацию баз данных, которая прежде была написана на SASе для пригородных данных, с использованием сложных и медленных процедур, никак не укладывавшихся в SQL-запросы и требовавших специальной обработки.

1. Оптимизация работ локомотивов.

Исходной предпосылкой для проведения работы послужила статья в газете Гудок от 3 февраля 2014г “Плясать от тепловоза” ( <http://www.oktmag.ru/archive/?aday=31&amonth=01&ayear=2014> , ныне в интернете недоступна по давности лет). В ней рассказывалось, что на одном участке Октябрьской ЖД (Янисьярве – Лодейное поле – Волховстрой) провели ручную оптимизацию движения локомотивов, в результате чего высвободилось несколько локомотивов, и имелся большой экономический эффект. Насколько я понял из статьи, хоть была проведена и нетривиальная работа, но в целом её можно описать достаточно кратко – поиск оптимального маршрута работы нескольких локомотивов, для достижения минимума издержек, который вполне можно алгоритмизировать.

Когда вскоре начальством был брошен клич «Мы ищем идеи для развития», я взял за основу эту статью, переформулировал задачу в терминах всем известной «задачи коммивояжёра», и запрограммировал её оптимизацию методом муравьиной тропы (алгоритм был почёрпнут из статьи в журнале Компьютерра <http://www.computerra.ru/60971/ants/> ).



В чём суть вопроса. По ЖД ходят поезда и возят грузы, но ходят они не самостоятельно, а их везут локомотивы. Один поезд может пройти путь от Москвы до Сеула почти 10000км за неделю, при этом в разных местах его будут тянуть 5-20 разных локомотивов, в зависимости от текущей ситуации. И в каждом локомотиве работает своя локомотивная бригада, которая в большинстве случаев в конце смены должна вместе с локомотивом оказаться пункте смены бригад, по месту их проживания, а сам локомотив перед новыми сутками работы должен там же пройти ТО, и возможно заправку. И при этом каждый локомотив не может покидать зоны действия своей ТЧ (тяговой части) – например локомотив на электротяге не может заехать в неэлектрифицированную зону. И между проведением разных работ (перевозкой поездов) локомотив либо находится на месте (простой), либо перемещается в другой пункт для прицепки к другому поезду (холостой пробег). В итоге, разным образом организовав работу локомотивов можно как минимум удешевить работу, благодаря уменьшению холостого пробега (который существенно дороже простоя), а можно и вовсе избавиться от потребности в некоторых локомотивов, переведя их в горячий или холодный резерв (это ещё дешевле), либо переведя на другие более востребованные участки РЖД (выполнение полезной работы вместо бесполезного стояния).

В итоге мною была написана программа на языке SAS, которая сперва научилась проводить оптимизацию работ при случайном разбиении сети РЖД на разные ТЧ, со случайным расписанием движения составов, в рамках которого надо было оптимизировать перевозку 100 поездов, маршруты которых были разбиты на 1000 отрезков (аналог задачи коммивояжёра для 1000 городов). А когда программа была полностью отлажена, и протестирована её работоспособность на ещё вдесятеро большем объёме данных, было получено одобрение непосредственного начальства, и была сделана попытка «ретроспективной оптимизации» работы одного конкретного ТЧ, для исполнения работ, реально проведённых в течение одного месяца. Результатом ретроспективного исследования стало выявление возможности существенной экономии сил и средств, порядка сотен миллионов рублей в год в рамках одного ТЧ. При этом, была проведена оптимизация только внутри одного ТЧ, без возможности перераспределения работ между двумя соседними ТЧ (территории работ любых двух ТЧ обычно имеют больше пересечение, на котором проведение работ может быть осуществлено локомотивами любого из них), а значит более полная оптимизация могла дать ещё больший эффект.

По результату проведённых вычислений в 2018г я выступил с докладом во ВНИИЖТе в рамках конкурса проектов, и получил второе место. Первое место занял проект «введение заградительных тарифов», который позволял экономить на логистике благодаря тому, что пассажиры в теории могли выбирать обед из 3 вариантов, но реально был только 1 вариант за нормальную цену, а 2 прочих варианта – за космическую стоимость, благодаря чему их никто и не покупал, а значит отпадала надобность в их закупке. Как следствие, никакого дальнейшего развития мой проект не получил.

1. Нейросетевое прогнозирование.

Примерно с 2005г второй основной моей задачей стало построение системы нейросетевого прогнозирования пассажиропотоков на РЖД, по пассажирскому и пригородному движению.

Первые версии ПО писались на SASе – язык очень удобный, но медленный, и не поддерживает распараллеливания вычислений. Но так как именно на нём был написан АРМ обработки пригородной отчётности, то программа краткосрочного прогнозирования пригородных перевозок (на срок в 1 месяц), созданная по заявке СЗППК, была написана именно на этом комплексе, и работает до сих пор.

С 2015г по заявке компании ДОСС в компании Транс-ИТ я занимался развитием прогнозирования уже на более мощном языке R. В результате был накоплен массив исходных данных, позволивший построить систему конкурентного прогнозирования, осуществляющий выдачу максимально релевантных прогнозов. Конкретно – благодаря скорости вычислений на многоядерных процессорах, с возможностью распараллеливания вычислений, была написана программа генетического развития лучших нейросетей, с отсечением худших вариантов, в которой нейросети конкурировали друг с другом по критерию качества выдаваемого прогноза. В этой программе каждая нейросеть могла дать прогноз по какой-то части исходных данных (например на конкретное направление или поезд, или дни недели, или время суток, или сезон, или класс вагона, и любое сочетание из перечисленных и прочих факторов). И после получения всех прогнозов, на каждую конкретную дату брались только лучшие из достижений, и уже они передавались заказчику.

В результате этот комплекс нейросетевого прогнозирования, работая на персоналке с 4-ядерным процессором за сутки успевал обсчитать и настроить порядка 10тысяч нейросетей, после чего выдать прогноз заполнения на примерно 500 ежедневных поездов на неделю вперёд, указывая в какую сторону и как изменится прогноз при небольшом изменении стоимости продажи билета в любую сторону.

К сожалению, в связи с развалом компании Транс-ИТ, заказчик потерял интерес к разработке.

Впоследствии, в рамках разработки системы Экспресс-НП уже во ВНИИЖТе, была попытка сделать хотя бы слабое прогнозирование на базе возможностей SQL-программирования, но из за постоянных срывов сроков сдачи системы Экспресс-НП в эксплуатацию (изначально планировалось в 2018г, самые последние планы были в 2025г), меня переориентировали на более срочные работы, а прогнозирование пассажирооборота на РЖД было похоронено совсем.

1. Выявление мошенничества на сахалинской ЖД

В 2015г, в ходе построения и исследования возможностей разных прогнозных моделей на предложенном тестовом полигоне (Сахалинская ЖД), я провёл полный анализ первичных данных о проданных билетах, и выявил огромное расхождение между количеством пассажиров и суммой выручки. Если в первом приближении – при увеличении пассажиропотока вдвое в одни сутки по сравнению с другими, сумма выручки вырастала всего на 30-50%, а не вдвое.

В ходе рассмотрения всех возможных факторов, я выявил, что всему виной была продажа билетов в арендованные вагоны, которые всегда арендовала одна и та же сторонняя фирма. В результате была выявлена следующая схема хищений.

Так как первичный прогноз пассажиропотоков в разные сезоны можно было провести легко и просто, и тем самым грубо предсказать, будет полное количество пассажиров в поезде составлять только 3-5 вагонов, или 8-15 и выше, то кто-то именно это и проделал. И на те дни, когда была гарантия большого пассажиропотока, сторонняя фирма делала заказ на прицепку к поезду дополнительных арендованных вагонов, после чего уже самостоятельно реализовывала на них билеты по своим ценам, предоставляя свой уровень сервиса, отличный от РЖД. Но так как вагоны в аренду брались за 20% себестоимости, эта фирма на пустом месте получала огромные прибыли – оттягивая на себя платёжеспособный спрос в периоды массовых поездок, и оставляя за РЖД право возить полупустые поезда в периоды низкой нагруженности – по правилам состав не имеет право быть короче 8 вагонов, а пассажиров в нём иногда всего на 3-4 вагона.

В результате, РЖД ежегодно теряла на Сахалинской дороге порядка 200-300 миллионов рублей недополученной прибыли. После этого я провёл анализ продаж по этим же алгоритмам по всей сети РЖД. и выявил ежегодные потери порядка 1млрд рублей.

Все полученные выводы и доказательства я предоставил вышестоящему начальству. Воспользовались ими или нет – я не знаю. Но примерно через год руководство этой дороги было сменено в полном составе.

10 Собеседование в Газпром на должность программиста-математика

После увольнения с РЖД. в феврале 2024г на меня вышли представители Газпрома, и предложили вакантное место программиста-математика. Для подтверждения квалификации мне было дано задание за 10 дней решить 3 задачи, и оформить их в форме презентаций. По итогу проделанной работы меня пригласили на очное собеседование, с которым я успешно справился, после чего мне предложили подать все документы для приёма на работу, но моя кандидатура была заблокирована отделом безопасности – на тот момент против меня уже завели уголовное дело, из за антивоенной деятельности, из за которого мне и пришлось бежать из России.

Полный архив описания вакансии, условий задач и их решений состоит из файлов на русском языке, могу прислать его по требованию.

Краткое описание задач:

10\_1

Дан искусственно созданный временной ряд, как на графике, надо спрогнозировать его поведение на ближайшее будущее. Мне не удалось придумать, какой точно формулой он был описан, я только нашёл два очень хороших к нему приближения, и показал, что на требуемом интервале будущего они расходятся на очень малое значение – на 1.5% от размера случайного отклонения. На графике показаны: чёрным – исходные значения, красным – аппроксимирующая функция, зеленым – разница между ними.



10\_2

Вторая задача была ближе к реальности – исследовать варианты использования двух ПХГ (подземных хранилищ газа), и найти оптимальный вариант, при котором к концу срока ежесуточная отдача была бы максимально возможной. При этом возможность газоотдачи каждого хранилища была описана своей формулой.

По свидетельству представителя компании, с которым я общался, Я был первым кандидатом, который сперва сделал именно точное аналитическое решение этой задачи. Но так как в реальной жизни таких хранилищ много. и каждое имеет свою уникальную функцию газоотдачи, пришлось написать и программу, решающую задачу оптимизации в общем случае множества газохранилищ с совершенно разными графиками газоотдачи.

10\_3

Последняя задача была самой сложной, потому что основывалась на реальных данных – по графику стоимости газа в течение почти 3 лет, надо было вычислить оптимальную стоимость опциона поставки газа на следующие полгода, при фиксированной цене поставки. Здесь мне уже удалось дать только интервал самых вероятных значений будущего поведения цены за газ, и соответственно реальной стоимости опциона. Итоговый ответ звучал следующим образом: «Средняя выручка по контракту в разных предположениях находится в диапазоне 5.5-11.2млн евро, и более точно в рамках имеющегося инструментария её вычислить невозможно.»

На предоставленном графике – реальные данные цены на газ за исходно данный период, и пример вычисления стоимости контракта, если бы он был заключён на полгода раньше требуемой даты.



Все графики взяты из итоговых презентаций для собеседования.

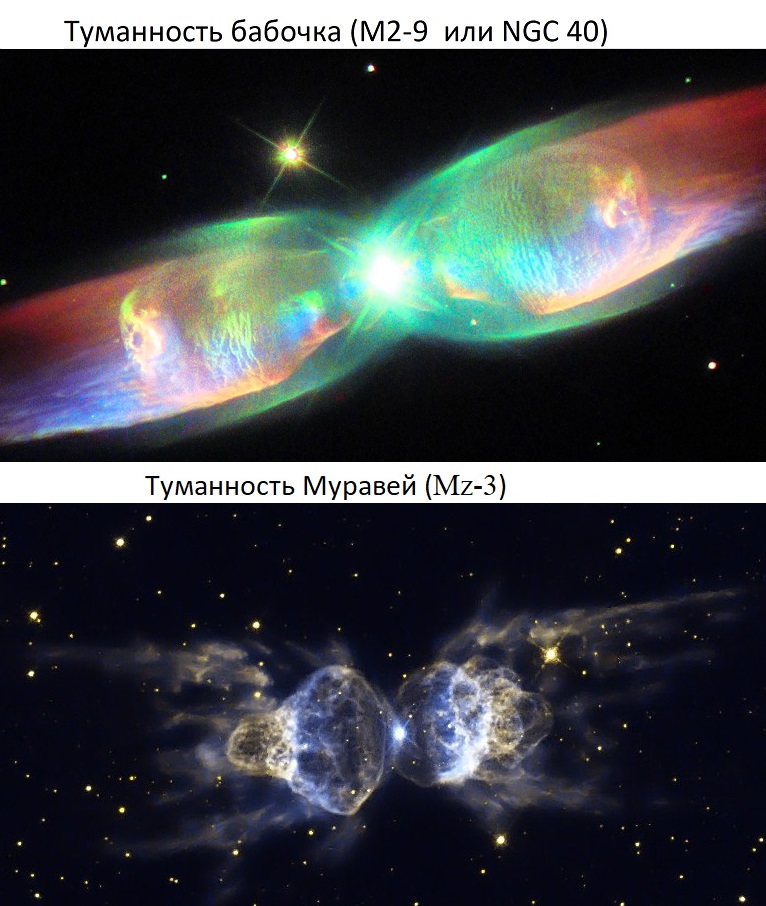
1. Написание книги по астрофизике

Несколько лет назад я увлёкся вопросами астрофизики, меня не удовлетворили общие слова о процессе формирования звёзд, и будучи знатоком математики и физики, я создал свою теорию процесса формирования карликовых звёзд. Как впоследствии оказалось. она в некоторых чертах похожа на теорию выдвигавшуюся американским астрофизиком Карлом Роузом, но с серьёзными отличиями.

Я рассмотрел именно процесс формирования звезды, и пришёл к выводу, что в процессе её образования в ней просто обязано было получиться центральное ядро, состоящее из тяжёлых элементов. Так как такая картина строения звезды противоречит всем современным представлениям о ней, то я начал изыскания, по поводу возможности подтверждения или опровержения моей теории. В результате я нашёл ей множество подтверждений, и ни одного опровержения. Попутно. Исходя именно из этой теории я смог дать объяснения многим непонятным современной науке явлениям. Кратко перечислю основные из них.

11\_1 Моя теория объясняет не только процесс формирования звёзд, но и планет вокруг них, и в результате даёт объяснение, посему у внутренних планет (Меркурия, Венеры, Земли и Луны) есть металлическое ядро. А Марс, имеющий большую нежели Луна среднюю плотность, и большее содержание железа на поверхности планеты, его лишён. Так же моя теория показывает, почему общее содержание разных элементов на планетах именно таково, каким мы его сейчас наблюдаем.

11\_2 Моя теория объясняет процесс формирования биполярных планетарных туманностей, полностью описывая их внутреннюю структуру, и причину серьёзной разницы в химическом составе разных частей газовой оболочки. Примеры таких туманностей на иллюстрации.



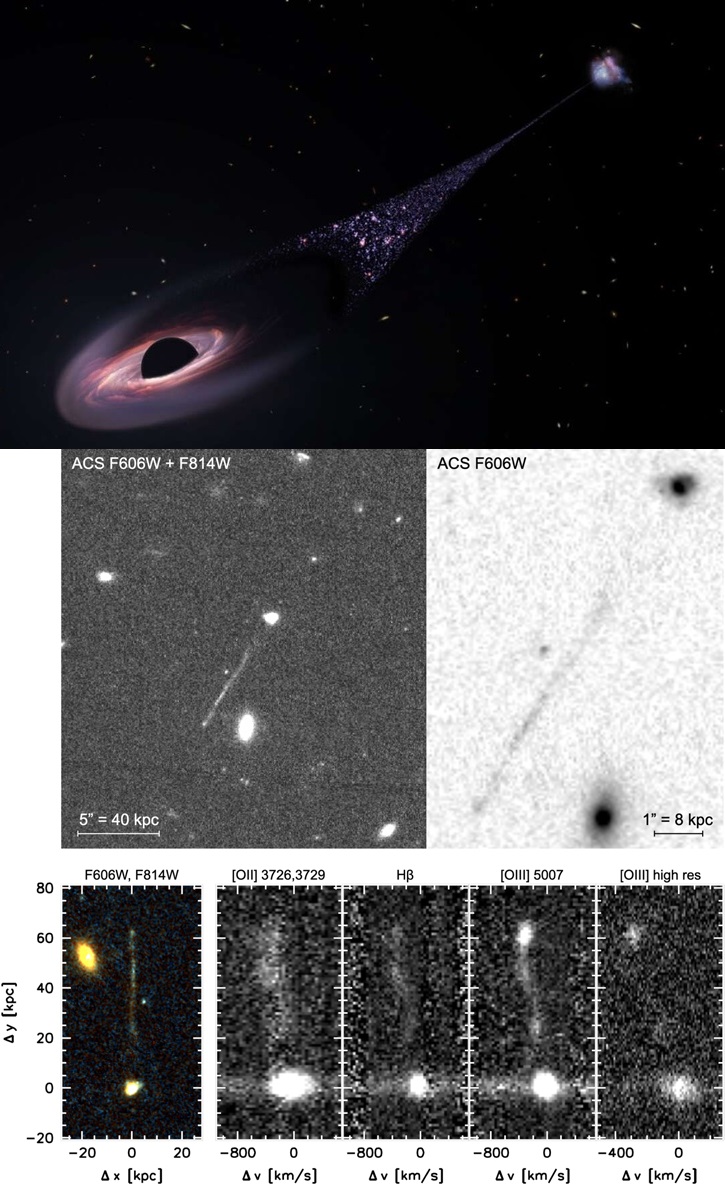
11\_3 Моя теория объясняет причину именно такой структуры туманности Столпы творения, какую наблюдают астрономы – она образована не только взрывом близкой сверхновой, но и множеством гравитирующих протозвёзд, сложенных из тяжёлых элементов.



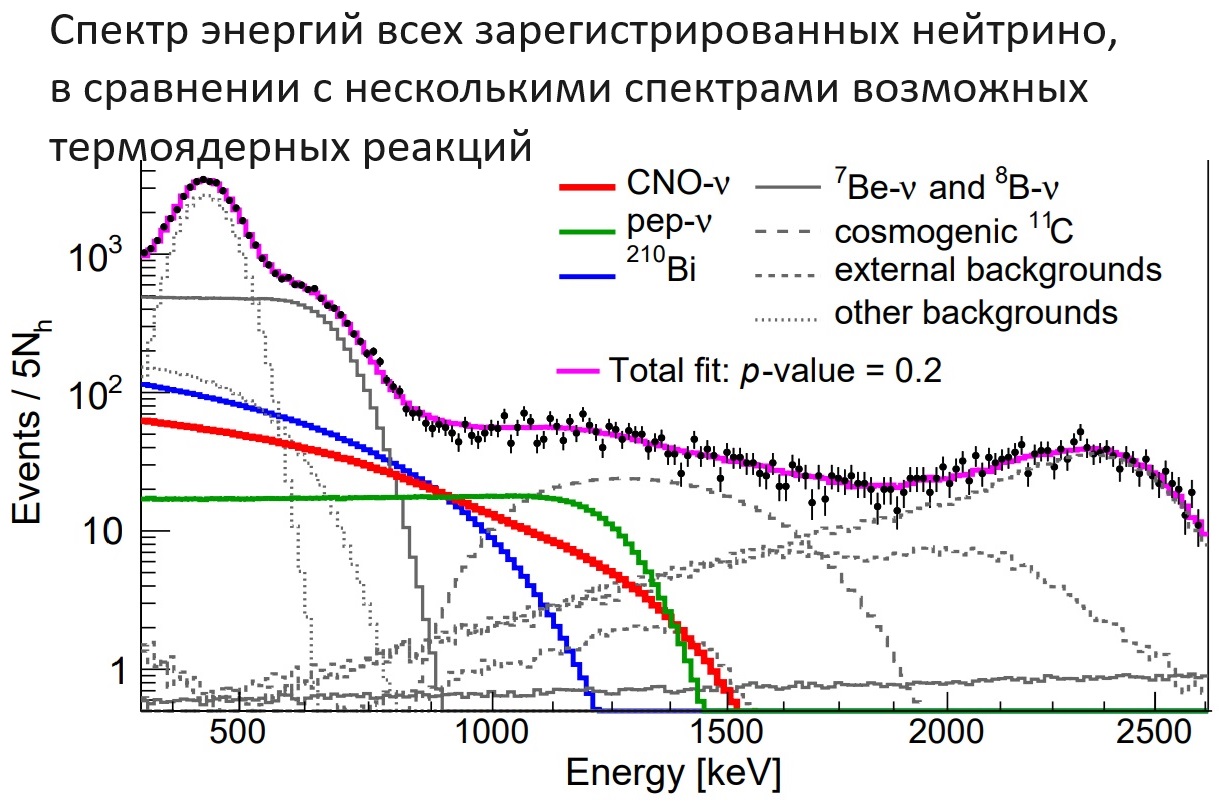
11\_4 Моя теория показывает ошибочность представлений о «сверхстарых звёздах, возрастом почти как наша вселенная, а иногда и старше неё», показывая в частности, что для светимости этих звёзд в том размере, который наблюдают астрономы, термоядерного горючего (водорода) имеющегося в их составе хватило бы только на несколько процентов их времени жизни, после чего они были бы вынуждены погаснуть – а они наоборот светят многократно сильнее нашего солнца, имея меньшую нежели у солнца массу.

11\_5 Моя теория показывает ошибочность теории большого взрыва – я чисто математически доказал, что эта теория противоречит наличию вращающихся галактик. То есть, при правдивости гипотезы Большого взрыва, вещество разделённое на галактики должно было образовать не вращающиеся комки, что противоречит наблюдениям. Моя теория наоборот подтверждает теорию Большого Отскока, то есть циклической вселенной. И показывает, что в этом случае соседние галактики должны вращаться согласованно – это было предсказано мною до экспериментального обнаружения данного факта летом 2023 года – современная астрофизика пока не может дать этому объяснения, а я наоборот это предсказал, и показал механизм подобного явления.

11\_6 Моя теория объясняет наблюдение - почему одна чёрная дыра, вылетевшая из своей галактики на 250тысяч световых лет, оставила за собой светящийся след, показывает механизм образования этого свечения, и объясняет его видимую внутреннюю структуру.



11\_7 Моя теория объясняет, почему в фотосфере Солнца с одной стороны совершенно нет даже следов таких элементов, как литий или бериллий, но с другой стороны – нейтринная астрономия показала, что 8% всех солнечных нейтрино рождены именно в реакции водорода с бериллием – отсутствующего в солнечной короне элемента. Тем самым моя теория буквально доказывает неизбежность существования центрального ядра, полностью лишённого водорода.



11\_8 Моя теория так же объясняет ставящую в тупик астрофизиков структуру туманности Pa-30. Образовавшуюся после столкновения двух карликовых звёзд.

